T2

Ejercicio 1

#Apartado 1

```
x1 <- c(0:3)
x1
```

```
## [1] 0 1 2 3
```

```
p_x1 <- c(64/125,48/125,12/125,1/125)
p_x1
```

```
## [1] 0.512 0.384 0.096 0.008
```

```
theo_mean = (1*(48/125) + 2 * (12/125) + 3 * (1/125))
theo_mean
```

```
## [1] 0.6
```

```
theo_var <- sum((x1 - theo_mean)^2*p_x1)
theo_var
```

```
## [1] 0.48
```

#Apartado 2

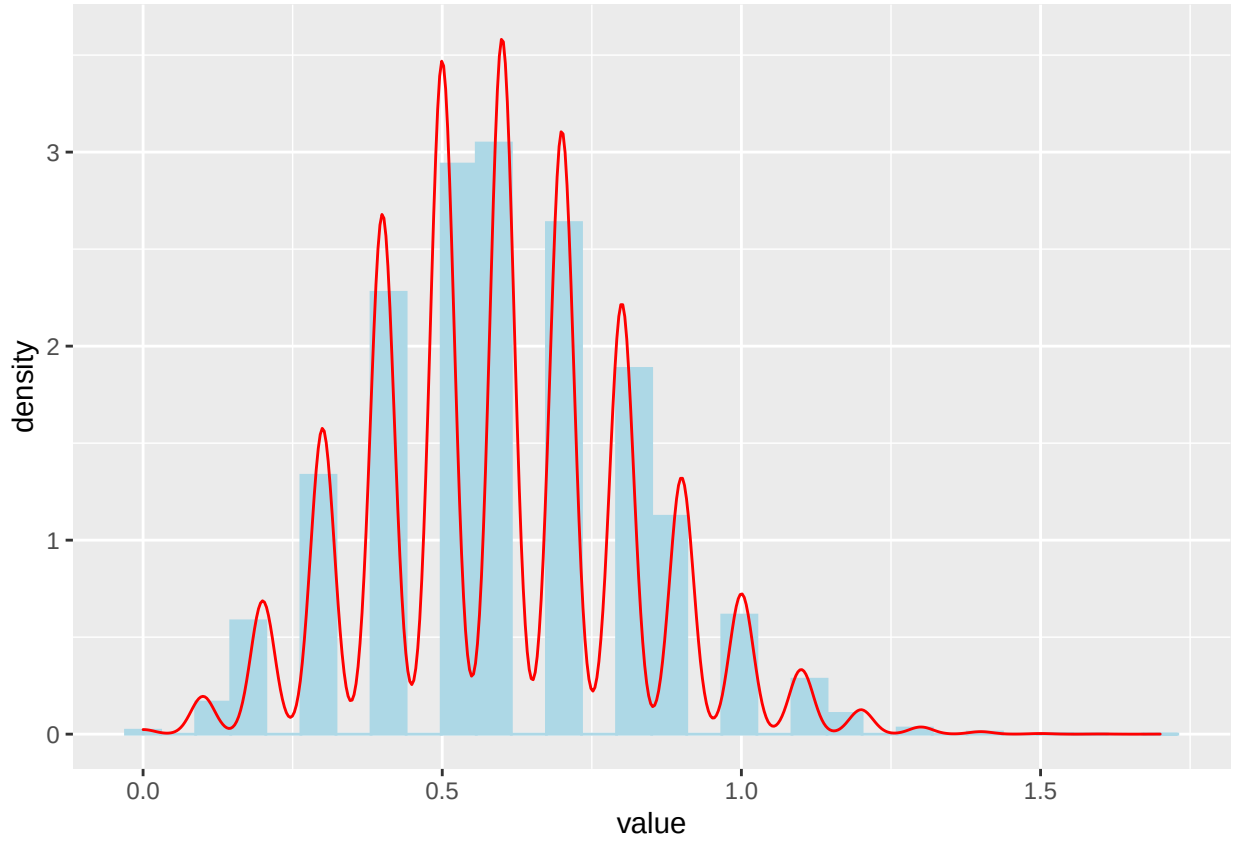
```
k = 100000
mediasMuestrales = replicate(k, {
  muestra = sample(0:3, size=10, replace=TRUE, prob = c(64,48,12,1))
  mean(muestra)
})
head(mediasMuestrales,10)
```

```
##  [1] 0.9 0.5 0.6 0.5 0.6 0.6 0.4 0.3 0.5 0.6
```

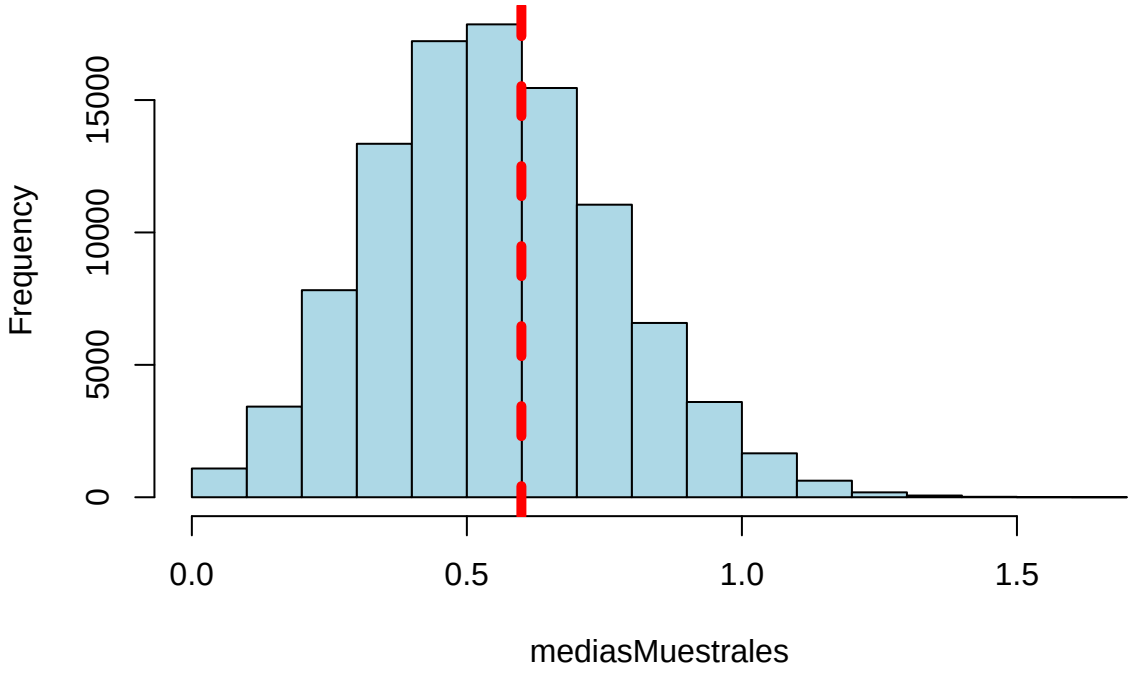
```
mm_tibble <- as_tibble(mediasMuestrales)
```

```
mm_tibble %>%
  ggplot() +
    geom_histogram(mapping = aes(x=value, y = stat(density)), fill="lightblue", color ="lightblue") +
    geom_density(mapping = aes(value), color = "red")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



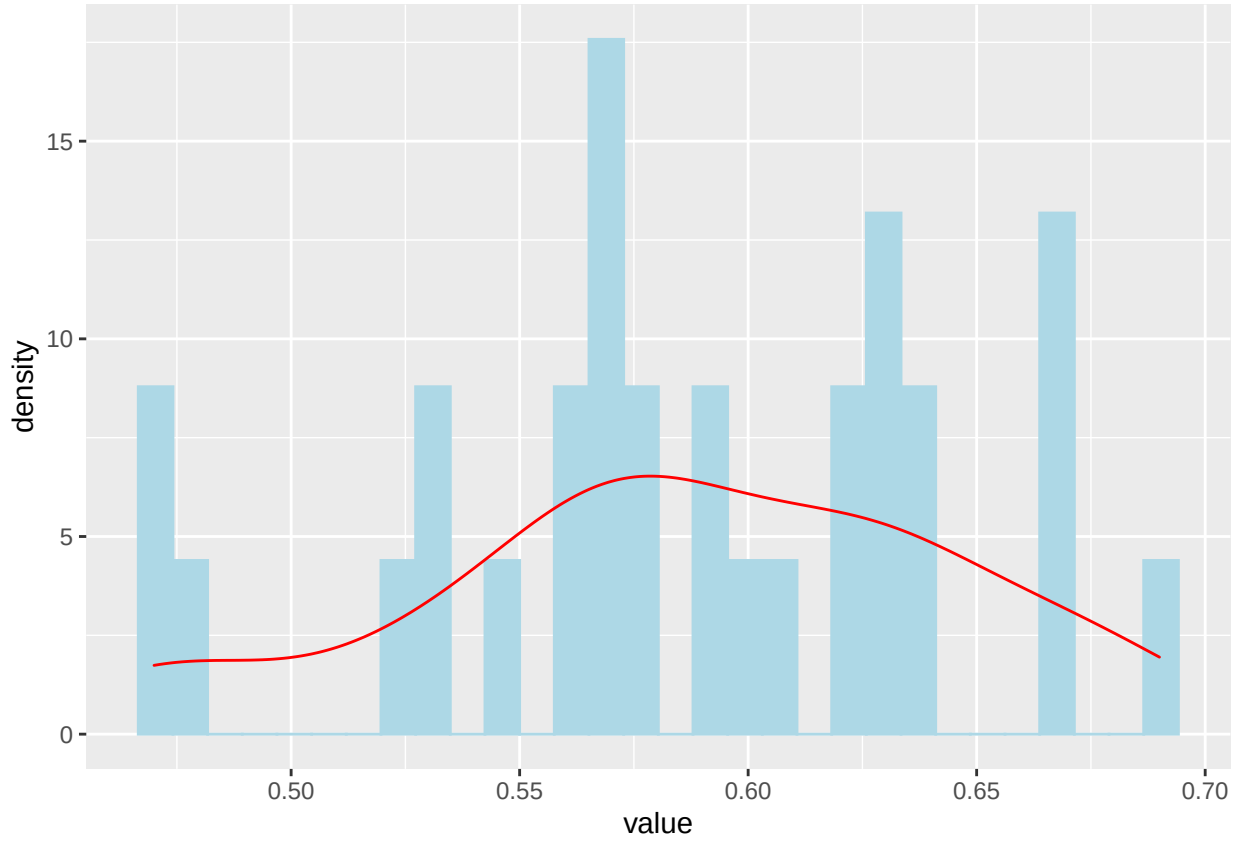

