Manipulación de datos - dplyr

Josman

10 de octubre de 2014

El paquete dplyr es una gran herramienta para la exploración y manipulación de datos. Cuenta con 5 principales funciones:

- filter()
- select()
- arrange()
- mutate()
- summarise()
- extra: group_by()

Para explorar las funciones de dplyr cargaremos una base de datos grande (21 variables, 227496 observaciones).

```
# Cargamos paquetes
suppressMessages(library(dplyr))
library(hflights)

# Exploramos datos
data(hflights)
head(hflights)
```

```
##      Year Month DayofMonth DayOfWeek DepTime ArrTime UniqueCarrier
## 5424 2011     1          1         6    1400    1500            AA
## 5425 2011     1          2         7    1401    1501            AA
## 5426 2011     1          3         1    1352    1502            AA
## 5427 2011     1          4         2    1403    1513            AA
## 5428 2011     1          5         3    1405    1507            AA
## 5429 2011     1          6         4    1359    1503            AA
##      FlightNum TailNum ActualElapsedTime AirTime ArrDelay DepDelay Origin
## 5424       428  N576AA                60      40      -10        0    IAH
## 5425       428  N557AA                60      45       -9        1    IAH
## 5426       428  N541AA                70      48       -8       -8    IAH
## 5427       428  N403AA                70      39        3        3    IAH
## 5428       428  N492AA                62      44       -3        5    IAH
## 5429       428  N262AA                64      45       -7       -1    IAH
##      Dest Distance TaxiIn TaxiOut Cancelled CancellationCode Diverted
## 5424  DFW      224      7      13         0                          0
## 5425  DFW      224      6       9         0                          0
## 5426  DFW      224      5      17         0                          0
## 5427  DFW      224      9      22         0                          0
## 5428  DFW      224      9       9         0                          0
## 5429  DFW      224      6      13         0                          0
```

