Principios y Herramientas de Programación

Dra. Jessica Andrea Carballido

jac@cs.uns.edu.ar

Dpto. de Ciencias e Ingeniería de la Computación

UNIVERSIDAD NACIONAL DEL SUR


```
    5
ames(sort(apply(ejemplo,1,sum)))[1]
1] "d"
> apply(ejemplo,1,sum)
a b c d e f g h
7 6 5 4 5 6 7 8
> sort(apply(ejemplo,1,sum))
d c e b f a g h
4 5 5 6 6 7 7 8
sort(apply(ejemplo,1,sum))[1]

y(ejemplo,1,sum))
```

set.seed

```
> set.seed(5)
> rnorm(5)
[1] -0.84085548  1.38435934 -1.25549186  0.07014277  1.71144087
>
```

set.seed(seed)

Set the seed of R's random number generator, which is useful for creating simulations or random objects that can be reproduced.

- seed – A number.

Example. Create simulated values that are reproducible.

```
> set.seed(5)
> rnorm(5)
[1] -0.84085548  1.38435934 -1.25549186  0.07014277  1.71144087
> set.seed(5)
> rnorm(5)
[1] -0.84085548  1.38435934 -1.25549186  0.07014277  1.71144087
```

The random numbers are the same, and they would continue to be the same no matter how far out in the sequence we went.

Tip. Use the set.seed function when running simulations to ensure all results, figures, etc are reproducible.

```
> m=matrix(round(runif(12,1,10)),3)
> m
     [,1] [,2] [,3] [,4]
[1,]   10    8    2    8
[2,]    9    9    8    6
[3,]    3    9    3   10
```

Insertar una fila en la tercera fila:

```
> rbind(m[1:2,],77,m[3,])
     [,1] [,2] [,3] [,4]
[1,]   10    8    2    8
[2,]    9    9    8    6
[3,]   77   77   77   77
[4,]    3    9    3   10
```

```
insertarFila=function(m,fila,pos){
  if(nrow(m)<pos) "No se puede, no existe esa posicion"
  else
   rbind(m[1:(pos-1),],fila,m[pos:nrow(m),])
}
```

```
> m
     [,1] [,2] [,3] [,4]
[1,]   10    8    2    8
[2,]    9    9    8    6
[3,]    3    9    3   10
```

```
> insertarFila(m,40,2)
     [,1] [,2] [,3] [,4]
       10    8    2    8
fila   40   40   40   40
        9    9    8    6
        3    9    3   10
```

Dra. Jessica Andrea Carballido
CONICET - DCIC (UNS)

MANEJO DE NA

```
> v<-c(1,2,3,NA,4)
> mean(v)
[1] NA
> mean(v,na.rm=T)
[1] 2.5
```

```
> v=c(1,2,46,NA,34,NA)
> na.omit(v)
[1]  1  2 46 34
attr(,"na.action")
[1] 4 6
attr(,"class")
[1] "omit"
> sum(v)
[1] NA
> sum(na.omit(v))
[1] 83
```

```
> v[!is.na(v)]
[1]  1  2 46 34
> v[is.na(v)]
[1] NA NA
```

Dra. Jessica Andrea Carballido
CONICET - DCIC (UNS)

Listas

Las listas son estructuras de datos
que permiten concatenar objetos de distintos tipos.

```
>
> familia=list(padre="juan", madre="maria", cantHijos=3,
+ nombresHijos=c("lola","eva","juanJr"), edadesHijos=c(5,7,3))
> familia
$padre
[1] "juan"

$madre
[1] "maria"

$cantHijos
[1] 3

$nombresHijos
[1] "lola"   "eva"    "juanJr"

$edadesHijos
[1] 5 7 3

> familia$madre
[1] "maria"
> familia[[2]]
[1] "maria"
> names(familia)
[1] "padre"        "madre"        "cantHijos"    "nombresHijos" "edadesHijos"
>
```

```
> gastos=list("enero",150,"GAS")
> gastos
[[1]]
[1] "enero"

[[2]]
[1] 150

[[3]]
[1] "GAS"

> names(gastos)=c("mes","importe","rubro")
> gastos
$mes
[1] "enero"

$importe
[1] 150

$rubro
[1] "GAS"
```

Los elementos se acceden por su nombre con $...
o por su posición con [[…]]