```
hist(mediasMuestrales, main=" ", col="lightblue")
abline(v = mean(mediasMuestrales), lty=2, lwd=5, col="red")
```




```
k = 30
mediasMuestrales_30 = replicate(k, {
  muestra = sample(0:3, size=100, replace=TRUE, prob = c(64,48,12,1))
  mean(muestra)
})
```

```
mediasMuestrales_30 %>%
  as_tibble %>%
ggplot() +
  geom_histogram(mapping = aes(x=value, y = stat(density)), fill="lightblue", color ="lightblue") +
  geom_density(mapping = aes(value), color = "red")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



##Apartado 3

##Sabiendo que las variables x1 y x2 son independientes, la suma x1 + x2 puede tomar cualquier valor entre 0 y 5. El valor mínimo se calculca sumando los dos mas pequeños que x1 y x2 pueden tomar. En este caso es 0+0. El valor máximo que la suma de estas variables pueden tomar es 5, ya que si sumamos los valores mas grandes de las variables (3+2) el resultado nos da 5.

##Tabla de probabilidad

```
x2 <- c(0:2)
x2
```

```
## [1] 0 1 2
```

```
p_x2 <- c(1/2,1/4,1/4)
p_x2
```

```
## [1] 0.50 0.25 0.25
```

```
p = c(64/125,48/125,12/125,1/125)*rep(c(1/2,1/4,1/4), each = 4)
p
```

```
##  [1] 0.256 0.192 0.048 0.004 0.128 0.096 0.024 0.002 0.128 0.096 0.024 0.002
```

```
(X1 = rep(0:3, each =3))
```

```
##  [1] 0 0 0 1 1 1 2 2 2 3 3 3
```

```
(X2 = rep(0:2, each = 4))
```

```
##  [1] 0 0 0 0 1 1 1 1 2 2 2 2
```

```
a = X1 + X2
(tabla = data.frame(a, X1, X2, p))
```

```
##    a X1 X2     p
## 1  0  0  0 0.256
## 2  0  0  0 0.192
## 3  0  0  0 0.048
## 4  1  1  0 0.004
## 5  2  1  1 0.128
## 6  2  1  1 0.096
## 7  3  2  1 0.024
## 8  3  2  1 0.002
## 9  4  2  2 0.128
## 10 5  3  2 0.096
## 11 5  3  2 0.024
## 12 5  3  2 0.002
```

Apartado 4

```
(media_x2 = sum(x2*p_x2))
```

```
## [1] 0.75
```

```
(media_teorica_de_la_suma = media_x2 + theo_mean)
```

```
## [1] 1.35
```

```
set.seed(1)
k=100000
suma_medias = replicate(k, {
  m = sample(0:3, size = 1, replace = TRUE, prob = c(64/125, 48/125, 12/125, 1/125))
  + sample(0:2, size = 1, replace = TRUE, prob = c(1/2,1/4,1/4))
  mean(m)

})
head(suma_medias)
```

```
## [1] 0 1 0 2 1 0
```

```
mean(suma_medias)
```

```
## [1] 0.59943
```

Ejercicio 2: Datos limpios

```
test <- read.csv(file = 'testResults.csv')
```

Ejercicio 2

```
resultados <- read.csv(file = "testResults.csv")
head(resultados,10)
```

```
##           name  id gender_age test_number week1 week2 week3 week4 week5
## 1        Jacob 108       m_20           1     8     5     7     5     6
## 2        Jacob 108       m_20           2     2     2     4     0     3
## 3      Michael 490       m_19           1    10     0     5     4     0
## 4      Michael 490       m_19           2     9    10     8    10     9
## 5      Matthew 424       m_18           1     6     0     0     1    10
## 6      Matthew 424       m_18           2     3     4     2     5     8
```

```
## 7       Joshua 734       m_17           1    10     2     2     0     6
## 8       Joshua 734       m_17           2    10     0     6     8     9
## 9  Christopher 928       m_20           1     5     2     0     0     0
## 10 Christopher 928       m_20           2     9     9     3    10     4
```

##Para que un conjunto de datos se considere limpio debe de cumplir una serie de requisitos.

1) Cada variable debe de tener su propia columna

2) Cada observación debe de tener su propia fila

3) Cada valor su propia celda

##En este caso, la tabla no cumple los principios de tidy data ya que hay columnas que no representan una variable, sino valores, como por ejemplo (week 1, week 2, week 3, week 4, week 5). Estas columnas representan la nota (valor) sacada en un examen. Para converitr la tabla a una tidy, debemos crear otra columna (resultado del test), y la semana en la que se ha hecho el test.

```
t_resul <- as_tibble(resultados)
head(t_resul,10)
```

```
## # A tibble: 10 x 9
##    name           id gender_age test_number week1 week2 week3 week4 week5
##    <chr>       <int> <chr>            <int> <int> <int> <int> <int> <int>
##  1 Jacob         108 m_20                 1     8     5     7     5     6
##  2 Jacob         108 m_20                 2     2     2     4     0     3
##  3 Michael       490 m_19                 1    10     0     5     4     0
##  4 Michael       490 m_19                 2     9    10     8    10     9
##  5 Matthew       424 m_18                 1     6     0     0     1    10
##  6 Matthew       424 m_18                 2     3     4     2     5     8
##  7 Joshua        734 m_17                 1    10     2     2     0     6
##  8 Joshua        734 m_17                 2    10     0     6     8     9
##  9 Christopher   928 m_20                 1     5     2     0     0     0
## 10 Christopher   928 m_20                 2     9     9     3    10     4
```

Vamos a usar separate para separar gender_age en dos variables nuevas, una para la edad y otra para el genero

```
t <- t_resul %>%
  separate(col  =  gender_age, into =  c("gender", "age"), sep  =  "_", convert = TRUE)

head(t,10)
```

```
## # A tibble: 10 x 10
##    name           id gender   age test_number week1 week2 week3 week4 week5
##    <chr>       <int> <chr>  <int>       <int> <int> <int> <int> <int> <int>
##  1 Jacob         108 m         20           1     8     5     7     5     6
##  2 Jacob         108 m         20           2     2     2     4     0     3
##  3 Michael       490 m         19           1    10     0     5     4     0
##  4 Michael       490 m         19           2     9    10     8    10     9
##  5 Matthew       424 m         18           1     6     0     0     1    10
##  6 Matthew       424 m         18           2     3     4     2     5     8
##  7 Joshua        734 m         17           1    10     2     2     0     6
##  8 Joshua        734 m         17           2    10     0     6     8     9
##  9 Christopher   928 m         20           1     5     2     0     0     0
## 10 Christopher   928 m         20           2     9     9     3    10     4
```

Hacemos un pivot_longer, con más filas, una para cada semana, y metemos los resultados del test en la variable

```
t %>%
  pivot_longer(c("week1", "week2", "week3", "week4", "week5"), values_to = "resultado del test", names_
```

```
## # A tibble: 1,000 x 7
##    name     id gender   age test_number semana `resultado del test`
##    <chr> <int> <chr>  <int>       <int> <chr>                  <int>
##  1 Jacob   108 m         20           1 week1                      8
##  2 Jacob   108 m         20           1 week2                      5
##  3 Jacob   108 m         20           1 week3                      7
##  4 Jacob   108 m         20           1 week4                      5
##  5 Jacob   108 m         20           1 week5                      6
##  6 Jacob   108 m         20           2 week1                      2
##  7 Jacob   108 m         20           2 week2                      2
##  8 Jacob   108 m         20           2 week3                      4
##  9 Jacob   108 m         20           2 week4                      0
## 10 Jacob   108 m         20           2 week5                      3
## # ... with 990 more rows
```