Convertimos los datos en un local data frame. Esto hace que nuestros datos se impriman de manera más amigable.

```r
# Convertimos en un data frame local
flights <- tbl_df(hflights)

# Imprime únicamente los renglones que se ajustan a tu pantalla
flights
```

```
## Source: local data frame [227,496 x 21]
##
##      Year Month DayofMonth DayOfWeek DepTime ArrTime UniqueCarrier
## 5424 2011     1          1         6    1400    1500            AA
## 5425 2011     1          2         7    1401    1501            AA
## 5426 2011     1          3         1    1352    1502            AA
## 5427 2011     1          4         2    1403    1513            AA
## 5428 2011     1          5         3    1405    1507            AA
## 5429 2011     1          6         4    1359    1503            AA
## 5430 2011     1          7         5    1359    1509            AA
## 5431 2011     1          8         6    1355    1454            AA
## 5432 2011     1          9         7    1443    1554            AA
## 5433 2011     1         10         1    1443    1553            AA
## ..    ...   ...        ...       ...     ...     ...           ...
## Variables not shown: FlightNum (int), TailNum (chr), ActualElapsedTime
##   (int), AirTime (int), ArrDelay (int), DepDelay (int), Origin (chr), Dest
##   (chr), Distance (int), TaxiIn (int), TaxiOut (int), Cancelled (int),
##   CancellationCode (chr), Diverted (int)
```

```r
# Podemos espicificar que queremos ver más renglones
print(flights, n=20)
```

```
## Source: local data frame [227,496 x 21]
##
##      Year Month DayofMonth DayOfWeek DepTime ArrTime UniqueCarrier
## 5424 2011     1          1         6    1400    1500            AA
## 5425 2011     1          2         7    1401    1501            AA
## 5426 2011     1          3         1    1352    1502            AA
## 5427 2011     1          4         2    1403    1513            AA
## 5428 2011     1          5         3    1405    1507            AA
## 5429 2011     1          6         4    1359    1503            AA
## 5430 2011     1          7         5    1359    1509            AA
## 5431 2011     1          8         6    1355    1454            AA
## 5432 2011     1          9         7    1443    1554            AA
## 5433 2011     1         10         1    1443    1553            AA
## 5434 2011     1         11         2    1429    1539            AA
## 5435 2011     1         12         3    1419    1515            AA
## 5436 2011     1         13         4    1358    1501            AA
## 5437 2011     1         14         5    1357    1504            AA
## 5438 2011     1         15         6    1359    1459            AA
## 5439 2011     1         16         7    1359    1509            AA
## 5440 2011     1         17         1    1530    1634            AA
## 5441 2011     1         18         2    1408    1508            AA
## 5442 2011     1         19         3    1356    1503            AA
## 5443 2011     1         20         4    1507    1622            AA
## ..    ...   ...        ...       ...     ...     ...           ...
## Variables not shown: FlightNum (int), TailNum (chr), ActualElapsedTime
```

```
##   (int), AirTime (int), ArrDelay (int), DepDelay (int), Origin (chr), Dest
##   (chr), Distance (int), TaxiIn (int), TaxiOut (int), Cancelled (int),
##   CancellationCode (chr), Diverted (int)
```

```r
# Convertimos en un data frame 'normal'
data.frame(head(flights))
```

```
##      Year Month DayofMonth DayOfWeek DepTime ArrTime UniqueCarrier
## 5424 2011     1          1         6    1400    1500            AA
## 5425 2011     1          2         7    1401    1501            AA
## 5426 2011     1          3         1    1352    1502            AA
## 5427 2011     1          4         2    1403    1513            AA
## 5428 2011     1          5         3    1405    1507            AA
## 5429 2011     1          6         4    1359    1503            AA
##      FlightNum TailNum ActualElapsedTime AirTime ArrDelay DepDelay Origin
## 5424       428  N576AA                60      40      -10        0    IAH
## 5425       428  N557AA                60      45       -9        1    IAH
## 5426       428  N541AA                70      48       -8       -8    IAH
## 5427       428  N403AA                70      39        3        3    IAH
## 5428       428  N492AA                62      44       -3        5    IAH
## 5429       428  N262AA                64      45       -7       -1    IAH
##      Dest Distance TaxiIn TaxiOut Cancelled CancellationCode Diverted
## 5424  DFW      224      7      13         0                          0
## 5425  DFW      224      6       9         0                          0
## 5426  DFW      224      5      17         0                          0
## 5427  DFW      224      9      22         0                          0
## 5428  DFW      224      9       9         0                          0
## 5429  DFW      224      6      13         0                          0
```

