



R MARKDOWN

*Cómo crear Tablas
de información
en R Markdown*

Pedro Luis Luque Calvo

Sevilla, Septiembre de 2019

Índice general

| | |
|--|----------|
| 1. Cómo crear tablas de información en R Markdown | 1 |
| 1.1. Ejemplos de tablas con información en un documento | 1 |
| 1.1.1. Ejemplo: tabla con información numérica (tasas específicas) | 2 |
| 1.1.2. Ejemplo: tabla con información numérica (peso relativo) | 2 |
| 1.1.3. Ejemplo: tablas con colores y diferentes tamaños de letra | 3 |
| 1.1.4. Ejemplos: tablas con agrupamiento de columnas y de filas | 3 |
| 1.1.5. Ejemplo: tablas con notas a pie de tabla | 4 |
| 1.1.6. Ejemplo: tabla con información en varias páginas (tabla mortalidad) | 5 |
| 1.1.7. Ejemplo: tabla con dos columnas con información textual | 7 |
| 1.1.8. Ejemplo: tabla apaisada en página aparte | 7 |
| 1.1.9. Ejemplos: tablas con resúmenes estadísticos | 9 |
| 1.1.10. Ejemplos: información matemática en forma de tabla | 11 |
| 1.2. Introducción a la creación de tablas con R Markdown | 12 |
| 1.3. Sitios web recomendados | 13 |
| 1.4. No se dispone de la información en formato digital, ¿cómo la añado? | 14 |
| 1.4.1. Utilizando código markdown | 14 |
| 1.4.1.1. Una tabla sin fila de cabecera | 16 |
| 1.4.2. Utilizando código R | 17 |
| 1.4.2.1. Uso de objetos tibble o tribble | 17 |
| 1.4.3. Utilizando LaTeX | 21 |
| 1.5. Consejos sobre RStudio y los Addins | 24 |
| 1.5.1. Addins de la librería datapasta | 25 |
| 1.5.2. Addins de la librería inserttable | 28 |
| 1.5.3. Addins de la librería remedy | 32 |
| 1.5.4. Addins de la librería rhandsontable | 33 |
| 1.5.5. Asociar combinaciones de teclas a los Addins | 34 |
| 1.6. Ya se dispone de la información en formato digital | 36 |
| 1.6.1. Se tiene información en un objeto R | 36 |
| 1.6.2. Crear tablas con la función kable de la librería knitr | 37 |
| 1.6.2.1. Tablas sin cabeceras con kable | 39 |
| 1.6.3. Crear tablas con la librería kableExtra | 40 |
| 1.6.3.1. Ejemplo kableExtra: añadiendo una leyenda explicativa a la tabla y manejo de la posición FLOTANTE de la tabla | 44 |
| 1.6.3.2. Ejemplo kableExtra: tablas que ocupan varias páginas | 50 |
| 1.6.3.3. Ejemplo kableExtra: especificar características columnas/filas/cabeceras | 52 |
| 1.6.3.4. Ejemplo kableExtra: especificar ancho de columnas y modo apaisado | 54 |

| | | |
|------------|--|----|
| 1.6.3.5. | Ejemplo kableExtra: leer data.frame de fichero externo, tabla con colores y cabeceras compuestas | 57 |
| 1.6.3.6. | Ejemplo kableExtra: tablas con colores y diferentes tamaños de letra | 59 |
| 1.6.3.7. | Ejemplo kableExtra: tablas con agrupamiento | 60 |
| 1.6.3.8. | Ejemplo kableExtra: tablas con notas a pie de página | 64 |
| 1.6.3.9. | Ejemplo kableExtra: saltos de líneas en celdas | 65 |
| 1.6.4. | Crear tablas con la librería huxtable | 66 |
| 1.6.4.1. | Primeros pasos | 67 |
| 1.6.4.2. | Ejemplo huxtable: caracteres especiales | 70 |
| 1.6.4.3. | Temas con huxtable | 70 |
| 1.6.5. | Crear tablas con la librería pander | 73 |
| 1.6.5.1. | Personalizando celdas | 75 |
| 1.6.5.2. | Redondear y formatear números | 77 |
| 1.6.5.3. | Ejemplos pander: personalizando los resultados de modelos estadísticos | 79 |
| 1.6.5.4. | Ejemplo pander: con leyendas y referencias | 81 |
| 1.6.6. | Crear tablas con la librería stargazer | 84 |
| 1.6.6.1. | Ejemplo stargazer: tablas con resúmenes estadísticos descriptivos | 84 |
| 1.6.6.2. | Ejemplo stargazer: tablas con resúmenes estadísticos de modelos lineales | 84 |
| 1.6.7. | Algunas cuestiones específicas para tablas con salida html o word | 86 |
| 1.6.7.1. | Conversión de tablas kableExtra LaTeX para ser incluidas en html o word | 86 |
| 1.6.7.2. | Tablas interactivas | 87 |
| 1.6.7.2.1. | Crear tablas con la librería rhandsontable | 87 |
| 1.6.7.2.2. | Crear tablas con la librería DT | 88 |
| 1.7. | Crear funciones R personalizadas para crear tablas fácilmente | 89 |
| 1.8. | Tabla Resumen: construcción de tablas con knitr-kableExtra | 91 |

Bibliografía

95

Capítulo 1

Cómo crear tablas de información en R Markdown

Cuando se escribe un documento, ya sea un libro, un informe, un manual o un trabajo fin de estudios, además de la redacción del texto se realizan una serie de tareas como:

- añadir tablas con información,
- añadir gráficos,
- añadir referencias a una tabla, a un gráfico, a una ecuación, o a un libro.

Los procedimientos que se recogerán en este apartado se centrarán principalmente en la creación de tablas con información para documentos en formato “pdf”, aunque también se añadirán algunas explicaciones para otros tipos de formatos de documentos: “html” (páginas web) y “docx” (documentos Word).

1.1. Ejemplos de tablas con información en un documento

Antes de ver cómo se crean las tablas de información en R Markdown, a continuación se recopilan algunos ejemplos que mostrarán distintas cuestiones a considerar cuando se escribe un documento. En cada uno de los ejemplos, el título del apartado resalta el aspecto al que se hace referencia.

1.1.1. Ejemplo: tabla con información numérica (tasas específicas)

| Edades | PobMediaMuj2016 | NacMadres2016 | TEspFec |
|--------|-----------------|---------------|---------|
| 15 | 219626.5 | 343 | 1.56 |
| 16 | 216904.0 | 818 | 3.77 |
| 17 | 211281.0 | 1408 | 6.66 |
| 18 | 209683.5 | 2208 | 10.53 |
| 19 | 211549.5 | 3152 | 14.90 |
| 20 | 211355.0 | 4014 | 18.99 |
| 21 | 213269.5 | 4707 | 22.07 |
| 22 | 220661.5 | 5409 | 24.51 |
| 23 | 230947.5 | 6770 | 29.31 |
| 24 | 236064.5 | 7975 | 33.78 |
| 25 | 239599.5 | 9721 | 40.57 |
| 26 | 246143.0 | 11945 | 48.53 |
| 27 | 253044.0 | 14018 | 55.40 |
| 28 | 260030.5 | 16885 | 64.93 |
| 29 | 268029.0 | 20232 | 75.48 |
| 30 | 277724.5 | 23629 | 85.08 |
| 31 | 289286.0 | 26817 | 92.70 |
| 32 | 301347.5 | 28932 | 96.01 |
| 33 | 316106.5 | 30797 | 97.43 |
| 34 | 332612.0 | 31835 | 95.71 |
| 35 | 348698.5 | 31376 | 89.98 |
| 36 | 363271.0 | 28997 | 79.82 |
| 37 | 376800.0 | 25342 | 67.26 |
| 38 | 387068.5 | 21408 | 55.31 |
| 39 | 393461.0 | 17282 | 43.92 |
| 40 | 395821.5 | 12835 | 32.43 |
| 41 | 393157.5 | 8548 | 21.74 |
| 42 | 387439.5 | 5359 | 13.83 |
| 43 | 381809.0 | 3194 | 8.37 |
| 44 | 378583.5 | 2025 | 5.35 |
| 45 | 373687.5 | 1214 | 3.25 |
| 46 | 370302.0 | 561 | 1.51 |
| 47 | 367959.5 | 275 | 0.75 |
| 48 | 368404.5 | 195 | 0.53 |
| 49 | 365317.0 | 113 | 0.31 |

1.1.2. Ejemplo: tabla con información numérica (peso relativo)

| Grupos Edades | Partos Sencillos | Partos Dobles | Partos Triples | Partos CuadMas |
|------------------|------------------|---------------|----------------|----------------|
| Todas las edades | 97.76 | 2.21 | 0.03 | – |
| Menos de 15 | 99.11 | 0.89 | – | – |
| De 15 a 19 | 99.37 | 0.63 | – | – |
| De 20 a 24 | 99.01 | 0.96 | 0.03 | – |
| De 25 a 29 | 98.52 | 1.46 | 0.02 | – |
| De 30 a 34 | 97.92 | 2.05 | 0.03 | – |
| De 35 a 39 | 97.29 | 2.68 | 0.03 | – |
| De 40 a 44 | 96.12 | 3.86 | 0.02 | – |
| De 45 a 49 | 90.85 | 9.1 | 0.05 | – |
| De 50 y más | 89.17 | 10.83 | – | – |

1.1.3. Ejemplo: tablas con colores y diferentes tamaños de letra

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 5 | 3.4 | 1.5 | 0.2 | setosa |
| 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 4.9 | 3.1 | 1.5 | 0.1 | setosa |

1.1.4. Ejemplos: tablas con agrupamiento de columnas y de filas

| | Group 1 | | Group 2 | | Group 3 | |
|-------------------|---------|-----|---------|-----|---------|-------|
| | mpg | cyl | disp | hp | drat | wt |
| Mazda RX4 | 21.0 | 6 | 160 | 110 | 3.90 | 2.620 |
| Mazda RX4 Wag | 21.0 | 6 | 160 | 110 | 3.90 | 2.875 |
| Datsun 710 | 22.8 | 4 | 108 | 93 | 3.85 | 2.320 |
| Hornet 4 Drive | 21.4 | 6 | 258 | 110 | 3.08 | 3.215 |
| Hornet Sportabout | 18.7 | 8 | 360 | 175 | 3.15 | 3.440 |

| | mpg | cyl | disp | hp | drat | wt |
|-------------------|------|-----|-------|-----|------|-------|
| Mazda RX4 | 21.0 | 6 | 160.0 | 110 | 3.90 | 2.620 |
| Mazda RX4 Wag | 21.0 | 6 | 160.0 | 110 | 3.90 | 2.875 |
| Datsun 710 | 22.8 | 4 | 108.0 | 93 | 3.85 | 2.320 |
| Group 1 | | | | | | |
| Hornet 4 Drive | 21.4 | 6 | 258.0 | 110 | 3.08 | 3.215 |
| Hornet Sportabout | 18.7 | 8 | 360.0 | 175 | 3.15 | 3.440 |
| Valiant | 18.1 | 6 | 225.0 | 105 | 2.76 | 3.460 |
| Duster 360 | 14.3 | 8 | 360.0 | 245 | 3.21 | 3.570 |
| Group 2 | | | | | | |
| Merc 240D | 24.4 | 4 | 146.7 | 62 | 3.69 | 3.190 |
| Merc 230 | 22.8 | 4 | 140.8 | 95 | 3.92 | 3.150 |
| Merc 280 | 19.2 | 6 | 167.6 | 123 | 3.92 | 3.440 |

| | C3 | C4 |
|----------|----|----|
| a | | |
| c | 1 | 0 |
| | 2 | 1 |
| | 3 | 1 |
| | 4 | 0 |
| | 5 | 0 |
| | 6 | 1 |
| | 7 | 0 |
| d | 8 | 0 |
| | 9 | 1 |
| | 10 | 0 |
| b | | |
| c | 11 | 1 |
| | 12 | 0 |
| d | 13 | 0 |
| | 14 | 1 |
| | 15 | 0 |

1.1.5. Ejemplo: tablas con notas a pie de tabla

| | mpg | cyl | disp | hp | drat | wt |
|-------------------|------|-----|------|-----|------|-------|
| Mazda RX4 | 21.0 | 6 | 160 | 110 | 3.90 | 2.620 |
| Mazda RX4 Wag | 21.0 | 6 | 160 | 110 | 3.90 | 2.875 |
| Datsun 710 | 22.8 | 4 | 108 | 93 | 3.85 | 2.320 |
| Hornet 4 Drive | 21.4 | 6 | 258 | 110 | 3.08 | 3.215 |
| Hornet Sportabout | 18.7 | 8 | 360 | 175 | 3.15 | 3.440 |

General: Here is a general comments of the table.

Type I: ¹ Footnote 1; ² Footnote 2;

Type II: ^a Footnote A; ^b Footnote B;

Type III: * Footnote Symbol 1; † Footnote Symbol 2

1.1.6. Ejemplo: tabla con información en varias páginas (tabla mortalidad)

| | x | mx | qx | dx | lx | px | Lx | Tx | ex | Sx |
|----|----|---------|---------|-----|--------|---------|-------|---------|-------|---------|
| 0 | 0 | 0.00295 | 0.00295 | 295 | 100000 | 0.99705 | 99734 | 8284892 | 82.85 | 1.99384 |
| 1 | 1 | 0.00023 | 0.00092 | 92 | 99705 | 0.99908 | 99650 | 8185157 | 82.09 | 0.99925 |
| 2 | 2 | 0.00016 | 0.00064 | 64 | 99613 | 0.99936 | 99575 | 8085507 | 81.17 | 0.99948 |
| 3 | 3 | 0.00011 | 0.00044 | 44 | 99550 | 0.99956 | 99523 | 7985932 | 80.22 | 0.99958 |
| 4 | 4 | 0.00010 | 0.00040 | 40 | 99506 | 0.99960 | 99482 | 7886409 | 79.26 | 0.99979 |
| 5 | 5 | 0.00010 | 0.00010 | 10 | 99466 | 0.99990 | 99461 | 7786927 | 78.29 | 0.99991 |
| 6 | 6 | 0.00008 | 0.00008 | 8 | 99456 | 0.99992 | 99452 | 7687466 | 77.30 | 0.99992 |
| 7 | 7 | 0.00009 | 0.00009 | 9 | 99448 | 0.99991 | 99444 | 7588014 | 76.30 | 0.99992 |
| 8 | 8 | 0.00007 | 0.00007 | 7 | 99439 | 0.99993 | 99436 | 7488570 | 75.31 | 0.99995 |
| 9 | 9 | 0.00004 | 0.00004 | 4 | 99432 | 0.99996 | 99430 | 7389134 | 74.31 | 0.99995 |
| 10 | 10 | 0.00006 | 0.00006 | 6 | 99428 | 0.99994 | 99425 | 7289704 | 73.32 | 0.99994 |
| 11 | 11 | 0.00007 | 0.00007 | 7 | 99422 | 0.99993 | 99419 | 7190279 | 72.32 | 0.99993 |
| 12 | 12 | 0.00008 | 0.00008 | 8 | 99415 | 0.99992 | 99411 | 7090860 | 71.33 | 0.99991 |
| 13 | 13 | 0.00010 | 0.00010 | 10 | 99407 | 0.99990 | 99402 | 6991449 | 70.33 | 0.99988 |
| 14 | 14 | 0.00014 | 0.00014 | 14 | 99397 | 0.99986 | 99390 | 6892046 | 69.34 | 0.99987 |
| 15 | 15 | 0.00012 | 0.00012 | 12 | 99383 | 0.99988 | 99378 | 6792656 | 68.35 | 0.99988 |
| 16 | 16 | 0.00013 | 0.00013 | 13 | 99372 | 0.99987 | 99365 | 6693278 | 67.36 | 0.99987 |
| 17 | 17 | 0.00014 | 0.00014 | 14 | 99359 | 0.99986 | 99352 | 6593913 | 66.36 | 0.99983 |
| 18 | 18 | 0.00020 | 0.00020 | 20 | 99345 | 0.99980 | 99335 | 6494562 | 65.37 | 0.99978 |
| 19 | 19 | 0.00024 | 0.00024 | 24 | 99325 | 0.99976 | 99313 | 6395227 | 64.39 | 0.99974 |
| 20 | 20 | 0.00029 | 0.00029 | 29 | 99301 | 0.99971 | 99287 | 6295914 | 63.40 | 0.99971 |
| 21 | 21 | 0.00029 | 0.00029 | 29 | 99272 | 0.99971 | 99258 | 6196627 | 62.42 | 0.99971 |
| 22 | 22 | 0.00030 | 0.00030 | 30 | 99243 | 0.99970 | 99229 | 6097369 | 61.44 | 0.99971 |
| 23 | 23 | 0.00028 | 0.00028 | 28 | 99214 | 0.99972 | 99200 | 5998141 | 60.46 | 0.99972 |
| 24 | 24 | 0.00028 | 0.00028 | 28 | 99186 | 0.99972 | 99172 | 5898941 | 59.47 | 0.99972 |
| 25 | 25 | 0.00028 | 0.00028 | 28 | 99158 | 0.99972 | 99144 | 5799769 | 58.49 | 0.99972 |
| 26 | 26 | 0.00028 | 0.00028 | 28 | 99130 | 0.99972 | 99117 | 5700625 | 57.51 | 0.99971 |
| 27 | 27 | 0.00030 | 0.00030 | 30 | 99103 | 0.99970 | 99088 | 5601508 | 56.52 | 0.99970 |
| 28 | 28 | 0.00030 | 0.00030 | 30 | 99073 | 0.99970 | 99058 | 5502420 | 55.54 | 0.99970 |
| 29 | 29 | 0.00031 | 0.00031 | 31 | 99043 | 0.99969 | 99028 | 5403362 | 54.56 | 0.99966 |
| 30 | 30 | 0.00037 | 0.00037 | 37 | 99012 | 0.99963 | 98994 | 5304335 | 53.57 | 0.99961 |
| 31 | 31 | 0.00042 | 0.00042 | 42 | 98976 | 0.99958 | 98955 | 5205340 | 52.59 | 0.99959 |
| 32 | 32 | 0.00040 | 0.00040 | 40 | 98934 | 0.99960 | 98915 | 5106385 | 51.61 | 0.99957 |
| 33 | 33 | 0.00046 | 0.00046 | 45 | 98895 | 0.99954 | 98872 | 5007471 | 50.63 | 0.99955 |
| 34 | 34 | 0.00045 | 0.00045 | 44 | 98849 | 0.99955 | 98827 | 4908599 | 49.66 | 0.99955 |
| 35 | 35 | 0.00046 | 0.00046 | 45 | 98805 | 0.99954 | 98782 | 4809772 | 48.68 | 0.99954 |
| 36 | 36 | 0.00047 | 0.00047 | 46 | 98759 | 0.99953 | 98736 | 4710990 | 47.70 | 0.99951 |
| 37 | 37 | 0.00052 | 0.00052 | 51 | 98713 | 0.99948 | 98687 | 4612254 | 46.72 | 0.99944 |
| 38 | 38 | 0.00060 | 0.00060 | 59 | 98662 | 0.99940 | 98632 | 4513566 | 45.75 | 0.99939 |
| 39 | 39 | 0.00063 | 0.00063 | 62 | 98602 | 0.99937 | 98571 | 4414934 | 44.78 | 0.99930 |
| 40 | 40 | 0.00077 | 0.00077 | 76 | 98540 | 0.99923 | 98502 | 4316363 | 43.80 | 0.99918 |
| 41 | 41 | 0.00087 | 0.00087 | 86 | 98464 | 0.99913 | 98422 | 4217860 | 42.84 | 0.99909 |
| 42 | 42 | 0.00095 | 0.00095 | 93 | 98379 | 0.99905 | 98332 | 4119439 | 41.87 | 0.99898 |
| 43 | 43 | 0.00110 | 0.00110 | 108 | 98285 | 0.99890 | 98231 | 4021107 | 40.91 | 0.99883 |
| 44 | 44 | 0.00124 | 0.00124 | 122 | 98177 | 0.99876 | 98117 | 3922875 | 39.96 | 0.99864 |
| 45 | 45 | 0.00148 | 0.00148 | 145 | 98056 | 0.99852 | 97983 | 3824759 | 39.01 | 0.99847 |
| 46 | 46 | 0.00159 | 0.00159 | 156 | 97911 | 0.99841 | 97833 | 3726775 | 38.06 | 0.99833 |
| 47 | 47 | 0.00176 | 0.00176 | 172 | 97755 | 0.99824 | 97669 | 3628942 | 37.12 | 0.99812 |
| 48 | 48 | 0.00200 | 0.00200 | 195 | 97583 | 0.99800 | 97486 | 3531273 | 36.19 | 0.99786 |
| 49 | 49 | 0.00228 | 0.00228 | 222 | 97388 | 0.99772 | 97277 | 3433787 | 35.26 | 0.99766 |
| 50 | 50 | 0.00240 | 0.00240 | 233 | 97166 | 0.99760 | 97050 | 3336510 | 34.34 | 0.99748 |
| 51 | 51 | 0.00264 | 0.00264 | 256 | 96934 | 0.99736 | 96806 | 3239460 | 33.42 | 0.99720 |
| 52 | 52 | 0.00297 | 0.00297 | 287 | 96678 | 0.99703 | 96535 | 3142654 | 32.51 | 0.99693 |
| 53 | 53 | 0.00318 | 0.00317 | 306 | 96391 | 0.99683 | 96238 | 3046120 | 31.60 | 0.99663 |
| 54 | 54 | 0.00358 | 0.00357 | 343 | 96085 | 0.99643 | 95914 | 2949881 | 30.70 | 0.99624 |
| 55 | 55 | 0.00396 | 0.00395 | 378 | 95742 | 0.99605 | 95553 | 2853968 | 29.81 | 0.99593 |