Factores

Un factor es un tipo particular de vector
que se usa para especificar una **clasificación discreta**
de las componentes de otros vectores de igual longitud.

```
>
> estudiantesOrigen=factor(c("BB","BA","Cordoba","Santa Fe","BB","BB","Cordoba","Santa Fe","BA","BA","BA"))
> estudiantesOrigen
 [1] BB       BA       Cordoba  Santa Fe BB       BB       Cordoba  Santa Fe BA       BA       BA
Levels: BA BB Cordoba Santa Fe
> summary(estudiantesOrigen)
      BA       BB  Cordoba Santa Fe
       4        3        2        2
> estudiantesMatAp=c(5,3,5,7,9,7,5,9,3,6,8)
> tapply(estudiantesMatAp, estudiantesOrigen, mean) # Promedio de materias aprobadas por Ciudad
      BA       BB  Cordoba Santa Fe
       5        7        5        8
> |
```

tapply(vector, factor, función)

La función *tapply* se usa para aplicar una función
a cada grupo de componentes del primer argumento (vector),
definidos por los niveles del segundo argumento (factor).

Factores

```
> v=c("juan", "pedro", "juan")
> summary(v)
   Length     Class      Mode
        3 character character
> factor(v)
[1] juan  pedro juan
Levels: juan pedro
> summary(factor(v))
 juan pedro
    2     1
```

```
> v=c(1,1,2,3,5,1,3,1)
> summary(v)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   1.500   2.125   3.000   5.000
> f=factor(v)
> summary(f)
1 2 3 5
4 1 2 1
```

```
> tapply(v,v,sum)
1 2 3 5
4 2 6 5
```

Diferencia entre vector y factor.

Algunas operaciones hacen la conversión automática de vector a factor

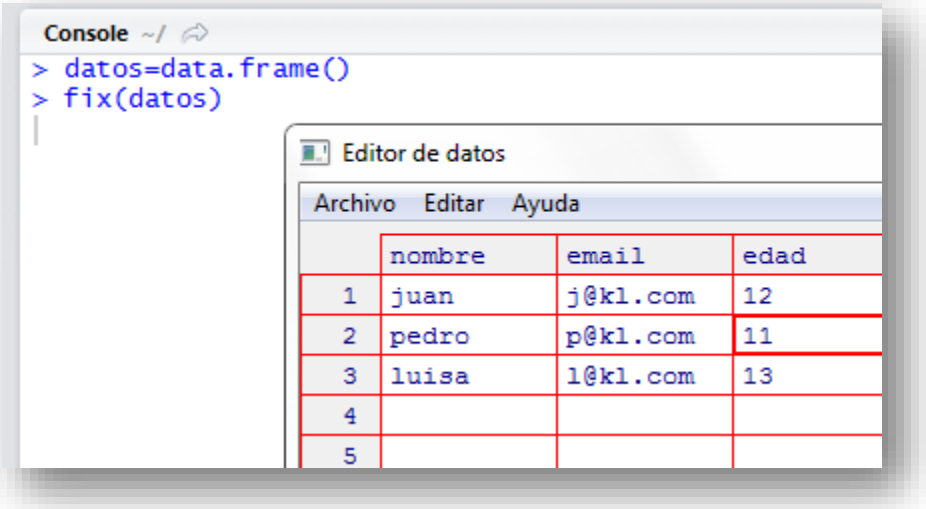
Data Frames

Los *data frames* son estructuras de datos que generalizan a las matrices en el sentido en que las columnas pueden ser de distinto tipo entre sí (cada una es un vector). Usamos la función **data.frame()** para generarlos.

```
>
> datos=matrix(c(20,65,174,22,70,180,19,68,170), nrow=3,byrow=T)
> dimnames(datos)=list(c("paco","pepe","kiko"),c("edad","peso","altura"))
> provincia=c("madrid","malaga","murcia")
> todosLosDatos=data.frame(datos,provincia)
> todosLosDatos
      edad peso altura provincia
paco    20   65    174    madrid
pepe    22   70    180    malaga
kiko    19   68    170    murcia
> mean(todosLosDatos[,"edad"])
[1] 20.33333
> apply(todosLosDatos[,1:3],2,mean)
      edad       peso    altura
 20.33333   67.66667 174.66667
> max(todosLosDatos$altura)     # lo accedo en modo lista indistintamente
[1] 180
```