filter

Indicamos un criterio para seleccionar renglones:

```r
# Modo tradicional para ver todos los vuelos de January 1
flights[flights$Month==1 & flights$DayofMonth==1, ]
```

```r
# Modo dplyr
# note: podemos usar coma o ampersand para representar AND
filter(flights, Month==1, DayofMonth==1)
```

```
## Source: local data frame [552 x 21]
##
##    Year Month DayofMonth DayOfWeek DepTime ArrTime UniqueCarrier FlightNum
## 1  2011     1          1         6    1400    1500            AA       428
## 2  2011     1          1         6     728     840            AA       460
## 3  2011     1          1         6    1631    1736            AA      1121
## 4  2011     1          1         6    1756    2112            AA      1294
## 5  2011     1          1         6    1012    1347            AA      1700
## 6  2011     1          1         6    1211    1325            AA      1820
## 7  2011     1          1         6     557     906            AA      1994
## 8  2011     1          1         6    1824    2106            AS       731
```

```
## 9  2011     1          1         6     654    1124            B6       620
## 10 2011     1          1         6    1639    2110            B6       622
## ..  ...   ...        ...       ...     ...     ...           ...       ...
## Variables not shown: TailNum (chr), ActualElapsedTime (int), AirTime
##   (int), ArrDelay (int), DepDelay (int), Origin (chr), Dest (chr),
##   Distance (int), TaxiIn (int), TaxiOut (int), Cancelled (int),
##   CancellationCode (chr), Diverted (int)
```

```
# Usamos 'pipe' para OR
filter(flights, UniqueCarrier=="AA" | UniqueCarrier=="UA")
```

```
## Source: local data frame [5,316 x 21]
##
##    Year Month DayofMonth DayOfWeek DepTime ArrTime UniqueCarrier FlightNum
## 1  2011     1          1         6    1400    1500            AA       428
## 2  2011     1          2         7    1401    1501            AA       428
## 3  2011     1          3         1    1352    1502            AA       428
## 4  2011     1          4         2    1403    1513            AA       428
## 5  2011     1          5         3    1405    1507            AA       428
## 6  2011     1          6         4    1359    1503            AA       428
## 7  2011     1          7         5    1359    1509            AA       428
## 8  2011     1          8         6    1355    1454            AA       428
## 9  2011     1          9         7    1443    1554            AA       428
## 10 2011     1         10         1    1443    1553            AA       428
## ..  ...   ...        ...       ...     ...     ...           ...       ...
## Variables not shown: TailNum (chr), ActualElapsedTime (int), AirTime
##   (int), ArrDelay (int), DepDelay (int), Origin (chr), Dest (chr),
##   Distance (int), TaxiIn (int), TaxiOut (int), Cancelled (int),
##   CancellationCode (chr), Diverted (int)
```

```
# También podemos usar el operador %in% para seleccionar distintos criterios
filter(flights, UniqueCarrier %in% c("AA", "UA"))
```

```
## Source: local data frame [5,316 x 21]
##
##    Year Month DayofMonth DayOfWeek DepTime ArrTime UniqueCarrier FlightNum
## 1  2011     1          1         6    1400    1500            AA       428
## 2  2011     1          2         7    1401    1501            AA       428
## 3  2011     1          3         1    1352    1502            AA       428
## 4  2011     1          4         2    1403    1513            AA       428
## 5  2011     1          5         3    1405    1507            AA       428
## 6  2011     1          6         4    1359    1503            AA       428
## 7  2011     1          7         5    1359    1509            AA       428
## 8  2011     1          8         6    1355    1454            AA       428
## 9  2011     1          9         7    1443    1554            AA       428
## 10 2011     1         10         1    1443    1553            AA       428
## ..  ...   ...        ...       ...     ...     ...           ...       ...
## Variables not shown: TailNum (chr), ActualElapsedTime (int), AirTime
##   (int), ArrDelay (int), DepDelay (int), Origin (chr), Dest (chr),
##   Distance (int), TaxiIn (int), TaxiOut (int), Cancelled (int),
##   CancellationCode (chr), Diverted (int)
```

select

Seleccionamos columnas por su nombre:

```
# Modo tradicional para seleccionar las columnas DepTime, ArrTime, y FlightNum
flights[, c("DepTime", "ArrTime", "FlightNum")]
```

```
# modo dplyr
select(flights, DepTime, ArrTime, FlightNum)
```

```
## Source: local data frame [227,496 x 3]
##
##      DepTime ArrTime FlightNum
## 5424    1400    1500       428
## 5425    1401    1501       428
## 5426    1352    1502       428
## 5427    1403    1513       428
## 5428    1405    1507       428
## 5429    1359    1503       428
## 5430    1359    1509       428
## 5431    1355    1454       428
## 5432    1443    1554       428
## 5433    1443    1553       428
## ..       ...     ...       ...
```

```
# Usamos ':' para seleccionar múltiples columnas continuas, y 'contains' para buscar columnas que coinc
# nota: `starts_with`, `ends_with`, y `matches` también puede ser usado para buscar columnas coincident
select(flights, Year:DayofMonth, contains("Taxi"), contains("Delay"))
```

```
## Source: local data frame [227,496 x 7]
##
##      Year Month DayofMonth TaxiIn TaxiOut ArrDelay DepDelay
## 5424 2011     1          1      7      13      -10        0
## 5425 2011     1          2      6       9       -9        1
## 5426 2011     1          3      5      17       -8       -8
## 5427 2011     1          4      9      22        3        3
## 5428 2011     1          5      9       9       -3        5
## 5429 2011     1          6      6      13       -7       -1
## 5430 2011     1          7     12      15       -1       -1
## 5431 2011     1          8      7      12      -16       -5
## 5432 2011     1          9      8      22       44       43
## 5433 2011     1         10      6      19       43       43
## ..    ...   ...        ...    ...     ...      ...      ...
```