(continúa)

| | x | mx | qx | dx | lx | px | Lx | Tx | ex | Sx |
|------|-----|---------|---------|------|-------|---------|-------|---------|-------|---------|
| 56 | 56 | 0.00420 | 0.00419 | 400 | 95364 | 0.99581 | 95164 | 2758415 | 28.93 | 0.99574 |
| 57 | 57 | 0.00434 | 0.00433 | 411 | 94964 | 0.99567 | 94758 | 2663251 | 28.04 | 0.99545 |
| 58 | 58 | 0.00478 | 0.00477 | 451 | 94553 | 0.99523 | 94327 | 2568493 | 27.16 | 0.99494 |
| 59 | 59 | 0.00537 | 0.00536 | 504 | 94102 | 0.99464 | 93850 | 2474166 | 26.29 | 0.99440 |
| 60 | 60 | 0.00586 | 0.00584 | 547 | 93598 | 0.99416 | 93324 | 2380316 | 25.43 | 0.99382 |
| 61 | 61 | 0.00654 | 0.00652 | 607 | 93051 | 0.99348 | 92748 | 2286992 | 24.58 | 0.99338 |
| 62 | 62 | 0.00675 | 0.00673 | 622 | 92444 | 0.99327 | 92133 | 2194245 | 23.74 | 0.99307 |
| 63 | 63 | 0.00716 | 0.00713 | 655 | 91822 | 0.99287 | 91495 | 2102111 | 22.89 | 0.99247 |
| 64 | 64 | 0.00795 | 0.00792 | 722 | 91167 | 0.99208 | 90806 | 2010616 | 22.05 | 0.99151 |
| 65 | 65 | 0.00910 | 0.00906 | 819 | 90445 | 0.99094 | 90036 | 1919810 | 21.23 | 0.99074 |
| 66 | 66 | 0.00951 | 0.00946 | 848 | 89626 | 0.99054 | 89202 | 1829774 | 20.42 | 0.99068 |
| 67 | 67 | 0.00922 | 0.00918 | 815 | 88778 | 0.99082 | 88370 | 1740573 | 19.61 | 0.98982 |
| 68 | 68 | 0.01126 | 0.01120 | 985 | 87963 | 0.98880 | 87470 | 1652202 | 18.78 | 0.98851 |
| 69 | 69 | 0.01185 | 0.01178 | 1025 | 86978 | 0.98822 | 86466 | 1564732 | 17.99 | 0.98756 |
| 70 | 70 | 0.01319 | 0.01310 | 1126 | 85953 | 0.98690 | 85390 | 1478266 | 17.20 | 0.98671 |
| 71 | 71 | 0.01358 | 0.01349 | 1144 | 84827 | 0.98651 | 84255 | 1392876 | 16.42 | 0.98649 |
| 72 | 72 | 0.01363 | 0.01354 | 1133 | 83683 | 0.98646 | 83116 | 1308621 | 15.64 | 0.98410 |
| 73 | 73 | 0.01847 | 0.01830 | 1511 | 82550 | 0.98170 | 81795 | 1225504 | 14.85 | 0.98166 |
| 74 | 74 | 0.01856 | 0.01839 | 1490 | 81039 | 0.98161 | 80294 | 1143710 | 14.11 | 0.98130 |
| 75 | 75 | 0.01920 | 0.01902 | 1513 | 79549 | 0.98098 | 78793 | 1063415 | 13.37 | 0.97693 |
| 76 | 76 | 0.02758 | 0.02720 | 2123 | 78036 | 0.97280 | 76975 | 984623 | 12.62 | 0.97109 |
| 77 | 77 | 0.03114 | 0.03066 | 2328 | 75913 | 0.96934 | 74749 | 907648 | 11.96 | 0.96909 |
| 78 | 78 | 0.03166 | 0.03117 | 2293 | 73586 | 0.96883 | 72439 | 832899 | 11.32 | 0.96748 |
| 79 | 79 | 0.03451 | 0.03392 | 2419 | 71292 | 0.96608 | 70083 | 760460 | 10.67 | 0.96326 |
| 80 | 80 | 0.04046 | 0.03966 | 2731 | 68874 | 0.96034 | 67508 | 690377 | 10.02 | 0.95815 |
| 81 | 81 | 0.04512 | 0.04412 | 2918 | 66142 | 0.95588 | 64683 | 622869 | 9.42 | 0.95377 |
| 82 | 82 | 0.04964 | 0.04844 | 3062 | 63224 | 0.95156 | 61693 | 558186 | 8.83 | 0.94803 |
| 83 | 83 | 0.05728 | 0.05569 | 3350 | 60161 | 0.94431 | 58486 | 496494 | 8.25 | 0.94088 |
| 84 | 84 | 0.06480 | 0.06277 | 3566 | 56811 | 0.93723 | 55028 | 438007 | 7.71 | 0.93307 |
| 85 | 85 | 0.07402 | 0.07138 | 3801 | 53245 | 0.92862 | 51345 | 382979 | 7.19 | 0.92462 |
| 86 | 86 | 0.08299 | 0.07968 | 3940 | 49445 | 0.92032 | 47475 | 331634 | 6.71 | 0.91451 |
| 87 | 87 | 0.09622 | 0.09180 | 4177 | 45505 | 0.90820 | 43416 | 284159 | 6.24 | 0.90272 |
| 88 | 88 | 0.10894 | 0.10331 | 4270 | 41327 | 0.89669 | 39193 | 240743 | 5.83 | 0.89076 |
| 89 | 89 | 0.12298 | 0.11586 | 4293 | 37058 | 0.88414 | 34911 | 201550 | 5.44 | 0.87626 |
| 90 | 90 | 0.14208 | 0.13266 | 4346 | 32764 | 0.86734 | 30591 | 166639 | 5.09 | 0.86064 |
| 91 | 91 | 0.15876 | 0.14708 | 4180 | 28418 | 0.85292 | 26328 | 136048 | 4.79 | 0.84773 |
| 92 | 92 | 0.17196 | 0.15835 | 3838 | 24238 | 0.84165 | 22319 | 109720 | 4.53 | 0.83682 |
| 93 | 93 | 0.18451 | 0.16893 | 3446 | 20400 | 0.83107 | 18677 | 87401 | 4.28 | 0.82404 |
| 94 | 94 | 0.20317 | 0.18443 | 3127 | 16954 | 0.81557 | 15391 | 68724 | 4.05 | 0.81210 |
| 95 | 95 | 0.21256 | 0.19214 | 2657 | 13827 | 0.80786 | 12499 | 53333 | 3.86 | 0.79315 |
| 96 | 96 | 0.25361 | 0.22507 | 2514 | 11170 | 0.77493 | 9913 | 40834 | 3.66 | 0.76845 |
| 97 | 97 | 0.27261 | 0.23991 | 2077 | 8656 | 0.76009 | 7618 | 30921 | 3.57 | 0.75341 |
| 98 | 98 | 0.29275 | 0.25537 | 1680 | 6580 | 0.74463 | 5739 | 23303 | 3.54 | 0.73836 |
| 99 | 99 | 0.31223 | 0.27007 | 1323 | 4899 | 0.72993 | 4238 | 17564 | 3.58 | 0.75872 |
| 100+ | 100 | 0.26836 | 1.00000 | 3576 | 3576 | 0.00000 | 13326 | 13326 | 3.73 | - |

1.1.7. Ejemplo: tabla con dos columnas con información textual

Tabla 1.4: Variables derived or modified from the ETS system for use in the analyses throughout this thesis.

| Created/modified variable | Description |
|---------------------------|---|
| Years since BCG | Derived using year of vaccination and year of notification. Categorised into ≤ 10 and $11+$ due to the evidence of waning protection for the BCG vaccine.[14] |
| Age at BCG | Derived using year of vaccination and age at vaccination. Categorised into < 1 , 1 to $x < 12$, 12 to $x < 16$ and ≤ 16 to capture historic vaccination policy.[2] |
| Successful treatment | For cases that had a recorded date of starting treatment, with their outcome recorded at the latest available follow up. Those that completed treatment are defined as successfully treated: treatment failure is defined as those that stopped treatment, were lost to follow up, those that died during follow up from TB, those that died during follow up were TB contributed to their death, and those who were still on treatment. Those that were not evaluated were treated as missing. |
| Mortality | Assessed via follow up at 12 and 24 months: mortality is defined as cases with an overall outcome of death, and survival is defined as those that completed treatment, were still on treatment, and stopped treatment. Those that were lost to follow up, or not evaluated were treated as missing |
| TB mortality | For cases with an overall outcome of died, and whose cause of death was known to be TB or to be related to TB. Those that were known to have not died, or who were known to have died from a cause other than from TB were defined to have not died from TB. |
| Death due to TB | Death due to TB is defined as those that died directly from TB, or where TB had contributed to their death with death not due to TB being cases that died from any other cause. Conditioned on all-cause mortality, for cases with a known cause of death. |

1.1.8. Ejemplo: tabla apaisada en página aparte

| Month | Sam | Supervisors |
|----------|---|---|
| Feburary | 1st Introduction 15th, The epidemiology of tuberculosis, and the role of BCG vaccination, in England & Developing and parameterising a dynamic transmission model of tuberculosis | 15th Introduction |
| March | 1st Reassessing the Evidence for Universal School-age Bacillus Calmette Guerin (BCG) Vaccination in England and Wales & Beneficial effects of BCG vaccination in outcomes for patients diagnosed with TB: observation study using the Enhanced Tuberculosis Surveillance System 2009-2015 15th Investigating the impact of the 2005 change in BCG vaccination policy using a fitted dynamic transmission model of TB & Estimating the effect of the 2005 UK BCG vaccination policy change: A retrospective cohort study using the Enhanced Tuberculosis Surveillance system, 2000-2015 | 1st The epidemiology of tuberculosis, and the role of BCG vaccination, in England & Developing and parameterising a dynamic transmission model of tuberculosis |
| April | 1st All chapters excepting the last two chapters with Supervisors, 15th Final two chapters updated based on supervisor comments. | 1st Investigating the impact of the 2005 change in BCG vaccination policy using a fitted dynamic transmission model of TB & Estimating the effect of the 2005 UK BCG vaccination policy change: A retrospective cohort study using the Enhanced Tuberculosis Surveillance system, 2000-2015 |
| May | 1st Final hand in + viva organisation | |

1.1.9. Ejemplos: tablas con resúmenes estadísticos

| | Model 1 | Model 2 |
|---------------------|-------------------|-------------------|
| (Intercept) | 5.03*** (0.22) | |
| groupTrt | -0.37 (0.31) | 4.66*** (0.22) |
| groupCtl | | 5.03*** (0.22) |
| R ² | 0.07 | 0.98 |
| Adj. R ² | 0.02 | 0.98 |
| Num. obs. | 20 | 20 |
| RMSE | 0.70 | 0.70 |

*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$

Tabla 1.5: Two linear models.

Tabla 1.6

| Statistic | N | Mean | St. Dev. | Min | Pctl(25) | Pctl(75) | Max |
|------------|----|--------|----------|-----|----------|----------|-----|
| rating | 30 | 64.633 | 12.173 | 40 | 58.8 | 71.8 | 85 |
| complaints | 30 | 66.600 | 13.315 | 37 | 58.5 | 77 | 90 |
| privileges | 30 | 53.133 | 12.235 | 30 | 45 | 62.5 | 83 |
| learning | 30 | 56.367 | 11.737 | 34 | 47 | 66.8 | 75 |
| raises | 30 | 64.633 | 10.397 | 43 | 58.2 | 71 | 88 |
| critical | 30 | 74.767 | 9.895 | 49 | 69.2 | 80 | 92 |
| advance | 30 | 42.933 | 10.289 | 25 | 35 | 47.8 | 72 |

Tabla 1.7: Results

| | <i>Dependent variable:</i> | | |
|-------------------------|----------------------------|-----------------------------|---------------------|
| | rating | | high.rating |
| | <i>OLS</i> | | <i>probit</i> |
| | (1) | (2) | (3) |
| complaints | 0.692*** (0.149) | 0.682*** (0.129) | |
| privileges | -0.104 (0.135) | -0.103 (0.129) | |
| learning | 0.249 (0.160) | 0.238* (0.139) | 0.164*** (0.053) |
| raises | -0.033 (0.202) | | |
| critical | 0.015 (0.147) | | -0.001 (0.044) |
| advance | | | -0.062 (0.042) |
| Constant | 11.011 (11.704) | 11.258 (7.318) | -7.476** (3.570) |
| Observations | 30 | 30 | 30 |
| R ² | 0.715 | 0.715 | |
| Adjusted R ² | 0.656 | 0.682 | |
| Log Likelihood | | | -9.087 |
| Akaike Inf. Crit. | | | 26.175 |
| Residual Std. Error | 7.139 (df = 24) | 6.863 (df = 26) | |
| F Statistic | 12.063*** (df = 5; 24) | 21.743*** (df = 3; 26) | |
| <i>Note:</i> | | *p<0.1; **p<0.05; ***p<0.01 | |

1.1.10. Ejemplos: información matemática en forma de tabla

El *polinomio característico* $\chi(\lambda)$ de la matriz 3×3 :

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

viene dado por la fórmula:

$$\chi(\lambda) = \begin{vmatrix} \lambda - a & -b & -c \\ -d & \lambda - e & -f \\ -g & -h & \lambda - i \end{vmatrix}.$$

1.2. Introducción a la creación de tablas con R Markdown

Algunas veces la información ya está disponible en un formato digital reconocible por R, como en ficheros: csv, Excel, RData, etc. En esta situación, el primer paso será importar los datos con ayuda de las funciones que R tiene en sus librerías básicas o en algunas librerías especializadas.

La otra situación, será aquella en la que tendremos que escribir la información o de algún otro modo podamos obtener la información, bien de una página web, escaneando documentos, etc. En este último caso, se tendría que valorar si por la cantidad de información que se intenta presentar en forma de tabla, es más rápido escribirla manualmente, utilizar alguna librería que contenga funciones que faciliten la tarea de convertirla en formato digital, o convertirla en un fichero gráfico (por ejemplo, haciendo una captura de pantalla) e incorporarlo como un gráfico (ver ejemplo en el apartado 1.8)

Una vez esté disponible en R la información digitalizada, se tendrá que decidir de qué modo presentar la tabla:

- apaisada en la página,
- ocupa varias páginas,
- centrada en el texto o aparece justificada a la derecha,
- tiene celdas agrupadas,
- tiene celdas coloreadas,
- utiliza líneas como bordes de las celdas,
- el ancho y la justificación del texto dentro de las columnas,
- etc.

En R existen muchas librerías que se especializan en la creación de información en forma de tablas, algunas de las más conocidas son:

- [kableExtra](#),
- [huxtable](#),
- [pander](#),
- [xtable](#),
- [gt](#),
- [texreg](#) (especializada en resúmenes estadísticos),
- [stargazer](#) (especializada en resúmenes estadísticos),
- [tabulizer](#) (extrae tablas de ficheros pdf),
- [janitor](#) (crea resúmenes estadísticos sin formatear),
- [texPreview](#) (compila código LaTeX), etc.

También se hará referencia a otras librerías R que se han especializado en facilitar la tarea de escribir la información digitalmente para presentarla en formato de tablas, añadiendo miniherramientas a RStudio conocidos también como “Addins”.

1.3. Sitios web recomendados

- **Tablesgenerator:** <http://tablesgenerator.com>. Un servicio web que te permite escribir los datos en forma de tabla, generarlos en distintos formatos de salida y por último copiarlos al portapapeles, lo que permitirá incorporarlos en cualquier editor, en particular en RStudio.

Además, permite incorporar datos desde un fichero csv, desde el portapapeles e **incluso de una tabla LaTeX creada previamente** (con algunas limitaciones), utilizando los menús disponibles.

También es posible seleccionar la tabla completa, o algunas celdas (con ayuda de la tecla “Ctrl” y haciendo click en las celdas o columnas/filas), y luego copiarla con “Ctrl+C”. La selección se puede pegar como texto separado por tabulaciones en cualquier aplicación (por ejemplo, en Excel o en un fichero de texto y guardarlo como csv, etc.).

Hay varios formatos de salida disponibles, los más usados son:

- Tablas LaTeX: http://tablesgenerator.com/latex_tables.
- Tablas html: http://tablesgenerator.com/html_tables
- Tablas markdown: http://tablesgenerator.com/markdown_tables

1.4. No se dispone de la información en formato digital, ¿cómo la añado?

1.4.1. Utilizando código markdown

Es muy sencillo construir una tabla básica con código markdown, tan solo tendremos que escribir el contenido, teniendo en cuenta lo siguiente:

- En cada fila, las distintas celdas se separan en columnas, con ayuda del código “|”.
- Cada fila de la tabla se coloca en una línea diferente utilizando un salto de línea (pulsar la tecla “Enter”).
- Para incluir líneas, que separen algunas filas (por ejemplo la de la cabecera), se utiliza varios códigos de guión: -----. Hay que tener en cuenta, que también es necesario incluir los separadores de columnas: -----|-----|-----.
- Se pueden usar los dos puntos “:” para alinear el contenido de las columnas en la línea inferior de la cabecera. De forma que:
 - |:----| ajustará el contenido a la izquierda (comportamiento por defecto).
 - |----:| ajustará el contenido a la derecha.
 - |:---:| ajustará el contenido al centro.

Por ejemplo con el siguiente código markdown:

```
| Edad | Nacimientos | Mujeres | TFR por 1000 |
|-----|-----|-----|-----|
| 20 | 20 | 100 | 200 |
| 21 | 25 | 100 | 250 |
| 22 | 30 | 100 | 300 |
| 23 | 35 | 100 | 350 |
| 24 | 40 | 100 | 400 |
| Total | 150 | 500 | 1500 |
| 20-24 | 150 | 500 | 300 |
```

se obtiene el siguiente resultado al generar el fichero de salida (“Knit” el documento Rmd):

| Edad | Nacimientos | Mujeres | TFR por 1000 |
|-------|-------------|---------|--------------|
| 20 | 20 | 100 | 200 |
| 21 | 25 | 100 | 250 |
| 22 | 30 | 100 | 300 |
| 23 | 35 | 100 | 350 |
| 24 | 40 | 100 | 400 |
| Total | 150 | 500 | 1500 |
| 20-24 | 150 | 500 | 300 |

Como puede apreciarse, la primera fila o cabecera de la tabla (nombres o descripción del contenido de cada columna) aparece entre dos líneas, y el final de la tabla termina con

otra línea. La información aparece en columnas ajustadas a la izquierda. Este modo de presentar información en forma de tabla es un estándar.

En este otro ejemplo, se presenta la misma información, pero como puede verse se han dejado más espacios entre cada celda para ver claramente a qué columna pertenece. El resultado es el mismo.

| Edad | Nacimientos | Mujeres | TFR por 1000 |
|-------|-------------|---------|--------------|
| 20 | 20 | 100 | 200 |
| 21 | 25 | 100 | 250 |
| 22 | 30 | 100 | 300 |
| 23 | 35 | 100 | 350 |
| 24 | 40 | 100 | 400 |
| Total | 150 | 500 | 1500 |
| 20-24 | 150 | 500 | 300 |

| Edad | Nacimientos | Mujeres | TFR por 1000 |
|-------|-------------|---------|--------------|
| 20 | 20 | 100 | 200 |
| 21 | 25 | 100 | 250 |
| 22 | 30 | 100 | 300 |
| 23 | 35 | 100 | 350 |
| 24 | 40 | 100 | 400 |
| Total | 150 | 500 | 1500 |
| 20-24 | 150 | 500 | 300 |

En este otro ejemplo, se usa el código “:” para justificar de forma diferente las columnas, como se indicó anteriormente:

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |

Este es otro ejemplo básico de construcción de una tabla:

```
| a| b| c|
|---:|---:|---:|
| 1| 2| 3|
| 2| 3| 4|
```

| a | b | c |
|---|---|---|
| 1 | 2 | 3 |
| 2 | 3 | 4 |

Nota importante. Cuando se utiliza código markdown para escribir tablas en formato “pdf” no se pueden construir sin una fila inicial de cabecera.

1.4.1.1. Una tabla sin fila de cabecera

En los siguientes ejemplos, se puede ver como introducir tablas sin cabecera. En la salida pdf, no queda bien resuelto como puede apreciarse (doble línea sin contenido). Luego, si nuestra tabla no tiene cabecera, se recomienda no utilizar código markdown.

```
[] ()      |      |      |      |      |      |
-----:   | ----: | ----: | ----: | ----: | ----: |
Demanda:   | 150  | 175  | 200  | 225  | 250  |
Probabilidad: | 0.10 | 0.30 | 0.40 | 0.15 | 0.05 |
>[] ()      |      |      |      |      |      |
```

| | | | | | |
|---------------|------|------|------|------|------|
| Demanda: | 150 | 175 | 200 | 225 | 250 |
| Probabilidad: | 0.10 | 0.30 | 0.40 | 0.15 | 0.05 |

```
<span> |
---    |    ---
Value  | Value
Value  | Value
```

| | |
|-------|-------|
| Value | Value |
| Value | Value |

1.4.2. Utilizando código R

En esta situación, se puede escribir el contenido de la tabla en un objeto R, tipo “data.frame” del sistema base, o los objetos “tibble” definidos en la librería “dplyr” o “tibble”.

En el primer ejemplo que se presenta se almacena la información en un objeto “data.frame”.

```
tb01 = data.frame(stringsAsFactors=FALSE,
  Posiciones = c("Fabricación","Pintura","Subensamblaje1",
    "Subensamblaje2","Ensamblaje1"),
  CoordenadaX = c(1,1,2.5,3,4),
  CoordenadaY = c(4,2,2,5,4),
  Demanda = c(12,24,13,7,17)
)
```

Vemos el contenido del objeto “data.frame” creado, al que se ha llamado: “tb01”:

```
##      Posiciones CoordenadaX CoordenadaY Demanda
## 1  Fabricación          1.0           4      12
## 2      Pintura          1.0           2      24
## 3 Subensamblaje1        2.5           2      13
## 4 Subensamblaje2        3.0           5       7
## 5   Ensamblaje1        4.0           4      17
```

Si queremos que nuestra tabla no tenga fila de cabecera:

```
tb01 = data.frame(stringsAsFactors=FALSE,
  Posiciones = c("Fabricacion","Pintura","Subensamblaje1",
    "Subensamblaje2","Ensamblaje1"),
  CoordenadaX = c(1,1,2.5,3,4),
  CoordenadaY = c(4,2,2,5,4),
  Demanda = c(12,24,13,7,17)
)
colnames(tb01) = rep(NA,4)
tb01
```

```
##      NA NA NA NA
## 1  Fabricacion 1.0 4 12
## 2      Pintura 1.0 2 24
## 3 Subensamblaje1 2.5 2 13
## 4 Subensamblaje2 3.0 5 7
## 5   Ensamblaje1 4.0 4 17
```

1.4.2.1. Uso de objetos tibble o tribble

También se puede crear un objeto “tibble” (recomendado por el sistema “tidyverse”) con la información que necesitemos. Entre las características que posee este objeto, destacamos: que no convierte por defecto cadenas de caracteres a objetos tipo “factor” (cosa que sí ocurre con los “data.frame”), pueden asignarse nombres a las columnas con caracteres como espacios, acentos, etc.

```
library(dplyr) # library(tibble)
tb01 = tibble(
  X = c("Partly cloudy.", "Partly cloudy.", "Possible shower.",
        "Partly cloudy.", "Shower or two. Possible storm.",
        "Possible shower.", "Partly cloudy.", "Mostly sunny.",
        "Partly cloudy.", "Possible shower.", "Partly cloudy."),
  Location = c("Brisbane", "Brisbane Airport", "Beaudesert", "Chermside",
              "Gatton", "Ipswich", "Logan Central", "Manly",
              "Mount Gravatt", "Oxley", "Redcliffe"),
  Min = c(19, 18, 15, 17, 15, 15, 18, 20, 17, 17, 19),
  Max = c(29, 27, 30, 29, 32, 30, 29, 26, 28, 30, 27)
)
```

```
tb01
```

```
## # A tibble: 11 x 4
##   X                Location      Min   Max
##   <chr>            <chr>      <dbl> <dbl>
## 1 Partly cloudy.   Brisbane    19     29
## 2 Partly cloudy.   Brisbane Airport 18     27
## 3 Possible shower. Beaudesert  15     30
## 4 Partly cloudy.   Chermside   17     29
## 5 Shower or two. Possible storm. Gatton      15     32
## 6 Possible shower. Ipswich     15     30
## 7 Partly cloudy.   Logan Central 18     29
## 8 Mostly sunny.    Manly        20     26
## 9 Partly cloudy.   Mount Gravatt 17     28
## 10 Possible shower. Oxley        17     30
## 11 Partly cloudy.  Redcliffe    19     27
```

En este otro ejemplo, se usa la función `tribble()` para introducir los mismos datos que en el ejemplo anterior, pero se introducen por filas (de ahí la “r” de la función “`tribble()`”). Como se verá más adelante, un Addins de la librería “datapasta” permite pegar los datos en el documento R Markdown como “tribble”.

```
tb01b = tibble::tribble(
  ~X, ~Location, ~Min, ~Max,
  "Partly cloudy.", "Brisbane", 19, 29,
  "Partly cloudy.", "Brisbane Airport", 18, 27,
  "Possible shower.", "Beaudesert", 15, 30,
  "Partly cloudy.", "Chermside", 17, 29,
  "Shower or two. Possible storm.", "Gatton", 15, 32,
  "Possible shower.", "Ipswich", 15, 30,
  "Partly cloudy.", "Logan Central", 18, 29,
  "Mostly sunny.", "Manly", 20, 26,
  "Partly cloudy.", "Mount Gravatt", 17, 28,
  "Possible shower.", "Oxley", 17, 30,
  "Partly cloudy.", "Redcliffe", 19, 27
)
```

```
tb01b
```

```
## # A tibble: 11 x 4
##   X                Location      Min   Max
##   <chr>            <chr>      <dbl> <dbl>
## 1 Partly cloudy.   Brisbane    19    29
## 2 Partly cloudy.   Brisbane Airport 18    27
## 3 Possible shower. Beaudesert  15    30
## 4 Partly cloudy.   Chermside   17    29
## 5 Shower or two. Possible storm. Gatton     15    32
## 6 Possible shower. Ipswich     15    30
## 7 Partly cloudy.   Logan Central 18    29
## 8 Mostly sunny.    Manly       20    26
## 9 Partly cloudy.   Mount Gravatt 17    28
## 10 Possible shower. Oxley      17    30
## 11 Partly cloudy.   Redcliffe   19    27
```

También se podrían introducir los datos fila a fila con ayuda de la función `add_row()` del paquete “dplyr” (con el operador encadenamiento `%>%` habitual en el sistema “tidyverse”), del siguiente modo:

```
library(dplyr) # library(tibble)
tb01b2 = tibble(
  X = "Partly cloudy.",
  Location = "Brisbane",
  Min = 19,
  Max = 29
) %>%
add_row(
  X = "Partly cloudy.",
  Location = "Brisbane Airport",
  Min = 18,
  Max = 27
) %>%
add_row(
  X = "Possible shower.",
  Location = "Beaudesert",
  Min = 15,
  Max = 30
) %>%
add_row(
  X = "Partly cloudy.",
  Location = "Chermside",
  Min = 17,
  Max = 29
) %>%
add_row(
  X = "Shower or two. Possible storm.",
  Location = "Gatton",
```

```

    Min =      15,
    Max =      32
  )

```

```
tb01b2
```

```

## # A tibble: 5 x 4
##   X                Location      Min   Max
##   <chr>            <chr>      <dbl> <dbl>
## 1 Partly cloudy.   Brisbane    19    29
## 2 Partly cloudy.   Brisbane Airport 18    27
## 3 Possible shower. Beaudesert  15    30
## 4 Partly cloudy.   Chermside   17    29
## 5 Shower or two. Possible storm. Gatton     15    32

```

Este modo de introducir la información puede ser útil cuando haya celdas que contengan mucho texto (ver ejemplo del apartado 1.1.8), ya que sería fácil seguir la fila que se está introduciendo (con “tribble” al haber columnas con celdas de ancho diferente, sería más difícil de seguir).