Data Frames

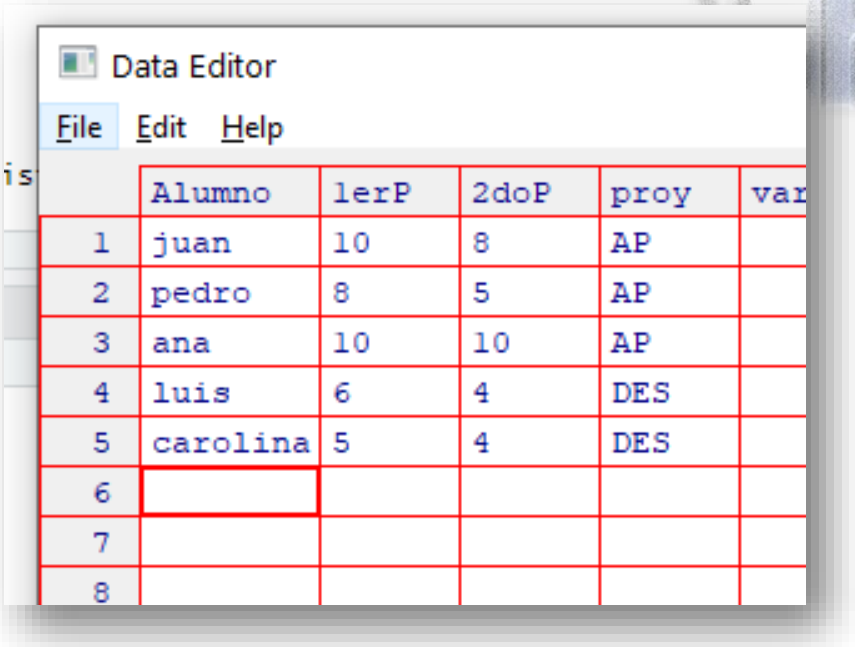
Otra forma de crear un *data frame*:



En general se usan para representar
conjuntos de datos con registros
de varios experimentos (filas) para varias variables (columnas).

También las filas pueden representar individuos.

Data Frames



Crear el *dataframe*.

Escribir expresiones para:

a) Calcular el promedio de cada parcial (apply).
b) Calcular cuantos aprobado alumnos aprobaron el proyecto.
c) Calcular, para el 1er parcial, el promedio de los aprobados y el promedio de los desaprobados (tapply).

Data Frames: datos externos

- Crear la hoja de datos en Excel, exportarla como archivo de texto (*text(MSDOS)*). Luego usar:

  datos=**read.table**(file.choose(), header=T)

Con header=T especificamos que la primera fila contiene los nombres de las columnas.

- Desde Excel, también se puede exportar la hoja en formato texto *delimitado por tabulaciones* (*tab-delimited*) o formato separado por comas, y usar **read.delim** o **read.csv** para importar los datos.

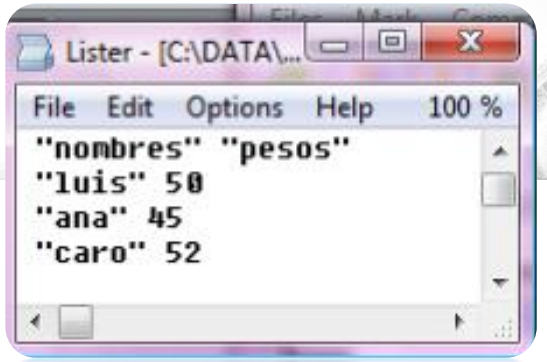
Datos externos

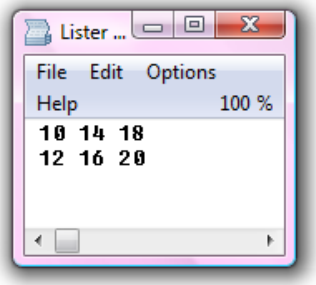
- Ingresar de teclado cada vector con scan()

```
>
> nombres=scan("",what="character")
1: "luis"
2: "ana"
3: "caro"
4:
Read 3 items
> pesos=scan()
1: 50
2: 45
3: 52
4:
Read 3 items
> PlanAlimenticioDatos=data.frame(nombres,pesos)
> PlanAlimenticioDatos
  nombres pesos
1    luis    50
2     ana    45
3    caro    52
>
```