Ejercicio 3 #Lectura de R4DS

#7.5.1.1 Exercises

```
library(nycflights13)
```

```
## Warning: package 'nycflights13' was built under R version 4.0.5
```

```
flights
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
##  1  2013     1     1      517            515         2      830            819
##  2  2013     1     1      533            529         4      850            830
##  3  2013     1     1      542            540         2      923            850
##  4  2013     1     1      544            545        -1     1004           1022
##  5  2013     1     1      554            600        -6      812            837
##  6  2013     1     1      554            558        -4      740            728
##  7  2013     1     1      555            600        -5      913            854
##  8  2013     1     1      557            600        -3      709            723
##  9  2013     1     1      557            600        -3      838            846
## 10  2013     1     1      558            600        -2      753            745
## # ... with 336,766 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
flights %>%
  mutate("cancelled" = is.na(dep_time))
```

```
## # A tibble: 336,776 x 20
##     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
##  1  2013     1     1      517            515         2      830            819
##  2  2013     1     1      533            529         4      850            830
##  3  2013     1     1      542            540         2      923            850
##  4  2013     1     1      544            545        -1     1004           1022
```

```
##  5  2013     1     1      554            600        -6      812            837
##  6  2013     1     1      554            558        -4      740            728
##  7  2013     1     1      555            600        -5      913            854
##  8  2013     1     1      557            600        -3      709            723
##  9  2013     1     1      557            600        -3      838            846
## 10  2013     1     1      558            600        -2      753            745
## # ... with 336,766 more rows, and 12 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>,
## #   cancelled <lgl>
```

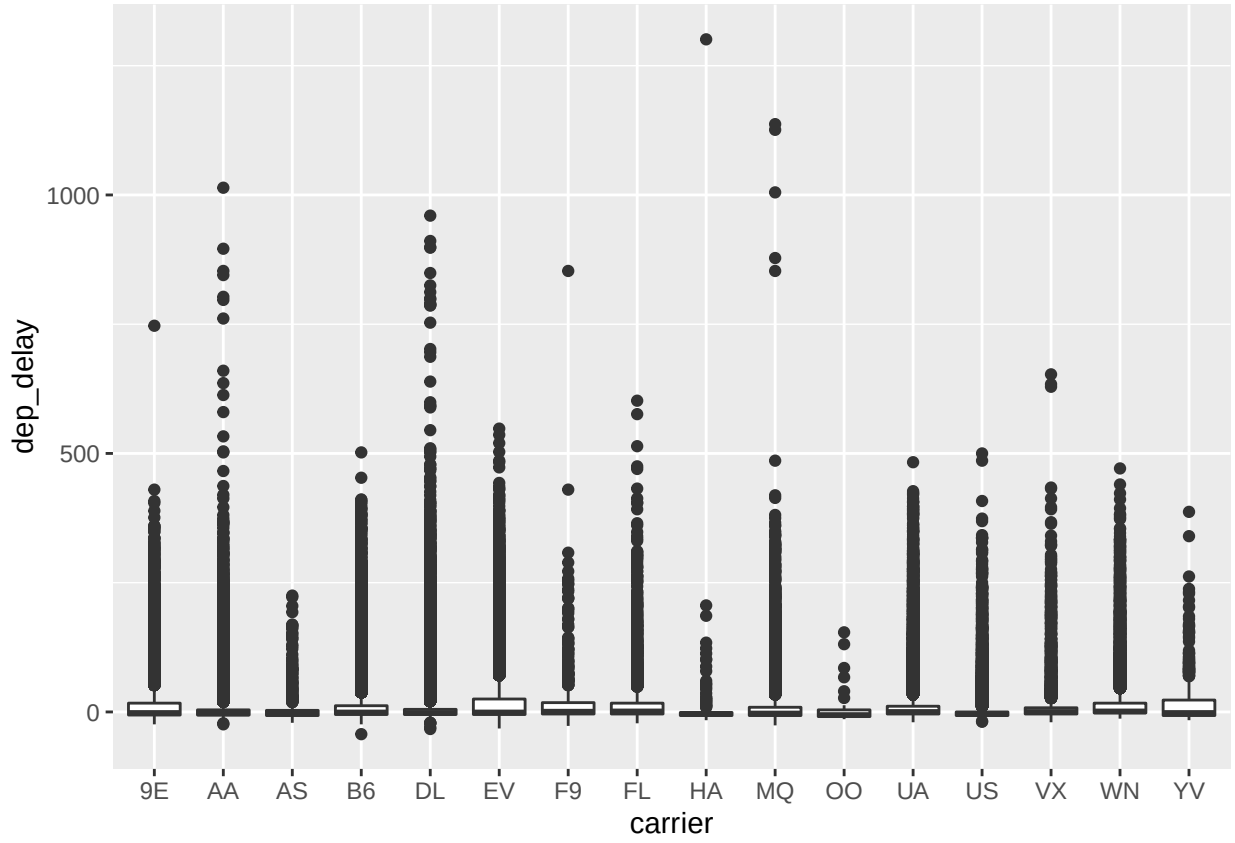
```
flights
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
##  1  2013     1     1      517            515         2      830            819
##  2  2013     1     1      533            529         4      850            830
##  3  2013     1     1      542            540         2      923            850
##  4  2013     1     1      544            545        -1     1004           1022
##  5  2013     1     1      554            600        -6      812            837
##  6  2013     1     1      554            558        -4      740            728
##  7  2013     1     1      555            600        -5      913            854
##  8  2013     1     1      557            600        -3      709            723
##  9  2013     1     1      557            600        -3      838            846
## 10  2013     1     1      558            600        -2      753            745
## # ... with 336,766 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
cancelled_flights <- flights %>%
  filter(cancelled = TRUE)
```

```
ggplot(data = flights, mapping = aes(x=carrier, y=dep_delay)) +
  geom_boxplot()
```

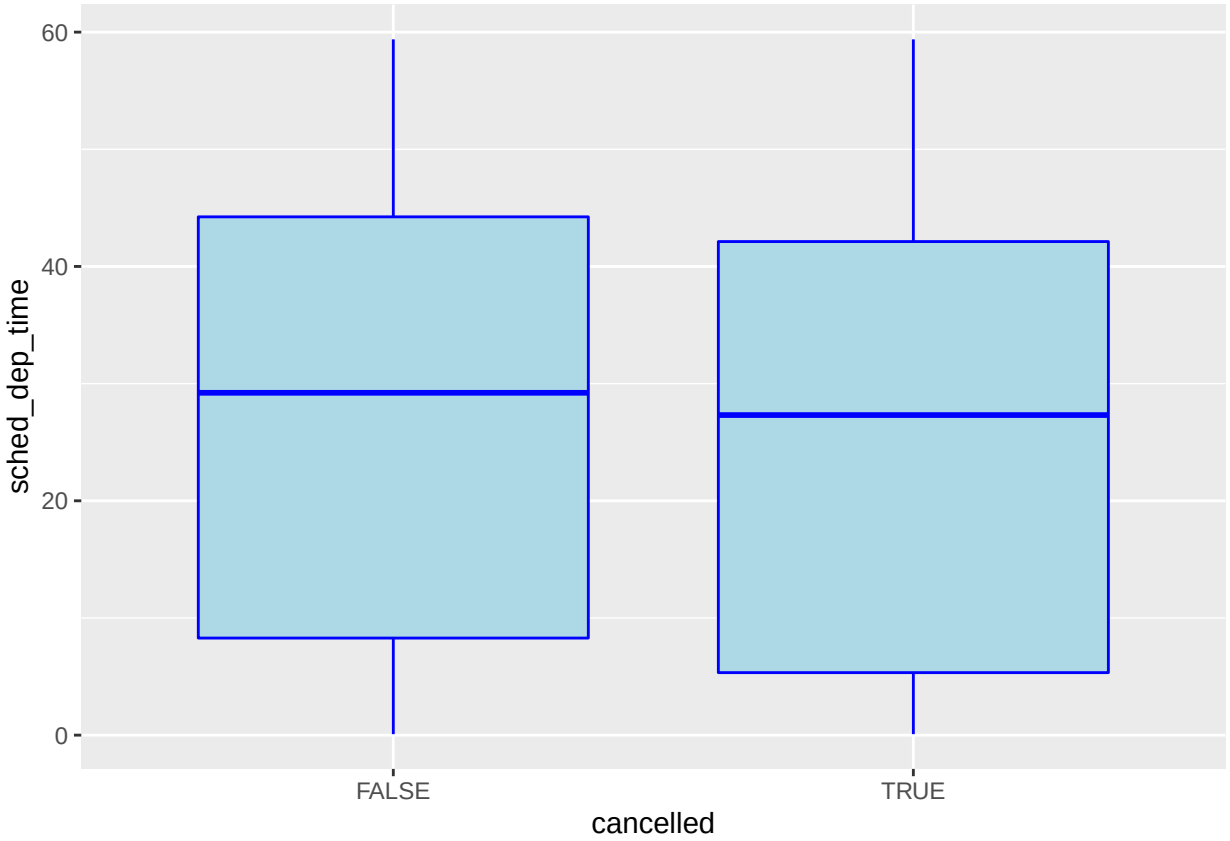
```
## Warning: Removed 8255 rows containing non-finite values (stat_boxplot).
```




```
flights
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
##  1  2013     1     1      517            515         2      830            819
##  2  2013     1     1      533            529         4      850            830
##  3  2013     1     1      542            540         2      923            850
##  4  2013     1     1      544            545        -1     1004           1022
##  5  2013     1     1      554            600        -6      812            837
##  6  2013     1     1      554            558        -4      740            728
##  7  2013     1     1      555            600        -5      913            854
##  8  2013     1     1      557            600        -3      709            723
##  9  2013     1     1      557            600        -3      838            846
## 10  2013     1     1      558            600        -2      753            745
## # ... with 336,766 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
flights %>%
  mutate(cancelled = is.na(dep_time),
  sched_hour = sched_dep_time %/% 100,
  sched_min = sched_dep_time %% 100,
  sched_dep_time = sched_min + sched_hour /60) %>%
  ggplot() +
  geom_boxplot(mapping = aes(x = cancelled, y = sched_dep_time), fill = "lightblue", color = "blue")
```