Encadenando

Normalmente cuando aplicamos una secuencia de funciones a los datos vamos anidandolas una dentro de otra, lo cual puede ser muy confuso y sucio para el lector. Con este paquete podremos aplicar una secuencia de funciones 'encadenándolas':

```r
# Modo anidado para seleecionar las columnas UniqueCarrier y DepDelay y filtrarlas para delays mayores
filter(select(flights, UniqueCarrier, DepDelay), DepDelay > 60)
```

```r
# Modo encadenado
flights %>%
    select(UniqueCarrier, DepDelay) %>%
    filter(DepDelay > 60)
```

```
## Source: local data frame [10,242 x 2]
##
##    UniqueCarrier DepDelay
## 1             AA       90
## 2             AA       67
## 3             AA       74
## 4             AA      125
## 5             AA       82
## 6             AA       99
## 7             AA       70
## 8             AA       61
## 9             AA       74
## 10            AS       73
## ..           ...      ...
```

arrange

Reordenar renglones:

```r
# Modo tradicional para seleccionar las columnas UniqueCarrier y DepDelay y ornenarlas según el DepDela
flights[order(flights$DepDelay), c("UniqueCarrier", "DepDelay")]
```

```r
# Modo dplyr + encadenado
flights %>%
    select(UniqueCarrier, DepDelay) %>%
    arrange(DepDelay)
```

```
## Source: local data frame [227,496 x 2]
##
##    UniqueCarrier DepDelay
## 1             OO      -33
## 2             MQ      -23
## 3             XE      -19
## 4             XE      -19
## 5             CO      -18
## 6             EV      -18
## 7             XE      -17
## 8             CO      -17
## 9             XE      -17
## 10            MQ      -17
## ..           ...      ...
```

```
# Usamos `desc` para orden descendiente
flights %>%
    select(UniqueCarrier, DepDelay) %>%
    arrange(desc(DepDelay))
```

```
## Source: local data frame [227,496 x 2]
##
##    UniqueCarrier DepDelay
## 1             CO      981
## 2             AA      970
## 3             MQ      931
## 4             UA      869
## 5             MQ      814
## 6             MQ      803
## 7             CO      780
## 8             CO      758
## 9             DL      730
## 10            MQ      691
## ..           ...      ...
```

mutate

Agrega nuevas variables, las cuales son funciones de las variables ya existentes:

```
# Modo tradicional para crear la variable Speed
flights$Speed <- flights$Distance / flights$AirTime*60
flights[, c("Distance", "AirTime", "Speed")]
```

```
# Modo dplyr
flights %>%
    select(Distance, AirTime) %>%
    mutate(Speed = Distance/AirTime*60)
```

```
## Source: local data frame [227,496 x 3]
##
##    Distance AirTime    Speed
## 1       224      40 336.0000
## 2       224      45 298.6667
## 3       224      48 280.0000
## 4       224      39 344.6154
## 5       224      44 305.4545
## 6       224      45 298.6667
## 7       224      43 312.5581
## 8       224      40 336.0000
## 9       224      41 327.8049
## 10      224      45 298.6667
## ..      ...     ...      ...
```

```
# Guardamos la nueva variable
flights <- flights %>% mutate(Speed = Distance/AirTime*60)
```