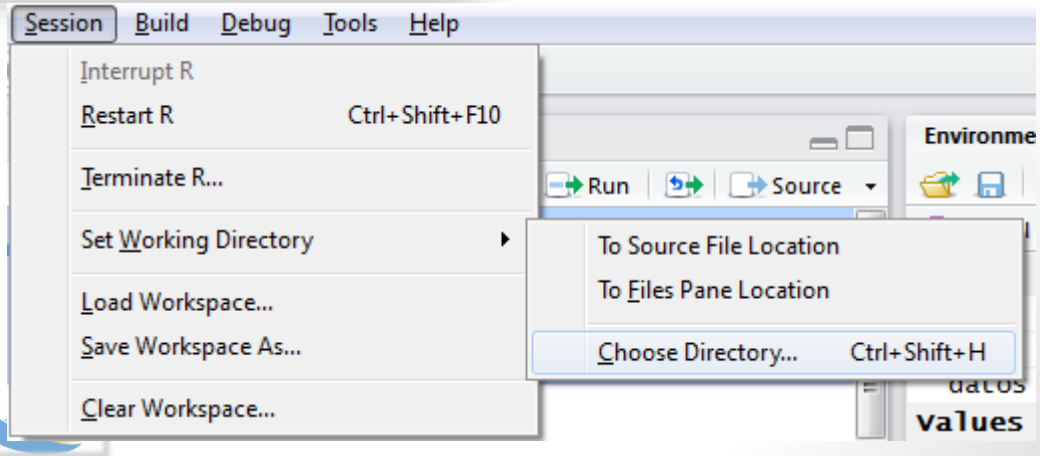
Exportar datos

```
>
> PlanAlimenticioDatos
  nombres pesos
1    luis    50
2     ana    45
3    caro    52
> write.table(PlanAlimenticioDatos,file="c:/DATA/plan.txt",row.names=F,col.names=T)
>
```






```
>
> x=matrix(seq(10,20,length=6), 2)
> x
     [,1] [,2] [,3]
[1,]   10   14   18
[2,]   12   16   20
> write.table(x, "c:/DATA/matriz.txt", row.names=F,col.names=F)
>
```



Dra. Jessica Andrea Carballido
CONICET - DCIC (UNS)

Conjuntos de datos incluidos en R

- Con la función data() obtenemos un listado de los *datasets* de R. Con el nombre del *dataset* podemos ver los valores almacenados.

R Console

```
> AirPassengers
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1949 112 118 132 129 121 135 148 148 136 119 104 118
1950 115 126 141 135 125 149 170 170 158 133 114 140
1951 145 150 178 163 172 178 199 199 184 162 146 166
1952 171 180 193 181 183 218 230 242 209 191 172 194
1953 196 196 236 235 229 243 264 272 237 211 180 201
1954 204 188 235 227 234 264 302 293 259 229 203 229
1955 242 233 267 269 270 315 364 347 312 274 237 278
1956 284 277 317 313 318 374 413 405 355 306 271 306
1957 315 301 356 348 355 422 465 467 404 347 305 336
1958 340 318 362 348 363 435 491 505 404 359 310 337
1959 360 342 406 396 420 472 548 559 463 407 362 405
1960 417 391 419 461 472 535 622 606 508 461 390 432
> HairEyeColor
, , Sex = Male

       Eye
Hair    Brown Blue Hazel Green
  Black    32   11    10     3
  Brown    53   50    25    15
  Red      10   10     7     7
  Blond     3   30     5     8

, , Sex = Female

       Eye
Hair    Brown Blue Hazel Green
  Black    36    9     5     2
  Brown    66   34    29    14
  Red      16    7     7     7
  Blond     4   64     5     8
```