Los mismos datos guardados en un data.frame:

```

tb01c = data.frame(stringsAsFactors=FALSE,
  X = c("Partly cloudy.", "Partly cloudy.", "Possible shower.",
        "Partly cloudy.", "Shower or two. Possible storm.",
        "Possible shower.", "Partly cloudy.", "Mostly sunny.",
        "Partly cloudy.", "Possible shower.", "Partly cloudy."),
  Location = c("Brisbane", "Brisbane Airport", "Beaudesert", "Chermside",
              "Gatton", "Ipswich", "Logan Central", "Manly",
              "Mount Gravatt", "Oxley", "Redcliffe"),
  Min = c(19, 18, 15, 17, 15, 15, 18, 20, 17, 17, 19),
  Max = c(29, 27, 30, 29, 32, 30, 29, 26, 28, 30, 27)
)

```

```
tb01c
```

```

##           X      Location Min Max
## 1      Partly cloudy.   Brisbane 19 29
## 2      Partly cloudy. Brisbane Airport 18 27
## 3      Possible shower.   Beaudesert 15 30
## 4      Partly cloudy.   Chermside 17 29
## 5 Shower or two. Possible storm.   Gatton 15 32
## 6      Possible shower.   Ipswich 15 30
## 7      Partly cloudy.   Logan Central 18 29
## 8      Mostly sunny.     Manly 20 26
## 9      Partly cloudy.   Mount Gravatt 17 28
## 10     Possible shower.   Oxley 17 30
## 11     Partly cloudy.   Redcliffe 19 27

```

1.4.3. Utilizando LaTeX

Consultar la ayuda en [Wikipedia sobre tablas con LaTeX](#).

Consejo LaTeX 1. Algunas veces en R Markdown es conveniente utilizar comandos LaTeX para forzar a utilizar tamaños de letras más pequeños o más grandes. A continuación, se enumeran los comandos para selección de tamaños de fuentes desde el tamaño más pequeño al mayor:

```
\tiny
\scriptsize
\footnotesize
\small
\normalsize
\large
\Large
\LARGE
\huge
\Huge
```

Consejo LaTeX 2. En algunas ocasiones es conveniente forzar una pequeña separación vertical entre elementos, y para ello pueden usarse en R Markdown los siguientes comandos LaTeX de menor a mayor espacio vertical:

```
\medskip
\bigskip
```

También es posible introducir información en formato tabla, como puede verse en los siguientes ejemplos, con ayuda de un entorno LaTeX directamente:

- `$$\begin{array}{l} \dots \end{array}$$`.
- `\begin{center}\begin{tabular}{l} \dots \end{tabular}\end{center}`.

Básicamente se puede crear una tabla LaTeX teniendo en cuenta lo siguiente:

- Entre las llaves que siguen a “array” o “tabular” se indican las columnas que se van a necesitar, colocando por cada columna una: “c” (columna justificada de forma centrada), “l” (justificada a la izquierda), “r” (justificada a la derecha) o “p{2cm}” (justificada a la izquierda, pero con el ancho de columna que indiquemos). Además se pueden añadir | para indicar que hay líneas verticales en esa columna.
- Para separar las celdas dentro de cada columna se emplea el signo &.
- Para indicar que acabamos una fila se utilizan \\.
- Con el comando `\hline` se traza una línea horizontal del ancho de la tabla en la posición en la que se haya escrito. **Nota.** El comando `\cline{1-3}` permite trazar una línea horizontal bajo las columnas 1 a 3.
- Y con el comando `\multicolumn{numcol}{formatocolumnas}{Texto}` se pueden agrupar celdas continuas en una misma fila, al indicar respectivamente: número de columnas que agrupamos, el justificado de la columna agrupada (cualquiera de los señalados anteriormente), y el texto que se colocará en la celda agrupada.

El siguiente ejemplo de código LaTeX, que solamente pretende utilizar las posibilidades que se acaban de explicar

```
\begin{center}
\begin{tabular}{|p{2cm}|c|l|r|}
\hline
Edad      & & Nacimientos      & & Mujeres      & & TFR por 1000      & & \\
20        & & 20                & & 100          & & 200               & & \\
21        & & 25                & & 100          & & 250               & & \\
22        & & 30                & & 100          & & 300               & & \\
23        & & 35                & & 100          & & 350               & & \\
24        & & 40                & & 100          & & 400               & & \\
\multicolumn{1}{|c|}{Total} & & & & & & & & \\
\multicolumn{1}{r|}{150} & & & & & & & & \\
\multicolumn{1}{c|}{500} & & & & & & & & \\
\multicolumn{1}{c|}{1500} & & & & & & & & \\
\hline
20-24     & & 150               & & 500          & & 300               & & \\
\end{tabular}
\end{center}
```

produce el siguiente resultado:

| Edad | Nacimientos | Mujeres | TFR por 1000 |
|-------|-------------|---------|--------------|
| 20 | 20 | 100 | 200 |
| 21 | 25 | 100 | 250 |
| 22 | 30 | 100 | 300 |
| 23 | 35 | 100 | 350 |
| 24 | 40 | 100 | 400 |
| Total | 150 | 500 | 1500 |
| 20-24 | 150 | 500 | 300 |

Consejo LaTeX 3. También puede resultar muy útil incluir alguno de los siguientes comandos LaTeX para producir un salto de página en documentos Rmd con salida latex-pdf:

```
\newpage
\clearpage
```

Se recomienda el uso de `\clearpage`.

Se suelen utilizar una vez que ya se ha escrito el documento completo y se quiere ajustar la presentación del contenido final. Por ejemplo, si el título de un apartado aparece con unas pocas líneas al final de una página, se puede decidir introducir justo antes del título un salto de página, con `\clearpage`, para pasarlo a la siguiente página. Esto puede provocar un efecto en cadena sobre las siguientes páginas, que habrá que ajustar avanzando en el documento introduciendo nuevos saltos de página.

También es posible introducir información matemática en formato tabla, como puede verse en los siguientes ejemplos, con ayuda del entorno LaTeX: `$$\begin{array}{...} \end{array}$$`. **Nota:** RStudio, en sus últimas versiones, nos muestra la expresión matemática justo a continuación del lugar donde se haya escrito la segunda pareja de `$$`.

El `\emph{polinomio característico}` $\chi(\lambda)$ de la matriz `3 \times 3`:

```

$$
\left( \begin{array}{ccc}
a & b & c \\
d & e & f \\
g & h & i \end{array} \right)
$$

```

viene dado por la fórmula:

```

$$
\chi(\lambda) = \left| \begin{array}{ccc}
\lambda - a & -b & -c \\
-d & \lambda - e & -f \\
-g & -h & \lambda - i \end{array} \right|.
$$

```

El resultado del código LaTeX anterior es el siguiente:

El *polinomio característico* $\chi(\lambda)$ de la matriz 3×3 :

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

viene dado por la fórmula:

$$\chi(\lambda) = \begin{vmatrix} \lambda - a & -b & -c \\ -d & \lambda - e & -f \\ -g & -h & \lambda - i \end{vmatrix}.$$

1.5. Consejos sobre RStudio y los Addins

Los “Addins de RStudio” son miniherramientas que se añaden a RStudio, en la barra de herramientas como un desplegable, llamado: “Addins”, que facilitan las tareas más habituales.

Consejo 1. El editor de RStudio nos permite seleccionar texto de dos modos principalmente:

- selección de texto en modo fila (el habitual)
- selección de texto en modo columna, el cual se activa presionando la tecla “Alt” (el cursor se convertirá en una “cruz”) y moviendo el ratón sobre las columnas que se quieran seleccionar.

También es aconsejable el uso de la tecla de “Tabulación” para indentar-separar la información en distintas columnas y la combinación de teclas “Mayúsculas+Tabulación” para eliminar la indentación. Las dos opciones se pueden utilizar también para indentar o desindentar el texto seleccionado.

Consejo 2. También es interesante utilizar identificadores en los trozos de código R (RStudio utiliza el término inglés: “**chunk**”) que usemos para presentar las tablas, ya que RStudio en la parte inferior izquierda del editor nos permite navegar por todos los “chunks” bien utilizando su identificador o un número que corresponde a la ordenación de arriba a abajo en el texto.

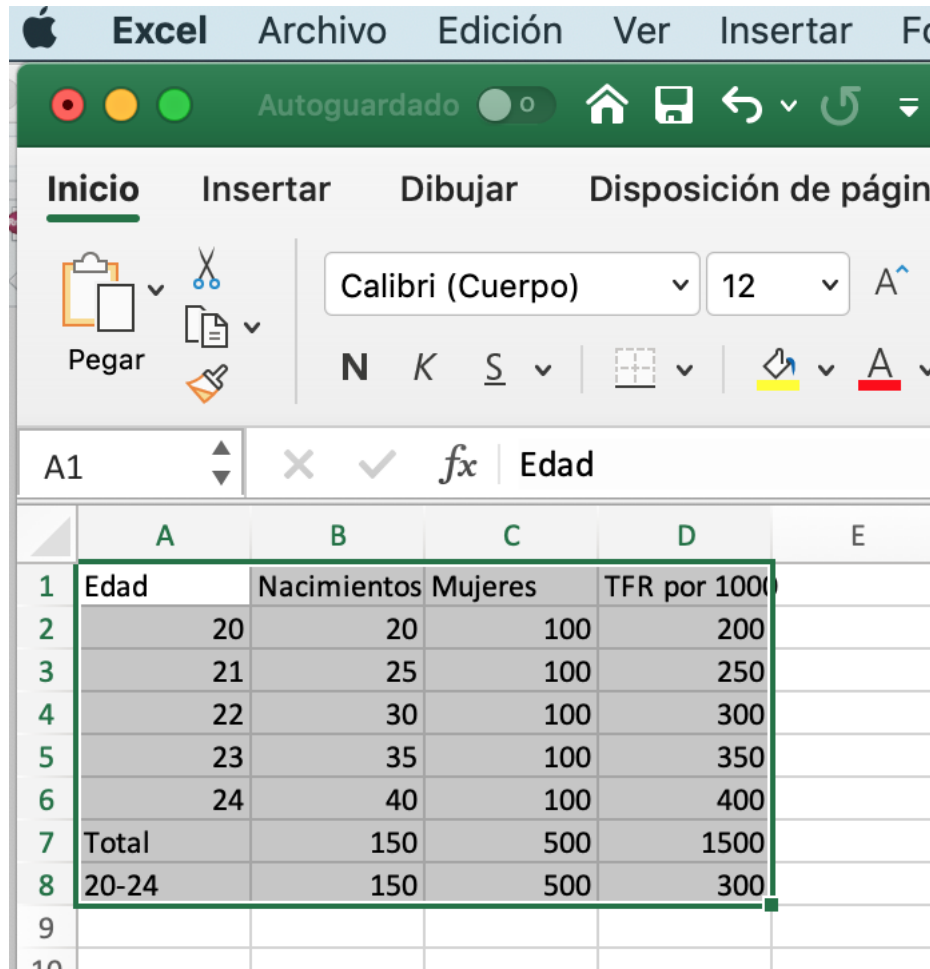
En el siguiente ejemplo se ha usado como identificador “tabla01”.

```
```${r tabla01}
head(iris,7)
```
```

1.5.1. Addins de la librería datapasta

Esta librería tiene una serie de addins que nos permite utilizar la información que se encuentre en el portapapeles (por ejemplo, unos datos copiados de una selección en Excel), la escribe en el editor de RStudio convirtiéndola en el código de un objeto R del tipo:

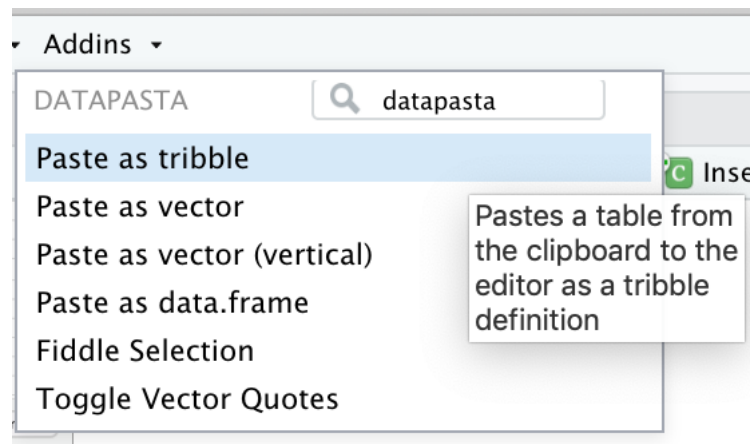
- `tribble`
- `vector`
- `vector (vertical)`
- `data.frame`



| | A | B | C | D | E |
|----|-------|-------------|---------|--------------|---|
| 1 | Edad | Nacimientos | Mujeres | TFR por 1000 | |
| 2 | 20 | 20 | 100 | 200 | |
| 3 | 21 | 25 | 100 | 250 | |
| 4 | 22 | 30 | 100 | 300 | |
| 5 | 23 | 35 | 100 | 350 | |
| 6 | 24 | 40 | 100 | 400 | |
| 7 | Total | 150 | 500 | 1500 | |
| 8 | 20-24 | 150 | 500 | 300 | |
| 9 | | | | | |
| 10 | | | | | |

Consejo. Un objeto R (“matrix”, “data.frame”, “tibble”, etc) podría pegarse al portapapeles con ayuda de la función `clipr::write_clip()`, de la librería “clipr”. Por ejemplo, se podría copiar parte de un `data.frame` en el portapapeles `clipr::write_clip(mtcars[1:10,1:4])` y luego pegarlo con los addins de la librería “datapasta” para poder modificarlos en el código R más cómodamente.

Podemos pegarlos en RStudio con ayuda del addins “Paste as tribble”, para obtener:



```
tibble::tribble(
  ~Edad, ~Nacimientos, ~Mujeres, ~TFR.por.1000,
  "20",      20,      100,      200,
  "21",      25,      100,      250,
  "22",      30,      100,      300,
  "23",      35,      100,      350,
  "24",      40,      100,      400,
  "Total",   150,     500,     1500,
  "20-24",   150,     500,      300
)
```

Podríamos ponerle un nombre (“tb01”) y presentarlo en formato tabla, por ejemplo con ayuda de la función `kable()` del paquete “knitr”:

```
tb01 = tibble::tribble(
  ~Edad, ~Nacimientos, ~Mujeres, ~TFR.por.1000,
  "20",      20,      100,      200,
  "21",      25,      100,      250,
  "22",      30,      100,      300,
  "23",      35,      100,      350,
  "24",      40,      100,      400,
  "Total",   150,     500,     1500,
  "20-24",   150,     500,      300
)

knitr::kable(tb01)
```

| Edad | Nacimientos | Mujeres | TFR.por.1000 |
|-------|-------------|---------|--------------|
| 20 | 20 | 100 | 200 |
| 21 | 25 | 100 | 250 |
| 22 | 30 | 100 | 300 |
| 23 | 35 | 100 | 350 |
| 24 | 40 | 100 | 400 |
| Total | 150 | 500 | 1500 |
| 20-24 | 150 | 500 | 300 |

Si hubiésemos seleccionado “Paste as data.frame” el texto que se obtendría en RStudio sería:

```
data.frame(stringsAsFactors=FALSE,
  Edad = c("20", "21", "22", "23", "24", "Total", "20-24"),
  Nacimientos = c(20, 25, 30, 35, 40, 150, 150),
  Mujeres = c(100, 100, 100, 100, 100, 500, 500),
  TFR.por.1000 = c(200, 250, 300, 350, 400, 1500, 300)
)
```

Los siguientes dos addins de “datapasta” tienen una utilidad diferente:

- **Fiddle Selection**: coge la selección actual e intenta convertirlo a algo mejor formateado (si es posible): texto a un vector, tribble o data.frame mejora la indentación en la que se presentan.

Por ejemplo, el siguiente objeto `tribble` de R, al seleccionarlo

```
tibble::tribble(
  ~Edad, ~Nacimientos, ~Mujeres, ~TFR.por.1000,
  "20", 20, 100, 200,
  "21", 25, 100, 250,
  "22", 30, 100,300,
  "23", 35, 100, 350,
  "24", 40, 100, 400,
  "Total", 150, 500, 1500,
  "20-24", 150, 500, 300
)
```

Al elegir “Fiddle Selection”, lo convertiría en texto más fácil de ver:

```
tibble::tribble(
  ~Edad, ~Nacimientos, ~Mujeres, ~TFR.por.1000,
  "20", 20, 100, 200,
  "21", 25, 100, 250,
  "22", 30, 100, 300,
  "23", 35, 100, 350,
  "24", 40, 100, 400,
  "Total", 150, 500, 1500,
  "20-24", 150, 500, 300
)
```

- **Toggle Vector Quotes**: sobre un vector R de elementos seleccionado, alterna o bien colocando cada uno de los elementos entre dobles comillas, o bien eliminando las dobles comillas de todos los elementos.

```
c(32,87,98,3,23)
```

```
c("32","87","98","3","23")
```

```
c(32,87,98,3,23)
```

1.5.2. Addins de la librería inserttable

El addins “Insert Table” que contiene esta librería nos permite escribir el texto de la tabla usando una pequeña hoja de cálculo (es posible también pegarlo desde una hoja Excel), en la que podemos insertar/borrar filas/columnas. Una vez introducido todo el texto, automáticamente añade una tabla al documento R Markdown, con el formato de salida que hayamos seleccionado. Los formatos de salida están relacionados con los siguientes paquetes R:

- kable
- kableExtra (html o pdf)
- DT
- rhandsontable
- NONE

Desde un fichero “Rmd” y dentro de RStudio, se coloca el cursor sobre una línea vacía o se selecciona el nombre de un data.frame dentro de un “trozo de código R”, luego se hace click sobre “Addins” y seleccionamos “Insert Table”.

En el siguiente ejemplo, en un trozo de código R teníamos el cursor en una línea vacía, y llamamos al addins “Insert Table”. Editamos la tabla del addins y pulsamos sobre “Done”, obteniendo el siguiente código:

Insert Table Add-In

Cancel Select output format and edit the Table if you wish so Done

Select Table Name Select Output Format

my_tbl kable

Edit Table or cut and paste from spreadsheet

* The first row will be used as column names.
* Right click to add more lines or columns

Use first row as column names. (If unchecked, 'Col_1', 'Col_2', etc. are used)

| | | | |
|---|---|----|-----|
| 1 | 3 | 34 | 5.5 |
| 2 | 2 | 43 | 3.5 |
| 3 | 1 | 53 | 2.2 |
| 4 | 4 | 22 | 1.1 |

```
my_tbl <- tibble::tribble(  
  ~Col_1, ~Col_2, ~Col_3,  
    3,     34,    5.5,  
    2,     43,    3.5,  
    1,     53,    2.2,  
    4,     22,    1.1  
)  
  
require(knitr)  
kable(my_tbl, digits = 3, row.names = FALSE, align = "c",  
      caption = NULL)
```

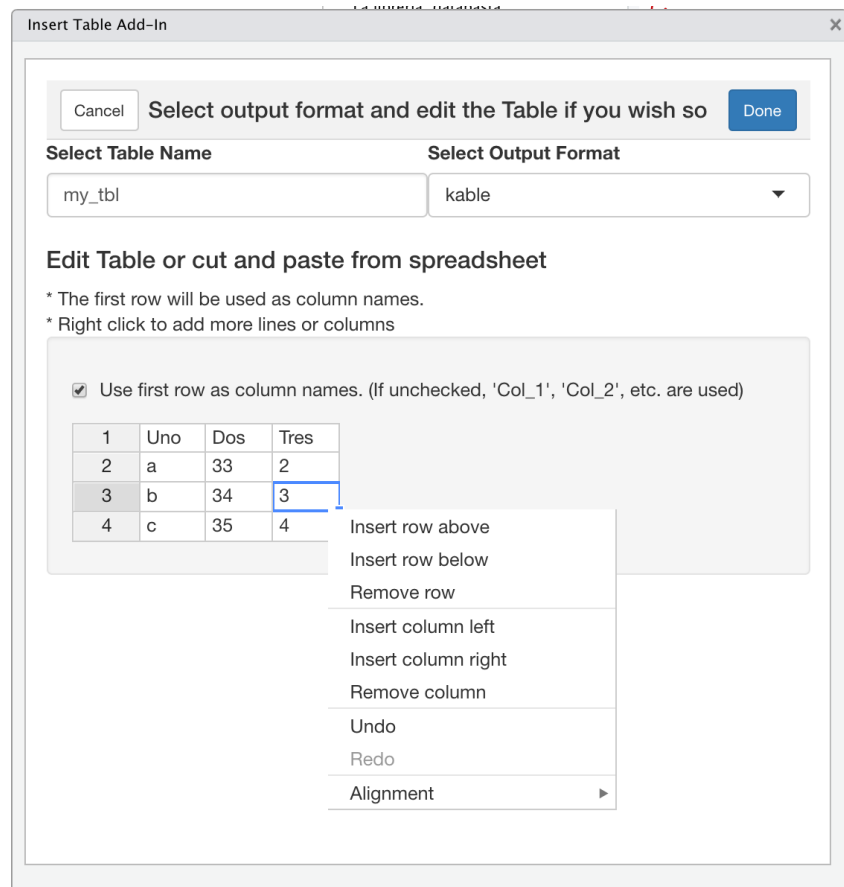
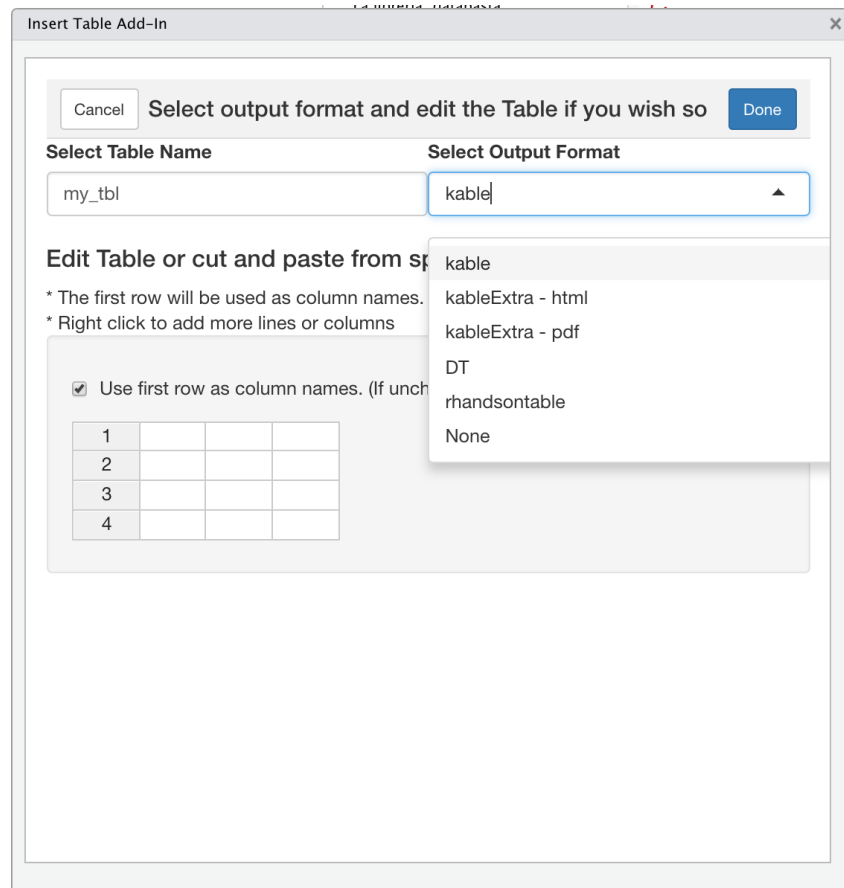
| Col_1 | Col_2 | Col_3 |
|-------|-------|-------|
| 3 | 34 | 5.5 |
| 2 | 43 | 3.5 |
| 1 | 53 | 2.2 |
| 4 | 22 | 1.1 |