Diamantes

```
diamonds
```

```
## # A tibble: 53,940 x 10
##    carat cut       color clarity depth table price     x     y     z
##    <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
##  1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
##  2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
##  3  0.23 Good      E     VS1      56.9    65   327  4.05  4.07  2.31
##  4  0.29 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
##  5  0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75
##  6  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
##  7  0.24 Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
##  8  0.26 Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
##  9  0.22 Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
## 10  0.23 Very Good H     VS1      59.4    61   338  4     4.05  2.39
## # ... with 53,930 more rows
```

```
cov_dia <-select(diamonds, -c(color, cut, clarity))
cov(cov_dia)
```

```
##              carat        depth        table         price           x
## carat 2.246867e-01   0.01916653    0.1923645  1.742765e+03    0.5184841
## depth 1.916653e-02   2.05240384   -0.9468399 -6.085371e+01   -0.0406413
## table 1.923645e-01  -0.94683994    4.9929481  1.133318e+03    0.4896429
## price 1.742765e+03 -60.85371214 1133.3180641  1.591563e+07 3958.0214908
## x     5.184841e-01  -0.04064130    0.4896429  3.958021e+03    1.2583472
```

```
## y     5.152478e-01  -0.04800857    0.4689723  3.943271e+03    1.2487893
## z     3.189168e-01   0.09596797    0.2379960  2.424713e+03    0.7684875
##                   y           z
## carat    0.51524782 3.189168e-01
## depth   -0.04800857 9.596797e-02
## table    0.46897228 2.379960e-01
## price 3943.27081043 2.424713e+03
## x        1.24878933 7.684875e-01
## y        1.30447161 7.673196e-01
## z        0.76731958 4.980109e-01
```

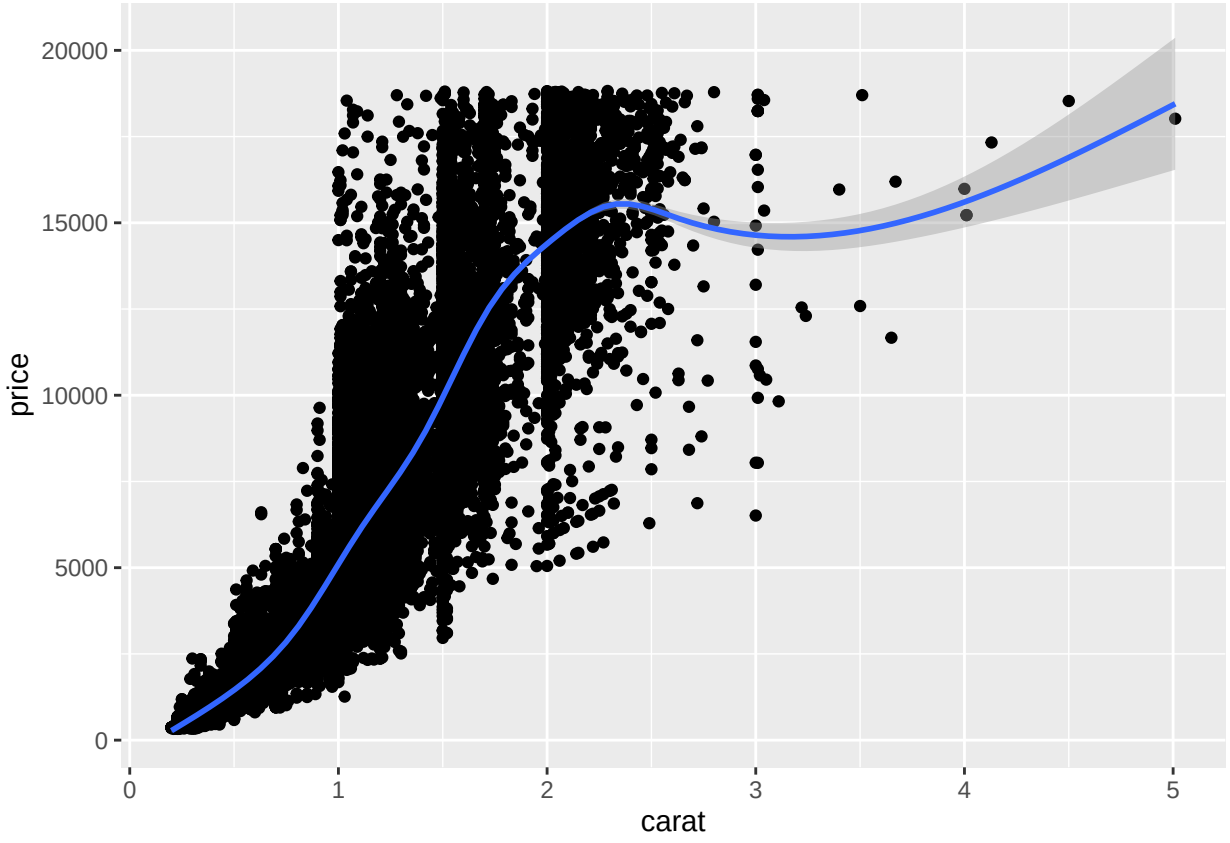
```
(cor(diamonds$price, select(diamonds, carat, depth, table, x, y, z)))
```

```
##          carat      depth     table         x         y         z
## [1,] 0.9215913 -0.0106474 0.1271339 0.8844352 0.8654209 0.8612494
```

#Las variable mas importante para predecir el precio de un diamante son los quilates

```
ggplot(data = diamonds, mapping = aes(x = carat, y = price), fill = "magenta") +
  geom_point() +
  geom_smooth(se = TRUE)
```

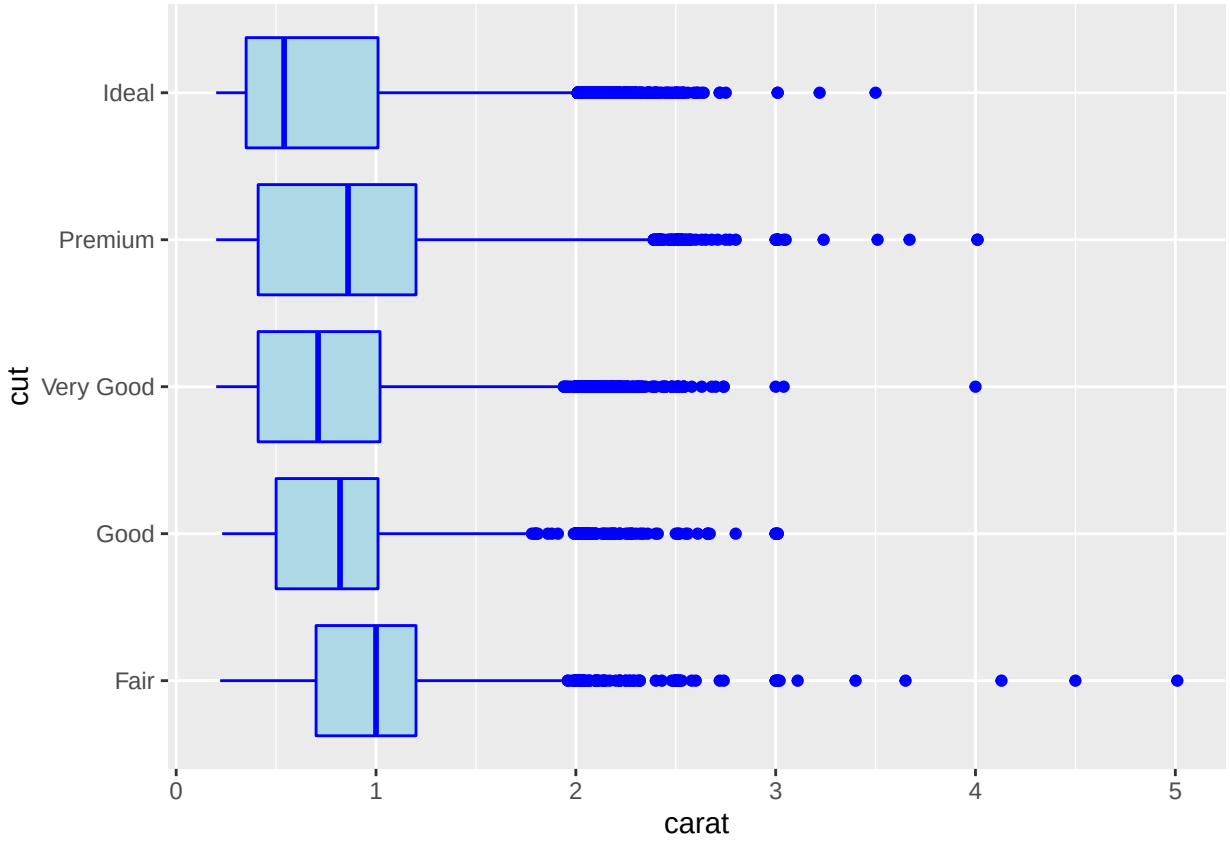
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




```
ggplot(data =diamonds, mapping = aes(x = carat, y = cut), fill="lightblue", color="blue") +
  geom_boxplot(fill="lightblue", color = "blue") +
  geom_smooth(se =FALSE)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Computation failed in `stat_smooth()`:
## NA/NaN/Inf en llamada a una función externa (arg 3)
```