summarise

Reducir variables a valores. Muy útil cuando los datos están agrupados por una o más variables. Con `group_by` crearemos grupos para dividir los datos, `summarise` nos permitirá presentar los datos.

```
# Modo tradicional para calcular el promedio de 'delay' en cada destino
head(with(flights, tapply(ArrDelay, Dest, mean, na.rm=TRUE)))
```

```
##       ABQ       AEX       AGS       AMA       ANC       ASE
##  7.226259  5.839437  4.000000  6.840095 26.080645  6.794643
```

```
head(aggregate(ArrDelay ~ Dest, flights, mean))
```

```
##   Dest  ArrDelay
## 1  ABQ  7.226259
## 2  AEX  5.839437
## 3  AGS  4.000000
## 4  AMA  6.840095
## 5  ANC 26.080645
## 6  ASE  6.794643
```

```
# Modo dplyr
flights %>%
    group_by(Dest) %>%
    summarise(avg_delay = mean(ArrDelay, na.rm=TRUE))
```

```
## Source: local data frame [116 x 2]
##
##    Dest  avg_delay
## 1   ABQ   7.226259
## 2   AEX   5.839437
## 3   AGS   4.000000
## 4   AMA   6.840095
## 5   ANC  26.080645
## 6   ASE   6.794643
## 7   ATL   8.233251
## 8   AUS   7.448718
## 9   AVL   9.973988
## 10  BFL -13.198807
## ..  ...         ...
```

```
# Para cada corrida calculamos el porcentaje de vuelos cancelados o desviados
flights %>%
    group_by(UniqueCarrier) %>%
    summarise_each(funs(mean), Cancelled, Diverted)
```

```
## Source: local data frame [15 x 3]
##
##    UniqueCarrier   Cancelled    Diverted
## 1             AA 0.018495684 0.001849568
## 2             AS 0.000000000 0.002739726
```

```
## 3             B6 0.025899281 0.005755396
## 4             CO 0.006782614 0.002627370
## 5             DL 0.015903067 0.003029156
## 6             EV 0.034482759 0.003176044
## 7             F9 0.007159905 0.000000000
## 8             FL 0.009817672 0.003272557
## 9             MQ 0.029044750 0.001936317
## 10            OO 0.013946828 0.003486707
## 11            UA 0.016409266 0.002413127
## 12            US 0.011268986 0.001469868
## 13            WN 0.015504047 0.002293629
## 14            XE 0.015495599 0.003449550
## 15            YV 0.012658228 0.000000000
```

```
# Para cada corrida calculamos el minimo y máximo de los retrasos
flights %>%
    group_by(UniqueCarrier) %>%
    summarise_each(funs(min(., na.rm=TRUE), max(., na.rm=TRUE)), matches("Delay"))
```

```
## Source: local data frame [15 x 5]
##
##    UniqueCarrier ArrDelay_min DepDelay_min ArrDelay_max DepDelay_max
## 1             AA          -39          -15          978          970
## 2             AS          -43          -15          183          172
## 3             B6          -44          -14          335          310
## 4             CO          -55          -18          957          981
## 5             DL          -32          -17          701          730
## 6             EV          -40          -18          469          479
## 7             F9          -24          -15          277          275
## 8             FL          -30          -14          500          507
## 9             MQ          -38          -23          918          931
## 10            OO          -57          -33          380          360
## 11            UA          -47          -11          861          869
## 12            US          -42          -17          433          425
## 13            WN          -44          -10          499          548
## 14            XE          -70          -19          634          628
## 15            YV          -32          -11           72           54
```

```
# Para cada día contamos el número de vuelos y los ordenamos
flights %>%
    group_by(Month, DayofMonth) %>%
    summarise(flight_count = n()) %>%
    arrange(desc(flight_count))
```

```
## Source: local data frame [365 x 3]
## Groups: Month
##
##    Month DayofMonth flight_count
## 1      8          4          706
## 2      8         11          706
## 3      8         12          706
## 4      8          5          705
## 5      8          3          704
```

```
## 6      8         10          704
## 7      1          3          702
## 8      7          7          702
## 9      7         14          702
## 10     7         28          701
## ..   ...        ...          ...
```

```
# Otra forma más sencilla de escribirlo con la función 'tally'
flights %>%
    group_by(Month, DayofMonth) %>%
    tally(sort = TRUE)
```

```
## Source: local data frame [365 x 3]
## Groups: Month
##
##    Month DayofMonth   n
## 1      8          4 706
## 2      8         11 706
## 3      8         12 706
## 4      8          5 705
## 5      8          3 704
## 6      8         10 704
## 7      1          3 702
## 8      7          7 702
## 9      7         14 702
## 10     7         28 701
## ..   ...        ... ...
```

```
# Para cada destino contamos el núero de vuelos y número de aviones
flights %>%
    group_by(Dest) %>%
    summarise(flight_count = n(), plane_count = n_distinct(TailNum))
```

```
## Source: local data frame [116 x 3]
##
##    Dest flight_count plane_count
## 1   ABQ         2812         716
## 2   AEX          724         215
## 3   AGS            1           1
## 4   AMA         1297         158
## 5   ANC          125          38
## 6   ASE          125          60
## 7   ATL         7886         983
## 8   AUS         5022        1015
## 9   AVL          350         142
## 10  BFL          504          70
## ..  ...          ...         ...
```

```
# Para cada destino mostramos el número de vuelos cancelados y no cancelados
flights %>%
    group_by(Dest) %>%
    select(Cancelled) %>%
    table() %>%
    head()
```

```
##      Cancelled
## Dest     0  1
##   ABQ 2787 25
##   AEX  712 12
##   AGS    1  0
##   AMA 1265 32
##   ANC  125  0
##   ASE  120  5
```

