Practical: Data Preparation

Laura Rodriguez Navas

February 2020

Exercise 1: Inspection of data.

The "titanic.csv" file (available on the platform) contains data on the sinking of the Titanic. Copy the file in your working directory. Then, go to R and use the command:

```
titanic <- read.csv("titanic.csv",header=TRUE, sep=",")
```

Show the names of the columns.

```
head(titanic, 1)
```

```
##   X Class  Sex   Age Survived Freq
## 1 1   1st Male Child       No    0
```

Observe that the first column (whose name is "X") is redundant (it denotes the identifier of each in- stance) so it could be removed. To do this, use the subset command as follows (use help if needed):

```
titanic <- subset(titanic, select=-X)
```

Now try the following commands:

```
titanic
```

```
##    Class    Sex   Age Survived Freq
## 1    1st   Male Child       No    0
## 2    2nd   Male Child       No    0
## 3    3rd   Male Child       No   35
## 4   Crew   Male Child       No    0
## 5    1st Female Child       No    0
## 6    2nd Female Child       No    0
## 7    3rd Female Child       No   17
## 8   Crew Female Child       No    0
## 9    1st   Male Adult       No  118
## 10   2nd   Male Adult       No  154
## 11   3rd   Male Adult       No  387
## 12  Crew   Male Adult       No  670
## 13   1st Female Adult       No    4
## 14   2nd Female Adult       No   13
## 15   3rd Female Adult       No   89
## 16  Crew Female Adult       No    3
## 17   1st   Male Child      Yes    5
## 18   2nd   Male Child      Yes   11
## 19   3rd   Male Child      Yes   13
## 20  Crew   Male Child      Yes    0
## 21   1st Female Child      Yes    1
## 22   2nd Female Child      Yes   13
```

```
## 23   3rd Female Child      Yes   14
## 24  Crew Female Child      Yes    0
## 25   1st   Male Adult      Yes   57
## 26   2nd   Male Adult      Yes   14
## 27   3rd   Male Adult      Yes   75
## 28  Crew   Male Adult      Yes  192
## 29   1st Female Adult      Yes  140
## 30   2nd Female Adult      Yes   80
## 31   3rd Female Adult      Yes   76
## 32  Crew Female Adult      Yes   20
```

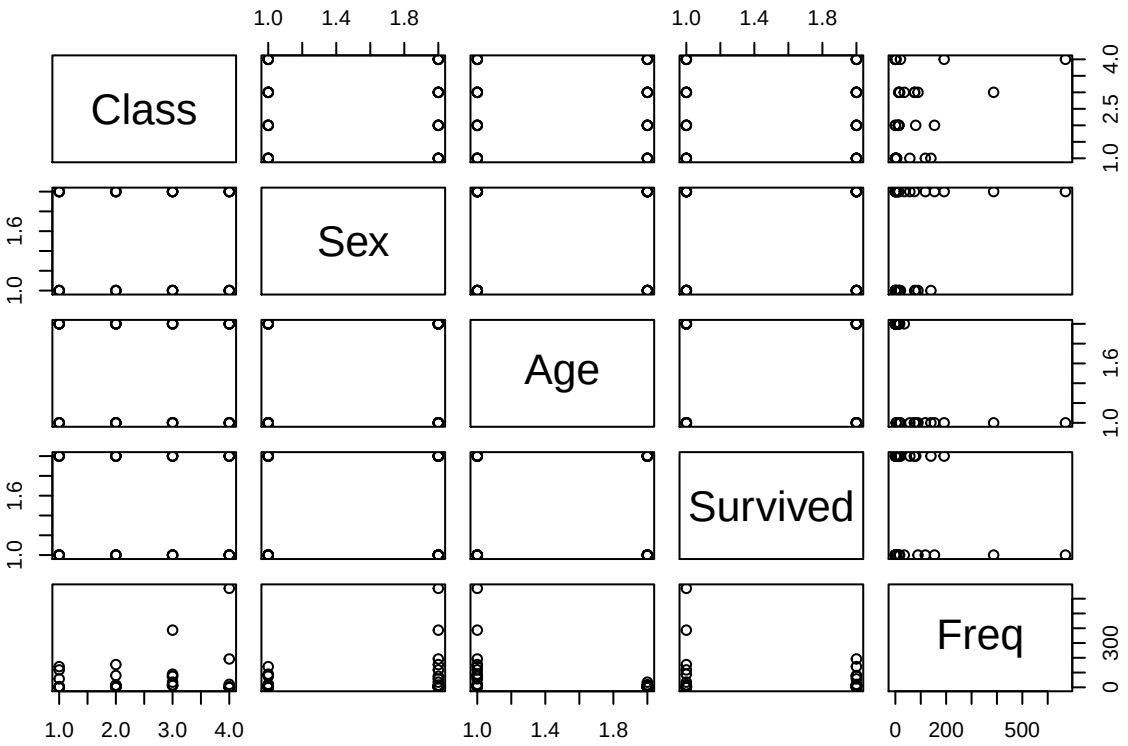
```
head(titanic)
```

```
##   Class    Sex   Age Survived Freq
## 1   1st   Male Child       No    0
## 2   2nd   Male Child       No    0
## 3   3rd   Male Child       No   35
## 4  Crew   Male Child       No    0
## 5   1st Female Child       No    0
## 6   2nd Female Child       No    0
```

```
summary(titanic)
```

```
##   Class       Sex         Age      Survived      Freq
##  1st :8   Female:16   Adult:16   No :16   Min.   :  0.00
##  2nd :8   Male  :16   Child:16   Yes:16   1st Qu.:  0.75
##  3rd :8                                   Median : 13.50
##  Crew:8                                   Mean   : 68.78
##                                           3rd Qu.: 77.00
##                                           Max.   :670.00
```

```
plot(titanic)
```