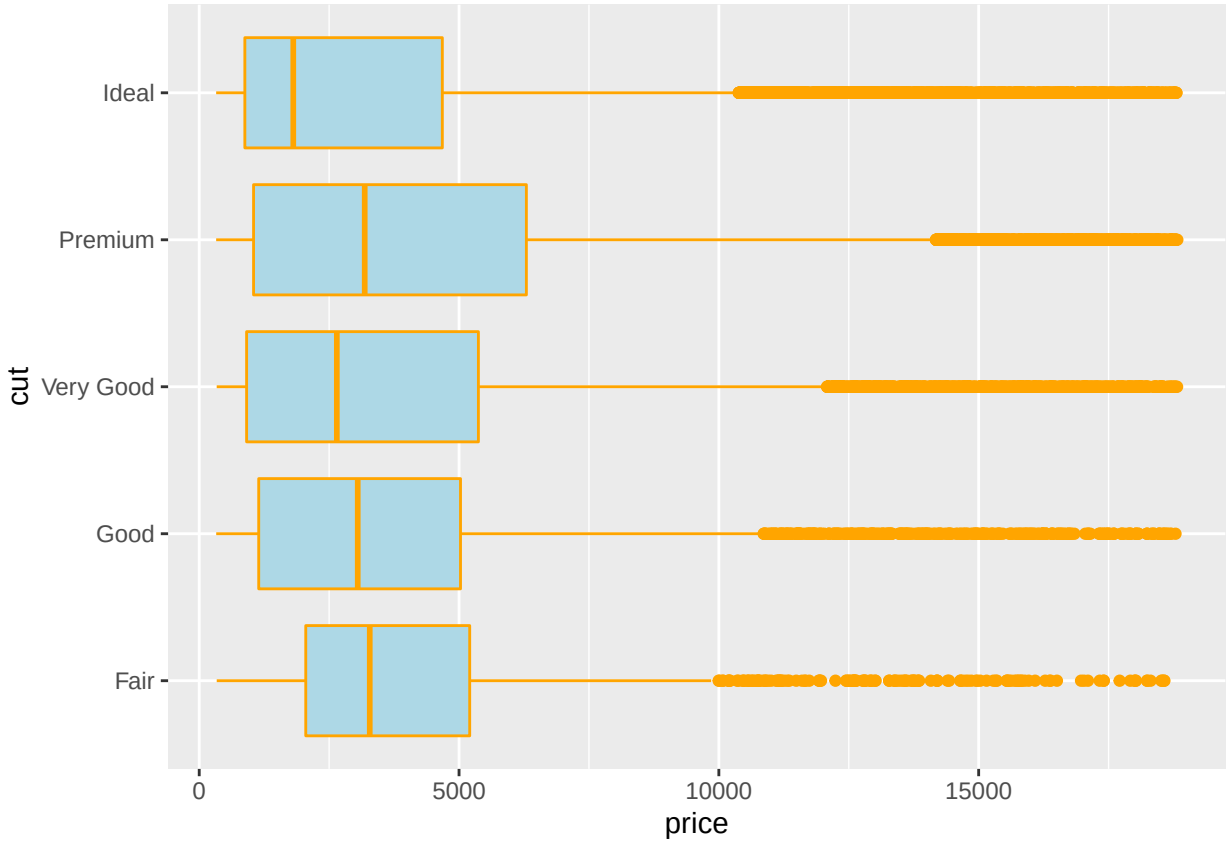

```
library(ggstance)
```

```
## Warning: package 'ggstance' was built under R version 4.0.5
```

```
##
## Attaching package: 'ggstance'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh
```

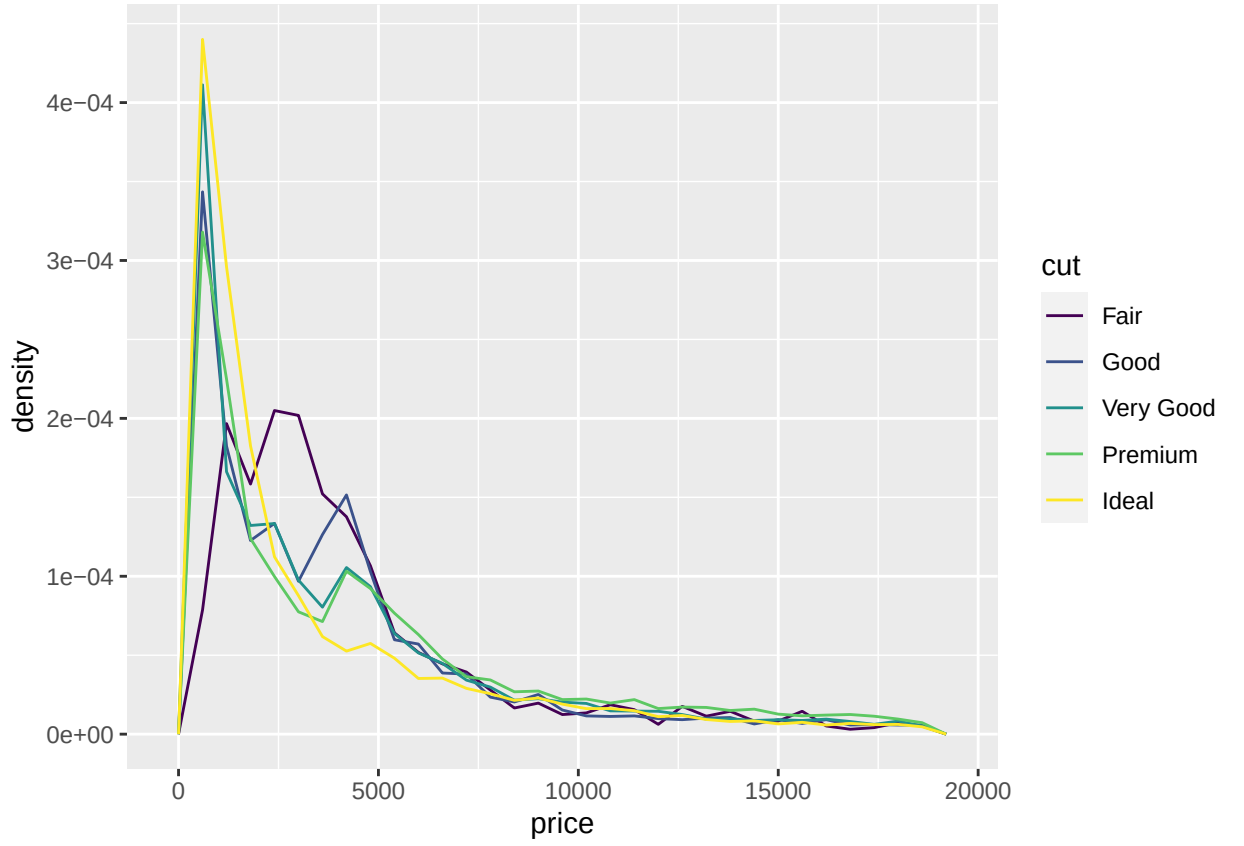
```
ggplot(diamonds, mapping = aes(x=price, y=cut)) +
  geom_boxploth(fill = "lightblue", color = "orange")
```