El siguiente código se generó sobre el objeto “mtcars” que se tenía seleccionado en un “trozo de código R”:

```
require(knitr)
kable(mtcars, digits = 3, row.names = FALSE, align = "c",
      caption = NULL)
```

| mpg | cyl | disp | hp | drat | wt | qsec | vs | am | gear | carb |
|------|-----|-------|-----|------|-------|-------|----|----|------|------|
| 21.0 | 6 | 160.0 | 110 | 3.90 | 2.620 | 16.46 | 0 | 1 | 4 | 4 |
| 21.0 | 6 | 160.0 | 110 | 3.90 | 2.875 | 17.02 | 0 | 1 | 4 | 4 |
| 22.8 | 4 | 108.0 | 93 | 3.85 | 2.320 | 18.61 | 1 | 1 | 4 | 1 |
| 21.4 | 6 | 258.0 | 110 | 3.08 | 3.215 | 19.44 | 1 | 0 | 3 | 1 |
| 18.7 | 8 | 360.0 | 175 | 3.15 | 3.440 | 17.02 | 0 | 0 | 3 | 2 |
| 18.1 | 6 | 225.0 | 105 | 2.76 | 3.460 | 20.22 | 1 | 0 | 3 | 1 |
| 14.3 | 8 | 360.0 | 245 | 3.21 | 3.570 | 15.84 | 0 | 0 | 3 | 4 |
| 24.4 | 4 | 146.7 | 62 | 3.69 | 3.190 | 20.00 | 1 | 0 | 4 | 2 |
| 22.8 | 4 | 140.8 | 95 | 3.92 | 3.150 | 22.90 | 1 | 0 | 4 | 2 |
| 19.2 | 6 | 167.6 | 123 | 3.92 | 3.440 | 18.30 | 1 | 0 | 4 | 4 |
| 17.8 | 6 | 167.6 | 123 | 3.92 | 3.440 | 18.90 | 1 | 0 | 4 | 4 |
| 16.4 | 8 | 275.8 | 180 | 3.07 | 4.070 | 17.40 | 0 | 0 | 3 | 3 |
| 17.3 | 8 | 275.8 | 180 | 3.07 | 3.730 | 17.60 | 0 | 0 | 3 | 3 |
| 15.2 | 8 | 275.8 | 180 | 3.07 | 3.780 | 18.00 | 0 | 0 | 3 | 3 |
| 10.4 | 8 | 472.0 | 205 | 2.93 | 5.250 | 17.98 | 0 | 0 | 3 | 4 |
| 10.4 | 8 | 460.0 | 215 | 3.00 | 5.424 | 17.82 | 0 | 0 | 3 | 4 |
| 14.7 | 8 | 440.0 | 230 | 3.23 | 5.345 | 17.42 | 0 | 0 | 3 | 4 |
| 32.4 | 4 | 78.7 | 66 | 4.08 | 2.200 | 19.47 | 1 | 1 | 4 | 1 |
| 30.4 | 4 | 75.7 | 52 | 4.93 | 1.615 | 18.52 | 1 | 1 | 4 | 2 |
| 33.9 | 4 | 71.1 | 65 | 4.22 | 1.835 | 19.90 | 1 | 1 | 4 | 1 |
| 21.5 | 4 | 120.1 | 97 | 3.70 | 2.465 | 20.01 | 1 | 0 | 3 | 1 |
| 15.5 | 8 | 318.0 | 150 | 2.76 | 3.520 | 16.87 | 0 | 0 | 3 | 2 |
| 15.2 | 8 | 304.0 | 150 | 3.15 | 3.435 | 17.30 | 0 | 0 | 3 | 2 |
| 13.3 | 8 | 350.0 | 245 | 3.73 | 3.840 | 15.41 | 0 | 0 | 3 | 4 |
| 19.2 | 8 | 400.0 | 175 | 3.08 | 3.845 | 17.05 | 0 | 0 | 3 | 2 |
| 27.3 | 4 | 79.0 | 66 | 4.08 | 1.935 | 18.90 | 1 | 1 | 4 | 1 |
| 26.0 | 4 | 120.3 | 91 | 4.43 | 2.140 | 16.70 | 0 | 1 | 5 | 2 |
| 30.4 | 4 | 95.1 | 113 | 3.77 | 1.513 | 16.90 | 1 | 1 | 5 | 2 |
| 15.8 | 8 | 351.0 | 264 | 4.22 | 3.170 | 14.50 | 0 | 1 | 5 | 4 |
| 19.7 | 6 | 145.0 | 175 | 3.62 | 2.770 | 15.50 | 0 | 1 | 5 | 6 |
| 15.0 | 8 | 301.0 | 335 | 3.54 | 3.570 | 14.60 | 0 | 1 | 5 | 8 |
| 21.4 | 4 | 121.0 | 109 | 4.11 | 2.780 | 18.60 | 1 | 1 | 4 | 2 |

En las siguientes dos capturas puede verse algunas de las opciones disponibles:



1.5.3. Addins de la librería remedy

La librería “remedy” tiene un addins llamado “Table” que permite insertar rápidamente una tabla en formato markdown.

```
| a| b| c|
|---:|---:|---:|
| 1| 2| 3|
| 2| 3| 4|
```

Si tenemos seleccionado previamente un objeto R (data.frame, tribble) lo convierte a una tabla en formato markdown:

```
mtcars[1:10,1:5]
```

Lo convertiría a:

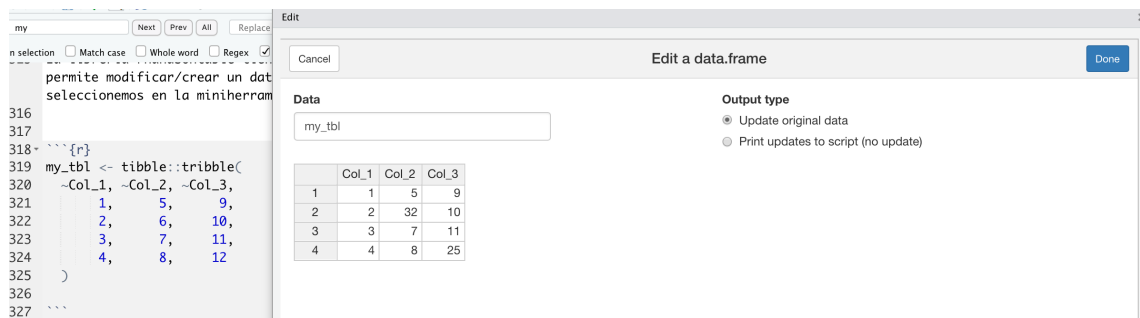
```
|          | mpg| cyl|  disp|  hp| drat|
|:-----|----:|---:|-----:|---:|----:|
|Mazda RX4      | 21.0|  6| 160.0| 110| 3.90|
|Mazda RX4 Wag  | 21.0|  6| 160.0| 110| 3.90|
|Datsun 710     | 22.8|  4| 108.0|  93| 3.85|
|Hornet 4 Drive | 21.4|  6| 258.0| 110| 3.08|
|Hornet Sportabout | 18.7|  8| 360.0| 175| 3.15|
|Valiant        | 18.1|  6| 225.0| 105| 2.76|
|Duster 360     | 14.3|  8| 360.0| 245| 3.21|
|Merc 240D      | 24.4|  4| 146.7|  62| 3.69|
|Merc 230       | 22.8|  4| 140.8|  95| 3.92|
|Merc 280       | 19.2|  6| 167.6| 123| 3.92|
```

1.5.4. Addins de la librería rhandsontable

La librería rhandsontable tiene un addin “Edit a data.frame” que permite modificar/crear un data.frame que tengamos seleccionado o que seleccionemos en la miniherramienta que aparece.

```
my_tbl <- tibble::tribble(
  ~Col_1, ~Col_2, ~Col_3,
    1,      5,      9,
    2,      6,     10,
    3,      7,     11,
    4,      8,     12
)
```

En la siguiente captura se puede ver que algunas celdas se han modificado (también se podrían haber añadido nuevas filas o columnas, con ayuda del submenú que se activa al pulsar el botón derecho):



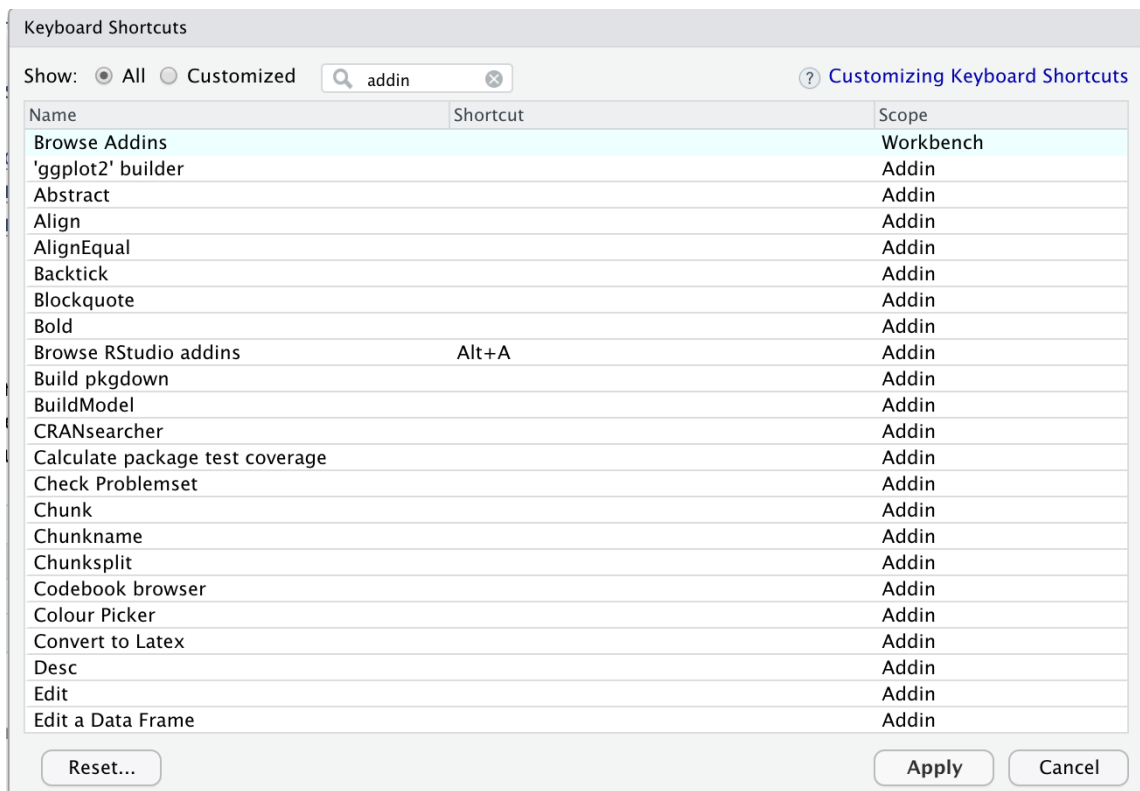
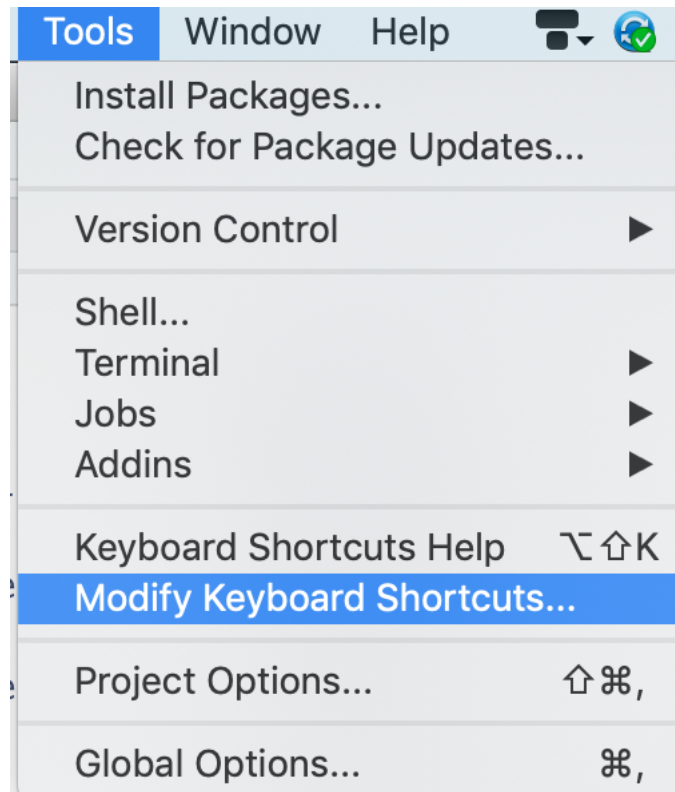
Al seleccionar la segunda opción “Print update to script (no update)” escribe el código R con los cambios realizados (nota: se produce un comportamiento extraño porque lo pega al final del documento Rmd):

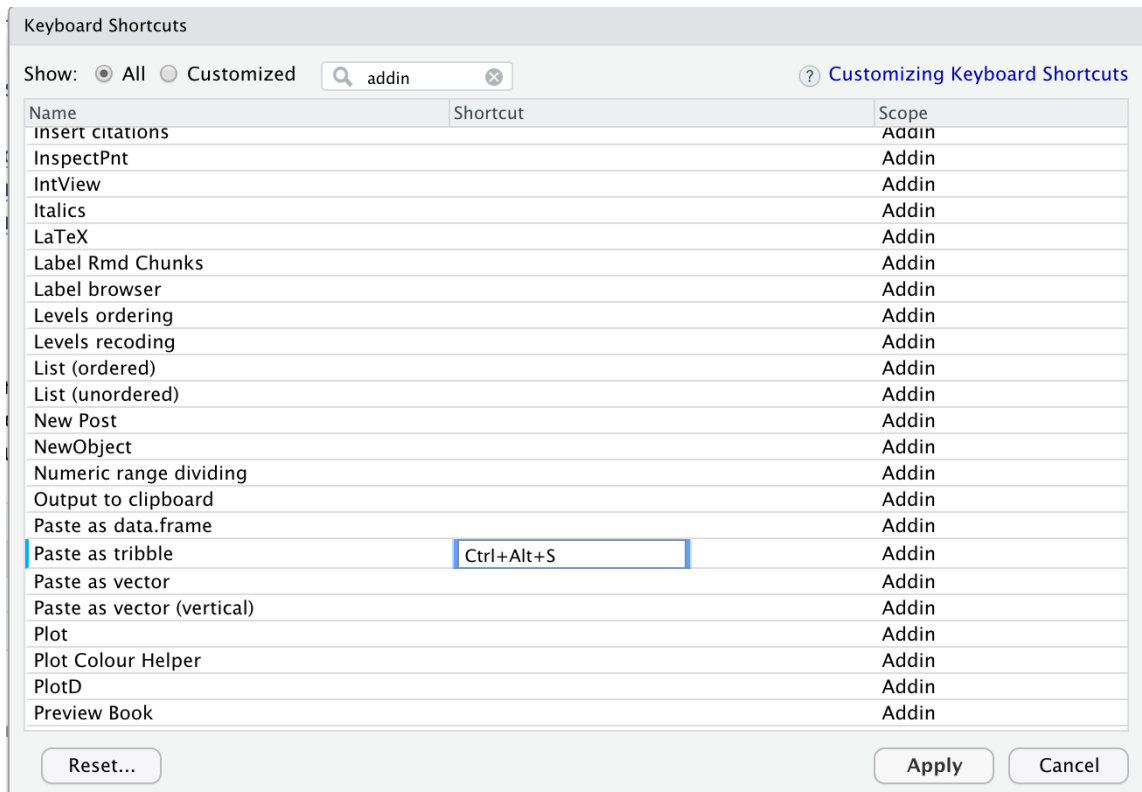
```
my_tbl = structure(list(Col_1 = 1:4,
  Col_2 = c(5L, 32L, 7L, 8L),
  Col_3 = c(9L, 10L, 11L, 25L)),
  row.names = c(NA, -4L),
  class = "data.frame")
my_tbl
```

```
##   Col_1 Col_2 Col_3
## 1     1     5     9
## 2     2    32    10
## 3     3     7    11
## 4     4     8    25
```

1.5.5. Asociar combinaciones de teclas a los Addins

Es posible asociar a los que más utilicemos una combinación de teclas para realizar más rápidamente. En las siguientes capturas puede verse cómo definir las.





1.6. Ya se dispone de la información en formato digital

1.6.1. Se tiene información en un objeto R

Para mostrar un “data.frame” o “tibble” en un trozo de código R y cuando el formato de salida es de tipo “html” o “Rnotebook”, en la cabecera yaml del documento “Rmd” se pueden incluir algunas de las siguientes opciones:

1. `df_print: paged` (lo muestra utilizando la función `rmarkdown::paged_table`, para html)
2. `df_print: kable` (lo muestra utilizando la función `knitr::kable()`)
3. `df_print: tibble` (lo muestra tratándolo como un objeto de tipo “tibble”)
4. `df_print: default` (lo muestra como en la consola de R)

Por ejemplo, si hemos utilizado la opción (2), es decir: `df_print: kable`, al utilizar un trozo de código R en el que se solicita que se muestre algún “data.frame” o “tibble”, como vemos a continuación

```
```{r}
head(iris,7)
```
```

```
```{r}
iris[1:7,]
```
```

Automáticamente mostrará en la salida generada, el siguiente resultado (lo ha sustituido por: `knitr::kable(head(iris,7))` o `knitr::kable(iris[1:7,])`, respectivamente):

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | setosa |

1.6.2. Crear tablas con la función kable de la librería knitr

Por ejemplo, queremos mostrar el contenido del data.frame “iris” que se encuentra disponible en cualquier instalación de R.

En este caso, con ayuda de la librería “knitr” y su función kable puede presentarse la información en forma de tabla, y además especificar una serie de argumentos que personalizan cómo se ve.

```
require(knitr)
kable(head(iris,10),
      digits = 3, row.names = FALSE, align = "c",
      caption = NULL)
```

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 4.9 | 3.1 | 1.5 | 0.1 | setosa |

La sintaxis de la función kable es la siguiente:

```
kable(x, format, digits = getOption("digits"),
      row.names = NA, col.names = NA, align,
      caption = NULL, label = NULL,
      format.args = list(), escape = TRUE, ...)
```

Nota. Resaltar, que es muy conveniente añadir el argumento: `booktabs=TRUE`. Esta opción presenta la tabla siguiendo un esquema muy habitual en los estándares de la imprenta:

- La cabecera en negrita y entre líneas
- Las columnas numéricas justificadas a la derecha
- Las columnas de tipo carácter justificadas a la izquierda.

```
require(knitr)
kable(head(iris,10), booktabs = TRUE)
```


| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 4.9 | 3.1 | 1.5 | 0.1 | setosa |

Con la opción “col.names” de la función `kable()` se pueden redefinir los nombres de las columnas del data.frame o tibble, utilizando caracteres especiales, como por ejemplo espacios en blanco, acentos, expresiones matemáticas (si aparece el símbolo `$`: `escape=FALSE`),

...

```
require(knitr)
kable(head(iris,10), booktabs = TRUE, align = c("l","c","r","r","c"),
      col.names = c("Longitud Sépalo", "Anchura Sépalo",
                    "Longitud Pétalo", "Anchura Pétalo",
                    "Especies  $\mathcal{E}$ "),
      escape=FALSE)
```

| Longitud Sépalo | Anchura Sépalo | Longitud Pétalo | Anchura Pétalo | Especies \mathcal{E} |
|-----------------|----------------|-----------------|----------------|------------------------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 4.9 | 3.1 | 1.5 | 0.1 | setosa |

También se ha usado la opción `align()` para especificar la justificación del contenido de cada una de las columnas, donde: “c” es para centrar, “l” es para justificar a la izquierda y “r” es para justificar a la derecha.

1.6.2.1. Tablas sin cabeceras con kable

Si no se quiere mostrar los nombres de las columnas o variables, se debe usar el argumento `col.names=NULL` en la llamada a la función `kable()`.

```
my_tbl <- tibble::tribble(  
  ~Col_1, ~Col_2, ~Col_3,  
    1,     5,     9,  
    2,     6,    10,  
    3,     7,    11,  
    4,     8,    12  
)  
  
require(knitr)  
kable(my_tbl, digits = 3, row.names = FALSE, align = "c", col.names = NULL,  
      caption = NULL)
```

| | | |
|---|---|----|
| 1 | 5 | 9 |
| 2 | 6 | 10 |
| 3 | 7 | 11 |
| 4 | 8 | 12 |

1.6.3. Crear tablas con la librería `kableExtra`

La información que se ha utilizado en este material principalmente ha sido extraída del siguiente enlace pdf sobre [ejemplos de uso de la librería `kableExtra`](#).

El objetivo de `kableExtra` es ayudar a construir tablas complejas y manipular estilos de tabla. Importa el símbolo de tubería `%>%` de `magrittr` y verbaliza todas las funciones para que puedan aplicarse a la función `kable()` de la librería “knitr”, al estilo de paquetes como: `dplyr`, `ggplot2` y `plotly`.