R data sets

```
Data sets in package 'datasets':

AirPassengers                Monthly Airline Pass
BJsales                      Sales Data with Lead
BJsales.lead (BJsales)       Sales Data with Lead
BOD                          Biochemical Oxygen D
CO2                          Carbon Dioxide Uptak
ChickWeight                  Weight versus age of
DNase                        Elisa assay of DNase
EuStockMarkets               Daily Closing Prices
Formaldehyde                 Determination of For
HairEyeColor                 Hair and Eye Color o
Harman23.cor                 Harman Example 2.3
Harman74.cor                 Harman Example 7.4
Indometh                     Pharmacokinetics of
InsectSprays                 Effectiveness of Ins
JohnsonJohnson               Quarterly Earnings p
LakeHuron                    Level of Lake Huron
LifeCycleSavings             Intercountry Life-Cy
Loblolly                     Growth of Loblolly p
Nile                         Flow of the River Ni
Orange                       Growth of Orange Tre
OrchardSprays                Potency of Orchard S
PlantGrowth                  Results from an Expe
Puromycin                    Reaction Velocity of
```

Conjuntos de datos incluidos en R

4 dimensiones: Class, Sex, Age y Survived

```
> Titanic
, , Age = Child, Survived = No

      Sex
Class  Male Female
  1st     0      0
  2nd     0      0
  3rd    35     17
  Crew    0      0

, , Age = Adult, Survived = No

      Sex
Class  Male Female
  1st   118      4
  2nd   154     13
  3rd   387     89
  Crew  670      3

, , Age = Child, Survived = Yes

      Sex
Class  Male Female
  1st     5      1
  2nd    11     13
  3rd    13     14
  Crew    0      0

, , Age = Adult, Survived = Yes

      Sex
Class  Male Female
  1st    57    140
  2nd    14     80
  3rd    75     76
  Crew  192     20
```

```
> Titanic["1st", "Male", "Child", "Yes"]
[1] 5
```

```
> apply(Titanic[,,"Adult","Yes"],2,sum)
  Male Female
   338    316
> names(sort(Titanic[,"Male","Child","Yes"],decreasing=T))[1]
[1] "3rd"
> max(Titanic[,"Male","Child","Yes"])
[1] 13
>
```

```
> Titanic["1st",,"Adult",]
        Survived
Sex       No Yes
  Male   118  57
  Female   4 140
>
```

```
> Titanic["1st","Male","Adult",]
 No Yes
118  57
>
```

```
> Orange
   Tree  age circumference
1     1  118            30
2     1  484            58
3     1  664            87
4     1 1004           115
5     1 1231           120
6     1 1372           142
7     1 1582           145
8     2  118            33
9     2  484            69
10    2  664           111
11    2 1004           156
12    2 1231           172
13    2 1372           203
14    2 1582           203
15    3  118            30
16    3  484            51
17    3  664            75
18    3 1004           108
19    3 1231           115
20    3 1372           139
21    3 1582           140
22    4  118            32
23    4  484            62
24    4  664           112
25    4 1004           167
26    4 1231           179
27    4 1372           209
28    4 1582           214
29    5  118            30
30    5  484            49
31    5  664            81
32    5 1004           125
33    5 1231           142
34    5 1372           174
35    5 1582           177
```

```
> subset(Orange,Orange$age==118)
   Tree age circumference
1     1 118            30
8     2 118            33
15    3 118            30
22    4 118            32
29    5 118            30
> subset(Orange,Orange$age==118, select=circumference)
   circumference
1             30
8             33
15            30
22            32
29            30
```

```
> nrow(subset(Orange,Orange$age==118))
[1] 5
```

subset: retorna un sub-grupo de filas del 1er argumento, de acuerdo al criterio mostrado en el 2do argumento, **con todas las columnas salvo que se indiquen solo algunas con el 3er argumento**

Instalación de paquetes

Los paquetes contienen primitivas y pueden contener datasets.

- 2 formas (hay más)

1. Desde internet:

   install.packages("nombrePaquete")

2. Desde un .zip local:

Bajar el ZIP que contiene el paquete desde la página de R. Opción CRAN y en la ventana que se abre elegís el país de tu preferencia y en las opciones de la izquierda "Packages". Inmediatamente se abrirá una página con todos los paquetes disponibles, buscas el de tu preferencia y luego lo bajas según tu sistema operativo. En cada página del paquete hay un manual del mismo, es recomendable que le des una mirada

Cuando el paquete ya está en tu disco duro recién vas al Rstudio y elegís la opción paquetes, se desplegará el menú y al final está la opción "Instalar paquetes desde archivos ZIP locales", abre el menú de búsqueda, y elegís el archivo.