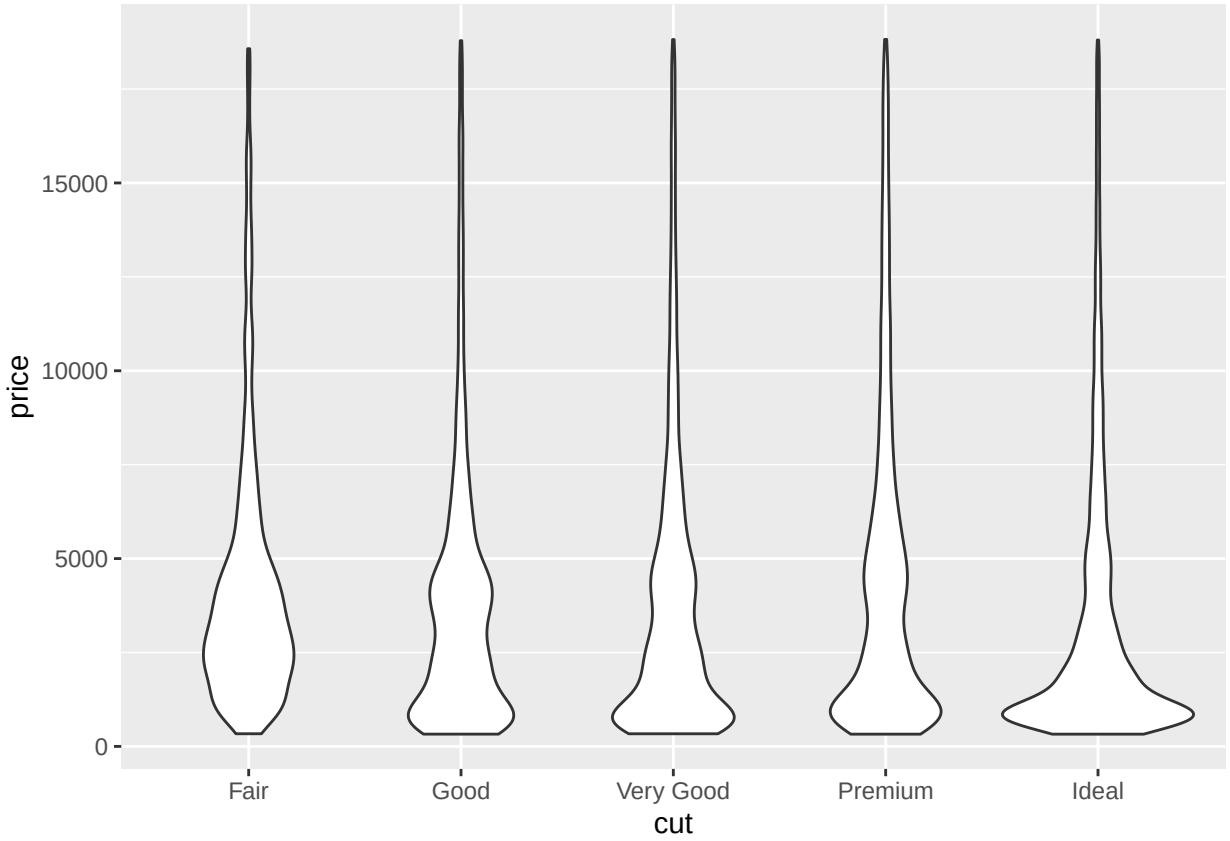

```
#ggplot(diamonds) +
  #lvplot::geom_lv(aes(x= cut, y=price, fill= ..LV..)
```

##El geom_lv es similar al geom_boxplot, pero separa los datos en 12 grupos diferentes, en vez de en cuartiles (como el box plot). Geom_lv se utiliza cuando la cantidad de datos es relativamente mas grande, ya que se pueden separar en mas grupos.

```
ggplot(data=diamonds, mapping = aes(x=price, y=stat(density))) +
  geom_freqpoly(mapping = aes(color=cut), binwidth = 600)
```

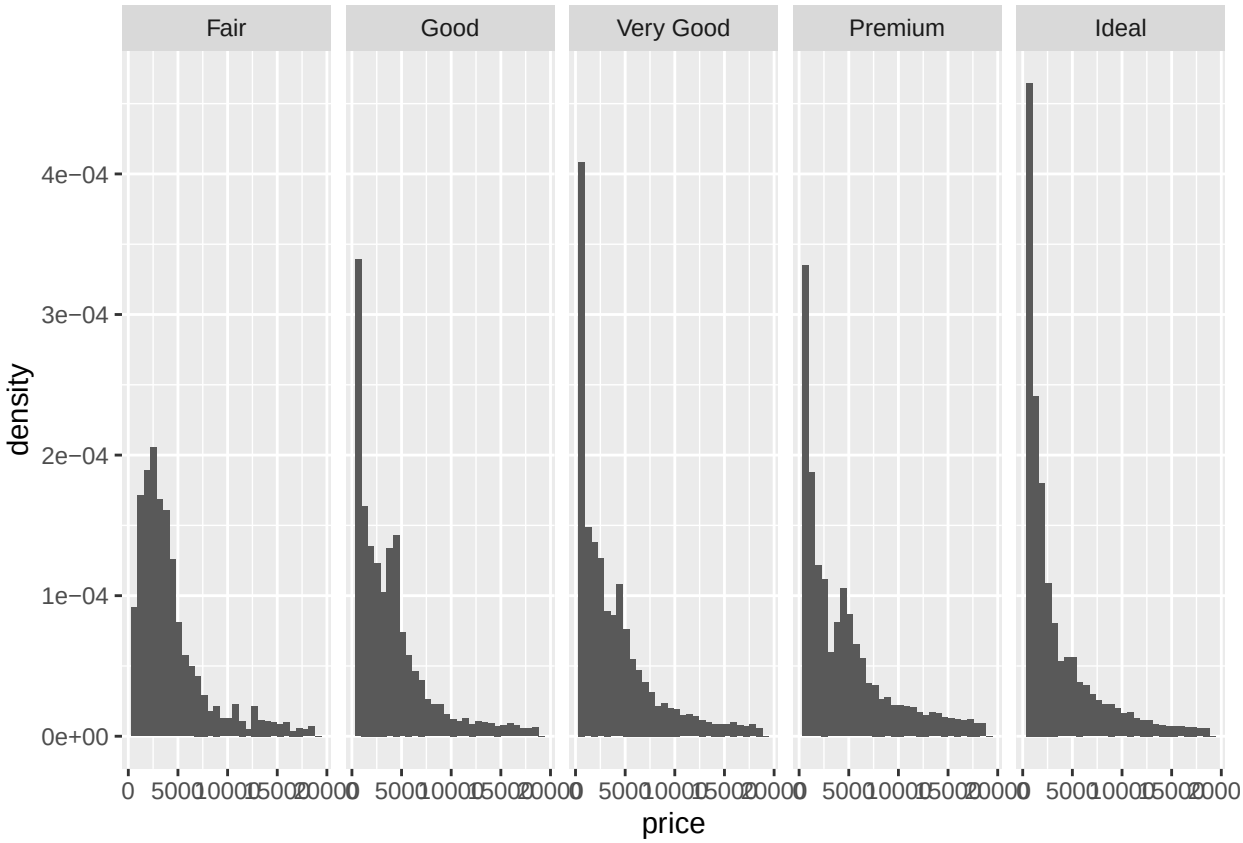



```
ggplot(data=diamonds, mapping = aes(x=cut, y=price)) +
  geom_violin()
```




```
ggplot(diamonds, mapping = aes(x=price, y=stat(density)), color = "blue") +
  geom_histogram() +
  facet_wrap(~cut,ncol = 5)
```