En el siguiente ejemplo, vemos como se mejora la tabla presentada en el apartado de “knitr::kable()”, con ayuda de la función `kable_styling()`:

- `latex_options`: un vector con varias opciones: con “striped” las filas alternan colores, y con “condensed” la altura de las filas se hace más estrecha.
- `position`: con la opción “center” la tabla aparece centrada en el texto (otras opciones son: “left”, “right”, y “float_left” y “float_right”, estas últimas dejan que el texto fluya a la izquierda o a la derecha respectivamente).
- `full_width`: con “FALSE” le indica que la tabla **no** se ensanche para ocupar el ancho de la página, con “TRUE” **sí** se ajustan las columnas para que toda la tabla ocupe el ancho completo de la página.

```
require(knitr)
require(kableExtra)
head(iris,10) %>%
  kable(booktabs = TRUE,format = "latex") %>%
  kable_styling(
    latex_options = c("striped", "condensed"),
    position = "center",
    full_width = FALSE)
```

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 4.9 | 3.1 | 1.5 | 0.1 | setosa |

Otra forma alternativa de escribir ese mismo código, sin utilizar el símbolo de tubería `%>%`, sería:

```
kable_styling(
  kable(head(iris,10), format = "latex"),
  latex_options = c("striped", "condensed"),
  position = "center",
  full_width = FALSE)
```

Para obligar a que la salida de `kableExtra` sea en formato html habría que cambiar algunas opciones:

- `format="html"`,
- y utilizar: `bootstrap_options=` en lugar de `latex_options=`, principalmente.

La sintaxis de la función `kable_styling()` es la siguiente:

```
kable_styling(kable_input, bootstrap_options = "basic",
  latex_options = "basic", full_width = NULL, position = "center",
  font_size = NULL, row_label_position = "1",
  repeat_header_text = "\\textit{(continued)}",
  repeat_header_method = c("append", "replace"),
  repeat_header_continued = FALSE, stripe_color = "gray!6",
  stripe_index = NULL, latex_table_env = NULL, protect_latex = TRUE,
  table.envir = "table", fixed_thead = FALSE)
```

Comentamos algunas de estas opciones:

- `stripe_color`: (por defecto es: “gray!6”) en la salida latex se puede especificar un color diferente para las líneas sombreadas. Esta opción no está disponible en HTML.
- `stripe_index`: (por defecto es: NULL, lo que significa que alterna empezando a colorear la primera línea que no es de la cabecera) en la salida latex se puede especificar que filas aparecen coloreadas (se indica con un vector de filas).
- `font_size`: (por defecto es: NULL) un valor numérico para indicar el tamaño de la fuente que se usará en toda la tabla.

En este ejemplo, se obliga a que la tabla ocupe el ancho de la página con `full_width=TRUE`:

```
library(kableExtra)
head(iris,10) %>%
  kable(booktabs = TRUE,format = "latex",
        digits = c(0,1,0,1,NA)) %>%
  kable_styling(
    latex_options = c("striped", "condensed"),
    position = "center",
    full_width = TRUE)
```

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5 | 3.5 | 1 | 0.2 | setosa |
| 5 | 3.0 | 1 | 0.2 | setosa |
| 5 | 3.2 | 1 | 0.2 | setosa |
| 5 | 3.1 | 2 | 0.2 | setosa |
| 5 | 3.6 | 1 | 0.2 | setosa |
| 5 | 3.9 | 2 | 0.4 | setosa |
| 5 | 3.4 | 1 | 0.3 | setosa |
| 5 | 3.4 | 2 | 0.2 | setosa |
| 4 | 2.9 | 1 | 0.2 | setosa |
| 5 | 3.1 | 2 | 0.1 | setosa |

Además, en este último ejemplo se ha controlado el número de dígitos que se muestran en las columnas numéricas con la opción: `digits=` de la función `kable()`. En este caso se ha usado un vector, `digits = c(0,1,0,1,NA)` para indicar qué número de cifras decimales se muestran en cada columna (NA para la columna de tipo `character`). Si se usa un único valor, sería para todas las columnas (`digits = 1`).

```
library(kableExtra)
head(iris,10) %>%
  kable(booktabs = TRUE,format = "latex") %>%
  kable_styling(
    latex_options = c("striped", "condensed"),
    position = "float_left",
    font_size=8,
    full_width = FALSE)
```

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 4.9 | 3.1 | 1.5 | 0.1 | setosa |

En este otro ejemplo, la tabla aparece justificada a la izquierda permitiendo que el texto que hemos escrito justo después del código R aparezca de forma fluida a su derecha (hemos usado antes del código R, el comando LaTeX `\bigskip` para separar las dos tablas).

También se ha cambiado el tamaño de fuente a 8: `font_size=8`.

Si se usa `float_right` el texto se debe colocar justo antes del código R.

En el siguiente ejemplo se usa la opción `scale_down` en `latex_options`, lo que ajustará automáticamente el tamaño de la fuente para que quepa en el ancho de la página. Es adecuado su uso cuando es una tabla con muchas columnas:

```
library(kableExtra)

cbind(iris,iris,iris)[1:10,] %>%
  kable(booktabs = TRUE,format = "latex") %>%
  kable_styling(
    latex_options = c("striped", "condensed","scale_down"),
    position = "center",
    full_width = FALSE)
```

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|--------------|-------------|--------------|-------------|---------|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa | 5.1 | 3.5 | 1.4 | 0.2 | setosa | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa | 4.9 | 3.0 | 1.4 | 0.2 | setosa | 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa | 4.7 | 3.2 | 1.3 | 0.2 | setosa | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa | 4.6 | 3.1 | 1.5 | 0.2 | setosa | 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa | 5.0 | 3.6 | 1.4 | 0.2 | setosa | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa | 5.4 | 3.9 | 1.7 | 0.4 | setosa | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | setosa | 4.6 | 3.4 | 1.4 | 0.3 | setosa | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 5.0 | 3.4 | 1.5 | 0.2 | setosa | 5.0 | 3.4 | 1.5 | 0.2 | setosa | 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| 4.4 | 2.9 | 1.4 | 0.2 | setosa | 4.4 | 2.9 | 1.4 | 0.2 | setosa | 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 4.9 | 3.1 | 1.5 | 0.1 | setosa | 4.9 | 3.1 | 1.5 | 0.1 | setosa | 4.9 | 3.1 | 1.5 | 0.1 | setosa |

Consejo: en situaciones extremas puede no resolverlo correctamente (en este caso se muestra con un tamaño de letra demasiado pequeño), de ahí que se aconseje que se subdivida la tabla en dos grupos de columnas y se presenten de forma separada, manteniendo posiblemente la primera columna.

Tabla 1.14: Leyenda explicativa de la primera tabla

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 4.9 | 3.1 | 1.5 | 0.1 | setosa |

1.6.3.1. Ejemplo `kableExtra`: añadiendo una leyenda explicativa a la tabla y manejo de la posición FLOTANTE de la tabla

Para incluir una leyenda explicativa a una tabla se usa el argumento `caption=`, de la función `kable()`, como puede verse en el siguiente código.

Pero la tabla o cuadro resultante puede aparecer no justo en el sitio que hemos escrito el código R (“chunk”). Habitualmente, las publicaciones generalmente prefieren que las tablas estén colocadas en determinadas posiciones (parte superior o parte inferior) y para referirse a ellas se recomienda utilizar las referencias numeradas que se añaden junto a la leyenda explicativa. En este caso, estamos hablando de la tabla o cuadro 1.14, y puede verse que la ha colocado en la parte superior de la página donde está el “chunk” de código R (la referencia utilizada deja claro a que tabla nos estamos refiriendo).

```
library(kableExtra)
head(iris,10) %>%
  kable(booktabs = TRUE,format = "latex",
        caption = "\\label{tabla01}Leyenda explicativa de
la primera tabla") %>%
  kable_styling(
    latex_options = c("striped", "condensed"),
    position = "center",
    full_width = FALSE)
```

Podemos hacer referencia a esta tabla usando `\ref{tabla01}` en el texto. Como puede verse en la tabla 1.14.

Nota LaTeX. Cuando se usa `caption=` con la función `kable()` al crear el fichero pdf se inserta el entorno LaTeX: `\begin{table}[posicion] ... \end{table}`, que hace que las tablas puedan tener asociada una leyenda con una referencia, y sean colocadas de forma flotante en posiciones del documento convenientes, según los cánones de las publicaciones.

En este otro ejemplo, usamos `"hold_position"` en `latex_options`, esto intentará poner la tabla en el lugar del “chunk” R, pero puede introducir espacios verticales que no

dejan un aspecto adecuado al documento. Estamos hablando de la tabla 1.15 (en algunas situaciones puede observarse un espacio vertical adicional no deseado que se ha añadido antes y después de la tabla).

```
library(kableExtra)
head(iris,10) %>%
  kable(booktabs = TRUE,format = "latex",
        caption = "\\label{tabla02}Leyenda explicativa de
la segunda tabla") %>%
  kable_styling(
    latex_options = c("striped", "condensed","hold_position"),
    position = "center",
    full_width = FALSE)
```

Tabla 1.15: Leyenda explicativa de la segunda tabla

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 4.9 | 3.1 | 1.5 | 0.1 | setosa |

Pero para forzar aún más, que la tabla irá justamente en el sitio en el que hayamos colocado el “chunk”, sin añadir espacios verticales adicionales, se recomienda usar la opción: "HOLD_position" en latex_options. Estamos hablando de la tabla 1.16.

```
library(kableExtra)
head(iris,10) %>%
  kable(booktabs = TRUE,format = "latex",
        caption = "\\label{tabla03}Leyenda explicativa de
la tercera tabla") %>%
  kable_styling(
    latex_options = c("striped", "condensed","HOLD_position"),
    position = "center",
    full_width = FALSE)
```


Tabla 1.16: Leyenda explicativa de la tercera tabla

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 4.9 | 3.1 | 1.5 | 0.1 | setosa |

Consejo LaTeX. Se recomienda usar el comando LaTeX `\clearpage` para provocar un salto de página en un documento R Markdown, en lugar del comando LaTeX: `\newpage`. Evitará que las tablas se coloquen en posiciones muy alejadas del “chunk” que las originó.

En este ejemplo, se ha utilizado `\newpage` (para demostrar el efecto perjudicial) y no se ha arreglado el problema. Si se hubiera utilizado `\clearpage` sí habría quedado mejor.

También, se recomienda usar siempre: `HOLD_position` en lugar de `hold_position`, en `latex_options`.

Consejo LaTeX. Cuando se utilizan leyendas en las tablas aparece un nombre inicial (por defecto en inglés: “Table” y en español: “Cuadro”) acompañado de una numeración. Para cambiar ese nombre se puede utilizar el siguiente comando LaTeX ([url información](#)), con el nuevo nombre “Tabla” y colocarlo justo después de la cabecera “yaml” del documento R Markdown.

```
\renewcommand\tablename{Tabla}
```

Con el paquete “babel-spanish” sería:

```
\renewcommand\spanishtablename{Tabla}
```

Consejo LaTeX. En algunas publicaciones se exige que las leyendas que van junto a una tabla o a una figura sigan una serie de normas en cuanto a su personalización. Para conseguir un mayor control en esta cuestión se recomienda utilizar el paquete LaTeX “caption”. A continuación, se incluyen algunas referencias para conocer mejor las posibilidades de este paquete y su incorporación en un documento R Markdown:

- [Enlace a CTAN-caption](#) y [Enlace documentación sobre caption](#)
- [Enlace R Markdown y caption](#)

Para poder usarlo, se debería incluir una referencia en la cabecera “yaml” del documento R Markdown como la siguiente:

```
header-includes:
- \usepackage[labelsep=period,labelfont=bf]{caption}
```

Y además se podría incluir en el documento R Markdown (salida latex-pdf) algunos de estos comandos, con la personalización requerida (consultar la documentación de “caption” para ver todas las posibilidades existentes).

```
\captionsetup{justification=raggedright,singlelinecheck=false}

\captionsetup[table]{format=hang,justification=raggedright,
  singlelinecheck=false}

\captionsetup[figure]{format=plain,justification=raggedleft,
  singlelinecheck=false}
```

Consejo LaTeX. El código LaTeX que se genera al crear la Tabla 1.14 es el siguiente:

```
\begin{table}[t]

\caption{\label{tab:}\label{tabla01}Leyenda explicativa de
  la primera tabla}

\centering
\begin{tabular}{rrrrl}
\toprule
Sepal.Length & Sepal.Width & Petal.Length & Petal.Width & Species\\
\midrule
\rowcolor{gray!6} 5.1 & 3.5 & 1.4 & 0.2 & setosa\\
4.9 & 3.0 & 1.4 & 0.2 & setosa\\
\rowcolor{gray!6} 4.7 & 3.2 & 1.3 & 0.2 & setosa\\
4.6 & 3.1 & 1.5 & 0.2 & setosa\\
\rowcolor{gray!6} 5.0 & 3.6 & 1.4 & 0.2 & setosa\\
\addlinespace
5.4 & 3.9 & 1.7 & 0.4 & setosa\\
\rowcolor{gray!6} 4.6 & 3.4 & 1.4 & 0.3 & setosa\\
5.0 & 3.4 & 1.5 & 0.2 & setosa\\
\rowcolor{gray!6} 4.4 & 2.9 & 1.4 & 0.2 & setosa\\
4.9 & 3.1 & 1.5 & 0.1 & setosa\\
\bottomrule
\end{tabular}
\end{table}
```

El entorno LaTeX que produce que la tabla pueda presentar una leyenda, pueda tener referencias y se posicione de forma flotante es: `\begin{table}[t] ... \end{table}`.

Los corchetes que aparecen a continuación de `\begin{table}`, en este caso: `[t]`, lleva una cadena de caracteres, “t”, que le indica donde puede colocar la tabla. Las opciones más habituales son:

- “t”: para colocar la tabla en la parte superior de la página.
- “b”: para colocar la tabla en la parte inferior de la página.
- “p”: para colocar la tabla en una página que solamente hay tablas.
- “h”: para colocar la tabla justo aquí donde esté el entorno `\begin{table} ... \end{table}`. Puede llevar un signo de admiración antes: “!h”.
- “H”: para colocar la tabla justo aquí, pero sin introducir espacios verticales adicionales

(necesita que esté cargado un paquete LaTeX llamado “float”. Si se usa “kableExtra” puede utilizarse).

También se puede utilizar una combinación de esos caracteres, por ejemplo: “!htbp”.

En particular, la opción `HOLD_position` corresponde a utilizar `[H]`, y `hold_position` corresponde a utilizar `[!h]`.

Consejo LaTeX. También sería posible guardar el código LaTeX que genera la tabla en un fichero latex (suelen tener extensión: “.tex”) y cargarlo en el documento R Markdown con la orden latex siguiente:

```
\input{ficherotabla.tex}
```

La siguiente tabla ha sido incluida en este documento R Markdown al insertar la tabla contenida en el fichero “codigoR/tablatex01.tex”, al escribir `\input{codigoR/tablatex01.tex}`.

| | | |
|------|------|--|
| test | test | |
|------|------|--|

Tabla 1.17: test caption

El código LaTeX que contiene ese fichero es:

```
\begin{table}![htbp]\centering
\begin{tabular}{|c|c|}
test & test\\
\end{tabular}
\caption{test caption}
\end{table}
```

Consejo R Markdown. Se pueden realizar referencias utilizando el identificador del “chunk” que genera la tabla, es decir, donde se encuentra la función `kable()` con el argumento `caption=`. Ya que se genera una etiqueta automáticamente del siguiente modo: `\label{tab:identificador}`, se puede hacer referencia a la tabla utilizando `\ref{tab:identificador}`. Es importante no olvidar escribir “**tab:**”. De esta forma no es necesario introducir `\\label{etiqueta}` en el texto asociado a `caption=`.

Lo siguiente solamente funciona con la salida: `bookdown::pdf_document2`: escribir en el texto `\@ref(tab:identificador)`.

Consejos sobre cómo escribir un libro. La mayoría de las editoriales recogen en sus páginas web una serie de consejos sobre cómo se debe escribir un documento, concretando de forma rigurosa cada uno de los aspectos que esta tarea conlleva.

A continuación se recoge una serie de enlaces que tiene disponibles la editorial “Springer” en su página web:

- [Manuscript preparation \(html\)](#)
- [Key Style Points \(pdf\)](#)
- [Key Style Points - Tables and Lists \(pdf\)](#)

Algunas de las recomendaciones que pueden encontrarse sobre tablas se recogen a continuación:

- La leyenda o subtítulos de la tabla comienzan con el término “Tabla” en negrita, seguido del número de la tabla, también en negrita.
- Si el material se ha publicado anteriormente, se debe identificar mediante una referencia a la fuente original al final de la leyenda.
- La leyenda de la tabla no tiene puntuación final ni punto después del número de la tabla.
- Las notas al pie de la tabla aparecen en la siguiente secuencia:
 - Comentarios de carácter general
 - Explicaciones de abreviaturas en un lista de inicio
 - notas al pie, organizadas primero por asterisco (valores p), luego en minúscula
- En la versión PDF, las tablas aparecerán en la parte superior de la página lo más cerca posible de su primera cita.

1.6.3.2. Ejemplo kableExtra: tablas que ocupan varias páginas

Cuando se quiere presentar una tabla que no cabe en una página, se puede utilizar la opción de la función `kable()`: `longtable=TRUE`. Si únicamente usamos esa opción, las filas de la tabla continuarán escribiéndose en las siguientes páginas pero no se repetirán los nombres de las columnas, una cuestión importante cuando se quiere seguir a qué columna pertenece una celda.

Para corregir esa situación, la librería “kableExtra” nos permite mejorar esta situación utilizando los siguientes argumentos, como puede comprobarse en el ejemplo que se muestra:

- Se ha añadido en `latex_options` la opción: `repeat_header`.
- Se ha indicado el texto que aparecerá antes de pasar a la siguiente página: `repeat_header_continued="contin\\'ua en la siguiente p\\'agina"`.
- Se ha indicado también el texto que aparecerá antes de escribir información de la tabla en la siguiente página: `repeat_header_text = "continuaci\\'on"`.

```
library(kableExtra)
iris[1:50,] %>%
  kable(booktabs = TRUE,format = "latex",digits = 1,
        longtable=TRUE) %>%
  kable_styling(
    latex_options = c("striped", "condensed","repeat_header"),
    position = "center",
    full_width = FALSE,
    repeat_header_text = "continuaci\\'on",
    repeat_header_continued="contin\\'ua en la siguiente p\\'agina")
```

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 4.9 | 3.1 | 1.5 | 0.1 | setosa |
| 5.4 | 3.7 | 1.5 | 0.2 | setosa |
| 4.8 | 3.4 | 1.6 | 0.2 | setosa |
| 4.8 | 3.0 | 1.4 | 0.1 | setosa |
| 4.3 | 3.0 | 1.1 | 0.1 | setosa |
| 5.8 | 4.0 | 1.2 | 0.2 | setosa |
| 5.7 | 4.4 | 1.5 | 0.4 | setosa |

continúa en la siguiente página

continuación

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.4 | 3.9 | 1.3 | 0.4 | setosa |
| 5.1 | 3.5 | 1.4 | 0.3 | setosa |
| 5.7 | 3.8 | 1.7 | 0.3 | setosa |
| 5.1 | 3.8 | 1.5 | 0.3 | setosa |
| 5.4 | 3.4 | 1.7 | 0.2 | setosa |
| 5.1 | 3.7 | 1.5 | 0.4 | setosa |
| 4.6 | 3.6 | 1.0 | 0.2 | setosa |
| 5.1 | 3.3 | 1.7 | 0.5 | setosa |
| 4.8 | 3.4 | 1.9 | 0.2 | setosa |
| 5.0 | 3.0 | 1.6 | 0.2 | setosa |
| 5.0 | 3.4 | 1.6 | 0.4 | setosa |
| 5.2 | 3.5 | 1.5 | 0.2 | setosa |
| 5.2 | 3.4 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.6 | 0.2 | setosa |
| 4.8 | 3.1 | 1.6 | 0.2 | setosa |
| 5.4 | 3.4 | 1.5 | 0.4 | setosa |
| 5.2 | 4.1 | 1.5 | 0.1 | setosa |
| 5.5 | 4.2 | 1.4 | 0.2 | setosa |
| 4.9 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.2 | 1.2 | 0.2 | setosa |
| 5.5 | 3.5 | 1.3 | 0.2 | setosa |
| 4.9 | 3.6 | 1.4 | 0.1 | setosa |
| 4.4 | 3.0 | 1.3 | 0.2 | setosa |
| 5.1 | 3.4 | 1.5 | 0.2 | setosa |
| 5.0 | 3.5 | 1.3 | 0.3 | setosa |
| 4.5 | 2.3 | 1.3 | 0.3 | setosa |
| 4.4 | 3.2 | 1.3 | 0.2 | setosa |
| 5.0 | 3.5 | 1.6 | 0.6 | setosa |
| 5.1 | 3.8 | 1.9 | 0.4 | setosa |
| 4.8 | 3.0 | 1.4 | 0.3 | setosa |
| 5.1 | 3.8 | 1.6 | 0.2 | setosa |
| 4.6 | 3.2 | 1.4 | 0.2 | setosa |
| 5.3 | 3.7 | 1.5 | 0.2 | setosa |
| 5.0 | 3.3 | 1.4 | 0.2 | setosa |

1.6.3.3. Ejemplo kableExtra: especificar características columnas/filas/cabeceras

Cuando tenga una tabla con muchos textos explicativos, es posible que desee especificar el ancho de columna para alguna columna, ya que el ajuste automático puede no funcionar correctamente. Además, a veces, puede resaltar una columna (por ejemplo, una columna “Total”) poniéndola en negrita. En estos escenarios, puede usar `column_spec()`. A continuación, aparece un ejemplo de uso:

```
text_tbl <- data.frame(
  Items = c("Item 1", "Item 2", "Item 3"),
  Features = c(
    "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vehicula tempor ex. Morbi malesuada sagittis turpis, at venenatis nisl luctus a. ",
    "In eu urna at magna luctus rhoncus quis in nisl. Fusce in velit varius, posuere risus et, cursus augue. Duis eleifend aliquam ante, a aliquet ex tincidunt in. ",
    "Vivamus venenatis egestas eros ut tempus. Vivamus id est nisi. Aliquam molestie erat et sollicitudin venenatis. In ac lacus at velit scelerisque mattis. "
  )
)

kable(text_tbl, "latex", booktabs = T) %>%
  kable_styling(full_width = F) %>%
  column_spec(1, bold = T, color = "red") %>%
  column_spec(2, width = "30em")
```

| Items | Features |
|---------------|--|
| Item 1 | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vehicula tempor ex. Morbi malesuada sagittis turpis, at venenatis nisl luctus a. |
| Item 2 | In eu urna at magna luctus rhoncus quis in nisl. Fusce in velit varius, posuere risus et, cursus augue. Duis eleifend aliquam ante, a aliquet ex tincidunt in. |
| Item 3 | Vivamus venenatis egestas eros ut tempus. Vivamus id est nisi. Aliquam molestie erat et sollicitudin venenatis. In ac lacus at velit scelerisque mattis. |

Similar a `column_spec`, pueden definirse especificaciones para filas. Puede poner en negrita o en cursiva una fila completa. Hay que tener en cuenta que, al igual que otras funciones relacionadas con las filas en `kableExtra`, para la posición de la fila de destino, no necesita contar las filas de encabezado o las filas de etiquetado de grupo.

```
dt <- mtcars[1:5, 1:6]
kable(dt, "latex", booktabs = T) %>%
  kable_styling("striped", full_width = F) %>%
  column_spec(7, border_left = T, bold = T) %>%
  row_spec(1, strikeout = T) %>%
  row_spec(3:5, bold = T, color = "white", background = "black")
```

| | mpg | cyl | disp | hp | drat | wt |
|--------------------------|-------------|----------|------------|------------|-------------|--------------|
| Mazda RX4 | 21.0 | 6 | 160 | 110 | 3.90 | 2.620 |
| Mazda RX4 Wag | 21.0 | 6 | 160 | 110 | 3.90 | 2.875 |
| Datsun 710 | 22.8 | 4 | 108 | 93 | 3.85 | 2.320 |
| Hornet 4 Drive | 21.4 | 6 | 258 | 110 | 3.08 | 3.215 |
| Hornet Sportabout | 18.7 | 8 | 360 | 175 | 3.15 | 3.440 |