- Una vez que el paquete está INSTALADO, hay que CARGARLO ejecutando:

  library("nombrePaquete")

- Instalar paquete **LearnBayes**
- Cargar el dataset *studentdata*
- Analizar el contenido del dataset (?studentdata)
- Escribir expresiones o asignaciones para:
  - **Eliminar** del dataset las filas que tienen algún NA. ¿Cuántas filas se eliminaron?
  - Obtener el promedio de altura agrupado de acuerdo al sexo
  - Obtener el promedio de horas trabajadas, agrupadas por el tipo de bebida que toman
  - **Reducir** el dataset a los estudiantes que trabajan (horas de trabajo distinto de 0) y las columnas Student Height Gender Haircut Job Drink
  - Obtener el número (Student) de alumno que tiene un trabajo con más horas. ¿Cuantas horas trabaja por semana?
  - Obtener el promedio de las columnas Height Haircut Job
  - Generar un sub conjunto de datos (llamarlo AGUA) de los estudiantes que toman agua, viendo solo las columnas Student Gender. ¿Cúantas mujeres y cuántos varones hay?
  - Generar un sub conjunto de datos (llamarlo ALGUNOS) de los estudiantes varones que toman leche o refresco. Ver los primeros elementos.

¿Para qué sirve la función “attach”?

Examen: Martes 19/11
8.15 hs.

Dra. Jessica Andrea Carballido
CONICET - DCIC (UNS)

```
> Orange
   Tree  age circumference
1     1  118            30
2     1  484            58
3     1  664            87
4     1 1004           115
5     1 1231           120
6     1 1372           142
7     1 1582           145
8     2  118            33
9     2  484            69
10    2  664           111
11    2 1004           156
12    2 1231           172
13    2 1372           203
14    2 1582           203
15    3  118            30
16    3  484
17    3  664
18    3 1004
19    3 1231
20    3 1372
21    3 1582
22    4  118            32
23    4  484            62
24    4  664           112
25    4 1004           167
26    4 1231           179
27    4 1372           209
28    4 1582           214
29    5  118            30
30    5
31    5
32    5
33    5
34    5
35    5
```

```
> tapply(Orange$circumference, as.factor(Orange$age), mean)
  118   484   664  1004  1231  1372  1582
 31.0  57.8  93.2 134.2 145.6 173.4 175.8
>
```

```
> nrow(Orange[Orange$Tree==1,])
[1] 7
> nrow(Orange[Orange$Tree==1&Orange$age<500,])
[1] 2
```

Accedidos con vector booleano

```
> nrow(orange[orange$circumference>100,])
[1] 22
```

```
> dim(Orange[,])
[1] 35  3
> dim(Orange[Orange$circumference>100,])
[1] 22  3
> dim(Orange[Orange$circumference>100,])[1] #cant de registros que lo cumplen
[1] 22
```

```
> airquality
    Ozone Solar.R Wind Temp Month Day
1      41     190  7.4   67     5   1
2      36     118  8.0   72     5   2
3      12     149 12.6   74     5   3
4      18     313 11.5   62     5   4
5      NA      NA 14.3   56     5   5
6      28      NA 14.9   66     5   6
7      23     299  8.6   65     5   7
8      19      99 13.8   59     5   8
9       8      19 20.1   61     5   9
10     NA     194  8.6   69     5  10
11      7      NA  6.9   74     5  11
12     16     256  9.7   69     5  12
13     11     290  9.2   66     5  13
14     14     274 10.9   68     5  14
15     18      65 13.2   58     5  15
16     14     334 11.5   64     5  16
17     34     307 12.0   66     5  17
18      6      78 18.4   57     5  18
```

subset: retorna un sub-grupo de filas del 1er argumento, de acuerdo al criterio mostrado en el 2do argumento, **con todas las columnas salvo que se indiquen solo algunas con el 3er argumento**

```
subset(airquality, Temp > 80, select = c(Ozone, Temp))

subset(airquality, Day == 1, select = -Temp)

subset(airquality, select = Ozone:Wind)
```

```
    Ozone Solar.R Wind Temp Month Day
26     NA     266 14.9   58     5  26
27     NA      NA  8.0   57     5  27
28     23      13 12.0   67     5  28
29     45     252 14.9   81     5  29
30    115     223  5.7   79     5  30
31     37     279  7.4   76     5  31
32     NA     286  8.6   78     6   1
33     NA     287  9.7   74     6   2
34     NA     242 16.1   67     6   3
```