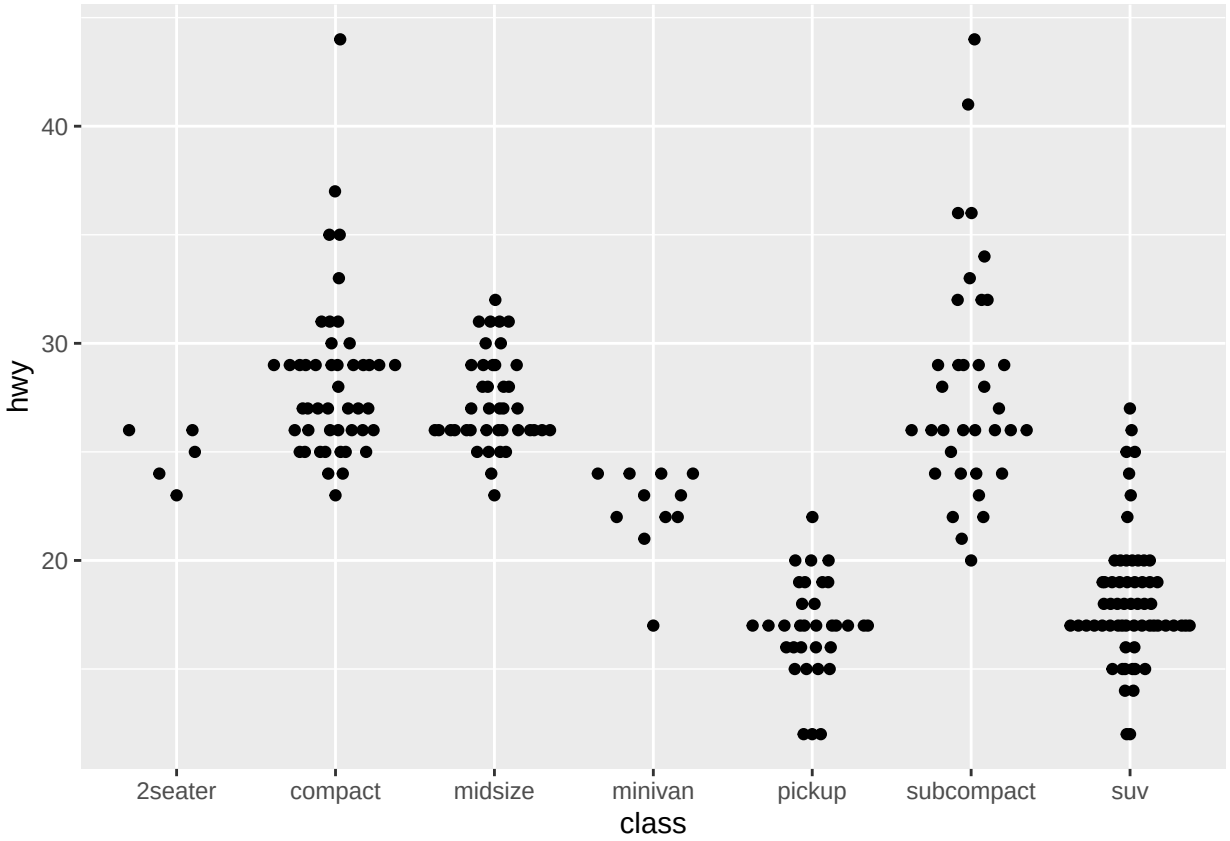
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
library(ggbeeswarm)
```

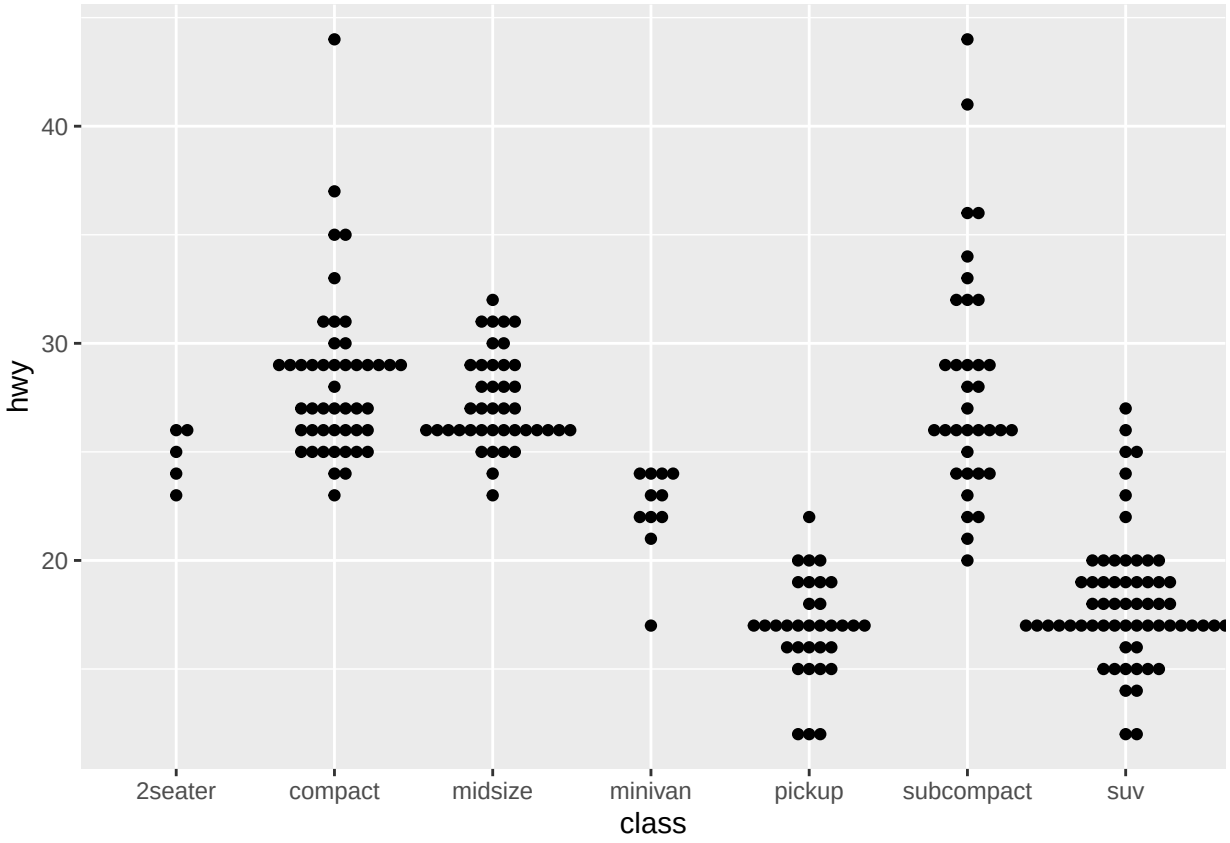
```
## Warning: package 'ggbeeswarm' was built under R version 4.0.5
```

```
ggplot(data=mpg, mapping = aes(y=hwy, x = class)) +
  geom_quasirandom()
```




```
ggplot(data=mpg, mapping = aes(y=hwy, x = class)) +
  geom_beeswarm(method="tukey")
```

```
## Warning: Ignoring unknown parameters: method
```



#Sección 12.6.1 R4DS

```
tidyr::who
```

```
## # A tibble: 7,240 x 60
##    country     iso2  iso3   year new_sp_m014 new_sp_m1524 new_sp_m2534 new_sp_m3544
##    <chr>       <chr> <chr> <int>       <int>        <int>        <int>        <int>
##  1 Afghanistan AF    AFG    1980          NA           NA           NA           NA
##  2 Afghanistan AF    AFG    1981          NA           NA           NA           NA
##  3 Afghanistan AF    AFG    1982          NA           NA           NA           NA
##  4 Afghanistan AF    AFG    1983          NA           NA           NA           NA
##  5 Afghanistan AF    AFG    1984          NA           NA           NA           NA
##  6 Afghanistan AF    AFG    1985          NA           NA           NA           NA
##  7 Afghanistan AF    AFG    1986          NA           NA           NA           NA
##  8 Afghanistan AF    AFG    1987          NA           NA           NA           NA
##  9 Afghanistan AF    AFG    1988          NA           NA           NA           NA
## 10 Afghanistan AF    AFG    1989          NA           NA           NA           NA
## # ... with 7,230 more rows, and 52 more variables: new_sp_m4554 <int>,
## #   new_sp_m5564 <int>, new_sp_m65 <int>, new_sp_f014 <int>,
## #   new_sp_f1524 <int>, new_sp_f2534 <int>, new_sp_f3544 <int>,
## #   new_sp_f4554 <int>, new_sp_f5564 <int>, new_sp_f65 <int>,
## #   new_sn_m014 <int>, new_sn_m1524 <int>, new_sn_m2534 <int>,
## #   new_sn_m3544 <int>, new_sn_m4554 <int>, new_sn_m5564 <int>,
## #   new_sn_m65 <int>, new_sn_f014 <int>, new_sn_f1524 <int>, ...
```

```
who
```

```
## # A tibble: 7,240 x 60
```

```
##    country     iso2  iso3   year new_sp_m014 new_sp_m1524 new_sp_m2534 new_sp_m3544
##    <chr>       <chr> <chr> <int>       <int>        <int>        <int>        <int>
##  1 Afghanistan AF    AFG    1980          NA           NA           NA           NA
##  2 Afghanistan AF    AFG    1981          NA           NA           NA           NA
##  3 Afghanistan AF    AFG    1982          NA           NA           NA           NA
##  4 Afghanistan AF    AFG    1983          NA           NA           NA           NA
##  5 Afghanistan AF    AFG    1984          NA           NA           NA           NA
##  6 Afghanistan AF    AFG    1985          NA           NA           NA           NA
##  7 Afghanistan AF    AFG    1986          NA           NA           NA           NA
##  8 Afghanistan AF    AFG    1987          NA           NA           NA           NA
##  9 Afghanistan AF    AFG    1988          NA           NA           NA           NA
## 10 Afghanistan AF    AFG    1989          NA           NA           NA           NA
## # ... with 7,230 more rows, and 52 more variables: new_sp_m4554 <int>,
## #   new_sp_m5564 <int>, new_sp_m65 <int>, new_sp_f014 <int>,
## #   new_sp_f1524 <int>, new_sp_f2534 <int>, new_sp_f3544 <int>,
## #   new_sp_f4554 <int>, new_sp_f5564 <int>, new_sp_f65 <int>,
## #   new_sn_m014 <int>, new_sn_m1524 <int>, new_sn_m2534 <int>,
## #   new_sn_m3544 <int>, new_sn_m4554 <int>, new_sn_m5564 <int>,
## #   new_sn_m65 <int>, new_sn_f014 <int>, new_sn_f1524 <int>, ...
```

```
who1 <- who %>%
  pivot_longer(
    cols = new_sp_m014:newrel_f65,
    names_to = "key",
    values_to = "cases",
    values_drop_na = TRUE
  )
who1
```

```
## # A tibble: 76,046 x 6
##    country     iso2  iso3   year key          cases
##    <chr>       <chr> <chr> <int> <chr>        <int>
##  1 Afghanistan AF    AFG    1997 new_sp_m014      0
##  2 Afghanistan AF    AFG    1997 new_sp_m1524    10
##  3 Afghanistan AF    AFG    1997 new_sp_m2534     6
##  4 Afghanistan AF    AFG    1997 new_sp_m3544     3
##  5 Afghanistan AF    AFG    1997 new_sp_m4554     5
##  6 Afghanistan AF    AFG    1997 new_sp_m5564     2
##  7 Afghanistan AF    AFG    1997 new_sp_m65       0
##  8 Afghanistan AF    AFG    1997 new_sp_f014      5
##  9 Afghanistan AF    AFG    1997 new_sp_f1524    38
## 10 Afghanistan AF    AFG    1997 new_sp_f2534    36
## # ... with 76,036 more rows
```

```
who1 %>%
  count(key)
```

```
## # A tibble: 56 x 2
##    key              n
##    <chr>        <int>
##  1 new_ep_f014   1032
##  2 new_ep_f1524  1021
##  3 new_ep_f2534  1021
##  4 new_ep_f3544  1021
##  5 new_ep_f4554  1017
##  6 new_ep_f5564  1017
```