Otras funciones muy útiles

```
# Obtener una muestra aleatoria de los datos por número de datos
flights %>% sample_n(5)
```

```
## Source: local data frame [5 x 22]
##
##        Year Month DayofMonth DayOfWeek DepTime ArrTime UniqueCarrier
## 35289  2011     2          4         5    1411    1515            WN
## 147919 2011     8         29         1    1305    1652            CO
## 28563  2011     2         24         4    1750    1845            XE
## 5763   2011     1          4         2    1335    1510            CO
## 75106  2011     5         28         6    1550    2007            CO
## Variables not shown: FlightNum (int), TailNum (chr), ActualElapsedTime
##   (int), AirTime (int), ArrDelay (int), DepDelay (int), Origin (chr), Dest
##   (chr), Distance (int), TaxiIn (int), TaxiOut (int), Cancelled (int),
##   CancellationCode (chr), Diverted (int), Speed (dbl)
```

```
# Muestra según una proporción del tamaño de los datos
flights %>% sample_frac(0.25, replace=TRUE)
```

```
## Source: local data frame [56,874 x 22]
##
##        Year Month DayofMonth DayOfWeek DepTime ArrTime UniqueCarrier
## 28264  2011     2         23         3    1313    1635            XE
## 66241  2011     4         18         1    1451    1802            XE
## 145950 2011     8         14         7    1759    2042            UA
## 212031 2011    12         14         3    1715    2049            CO
## 173437 2011    10         24         1    1540    1928            CO
## 23705  2011     2          6         7    2244    2340            CO
## 22921  2011     2         10         4    1052    1426            CO
## 115939 2011     7         18         1     930    1047            CO
## 115001 2011     7         23         6    1403    1740            CO
## 151731 2011     8          9         2    1906    2203            CO
## ..      ...   ...        ...       ...     ...     ...           ...
## Variables not shown: FlightNum (int), TailNum (chr), ActualElapsedTime
##   (int), AirTime (int), ArrDelay (int), DepDelay (int), Origin (chr), Dest
##   (chr), Distance (int), TaxiIn (int), TaxiOut (int), Cancelled (int),
##   CancellationCode (chr), Diverted (int), Speed (dbl)
```

```r
# Modo tradicional para ver la estructura de un objeto
str(flights)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    227496 obs. of  22 variables:
##  $ Year             : int  2011 2011 2011 2011 2011 2011 2011 2011 2011 2011 ...
##  $ Month            : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ DayofMonth       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ DayOfWeek        : int  6 7 1 2 3 4 5 6 7 1 ...
##  $ DepTime          : int  1400 1401 1352 1403 1405 1359 1359 1355 1443 1443 ...
##  $ ArrTime          : int  1500 1501 1502 1513 1507 1503 1509 1454 1554 1553 ...