Un caso especial de `row_spec` es que se puede especificar el formato de la fila del encabezado a través de `row_spec(row = 0, ...)`.

```
dt <- mtcars[1:5, 1:6]
kable(dt, "latex", booktabs = T, align = "c") %>%
  kable_styling(latex_options = "striped", full_width = F) %>%
  row_spec(0, angle = 45)
```

| | mpg | cyl | disp | hp | drat | wt |
|-------------------|------|-----|------|-----|------|-------|
| Mazda RX4 | 21.0 | 6 | 160 | 110 | 3.90 | 2.620 |
| Mazda RX4 Wag | 21.0 | 6 | 160 | 110 | 3.90 | 2.875 |
| Datsun 710 | 22.8 | 4 | 108 | 93 | 3.85 | 2.320 |
| Hornet 4 Drive | 21.4 | 6 | 258 | 110 | 3.08 | 3.215 |
| Hornet Sportabout | 18.7 | 8 | 360 | 175 | 3.15 | 3.440 |

1.6.3.4. Ejemplo kableExtra: especificar ancho de columnas y modo apaisado

Se crea en primer lugar el objeto tipo “tibble” con la información adecuada, en este caso se necesitan 3 columnas: “Month”, “Sam” y “Supervisors”, y se le van añadiendo filas al tibble con la función `add_row()` del paquete “tibble”.

```
library(tidyverse)
library(knitr)
library(kableExtra)

timeline_tab <- tibble(
  Month = c(
    "Feburary",
    ""),
  Sam = c(
    "1st Introduction",

    "15th, The epidemiology of tuberculosis,
    and the role of BCG vaccination, in England &
    Developing and parameterising a dynamic transmission
    model of tuberculosis"),
  Supervisors = c(
    "15th Introduction",
    "")
) %>%
add_row(Month = c(
  "March",
  ""),
  Sam = c("1st Reassessing the Evidence for Universal School-age
  Bacillus Calmette Guerin (BCG) Vaccination in England and
  Wales & Beneficial effects of BCG vaccination in outcomes
  for patients diagnosed with TB: observation study using the
  Enhanced Tuberculosis Surveillance System 2009-2015",

  "15th Investigating the impact of the 2005 change in BCG
  vaccination policy using a fitted dynamic transmission model
  of TB & Estimating the effect of the 2005 UK BCG vaccination
  policy change: A retrospective cohort study using the
  Enhanced Tuberculosis Surveillance system, 2000-2015"
  ),
  Supervisors = c(
    "1st The epidemiology of tuberculosis, and the role of BCG
    vaccination, in England & Developing and parameterising
    a dynamic transmission model of tuberculosis",
    "")
) %>%
add_row(Month = c(
  "April"),
  Sam = c(
    "1st All chapters excepting the last two chapters
    with Supervisors, 15th Final two chapters updated based
```

```

        on supervisor comments."
      ),
      Supervisors = c(
        "1st Investigating the impact of the 2005 change in BCG
        vaccination policy using a fitted dynamic transmission model
        of TB & Estimating the effect of the 2005 UK BCG vaccination
        policy change: A retrospective cohort study using the
        Enhanced Tuberculosis Surveillance system, 2000-2015"
      )
    ) %>%
  add_row(Month = c(
    "May"),
    Sam = c(
      "1st Final hand in + viva organisation"),
    Supervisors = c(""))

```

Una vez creado el objeto “tibble”: `timeline_tab` se le pasa a la función `kable` y se personaliza la tabla al especificar (ver la tabla o cuadro [1.19](#)):

- el ancho de algunas columnas, llamando en varias ocasiones a la función de `kableExtra`: `column_spec()`,
- y al ser la tabla demasiado ancha para un documento pdf con tamaño a4, se decide presentarla en modo apaisado al llamar a: `landscape()`.

```

timeline_tab %>%
  kable(booktabs = TRUE, caption="\\label{tab:apaisada}Esta tabla
    está apaisada y en una página separada") %>%
  column_spec(c(1), width = "2cm") %>%
  column_spec(c(2, 3), width = "8cm") %>%
  landscape()

```

Tabla 1.19: Esta tabla está apaisada y en una página separada

| Month | Sam | Supervisors |
|----------|---|---|
| February | 1st Introduction 15th, The epidemiology of tuberculosis, and the role of BCG vaccination, in England & Developing and parameterising a dynamic transmission model of tuberculosis | 15th Introduction |
| March | 1st Reassessing the Evidence for Universal School-age Bacillus Calmette Guerin (BCG) Vaccination in England and Wales & Beneficial effects of BCG vaccination in outcomes for patients diagnosed with TB: observation study using the Enhanced Tuberculosis Surveillance System 2009-2015 15th Investigating the impact of the 2005 change in BCG vaccination policy using a fitted dynamic transmission model of TB & Estimating the effect of the 2005 UK BCG vaccination policy change: A retrospective cohort study using the Enhanced Tuberculosis Surveillance system, 2000-2015 | 1st The epidemiology of tuberculosis, and the role of BCG vaccination, in England & Developing and parameterising a dynamic transmission model of tuberculosis |
| April | 1st All chapters excepting the last two chapters with Supervisors, 15th Final two chapters updated based on supervisor comments. | 1st Investigating the impact of the 2005 change in BCG vaccination policy using a fitted dynamic transmission model of TB & Estimating the effect of the 2005 UK BCG vaccination policy change: A retrospective cohort study using the Enhanced Tuberculosis Surveillance system, 2000-2015 |
| May | 1st Final hand in + viva organisation | |

1.6.3.5. Ejemplo kableExtra: leer data.frame de fichero externo, tabla con colores y cabeceras compuestas

Este ejemplo se ha obtenido de la siguiente [url](#).

En este ejemplo, el tratamiento necesario para obtener los datos que se quieren presentar se han guardado en un fichero de código R aparte y después de ese tratamiento se guardó el data.frame resultante en un fichero RDS. De esta forma, para presentar la tabla en un informe no tenemos que repetir el proceso de preparación de los datos, únicamente tenemos que leer los datos guardados y tratarlo con kable-kableExtra.

La tabla 1.20 muestra las poblaciones de cada provincia y Canadá en total para los años 2011 y 2016 (esta información incluye solo las 100 ciudades más grandes de Canadá).

```
library(dplyr)
library(knitr)
library(kableExtra)

# importar los datos
wiki_data_final_table <- readRDS(here::here("codigoR",
  "cleaned_wiki_data_for_table.rds"))
```

Una vez que se ha leído el fichero que contiene el objeto data.frame: “wiki_data_final_table”, construimos la tabla del siguiente modo:

```
# Colores para la tabla
blue_table_colour <- "#9BD4F5"
orange_table_colour <- "#FF9000"
light_stripping_table_colour <- "#DDDDDD"

wiki_data_final_table %>%
  knitr::kable(
    "latex",
    booktabs = TRUE,
    linesep = "",
    caption = "\\label{tab2:table-population-by-province}Number of cities
and population of each province (including only the largest 100 cities
in Canada)",
    col.names = c("Province", "Number of largest 100 cities in this province",
      rep(c("2011", "2016"), 1)),
    align = c("l", rep("r", 3))) %>%
  kable_styling(latex_options = "HOLD_position") %>%
  add_header_above(
    c(" " = 1, # tiene que ser un espacio aquí, como " ", y no como ""
      " " = 1,
      "Population" = 2),
    bold = TRUE,
    line = FALSE,
    background = blue_table_colour
  ) %>%
  column_spec(1,width = "6cm") %>%
  column_spec(2:4,width = "3cm") %>%
  footnote(general = "There are other cities in Canada not included
```

```

    in this table; the excluded cities are those smaller than
    the 100 largest cities in Canada.",
    threeparttable = TRUE,
    general_title = "Footnote:") %>%
row_spec(
  row = 0,
  background = blue_table_colour,
  bold = TRUE,
  align = "c"
) %>%
row_spec(
  row = c(2,4,6,8,10),
  background = light_stripping_table_colour
) %>%
row_spec(
  row = 11,
  background = orange_table_colour,
  bold = TRUE
) %>%
row_spec(
  row = 10,
  hline_after = TRUE) # Desafortunadamente, esta línea se oculta por el
# color naranja de la fila final, por lo que esta
# línea de código realmente no hace nada :(

```

Tabla 1.20: Number of cities and population of each province (including only the largest 100 cities in Canada)

| Province | Number of largest 100 cities in this province | Population | |
|---------------------------|--|-------------------|-------------------|
| | | 2011 | 2016 |
| Alberta | 13 | 2,538,547 | 2,891,712 |
| British Columbia | 14 | 3,254,203 | 3,471,292 |
| Manitoba | 2 | 715,649 | 760,249 |
| New Brunswick | 4 | 244,910 | 250,811 |
| Newfoundland and Labrador | 1 | 172,312 | 178,427 |
| Nova Scotia | 2 | 335,154 | 346,605 |
| Ontario | 39 | 10,140,286 | 10,659,522 |
| Prince Edward Island | 1 | 41,613 | 44,739 |
| Quebec | 20 | 5,140,554 | 5,337,846 |
| Saskatchewan | 4 | 479,228 | 527,638 |
| Canada | 100 | 23,062,456 | 24,468,841 |

Footnote:

There are other cities in Canada not included in this table; the excluded cities are those smaller than the 100 largest cities in Canada.

1.6.3.6. Ejemplo kableExtra: tablas con colores y diferentes tamaños de letra

Con el uso de la función `cell_spec()` se consigue personalizar las características de determinadas celdas. En este ejemplo, se cambian colores (letra y fondo) y tamaños de fuente (se ayuda de las funciones: `spec_font_size()` y `spec_color()`).

```
library(kableExtra)
iris[1:10, ] %>%
  mutate_if(is.numeric, function(x) {
    cell_spec(x, "latex", bold = T, color = spec_color(x, end = 0.9),
              font_size = spec_font_size(x))
  }) %>%
  mutate(Species = cell_spec(
    Species, "latex", color = "white", bold = T,
    background = spec_color(1:10, end = 0.9, option = "A", direction = -1)
  )) %>%
  kable("latex", escape = F, booktabs = T, linesep = "", align = "c") %>%
  kable_styling(position="center")
```

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 5 | 3.4 | 1.5 | 0.2 | setosa |
| 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 4.9 | 3.1 | 1.5 | 0.1 | setosa |

1.6.3.7. Ejemplo kableExtra: tablas con agrupamiento

En este ejemplo, se añade un línea de cabecera superior a través de la función `add_header_above()`.

```
dt <- mtcars[1:5, 1:6]
kable(dt, "latex", booktabs = T) %>%
  kable_styling() %>%
  add_header_above(c(" " = 1,
                    "Group 1" = 2,
                    "Group 2" = 2,
                    "Group 3" = 2))
```

| | Group 1 | | Group 2 | | Group 3 | |
|-------------------|---------|-----|---------|-----|---------|-------|
| | mpg | cyl | disp | hp | drat | wt |
| Mazda RX4 | 21.0 | 6 | 160 | 110 | 3.90 | 2.620 |
| Mazda RX4 Wag | 21.0 | 6 | 160 | 110 | 3.90 | 2.875 |
| Datsun 710 | 22.8 | 4 | 108 | 93 | 3.85 | 2.320 |
| Hornet 4 Drive | 21.4 | 6 | 258 | 110 | 3.08 | 3.215 |
| Hornet Sportabout | 18.7 | 8 | 360 | 175 | 3.15 | 3.440 |

En este ejemplo, se presentan agrupadas un grupo de filas, con ayuda de la función: `pack_rows()`.

```
kable(mtcars[1:10, 1:6], "latex", booktabs = T) %>%
  kable_styling() %>%
  pack_rows("Group 1", 4, 7) %>%
  pack_rows("Group 2", 8, 10, latex_gap_space = "2cm")
```

| | mpg | cyl | disp | hp | drat | wt |
|-------------------|------|-----|-------|-----|------|-------|
| Mazda RX4 | 21.0 | 6 | 160.0 | 110 | 3.90 | 2.620 |
| Mazda RX4 Wag | 21.0 | 6 | 160.0 | 110 | 3.90 | 2.875 |
| Datsun 710 | 22.8 | 4 | 108.0 | 93 | 3.85 | 2.320 |
| Group 1 | | | | | | |
| Hornet 4 Drive | 21.4 | 6 | 258.0 | 110 | 3.08 | 3.215 |
| Hornet Sportabout | 18.7 | 8 | 360.0 | 175 | 3.15 | 3.440 |
| Valiant | 18.1 | 6 | 225.0 | 105 | 2.76 | 3.460 |
| Duster 360 | 14.3 | 8 | 360.0 | 245 | 3.21 | 3.570 |
| Group 2 | | | | | | |
| Merc 240D | 24.4 | 4 | 146.7 | 62 | 3.69 | 3.190 |
| Merc 230 | 22.8 | 4 | 140.8 | 95 | 3.92 | 3.150 |
| Merc 280 | 19.2 | 6 | 167.6 | 123 | 3.92 | 3.440 |

En este otro ejemplo, se agrupan celdas verticalmente si se repiten valores de una columna. Consideremos los siguientes datos:

```
library(kableExtra)
collapse_rows_dt <- data.frame(
  C1 = c(rep("a", 10), rep("b", 5)),
  C2 = c(rep("c", 7), rep("d", 3), rep("c", 2), rep("d", 3)),
  C3 = 1:15,
  C4 = sample(c(0,1), 15, replace = TRUE))

kable(collapse_rows_dt,booktabs=T,align="c",linesep='') %>%
  column_spec(1, bold=T)
```

| C1 | C2 | C3 | C4 |
|-----------|----|----|----|
| a | c | 1 | 0 |
| a | c | 2 | 1 |
| a | c | 3 | 1 |
| a | c | 4 | 0 |
| a | c | 5 | 0 |
| a | c | 6 | 0 |
| a | c | 7 | 1 |
| a | d | 8 | 1 |
| a | d | 9 | 1 |
| a | d | 10 | 1 |
| b | c | 11 | 1 |
| b | c | 12 | 0 |
| b | d | 13 | 0 |
| b | d | 14 | 1 |
| b | d | 15 | 0 |

En este ejemplo, se agrupan celdas verticalmente con ayuda de la función `collapse_rows()`. También se usa la función `column_spec()` para personalizar la presentación de algunas columnas.

```
row_group_label_fonts <- list(
  list(bold = T, italic = T),
  list(bold = F, italic = F)
)

kable(collapse_rows_dt, "latex",
      booktabs = T, align = "c", linesep = '') %>%
  column_spec(1, bold=T,width = "2cm") %>%
  collapse_rows(1:3, latex_hline = 'custom', custom_latex_hline = 1:3,
               row_group_label_position = 'identity', # o 'stack'
               row_group_label_fonts = row_group_label_fonts)
```

| C1 | C2 | C3 | C4 |
|-----------|----|----|----|
| | | 1 | 0 |
| | | 2 | 1 |
| | | 3 | 1 |
| | | 4 | 0 |
| | c | 5 | 0 |
| | | 6 | 0 |
| a | | 7 | 1 |
| | | 8 | 1 |
| | d | 9 | 1 |
| | | 10 | 1 |
| | | 11 | 1 |
| | c | 12 | 0 |
| | | 13 | 0 |
| b | d | 14 | 1 |
| | | 15 | 0 |

1.6.3.8. Ejemplo kableExtra: tablas con notas a pie de página

En este ejemplo, se añaden notas al pie de la tabla, con ayuda de la función: `footnote()`. Según el argumento, el símbolo que se use para identificar las distintas notas a pie de tabla serán diferentes: `number` para números, `alphabet` para letras, etc.

```
library(kableExtra)
dt <- mtcars[1:5, 1:6]
kable(dt, "latex", align = "c", booktabs = T) %>%
  footnote(general = "Here is a general comments of the table. ",
          number = c("Footnote 1; ", "Footnote 2; "),
          alphabet = c("Footnote A; ", "Footnote B; "),
          symbol = c("Footnote Symbol 1; ", "Footnote Symbol 2"),
          general_title = "General: ", number_title = "Type I: ",
          alphabet_title = "Type II: ", symbol_title = "Type III: ",
          footnote_as_chunk = T, title_format = c("italic", "underline")
  )
```

| | mpg | cyl | disp | hp | drat | wt |
|-------------------|------|-----|------|-----|------|-------|
| Mazda RX4 | 21.0 | 6 | 160 | 110 | 3.90 | 2.620 |
| Mazda RX4 Wag | 21.0 | 6 | 160 | 110 | 3.90 | 2.875 |
| Datsun 710 | 22.8 | 4 | 108 | 93 | 3.85 | 2.320 |
| Hornet 4 Drive | 21.4 | 6 | 258 | 110 | 3.08 | 3.215 |
| Hornet Sportabout | 18.7 | 8 | 360 | 175 | 3.15 | 3.440 |

General: Here is a general comments of the table.

Type I: ¹ Footnote 1; ² Footnote 2;

Type II: ^a Footnote A; ^b Footnote B;

Type III: ^{*} Footnote Symbol 1; [†] Footnote Symbol 2

1.6.3.9. Ejemplo kableExtra: saltos de líneas en celdas

A diferencia de HTML, donde puede usar `
` en cualquier momento, en LaTeX, en realidad es bastante difícil hacer un salto de línea en una tabla. Para ello se debe utilizar la función `linebreak()` para facilitar este proceso. Para más detalles, ver: [Best Practice for Newline in LaTeX Table](#).

```
dt_lb <- data.frame(
  Item = c("Hello\nWorld", "This\nis a cat"),
  Value = c(10, 100)
)

dt_lb %>%
  mutate_all(linebreak) %>%
  kable("latex", booktabs = T, escape = F,
        col.names = linebreak(c("Item\n(Name)",
                                "Value\n(Number)"), align = "c"))
```

| Item (Name) | Value (Number) |
|------------------|-------------------|
| Hello World | 10 |
| This is a cat | 100 |

1.6.4. Crear tablas con la librería huxtable

Se puede profundizar más en esta librería en su [página oficial: huxtable](#) y en la [página de documentación](#).

```
library(knitr)
library(dplyr)
library(huxtable)
options(huxtable.knitr_print_df = FALSE)

is_latex <- guess_knitr_output_format() == 'latex'
# is_latex <- TRUE
knitr::knit_hooks$set(
  barrier = function(before, options, envir) {
    if (! before && is_latex) knitr::asis_output('\\FloatBarrier')
  }
)

if (is_latex) knitr::opts_chunk$set(barrier = TRUE)

huxtable::hux_logo(latex = is_latex, html = ! is_latex)
```

| | | | | |
|---|---|---|---|---|
| | | h | u | |
| | x | t | | a |
| b | | l | | |
| | | | | e |
| | | | | |

Huxtable es un paquete para crear *tablas de texto*. Es potente, pero fácil de usar. Las características de Huxtable incluyen:

- Exportar a LaTeX, HTML, Microsoft Word, Microsoft Excel, Microsoft Powerpoint, RTF y Markdown
- Fácil integración con documentos knitr y rmarkdown
- Visualización en consola formateada
- Celdas de varias filas y varias columnas
- Control detallado sobre el fondo de la celda, el espaciado, la alineación, el tamaño y los bordes
- Control sobre la fuente del texto, estilo, tamaño, color, alineación, formato de número y rotación
- Manipulación de tablas usando subconjuntos R estándar, o funciones dplyr como `filter` y `select`
- Formato condicional fácil basado en el contenido de la tabla
- Temas de tabla rápida
- Creación automática de tablas de salida de regresión con la función `huxreg`

1.6.4.1. Primeros pasos

Para crear un huxtable, use la función `huxtable`, `ohux` para abreviar. Esto funciona muy parecido a un `data.frame`:

```
library(huxtable)
ht <- hux(
  Employee      = c('John Smith', 'Jane Doe', 'David Hugh-Jones'),
  Salary        = c(50000L, 50000L, 40000L),
  add_colnames = TRUE
)
```

Puede convertirse un `data.frame` a huxtable con la función `as_hux`.

```
data(mtcars)
car_ht <- as_hux(mtcars)
```

Para imprimir un huxtable como LaTeX o HTML, simplemente llamar a `print_latex` o `print_html`. En documentos R Markdown, como este, simplemente puede evaluar el huxtable:

| Employee | Salary |
|------------------|--------|
| John Smith | 50000 |
| Jane Doe | 50000 |
| David Hugh-Jones | 40000 |

La salida predeterminada es una tabla simple. Lo personalizaremos modificando los encabezados de la tabla en negrita, dibujar una línea debajo de la fila del encabezado, y agregar espacio horizontal a las celdas y cambiar el formato de los números.