```
##  7 new_ep_f65    1014
##  8 new_ep_m014   1038
##  9 new_ep_m1524  1026
## 10 new_ep_m2534  1020
## # ... with 46 more rows
```

```
who2 <- who1 %>%
  mutate(key = stringr::str_replace(key, "newrel", "new_rel"))
who2
```

```
## # A tibble: 76,046 x 6
##    country     iso2  iso3   year key          cases
##    <chr>       <chr> <chr> <int> <chr>        <int>
##  1 Afghanistan AF    AFG    1997 new_sp_m014      0
##  2 Afghanistan AF    AFG    1997 new_sp_m1524    10
##  3 Afghanistan AF    AFG    1997 new_sp_m2534     6
##  4 Afghanistan AF    AFG    1997 new_sp_m3544     3
##  5 Afghanistan AF    AFG    1997 new_sp_m4554     5
##  6 Afghanistan AF    AFG    1997 new_sp_m5564     2
##  7 Afghanistan AF    AFG    1997 new_sp_m65       0
##  8 Afghanistan AF    AFG    1997 new_sp_f014      5
##  9 Afghanistan AF    AFG    1997 new_sp_f1524    38
## 10 Afghanistan AF    AFG    1997 new_sp_f2534    36
## # ... with 76,036 more rows
```

```
who3 <- who2 %>%
  separate(key, c("new", "type", "sexage"), sep = "_")
who3
```

```
## # A tibble: 76,046 x 8
##    country     iso2  iso3   year new   type  sexage cases
##    <chr>       <chr> <chr> <int> <chr> <chr> <chr>  <int>
##  1 Afghanistan AF    AFG    1997 new   sp    m014       0
##  2 Afghanistan AF    AFG    1997 new   sp    m1524     10
##  3 Afghanistan AF    AFG    1997 new   sp    m2534      6
##  4 Afghanistan AF    AFG    1997 new   sp    m3544      3
##  5 Afghanistan AF    AFG    1997 new   sp    m4554      5
##  6 Afghanistan AF    AFG    1997 new   sp    m5564      2
##  7 Afghanistan AF    AFG    1997 new   sp    m65        0
##  8 Afghanistan AF    AFG    1997 new   sp    f014       5
##  9 Afghanistan AF    AFG    1997 new   sp    f1524     38
## 10 Afghanistan AF    AFG    1997 new   sp    f2534     36
## # ... with 76,036 more rows
```

```
who4 <- who3 %>%
  select(-new, -iso2, -iso3)
who4
```

```
## # A tibble: 76,046 x 5
##    country      year type  sexage cases
##    <chr>       <int> <chr> <chr>  <int>
##  1 Afghanistan  1997 sp    m014       0
##  2 Afghanistan  1997 sp    m1524     10
##  3 Afghanistan  1997 sp    m2534      6
##  4 Afghanistan  1997 sp    m3544      3
##  5 Afghanistan  1997 sp    m4554      5
##  6 Afghanistan  1997 sp    m5564      2
```

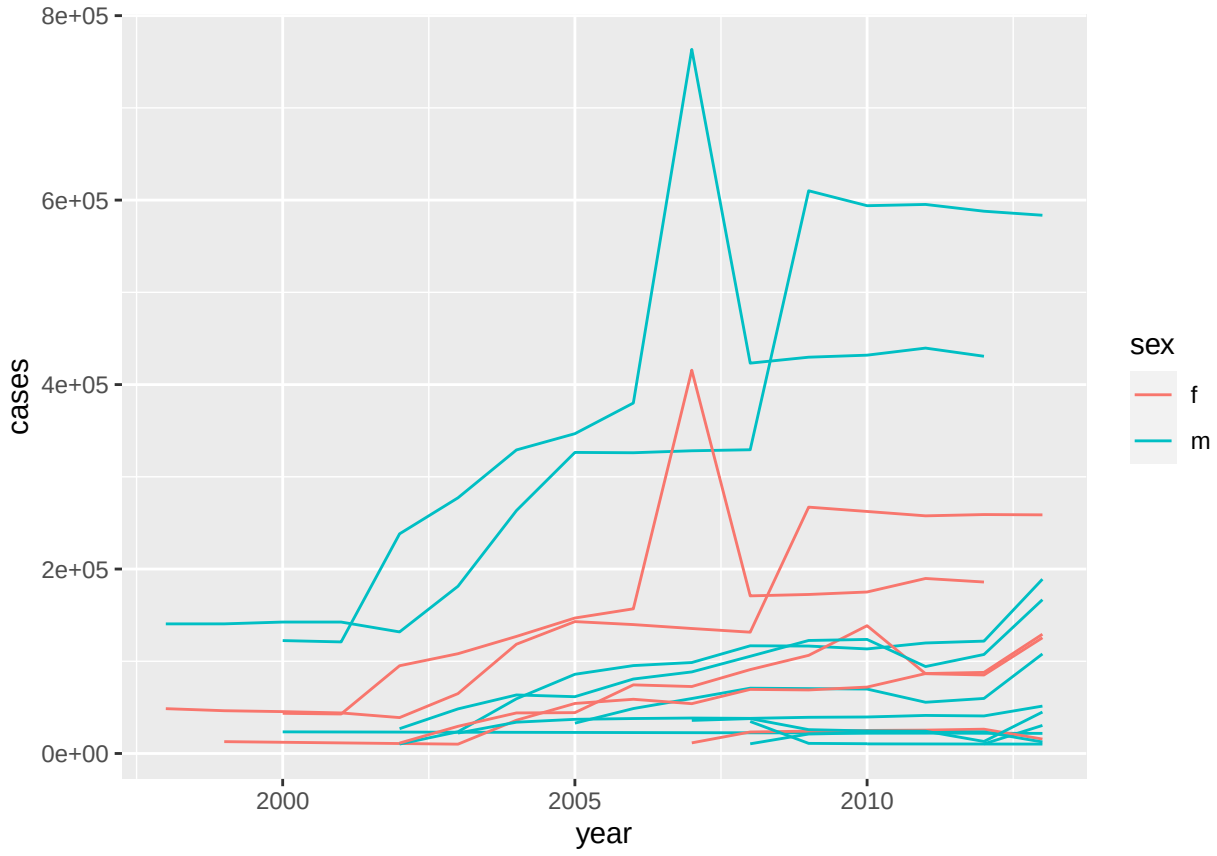
```
##  7 Afghanistan  1997 sp    m65        0
##  8 Afghanistan  1997 sp    f014       5
##  9 Afghanistan  1997 sp    f1524     38
## 10 Afghanistan  1997 sp    f2534     36
## # ... with 76,036 more rows
```

```
who5 <- who4 %>%
  separate(sexage, c("sex", "age"), sep = 1)
who5
```

```
## # A tibble: 76,046 x 6
##    country      year type  sex   age   cases
##    <chr>       <int> <chr> <chr> <chr> <int>
##  1 Afghanistan  1997 sp    m     014       0
##  2 Afghanistan  1997 sp    m     1524     10
##  3 Afghanistan  1997 sp    m     2534      6
##  4 Afghanistan  1997 sp    m     3544      3
##  5 Afghanistan  1997 sp    m     4554      5
##  6 Afghanistan  1997 sp    m     5564      2
##  7 Afghanistan  1997 sp    m     65        0
##  8 Afghanistan  1997 sp    f     014       5
##  9 Afghanistan  1997 sp    f     1524     38
## 10 Afghanistan  1997 sp    f     2534     36
## # ... with 76,036 more rows
```

```
who5 %>%
  group_by(country,year, sex) %>%
  filter(year>"1997") %>%
  filter(cases > 10000) %>%
  summarise(cases=sum(cases)) %>%
  unite(country_sex, country, sex, remove =FALSE) %>%
  ggplot(aes(x = year, y = cases, group = country_sex, colour = sex)) +
  geom_line()
```

```
## `summarise()` has grouped output by 'country', 'year'. You can override using the `.groups` argument
```




```
who5 %>%
  group_by(country, year, sex) %>%
  filter(year>"1997") %>%
  filter(cases < 100) %>%
  summarise(cases=sum(cases)) %>%
  unite(country_sex, country, sex, remove =FALSE) %>%
  ggplot(aes(x = year, y = cases, group = country_sex, colour = sex)) +
  geom_line()
```

```
## `summarise()` has grouped output by 'country', 'year'. You can override using the `.groups` argument
```


```
who5
```

```
## # A tibble: 76,046 x 6
##    country      year type  sex   age   cases
##    <chr>       <int> <chr> <chr> <chr> <int>
##  1 Afghanistan  1997 sp    m     014       0
##  2 Afghanistan  1997 sp    m     1524     10
##  3 Afghanistan  1997 sp    m     2534      6
##  4 Afghanistan  1997 sp    m     3544      3
##  5 Afghanistan  1997 sp    m     4554      5
##  6 Afghanistan  1997 sp    m     5564      2
##  7 Afghanistan  1997 sp    m     65        0
##  8 Afghanistan  1997 sp    f     014       5
##  9 Afghanistan  1997 sp    f     1524     38
## 10 Afghanistan  1997 sp    f     2534     36
## # ... with 76,036 more rows
```