##  $ UniqueCarrier    : chr  "AA" "AA" "AA" "AA" ...
##  $ FlightNum        : int  428 428 428 428 428 428 428 428 428 428 ...
##  $ TailNum          : chr  "N576AA" "N557AA" "N541AA" "N403AA" ...
##  $ ActualElapsedTime: int  60 60 70 70 62 64 70 59 71 70 ...
##  $ AirTime          : int  40 45 48 39 44 45 43 40 41 45 ...
##  $ ArrDelay         : int  -10 -9 -8 3 -3 -7 -1 -16 44 43 ...
##  $ DepDelay         : int  0 1 -8 3 5 -1 -1 -5 43 43 ...
##  $ Origin           : chr  "IAH" "IAH" "IAH" "IAH" ...
##  $ Dest             : chr  "DFW" "DFW" "DFW" "DFW" ...
##  $ Distance         : int  224 224 224 224 224 224 224 224 224 224 ...
##  $ TaxiIn           : int  7 6 5 9 9 6 12 7 8 6 ...
##  $ TaxiOut          : int  13 9 17 22 9 13 15 12 22 19 ...
##  $ Cancelled        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CancellationCode : chr  "" "" "" "" ...
##  $ Diverted         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Speed            : num  336 299 280 345 305 ...
```

```r
# Modo dplyr: mejor formato y se adapta a tu pantalla
glimpse(flights)
```

```
## Variables:
## $ Year              (int) 2011, 2011, 2011, 2011, 2011, 2011, 2011, 20...
## $ Month             (int) 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ DayofMonth        (int) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1...
## $ DayOfWeek         (int) 6, 7, 1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6,...
## $ DepTime           (int) 1400, 1401, 1352, 1403, 1405, 1359, 1359, 13...
## $ ArrTime           (int) 1500, 1501, 1502, 1513, 1507, 1503, 1509, 14...
## $ UniqueCarrier     (chr) "AA", "AA", "AA", "AA", "AA", "AA", "AA", "A...
## $ FlightNum         (int) 428, 428, 428, 428, 428, 428, 428, 428, 428,...
## $ TailNum           (chr) "N576AA", "N557AA", "N541AA", "N403AA", "N49...
## $ ActualElapsedTime (int) 60, 60, 70, 70, 62, 64, 70, 59, 71, 70, 70, ...
## $ AirTime           (int) 40, 45, 48, 39, 44, 45, 43, 40, 41, 45, 42, ...
## $ ArrDelay          (int) -10, -9, -8, 3, -3, -7, -1, -16, 44, 43, 29,...
## $ DepDelay          (int) 0, 1, -8, 3, 5, -1, -1, -5, 43, 43, 29, 19, ...
## $ Origin            (chr) "IAH", "IAH", "IAH", "IAH", "IAH", "IAH", "I...
## $ Dest              (chr) "DFW", "DFW", "DFW", "DFW", "DFW", "DFW", "D...
## $ Distance          (int) 224, 224, 224, 224, 224, 224, 224, 224, 224,...
## $ TaxiIn            (int) 7, 6, 5, 9, 9, 6, 12, 7, 8, 6, 8, 4, 6, 5, 6...
## $ TaxiOut           (int) 13, 9, 17, 22, 9, 13, 15, 12, 22, 19, 20, 11...
## $ Cancelled         (int) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ CancellationCode  (chr) "", "", "", "", "", "", "", "", "", "", "", ...
## $ Diverted          (int) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Speed             (dbl) 336.0000, 298.6667, 280.0000, 344.6154, 305....
```