Para hacer esto, estableceremos algunas **propiedades** en las celdas de la tabla. Establece propiedades se asignan al nombre de la propiedad, tal como se asignan en el sistema base R: `nombres(x) <- nuevos_nombres`.

```
right_padding(ht) <- 10
left_padding(ht) <- 10
number_format(ht) <- 2 # 2 decimal places
bold(ht)[1, ] <- TRUE
bottom_border(ht)[1, ] <- 1
caption(ht) <- 'Employee table'
```

Después de establecer esos cambios, el objeto huxtable mostrará la tabla actualiza:

Tabla 1.21: Employee table

| Employee | Salary |
|------------------|---------------|
| John Smith | 50000.00 |
| Jane Doe | 50000.00 |
| David Hugh-Jones | 40000.00 |

Con estilo tidyverse se puede especificar lo mismo del siguiente modo:

```
library(dplyr)
hux(
  Employee      = c('John Smith', 'Jane Doe', 'David Hugh-Jones'),
  Salary        = c(50000, 50000, 40000),
  add_colnames = TRUE
)
set_right_padding(10)
set_left_padding(10)
set_bold(1, 1:2, TRUE)
set_bottom_border(1, 1:2, 1)
set_align(1:4, 2, 'right')
set_number_format(2)
set_caption('Employee table')
```

Tabla 1.22: Employee table

| Employee | Salary |
|------------------|---------------|
| John Smith | 50000.00 |
| Jane Doe | 50000.00 |
| David Hugh-Jones | 40000.00 |

La tabla siguiente muestra una lista completa de propiedades.

```
sides <- c('left_', 'right_', 'top_', 'bottom_')
props <- list()
props[['Cell_Text']] <- sort(c('font', 'text_color', 'wrap', 'bold',
                              'italic', 'font', 'font_size', 'na_string',
                              'escape_contents', 'number_format',
                              'rotation'))

props[['Cell']] <- sort(c('align', 'valign', 'rowspan', 'colspan',
                          'background_color',
                          paste0(sides, 'border'),
                          paste0(sides, 'border_color'), paste0(sides, 'border_style'),
                          paste0(sides, 'padding')))
props[['Row']] <- 'row_height'
props[['Column']] <- 'col_width'
props[['Table']] <- sort(c('width', 'height', 'position', 'caption',
                           'caption_pos', 'tabular_environment', 'label',
                           'latex_float'))

maxl <- max(sapply(props, length))
props <- lapply(props, function(x) c(x, rep('', maxl - length(x))))

ss_font <- if (guess_knitr_output_format() == 'latex') 'cmtt' else 'courier'
```

Se puede usar el estilo “tidyverse” como sigue:

```
prop_hux <- hux(as.data.frame(props)) %>%
  add_colnames %>%
  {foo <- .; foo[1,] <- gsub('_', ' ', foo[1,]); foo} %>%
  set_font(-1, everywhere, ss_font) %>%
  set_font_size( 10) %>%
  set_bold(1, everywhere, TRUE) %>%
  set_width(0.9) %>%
  set_background_color(-1, evens, grey(.9)) %>%
  set_outer_borders(1) %>%
  set_bottom_border(1, everywhere, 1) %>%
  set_top_padding(2) %>%
  set_bottom_padding(4) %>%
  set_caption('Huxtable properties') %>%
  set_position('left') %>%
  set_col_width(c(.2, .25, .15, .15, .25))
```

```
prop_hux
```

Tabla 1.23: Huxtable properties

| Cell Text | Cell | Row | Column | Table |
|-----------------|---------------------|------------|-----------|---------------------|
| bold | align | row_height | col_width | caption |
| escape_contents | background_color | | | caption_pos |
| font | bottom_border | | | height |
| font | bottom_border_color | | | label |
| font_size | bottom_border_style | | | latex_float |
| italic | bottom_padding | | | position |
| na_string | colspan | | | tabular_environment |
| number_format | left_border | | | width |
| rotation | left_border_color | | | |
| text_color | left_border_style | | | |
| wrap | left_padding | | | |
| | right_border | | | |
| | right_border_color | | | |
| | right_border_style | | | |
| | right_padding | | | |
| | rowspan | | | |
| | top_border | | | |
| | top_border_color | | | |
| | top_border_style | | | |
| | top_padding | | | |
| | valign | | | |

1.6.4.2. Ejemplo huxtable: caracteres especiales

```
library(huxtable)
is_latex = T
code_ht <- if (is_latex) hux(c("Some maths", "$a^b$")) else
  hux(c("Copyright symbol", "&copy;"))

theme_article(code_ht)
```

| |
|------------|
| Some maths |
| \$a^b\$ |

```
escape_contents(code_ht)[2, 1] <- FALSE

theme_article(code_ht)
```

| |
|------------|
| Some maths |
| a^b |

1.6.4.3. Temas con huxtable

Las funciones que permiten cambiar el aspecto de la tabla que se quiere presentar con “huxtable” son las funciones del tipo: `theme_*()` (`theme_plain()`, `theme_stripped()`, etc). Los siguientes ejemplos se han extraído de la documentación de la librería “huxtable”.

```
library(huxtable)
jams <- cbind(jams, c("Sugar", "60%", "60%", "60%"))
latex_float(jams) = "H" # necesita paquete TeX: "float"
jams
```

| Type | Price | Sugar |
|------------|-------|---------|
| Strawberry | 1.90 | 60.00 % |
| Raspberry | 2.10 | 60.00 % |
| Plum | 1.80 | 60.00 % |

```
theme_plain(jams)
```

| Type | Price | Sugar |
|------------|-------|---------|
| Strawberry | 1.90 | 60.00 % |
| Raspberry | 2.10 | 60.00 % |
| Plum | 1.80 | 60.00 % |

`theme_basic(jams)`

| Type | Price | Sugar |
|------------|-------|---------|
| Strawberry | 1.90 | 60.00 % |
| Raspberry | 2.10 | 60.00 % |
| Plum | 1.80 | 60.00 % |

`theme_stripped(jams)`

| Type | Price | Sugar |
|------------|-------|---------|
| Strawberry | 1.90 | 60.00 % |
| Raspberry | 2.10 | 60.00 % |
| Plum | 1.80 | 60.00 % |

`theme_article(jams)`

| Type | Price | Sugar |
|------------|-------|---------|
| Strawberry | 1.90 | 60.00 % |
| Raspberry | 2.10 | 60.00 % |
| Plum | 1.80 | 60.00 % |

`theme_grey(jams)`

| Type | Price | Sugar |
|------------|-------|---------|
| Strawberry | 1.90 | 60.00 % |
| Raspberry | 2.10 | 60.00 % |
| Plum | 1.80 | 60.00 % |

`theme_blue(jams)`

| Type | Price | Sugar |
|------------|-------|---------|
| Strawberry | 1.90 | 60.00 % |
| Raspberry | 2.10 | 60.00 % |
| Plum | 1.80 | 60.00 % |

`theme_orange(jams)`

| Type | Price | Sugar |
|------------|-------|---------|
| Strawberry | 1.90 | 60.00 % |
| Raspberry | 2.10 | 60.00 % |
| Plum | 1.80 | 60.00 % |

`theme_green(jams)`

| Type | Price | Sugar |
|------------|-------|---------|
| Strawberry | 1.90 | 60.00 % |
| Raspberry | 2.10 | 60.00 % |
| Plum | 1.80 | 60.00 % |

`theme_mondrian(jams)`

| Type | Price | Sugar |
|------------|-------|---------|
| Strawberry | 1.90 | 60.00 % |
| Raspberry | 2.10 | 60.00 % |
| Plum | 1.80 | 60.00 % |

1.6.5. Crear tablas con la librería `pander`

El uso de la función `pandoc.table()` permite construir tablas al convertirlas en modo “markdown”. Para que se vean es necesario añadir en las opciones del código R: `results='asis'`.

- **Tablas multilínea:** `keep.line.breaks` a `TRUE` (no funciona en “pdf”):

```
library(pander)
m <- data.frame('Value\n1', 'Value\n2')
colnames(m) <- c('Multiline\nCol1', 'Multiline\nCol2')
pandoc.table(m, keep.line.breaks = TRUE)
```

| Multiline Col1 | Multiline Col2 |
|----------------|----------------|
| Value 1 | Value 2 |

```
m <- mtcars[1:3, 1:4]
pandoc.table(m)
```

| | mpg | cyl | disp | hp |
|---------------|------|-----|------|-----|
| Mazda RX4 | 21 | 6 | 160 | 110 |
| Mazda RX4 Wag | 21 | 6 | 160 | 110 |
| Datsun 710 | 22.8 | 4 | 108 | 93 |

- **Tablas simples:**

```
m <- mtcars[1:3, 1:4]
pandoc.table(m, style = 'simple')
```

| | mpg | cyl | disp | hp |
|---------------|------|-----|------|-----|
| Mazda RX4 | 21 | 6 | 160 | 110 |
| Mazda RX4 Wag | 21 | 6 | 160 | 110 |
| Datsun 710 | 22.8 | 4 | 108 | 93 |

```
m <- data.frame('Value\n1', 'Value\n2')
colnames(m) <- c('Multiline\nCol1', 'Multiline\nCol2')
pandoc.table(m, keep.line.breaks = TRUE, style='simple')
```

Esta tabla no se crea al producirse errores.

- Tablas grid:

```
m <- mtcars[1:3, 1:4]
pandoc.table(m, style = 'grid')
```

| | mpg | cyl | disp | hp |
|---------------|------|-----|------|-----|
| Mazda RX4 | 21 | 6 | 160 | 110 |
| Mazda RX4 Wag | 21 | 6 | 160 | 110 |
| Datsun 710 | 22.8 | 4 | 108 | 93 |

```
m <- data.frame('Value\n1', 'Value\n2')
colnames(m) <- c('Multiline\nCol1', 'Multiline\nCol2')
pandoc.table(m, keep.line.breaks = TRUE, style='grid')
```

| Multiline Col1 | Multiline Col2 |
|----------------|----------------|
| Value 1 | Value 2 |

- Tablas R markdown:

```
m <- mtcars[1:3, 1:4]
pandoc.table(m, style = 'rmarkdown')
```

| | mpg | cyl | disp | hp |
|---------------|------|-----|------|-----|
| Mazda RX4 | 21 | 6 | 160 | 110 |
| Mazda RX4 Wag | 21 | 6 | 160 | 110 |
| Datsun 710 | 22.8 | 4 | 108 | 93 |

```
m <- data.frame('Value\n1', 'Value\n2')
colnames(m) <- c('Multiline\nCol1', 'Multiline\nCol2')
pandoc.table(m, keep.line.breaks = TRUE, style='rmarkdown')
```

Esta tabla no se crea al producirse errores.

- Alineación de celdas

```
pandoc.table(head(iris[,1:3], 2), justify = 'right')
```

| Sepal.Length | Sepal.Width | Petal.Length |
|--------------|-------------|--------------|
| 5.1 | 3.5 | 1.4 |
| 4.9 | 3 | 1.4 |

```
pandoc.table(head(iris[,1:3], 2), justify = c('right', 'center', 'left'))
```

| Sepal.Length | Sepal.Width | Petal.Length |
|--------------|-------------|--------------|
| 5.1 | 3.5 | 1.4 |

| Sepal.Length | Sepal.Width | Petal.Length |
|--------------|-------------|--------------|
| 4.9 | 3 | 1.4 |

Otro modo de hacerlo es:

```
set.alignment('left', row.names = 'right')
# set only for next table since permanent parameter is false
pandoc.table(mtcars[1:2, 1:5])
```

| | mpg | cyl | disp | hp | drat |
|---------------|-----|-----|------|-----|------|
| Mazda RX4 | 21 | 6 | 160 | 110 | 3.9 |
| Mazda RX4 Wag | 21 | 6 | 160 | 110 | 3.9 |

1.6.5.1. Personalizando celdas

```
panderOptions('table.alignment.default',
  function(df)
    ifelse(sapply(df, mean) > 2, 'left', 'right'))
pandoc.table(head(iris[,1:3], 2))
```

| Sepal.Length | Sepal.Width | Petal.Length |
|--------------|-------------|--------------|
| 5.1 | 3.5 | 1.4 |
| 4.9 | 3 | 1.4 |

```
panderOptions('table.alignment.default', 'center')
```

Resaltando celdas

panderOptions pueden ser:

- `emphasize.italics.rows`
- `emphasize.italics.cols`
- `emphasize.italics.cells`
- `emphasize.strong.rows`
- `emphasize.strong.cols`
- `emphasize.strong.cells`
- `emphasize.verbatim.rows`
- `emphasize.verbatim.cols`
- `emphasize.verbatim.cells`

```
t <- mtcars[1:3, 1:5]
emphasize.italics.cols(1)
emphasize.italics.rows(1)
emphasize.strong.cells(which(t > 20, arr.ind = TRUE))
pandoc.table(t)
```

| | mpg | cyl | disp | hp | drat |
|---------------|-------------|-----|------------|------------|------|
| Mazda RX4 | 21 | 6 | 160 | 110 | 3.9 |
| Mazda RX4 Wag | 21 | 6 | 160 | 110 | 3.9 |
| Datsun 710 | 22.8 | 4 | 108 | 93 | 3.85 |

```
pandoc.table(t, emphasize.verbatim.rows = 1,
              emphasize.strong.cells = which(t > 20, arr.ind = TRUE))
```

| | mpg | cyl | disp | hp | drat |
|---------------|-------------|-----|------------|------------|------|
| Mazda RX4 | 21 | 6 | 160 | 110 | 3.9 |
| Mazda RX4 Wag | 21 | 6 | 160 | 110 | 3.9 |
| Datsun 710 | 22.8 | 4 | 108 | 93 | 3.85 |

Anchura de tablas y celdas:

```
pandoc.table(mtcars[1:2, ], style = "grid", caption = "Wide table to be split!")
```

Tabla 1.36: Wide table to be split! (continued below)

| | mpg | cyl | disp | hp | drat | wt | qsec |
|------------------|-----|-----|------|-----|------|-------|-------|
| Mazda RX4 | 21 | 6 | 160 | 110 | 3.9 | 2.62 | 16.46 |
| Mazda RX4 Wag | 21 | 6 | 160 | 110 | 3.9 | 2.875 | 17.02 |

| | vs | am | gear | carb |
|---------------|----|----|------|------|
| Mazda RX4 | 0 | 1 | 4 | 4 |
| Mazda RX4 Wag | 0 | 1 | 4 | 4 |

`split.table` por defecto a 80 caracteres y para desactivar su efecto, definir `split.table` a `Inf`:

```
pandoc.table(mtcars[1:2, ], style = "grid",
              caption = "Wide table to be split!", split.table = Inf)
```

Tabla 1.38: Wide table to be split!

| | mpg | cyl | disp | hp | drat | wt | qsec | vs | am | gear | carb |
|------------------|-----|-----|------|-----|------|-------|-------|----|----|------|------|
| Mazda RX4 | 21 | 6 | 160 | 110 | 3.9 | 2.62 | 16.46 | 0 | 1 | 4 | 4 |
| Mazda RX4 Wag | 21 | 6 | 160 | 110 | 3.9 | 2.875 | 17.02 | 0 | 1 | 4 | 4 |

```
df <- data.frame(a = 'Lorem ipsum', b = 'dolor sit', c = 'amet')
pandoc.table(df, split.cells = 5)
```

| a | b | c |
|----------------|-----------|------|
| Lorem ipsum | dolor sit | amet |

```
pandoc.table(df, split.cells = c(5, 20, 5))
```

| a | b | c |
|----------------|-----------|------|
| Lorem ipsum | dolor sit | amet |

```
pandoc.table(df, split.cells = c("80%", "10%", "10%"))
```

| a | b | c |
|-------------|-----------|------|
| Lorem ipsum | dolor sit | amet |

```
pandoc.table(df, split.cells = 5, style = 'simple')
```

| a | b | c |
|-------------|-----------|------|
| Lorem ipsum | dolor sit | amet |

```
pandoc.table(data.frame(baz = 'foobar', foo='accoutrements'),
  use.hyphening = TRUE, split.cells = 3)
```

| baz | foo |
|--------|---------------|
| foobar | accoutrements |

1.6.5.2. Redondear y formatear números

`pander/pandoc.table` trata con números formateados al especificar 4 parámetros:

- `round` para el número de cifras decimales.
- `digits` para especificar cuantos dígitos significativos serán usados por los números.
- `decimal.mark/big.mark` para especificar el carácter para el punto decimal/orden de magnitud (miles, millones, etc).

`round` y `digits` pueden ser un vector de valores específicos para cada columna (tiene que ser de la misma longitud que el número de columnas). Los valores para columnas no numéricas serán descartados.

Veamos algunos ejemplos:


```
r <- matrix(c(283764.97430, 29.12345678901, -7.1234, -100.1), ncol = 2)
pandoc.table(r, round = 2)
```

| | |
|--------|--------|
| 283765 | -7.12 |
| 29.12 | -100.1 |

```
pandoc.table(r, round = c(4,2)) # vector for each column
```

| | |
|--------|--------|
| 283765 | -7.12 |
| 29.12 | -100.1 |

```
pandoc.table(r, digits = 2)
```

| | |
|--------|------|
| 283765 | -7.1 |
| 29 | -100 |

```
pandoc.table(r, digits = c(0, 5)) # vector for each column
```

| | |
|--------|---------|
| 3.e+05 | -7.1234 |
| 29 | -100.1 |

```
pandoc.table(r, big.mark = ',')
```

| | |
|---------|--------|
| 283,765 | -7.123 |
| 29.12 | -100.1 |

```
pandoc.table(r, decimal.mark = ',')
```

| | |
|--------|--------|
| 283765 | -7,123 |
| 29,12 | -100,1 |

Otras opciones:

La funcionalidad descrita en otras secciones es más notable, pero `pander/pandoc.table` también tiene características ingeniosas más pequeñas que vale la pena mencionar:

- `plain.ascii` - permite tener la salida sin código markdown:

```
pandoc.table(mtcars[1:3, 1:4])
```

| | mpg | cyl | disp | hp |
|----------------------|------|-----|------|-----|
| Mazda RX4 | 21 | 6 | 160 | 110 |
| Mazda RX4 Wag | 21 | 6 | 160 | 110 |
| Datsun 710 | 22.8 | 4 | 108 | 93 |

```
pandoc.table(mtcars[1:3, 1:4], plain.ascii = TRUE)
```

| | mpg | cyl | disp | hp |
|---------------|------|-----|------|-----|
| Mazda RX4 | 21 | 6 | 160 | 110 |
| Mazda RX4 Wag | 21 | 6 | 160 | 110 |
| Datsun 710 | 22.8 | 4 | 108 | 93 |

- `caption` - define la leyenda (cadena) que se mostrará con la tabla:

```
pandoc.table(mtcars[1:3, 1:5], style = "grid", caption = "My caption!")
```

Tabla 1.52: My caption!

| | mpg | cyl | disp | hp | drat |
|----------------------|------|-----|------|-----|------|
| Mazda RX4 | 21 | 6 | 160 | 110 | 3.9 |
| Mazda RX4 Wag | 21 | 6 | 160 | 110 | 3.9 |
| Datsun 710 | 22.8 | 4 | 108 | 93 | 3.85 |

- `missing` - define una cadena que reemplazará los valores missing:

```
m <- mtcars[1:3, 1:5]
m$mpg <- NA
pandoc.table(m, missing = '?')
```

| | mpg | cyl | disp | hp | drat |
|----------------------|-----|-----|------|-----|------|
| Mazda RX4 | ? | 6 | 160 | 110 | 3.9 |
| Mazda RX4 Wag | ? | 6 | 160 | 110 | 3.9 |
| Datsun 710 | ? | 4 | 108 | 93 | 3.85 |

1.6.5.3. Ejemplos pander: personalizando los resultados de modelos estadísticos

Salidas de la librería “descr”:

```
library(pander)
library(descr, quietly = TRUE)
pander(CrossTable(mtcars$gear, mtcars$cyl))
```

| | mtcars\$cyl | | | |
|--------------|-------------|-----------|-----------|-----------|
| mtcars\$gear | 4 | 6 | 8 | Total |
| 3 | | | | |
| N | 1 | 2 | 12 | 15 |
| Chi-square | 3.3502 | 0.5003 | 4.5054 | |
| Row(%) | 6.6667 % | 13.3333 % | 80.0000 % | 46.8750 % |
| Column(%) | 9.0909 % | 28.5714 % | 85.7143 % | |
| Total(%) | 3.125 % | 6.250 % | 37.500 % | |

| mtcars\$gear | mtcars\$cyl | | | Total |
|--------------|-------------|-----------|-----------|-----------|
| | 4 | 6 | 8 | |
| 4 | | | | |
| N | 8 | 4 | 0 | 12 |
| Chi-square | 3.6402 | 0.7202 | 5.2500 | |
| Row(%) | 66.6667 % | 33.3333 % | 0.0000 % | 37.5000 % |
| Column(%) | 72.7273 % | 57.1429 % | 0.0000 % | |
| Total(%) | 25.000 % | 12.500 % | 0.000 % | |
| 5 | | | | |
| N | 2 | 1 | 2 | 5 |
| Chi-square | 0.0460 | 0.0080 | 0.0161 | |
| Row(%) | 40.0000 % | 20.0000 % | 40.0000 % | 15.6250 % |
| Column(%) | 18.1818 % | 14.2857 % | 14.2857 % | |
| Total(%) | 6.250 % | 3.125 % | 6.250 % | |
| Total | 11 | 7 | 14 | 32 |
| | 34.375 % | 21.875 % | 43.75 % | |

Salidas de la librería “tables”:

```
library(tables, quietly = TRUE)
tab <- tabular( (Species + 1) ~ (n=1) + Format(digits=2)*
  (Sepal.Length + Sepal.Width)*(mean + sd), data=iris )
pander(tab)
```

| Species | n | Sepal.Length | | Sepal.Width | |
|-------------------|-----|--------------|------|-------------|------|
| | | mean | sd | mean | sd |
| <i>setosa</i> | 50 | 5.01 | 0.35 | 3.43 | 0.38 |
| <i>versicolor</i> | 50 | 5.94 | 0.52 | 2.77 | 0.31 |
| <i>virginica</i> | 50 | 6.59 | 0.64 | 2.97 | 0.32 |
| <i>All</i> | 150 | 5.84 | 0.83 | 3.06 | 0.44 |

Construyendo tablas dentro de funciones vectorizadas o bucles.

Lo siguiente no funciona correctamente, no mostraría la información del modo deseado:

```
dfs <- list(mtcars[1:3, 1:4], mtcars[4:6, 1:4], mtcars[7:9, 1:4])
lapply(dfs, pander)
```

Para que funcione se tiene que especificar: `results='asis'` y `panderOptions('knitr.auto.asis', FALSE)`:

```
panderOptions('knitr.auto.asis', FALSE)
dfs <- list(mtcars[1:3, 1:4], mtcars[4:6, 1:4], mtcars[7:9, 1:4])
invisible(lapply(dfs, pander))
```

| | mpg | cyl | disp | hp |
|----------------------|-----|-----|------|-----|
| Mazda RX4 | 21 | 6 | 160 | 110 |
| Mazda RX4 Wag | 21 | 6 | 160 | 110 |

| | mpg | cyl | disp | hp |
|-------------------|------|-----|------|----|
| Datsun 710 | 22.8 | 4 | 108 | 93 |

| | mpg | cyl | disp | hp |
|--------------------------|------|-----|------|-----|
| Hornet 4 Drive | 21.4 | 6 | 258 | 110 |
| Hornet Sportabout | 18.7 | 8 | 360 | 175 |
| Valiant | 18.1 | 6 | 225 | 105 |

| | mpg | cyl | disp | hp |
|-------------------|------|-----|-------|-----|
| Duster 360 | 14.3 | 8 | 360 | 245 |
| Merc 240D | 24.4 | 4 | 146.7 | 62 |
| Merc 230 | 22.8 | 4 | 140.8 | 95 |

1.6.5.4. Ejemplo pander: con leyendas y referencias

```

created_var <- tibble(
  `Created/modified variable` = c('Years since BCG',
    'Age at BCG',
    'Successful treatment',
    'Mortality',
    'TB mortality',
    'Death due to TB'),
  Description = c('Derived using year of vaccination and year of
    notification. Categorized into  $\leq 10$  and 11+ due
    to the evidence of waning protection for the BCG
    vaccine. [R-dplyr]',
    'Derived using year of vaccination and age at
    vaccination. Categorized into < 1, 1 to x < 12, 12 to
    x < 16 and  $\leq 16$  to capture historic vaccination
    policy. [R-xtable]',
    'For cases that had a recorded date of starting
    treatment, with their outcome recorded at the latest
    available follow up. Those that completed treatment
    are defined as successfully treated: treatment failure
    is defined as those that stopped treatment, were lost
    to follow up, those that died during follow up from
    TB, those that died during follow up were TB
    contributed to their death, and those who were still
    on treatment. Those that were not evaluated were
    treated as missing.',
    'Assessed via follow up at 12 and 24 months: mortality
    is defined as cases with an overall outcome of death,
    and survival is defined as those that completed
  
```

```

treatment, were still on treatment, and stopped
treatment. Those that were lost to follow up, or not
evaluated were treated as missing',
'For cases with an overall outcome of died, and whose
cause of death was known to be TB or to be related to
TB. Those that were known to have not died, or who
were known to have died from a cause other than from
TB were defined to have not died from TB.',
'Death due to TB is defined as those that died
directly from TB, or where TB had contributed to their
death with death not due to TB being cases that died
from any other cause. Conditioned on all-cause
mortality, for cases with a known cause of death.')
)

pander::pander(created_var, caption = '\\label{tab:created-var} Variables
derived or modified from the ETS system for use in the
analyses throughout this thesis.',
booktabs = TRUE,
justify = 'left', split.cells = 50)

```

Tabla 1.59: Variables derived or modified from the ETS system for use in the analyses throughout this thesis.

| Created/modified variable | Description |
|---------------------------|--|
| Years since BCG | Derived using year of vaccination and year of notification. Categorized into ≤ 10 and $11+$ due to the evidence of waning protection for the BCG vaccine. ^[14] |
| Age at BCG | Derived using year of vaccination and age at vaccination. Categorized into < 1 , 1 to $x < 12$, 12 to $x < 16$ and ≤ 16 to capture historic vaccination policy. ^[2] |
| Successful treatment | For cases that had a recorded date of starting treatment, with their outcome recorded at the latest available follow up. Those that completed treatment are defined as successfully treated: treatment failure is defined as those that stopped treatment, were lost to follow up, those that died during follow up from TB, those that died during follow up were TB contributed to their death, and those who were still on treatment. |
| Mortality | Those that were not evaluated were treated as missing. Assessed via follow up at 12 and 24 months: mortality is defined as cases with an overall outcome of death, and survival is defined as those that completed treatment, were still on treatment, and stopped treatment. Those that were lost to follow up, or not evaluated were treated as missing |

| Created/modified variable | Description |
|---------------------------|--|
| TB mortality | For cases with an overall outcome of died, and whose cause of death was known to be TB or to be related to TB. Those that were known to have not died, or who were known to have died from a cause other than from TB were defined to have not died from TB. |
| Death due to TB | Death due to TB is defined as those that died directly from TB, or where TB had contributed to their death with death not due to TB being cases that died from any other cause. Conditioned on all-cause mortality, for cases with a known cause of death. |

1.6.6. Crear tablas con la librería stargazer

La librería “stargazer” está especializada en construir tablas preformateadas para presentar modelos estadísticos. Veamos con algunos ejemplos el uso de la función principal de este paquete: `stargazer()`.

1.6.6.1. Ejemplo stargazer: tablas con resúmenes estadísticos descriptivos

```
library(stargazer)
stargazer(attitude,header = FALSE,table.placement = "H")
```

Tabla 1.60

| Statistic | N | Mean | St. Dev. | Min | Pctl(25) | Pctl(75) | Max |
|-------------|----|--------|----------|-----|----------|----------|-----|
| rating | 30 | 64.633 | 12.173 | 40 | 58.8 | 71.8 | 85 |
| complaints | 30 | 66.600 | 13.315 | 37 | 58.5 | 77 | 90 |
| privileges | 30 | 53.133 | 12.235 | 30 | 45 | 62.5 | 83 |
| learning | 30 | 56.367 | 11.737 | 34 | 47 | 66.8 | 75 |
| raises | 30 | 64.633 | 10.397 | 43 | 58.2 | 71 | 88 |
| critical | 30 | 74.767 | 9.895 | 49 | 69.2 | 80 | 92 |
| advance | 30 | 42.933 | 10.289 | 25 | 35 | 47.8 | 72 |
| high.rating | 30 | 0.333 | 0.479 | 0 | 0 | 1 | 1 |

1.6.6.2. Ejemplo stargazer: tablas con resúmenes estadísticos de modelos lineales

```
## 2 OLS models
linear.1 <- lm(rating ~ complaints + privileges + learning + raises + critical,
              data=attitude)
linear.2 <- lm(rating ~ complaints + privileges + learning,
              data=attitude)
## create an indicator dependent variable, and run a probit model
attitude$high.rating <- (attitude$rating > 70)

probit.model <- glm(high.rating ~ learning + critical + advance,
                   data=attitude,
                   family = binomial(link = "probit"))

stargazer(linear.1, linear.2, probit.model,
          title="Results",
          align=TRUE,header = FALSE,
          table.placement = "H")
```

Tabla 1.61: Results

| | <i>Dependent variable:</i> | | |
|-------------------------|----------------------------|------------------------|---------------------|
| | rating | | high.rating |
| | <i>OLS</i> | | <i>probit</i> |
| | (1) | (2) | (3) |
| complaints | 0.692*** (0.149) | 0.682*** (0.129) | |
| privileges | -0.104 (0.135) | -0.103 (0.129) | |
| learning | 0.249 (0.160) | 0.238* (0.139) | 0.164*** (0.053) |
| raises | -0.033 (0.202) | | |
| critical | 0.015 (0.147) | | -0.001 (0.044) |
| advance | | | -0.062 (0.042) |
| Constant | 11.011 (11.704) | 11.258 (7.318) | -7.476** (3.570) |
| Observations | 30 | 30 | 30 |
| R ² | 0.715 | 0.715 | |
| Adjusted R ² | 0.656 | 0.682 | |
| Log Likelihood | | | -9.087 |
| Akaike Inf. Crit. | | | 26.175 |
| Residual Std. Error | 7.139 (df = 24) | 6.863 (df = 26) | |
| F Statistic | 12.063*** (df = 5; 24) | 21.743*** (df = 3; 26) | |

Note:

*p<0.1; **p<0.05; ***p<0.01

1.6.7.2. Tablas interactivas

1.6.7.2.1. Crear tablas con la librería rhandsontable

Solamente se muestra un sencillo ejemplo de uso. Ver más información en la [página oficial de rhandsontable](#).

```
my_tbl2 <- tibble::tribble(
  ~Col_1, ~Col_2, ~Col_3,
    1,     5,     9,
    2,     6,    10,
    3,     7,    11,
    4,     8,    12
)

require(rhandsontable)
rhandsontable(my_tbl2 <- tibble::tribble( ~Col_1, ~Col_2, ~Col_3,
                                           1,     5,     9,
                                           2,     6,    10,
                                           3,     7,    11,
                                           4,     8,    12 ),
  rowHeaders = NULL,
  digits = 3, useTypes = FALSE, search = FALSE,
  width = NULL, height = NULL)
```

| Col_1 | Col_2 | Col_3 |
|-------|-------|-------|
| 1 | 5 | 9 |
| 2 | 6 | 10 |
| 3 | 7 | 11 |
| 4 | 8 | 12 |

1.6.7.2.2. Crear tablas con la librería DT

Solamente se muestra un sencillo ejemplo de uso. Ver más información en la [página oficial de DT](#).

```
my_tbl2 <- tibble::tribble(
  ~Col_1, ~Col_2, ~Col_3,
    1,     5,     9,
    2,     6,    10,
    3,     7,    11,
    4,     8,    12
)
```

```
require(DT)
datatable(my_tbl2 <- tibble::tribble( ~Col_1, ~Col_2, ~Col_3,
                                       1,     5,     9,
                                       2,     6,    10,
                                       3,     7,    11,
                                       4,     8,    12 ),
  rownames = FALSE, caption = NULL,
  filter = "top", escape = FALSE, style = "default",
  width = NULL, height = NULL)
```

Show entries

Search:

| | Col_1 | Col_2 | Col_3 |
|----------------------------------|----------------------------------|----------------------------------|-------|
| <input type="text" value="All"/> | <input type="text" value="All"/> | <input type="text" value="All"/> | |
| | 1 | 5 | 9 |
| | 2 | 6 | 10 |
| | 3 | 7 | 11 |
| | 4 | 8 | 12 |

Showing 1 to 4 of 4 entries

Previous Next

1.7. Crear funciones R personalizadas para crear tablas fácilmente

En algunos documentos es habitual construir tablas de información con unas mismas características. Para evitar tener que repetir más de 3 veces un mismo código sobre un objeto “data.frame” o “tibble” para construir una tabla, se pueden crear funciones que reduzcan el código que se tendría que teclear y además si se decide finalmente cambiar las características comunes en algún aspecto, únicamente habría que modificar la definición de la función creada para que afecte a todo el documento.

A continuación se muestran dos ejemplos de definición de funciones personalizadas, y cómo se usan.

```
library(knitr)
library(dplyr)
library(kableExtra)

#####
##### kableExtra: salidas html y pdf (latex)
#####

func_salida_tablas = function(datos1,salida="latex",landscapelatex=FALSE) {
  if (salida=="latex") {
    if (landscapelatex) {
      datos1 %>%
        kable(salida,booktabs=TRUE) %>%
        kable_styling(latex_options = c("striped", "scale_down")) %>%
        landscape()
    } else {
      datos1 %>%
        kable(salida,longtable=TRUE,booktabs=TRUE) %>%
        kable_styling(latex_options = c("striped","repeat_header"),
                      font_size = 8,
                      repeat_header_text="(contin\\'ua)")
    }
  } else {
    datos1 %>%
      kable(salida) %>%
      kable_styling(bootstrap_options = c("striped", "hover",
                                         "condensed","responsive"),
                    full_width = T)
  }
}
```

Ejemplo de uso de la función `func_salida_tablas()` (aparecen varias formas de hacerlo mismo):

```
#func_salida_tablas(mtcars[1:10,], salida="latex", landscapelatex=FALSE)
#func_salida_tablas(mtcars[1:10,])
mtcars[1:10,] %>%
  func_salida_tablas()
```

| | mpg | cyl | disp | hp | drat | wt | qsec | vs | am | gear | carb |
|-------------------|------|-----|-------|-----|------|-------|-------|----|----|------|------|
| Mazda RX4 | 21.0 | 6 | 160.0 | 110 | 3.90 | 2.620 | 16.46 | 0 | 1 | 4 | 4 |
| Mazda RX4 Wag | 21.0 | 6 | 160.0 | 110 | 3.90 | 2.875 | 17.02 | 0 | 1 | 4 | 4 |
| Datsun 710 | 22.8 | 4 | 108.0 | 93 | 3.85 | 2.320 | 18.61 | 1 | 1 | 4 | 1 |
| Hornet 4 Drive | 21.4 | 6 | 258.0 | 110 | 3.08 | 3.215 | 19.44 | 1 | 0 | 3 | 1 |
| Hornet Sportabout | 18.7 | 8 | 360.0 | 175 | 3.15 | 3.440 | 17.02 | 0 | 0 | 3 | 2 |
| Valiant | 18.1 | 6 | 225.0 | 105 | 2.76 | 3.460 | 20.22 | 1 | 0 | 3 | 1 |
| Duster 360 | 14.3 | 8 | 360.0 | 245 | 3.21 | 3.570 | 15.84 | 0 | 0 | 3 | 4 |
| Merc 240D | 24.4 | 4 | 146.7 | 62 | 3.69 | 3.190 | 20.00 | 1 | 0 | 4 | 2 |
| Merc 230 | 22.8 | 4 | 140.8 | 95 | 3.92 | 3.150 | 22.90 | 1 | 0 | 4 | 2 |
| Merc 280 | 19.2 | 6 | 167.6 | 123 | 3.92 | 3.440 | 18.30 | 1 | 0 | 4 | 4 |

Otro ejemplo de definición de una función personalizada:

```
func_salida_tablas_landscape = function(datos1,salida="latex",
                                       variaspaginas=TRUE,fuente=8) {
  if (salida=="latex") {
    if (variaspaginas) {
      datos1 %>%
        kable(salida,longtable=TRUE,booktabs=TRUE) %>%
        kable_styling(latex_options = c("striped","repeat_header"),
                      font_size = fuente,
                      repeat_header_text="(contin\\'ua)") %>%
        landscape()
    } else {
      datos1 %>%
        kable(salida,booktabs=TRUE) %>%
        kable_styling(latex_options = c("striped", "scale_down")) %>%
        landscape()
    }
  } else {
    datos1 %>%
      kable(salida) %>%
      kable_styling(bootstrap_options = c("striped", "hover",
                                          "condensed","responsive"),
                    full_width = T)
  }
}
```

1.8. Tabla Resumen: construcción de tablas con knitr-kableExtra

A continuación se muestra dos trozos de código R (podrían ir en ficheros distintos) para construir la “tabla resumen apaisada sobre cómo construirla con ayuda de knitr-kableExtra”. **Nota.** Realmente se tuvo que modificar el fichero LaTeX que generaba este código R e incluirla en este documento con ayuda del paquete LaTeX: “pdfpages” y el comando:

```
\includepdf[scale=0.90,pages=1-,angle=90,pagecommand=,
  offset=00 -20]{chuleta_tabla_kableExtra_apaisada_texrevisado.pdf}

\includepdf[scale=0.95,pages=1-,pagecommand=,
  offset=00 -20]{chuleta_tabla_kableExtra_agrupada_texrevisado.pdf}
```

1. **Parte 1.** Introducir la información de cada una de las filas del objeto R que contendría el texto: `tbResumen_`. Este objeto R se grabó en un fichero “`tbResumen_.RDS`”, de forma que pudiese ser reutilizado.

```
tbResumen_ = tibble(
  Tarea = c("Estilo booktabs"),
  kable = c("booktabs=TRUE"),
  kable_styling = c(""),
  otrasfunciones = c("")
) %>%
add_row(
  Tarea = c("Formato salida"),
  kable = c("format='latex' o format='html'"),
  kable_styling = c(""),
  otrasfunciones = c("")
) %>%
add_row(
  Tarea = c("Posición tabla en párrafo"),
  kable = c(""),
  kable_styling = c("position='center' o position='float_right' o position='left'"),
  otrasfunciones = c("")
) %>%
add_row(
  Tarea = c("Ajusta tabla ancho página (NO/SÍ)"),
  kable = c(""),
  kable_styling = c("full_width=FALSE o full_width=TRUE"),
  otrasfunciones = c("")
) %>%
add_row(
  Tarea = c("Tabla Apaisada"),
  kable = c(""),
  kable_styling = c(""),
  otrasfunciones = c("landscape()")
) %>%
add_row(
  Tarea = c("Especificar características columnas"),
  kable = c(""),
  kable_styling = c(""),
  otrasfunciones = c("column_spec(1:2, bold=T, color='red', width='3cm')")
) %>%
add_row(
```

```
Tarea = c("Consejo: Ejemplo Básico-Paquetes"),
kable = c(""),
kable_styling = c(""),
otrasfunciones = '
  library(knitr); library(dplyr); library(kableExtra); \n
  tbResumen %>% \n
    kable(booktabs=TRUE,format="latex") %>% \n
    kable_styling(latex_options = c("striped"), \n
      font_size = 8) %>% \n
    column_spec(1,width = "4cm")
'
```

```
saveRDS(tbResumen_,file="tbResumen_.RDS")
```

2. **Parte 2.** El código R necesario para construir la tabla resumen en modo apaisado.

```
library(knitr)
library(dplyr)
library(kableExtra)
#knitr::opts_chunk$set(echo = FALSE, message=FALSE, warning = FALSE,
#  fig.align = "center", dev = "cairo_pdf", fig.pos = "H")

tbResumen = readRDS("tbResumen_.RDS")
tbResumen %>%
  kable(booktabs=TRUE,format="latex",
    col.names = c("Características de la Tabla",
      "kable()", "kable_styling()",
      "Otras funciones adicionales")) %>%
  kable_styling(latex_options = c("striped"),
    font_size = 8,
    stripe_color = "blue!9",
    stripe_index = seq(2,to = nrow(tbResumen)-1,by = 2)) %>%
  column_spec(1,width = "5cm",bold=T) %>%
  column_spec(2:3,width = "6cm") %>%
  column_spec(4,width = "7.5cm") %>%
  row_spec(0,bold=T,align="c") %>%
  row_spec(25,background = "green!5") %>%
  row_spec(26,background = "green!2") %>%
  #row_spec(27,background = "yellow!9") %>%
  footnote(general=c("\\\\textbf{Tabla Resumen}: Construcción de tablas
    con \\\\textbf{knitr-kableExtra} (2019-9-28)",
    "\\\\hfill\\\\hspace{6cm} \\\\textbf{Autor: Pedro Luis
    Luque Calvo} (\\\\url{http://destio.us.es/calvo})),
    footnote_as_chunk=TRUE, general_title = "",
    escape=FALSE) %>%
  landscape()
```

| Características de la Tabla | kable() | kable_styling() | Otras funciones adicionales |
|--------------------------------------|---|--|-----------------------------|
| Estilo booktabs | booktabs=TRUE | | |
| Formato salida | format='latex' o format='html' | | |
| Alineación de columnas | align = c('l', 'c', 'r') | | |
| Nombres de columna especiales | col.names=c('Ancho Sépalos', '\$\\mathcal{E}\$'), escape=FALSE | | |
| Mostrar nombres de filas | row.names=NA | | |
| Leyendas en tablas | caption= "\\label{etiqueta}En esta tabla ..." | | |
| Salto de línea en celdas | col.names = linebreak(c("Item\\n(Nombre)", "Valor\\n(Número)")) | | |
| Número de dígitos en columnas | digits = c(2,1,2,4) | | |
| Punto Decimal/Marcar órdenes | format.args=list(decimal.mark=".", big.mark = ,') | | |
| Tablas Multipáginas | longtable=TRUE | latex_options = c('repeat_header', repeat_header_text=(contin\\ua), repeat_header_continued='contin\\ua en la siguiente p\\'agna' | |
| Alto fila reducido | latex_options=c('condensed') | latex_options=c('condensed') | |
| Filas coloreadas automáticamente | stripe_color=gray[i], stripe_index=NULL | stripe_color=gray[i], stripe_index=NULL | |
| Posición tabla en párrafo | position='center' o position='float_right' o position='left' | position='center' o position='float_right' o position='left' | |
| Ajusta tabla ancho página (NO/SÍ) | full_width=FALSE o full_width=TRUE | full_width=FALSE o full_width=TRUE | |
| Tamaño fuente texto | font_size=NULL o font_size=8 | font_size=NULL o font_size=8 | |
| Ajuste fuentes respecto ancho página | latex_options=c('scale_down') | latex_options=c('scale_down') | |
| Ajuste vertical tabla (flotante) | latex_options=c('HOLD_position') o c('hold_position') | latex_options=c('HOLD_position') o c('hold_position') | |
| Tabla Apaisada | landscape() | landscape() | |
| Especificar características columnas | column_spec(1:2, bold=T, color='red', width='3cm') | column_spec(1:2, bold=T, color='red', width='3cm') | |
| Especificar características filas | row_spec(0, bold=T, background='orange', angle=45, align='c') | row_spec(0, bold=T, background='orange', angle=45, align='c') | |
| Añadir filas de cabecera | add_header_above(c(" "=1, "Población" = 2), bold=T, line=F, background='blue') | add_header_above(c(" "=1, "Población" = 2), bold=T, line=F, background='blue') | |
| Agrupar filas | pack_rows('Grupo 1', 4, 7, latex_gap_space='0.5cm' 2;'), number_title = "Tipo 1;") | pack_rows('Grupo 1', 4, 7, latex_gap_space='0.5cm' 2;'), number_title = "Tipo 1;") | |
| Notas a pie de tabla | as_image() o kable_as_image(file='fichero') saveRDS(tbResumen,file='tbResumen.RDS') y tbResumen = readRDS('tbResumen.RDS') | as_image() o kable_as_image(file='fichero') saveRDS(tbResumen,file='tbResumen.RDS') y tbResumen = readRDS('tbResumen.RDS') | |
| Insertar/Guardar como gráfico | | | |
| Consejo: Guardar/Leer datos objeto R | | | |
| Consejo: Tratamiento de NAs | options(knitr.kable.NA = ") | options(knitr.kable.NA = ") | |
| Consejo: Ejemplo Básico-Paquetes | <pre> ...{r echo=FALSE,message=FALSE} library(knitr); library(dplyr); library(kableExtra); library(tbsummen) %>% kable(booktabs=TRUE,format="latex") %>% kable_styling(latex_options = c("striped"), font_size = 8) %>% column_spec(1,width = "4cm") ... </pre> | | |

| Características de la Tabla | kable() - kable_styling() - Otras funciones adicionales |
|---|--|
| kable() | |
| Estilo booktabs | booktabs=TRUE |
| Formato salida | format='latex' o format='html' |
| Columnas: Alineación | align = c('l','c') |
| Columnas: Nombres especiales | col.names=c('Ancho Sépalo','\$\\mathcal{E}\$'), escape=FALSE |
| Filas: Mostrar nombres | row.names=NA |
| Tablas: Leyendas | caption='\\label{etiqueta}En esta tabla ...' |
| Celdas: Salto de línea | col.names = linebreak(c("Item\\n(Nombre)", "Valor\\n(Número)")) |
| Columnas: Número de dígitos | digits = c(2,1,2,4) |
| Punto Decimal/Marcar órdenes | format.args=list(decimal.mark='.', big.mark = ',') |
| Tablas: Multipáginas | longtable=TRUE — kable_styling(latex_options = c('repeat_header'), repeat_header_text='(contin\\'ua), repeat_header_continued='contin\\'ua en la siguiente p\\'agina') |
| kable_styling() | |
| Filas: Alto reducido | latex_options=c('condensed') |
| Filas: coloreadas automáticamente | latex_options=c('striped'), stripe_color='gray!6', stripe_index=NULL |
| Tabla: Posición en párrafo | position='center' o position='float_right' o position='left' |
| Tabla: Ajusta ancho página (NO/SÍ) | full_width=FALSE o full_width=TRUE |
| Tabla: Tamaño fuente texto | font_size=NULL o font_size=8 |
| Tabla: Ajuste fuentes respecto ancho página | latex_options=c('scale_down') |
| Tabla: Ajuste vertical (flotante) | latex_options=c('HOLD_position') o c('hold_position') |
| Otras funciones | |
| Tabla: Apaisada | landscape() |
| Columnas: Especificar características | column_spec(1:2, bold=T, color='red', width='3cm') |
| Filas: Especificar características | row_spec(0, bold=T, background='orange', angle=45, align='c') |
| Filas: Añadir cabeceras | add_header_above(c(" " =1, "Población" = 2), bold=T, line=F, background="blue") |
| Filas: Agrupar | pack_rows("Grupo 1", 4, 7, latex_gap_space="0.5cm") |
| Tabla: Notas a pie de tabla | footnote(general="texto", number = c("Nota 1; ", "Nota 2;"), number_title = "Tipo 1:") |
| Insertar/Guardar como gráfico | as_image() o kable_as_image(file="fichero") |
| Consejo: Guardar/Leer datos objeto R | saveRDS(tbResumen,file="tbResumen.RDS") y tbResumen = readRDS("tbResumen.RDS") |
| Consejo: Tratamiento de NAs | options(knitr.kable.NA = ") |
| EJEMPLO BÁSICO en R Markdown | |
| | <pre> ```{r echo=FALSE,message=FALSE} library(knitr); library(dplyr); library(kableExtra); tbResumen %>% kable(booktabs=TRUE,format="latex") %>% kable_styling(latex_options = c("striped"), font_size = 8) %>% column_spec(1,width = "4cm") ``` </pre> |

Tabla Resumen: Construcción de tablas con **knitr-kableExtra** (2019-9-28)
Autor: Pedro Luis Luque Calvo (<http://destio.us.es/calvo>)

Bibliografía

- [1] Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W. and Iannone, R. 2019. *Rmarkdown: Dynamic documents for r*.
- [2] Dahl, D.B., Scott, D., Roosen, C., Magnusson, A. and Swinton, J. 2019. *Xtable: Export tables to latex or html*.
- [3] Daróczi, G. and Tsegelskyi, R. 2018. *Pander: An r 'pandoc' writer*.
- [4] Firke, S. 2019. *Janitor: Simple tools for examining and cleaning dirty data*.
- [5] Hlavac, M. 2018. *Stargazer: Well-formatted regression and summary statistics tables*.
- [6] Hugh-Jones, D. 2019. *Hxtable: Easily create and style tables for latex, html and other formats*.
- [7] Iannone, R., Cheng, J. and Schloerke, B. 2019. *Gt: Easily create presentation-ready display tables*.
- [8] Leifeld, P. 2017. *Texreg: Conversion of r regression output to latex or html tables*.
- [9] Luque-Calvo, P.L. 2017. *Escribir un trabajo fin de estudios con r markdown*. Disponible en <http://destio.us.es/calvo>.
- [10] Owen, J. 2018. *Rhandsontable: Interface to the 'handsontable.js' library*.
- [11] R Core Team 2016. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing.
- [12] RStudio Team 2015. *RStudio: Integrated development environment for r*. RStudio, Inc.
- [13] Sidi, J. and Polhamus, D. 2019. *TexPreview: Compile and preview snippets of 'latex' in rstudio*.
- [14] Wickham, H., François, R., Henry, L. and Müller, K. 2019. *Dplyr: A grammar of data manipulation*.
- [15] Xie, Y. 2019. *Bookdown: Authoring books and technical documents with r markdown*.
- [16] Xie, Y. 2019. *Knitr: A general-purpose package for dynamic report generation in r*.
- [17] Xie, Y., Cheng, J. and Tan, X. 2019. *DT: A wrapper of the javascript library 'datatables'*.
- [18] Zhu, H. 2019. *KableExtra: Construct complex table with 'kable' and pipe syntax*.