Which variables are quantitative and which variables are categorical? How can we know it?

TODO

Exercise 2: Working with basic graphics.

Download the file "cars.csv" from the platform. This file contains information about the speed and stopping distances of cars.

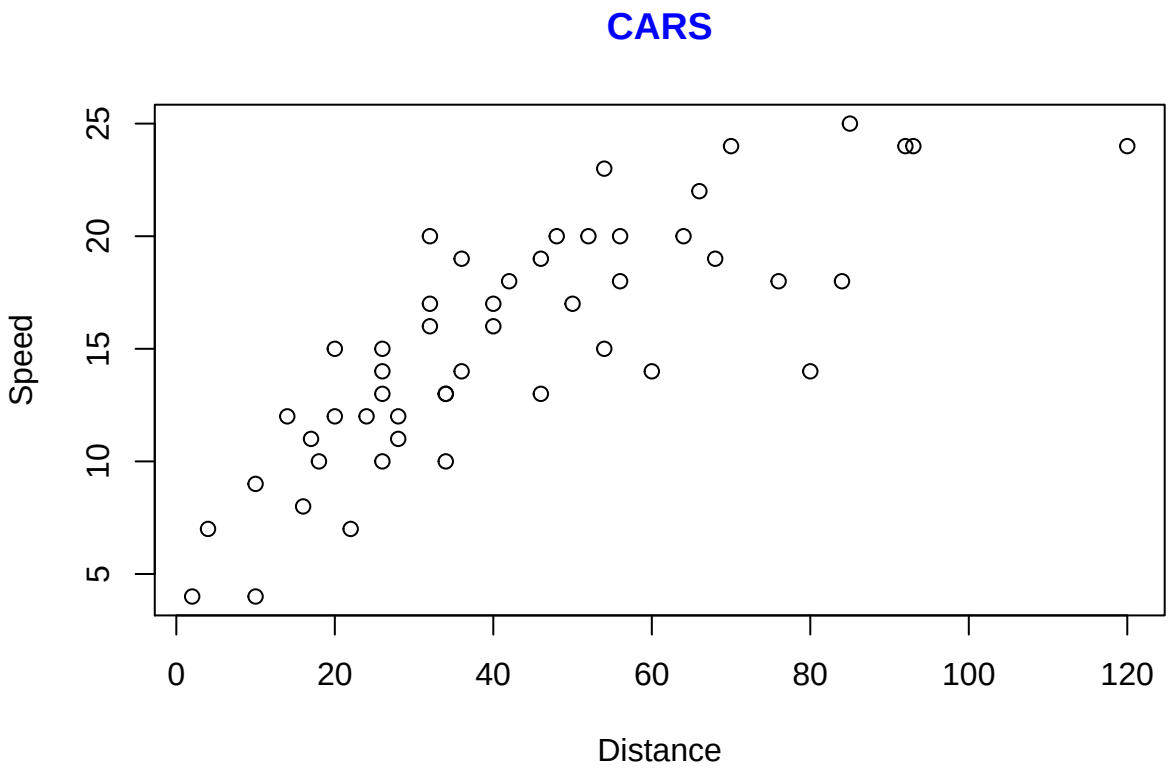
```
cars <- read.csv("cars.csv", header=TRUE, sep=",")
cars <- subset(cars, select=-X)
cars
```

```
##    speed dist
## 1      4    2
## 2      4   10
## 3      7    4
## 4      7   22
## 5      8   16
## 6      9   10
## 7     10   18
## 8     10   26
## 9     10   34
## 10    11   17
## 11    11   28
## 12    12   14
## 13    12   20
## 14    12   24
## 15    12   28
## 16    13   26
## 17    13   34
## 18    13   34
## 19    13   46
## 20    14   26
## 21    14   36
## 22    14   60
## 23    14   80
## 24    15   20
## 25    15   26
## 26    15   54
## 27    16   32
## 28    16   40
## 29    17   32
## 30    17   40
## 31    17   50
## 32    18   42
## 33    18   56
## 34    18   76
## 35    18   84
## 36    19   36
## 37    19   46
## 38    19   68
## 39    20   32
## 40    20   48
## 41    20   52
```

```
## 42    20   56
## 43    20   64
## 44    22   66
## 45    23   54
## 46    24   70
## 47    24   92
## 48    24   93
## 49    24  120
## 50    25   85
```

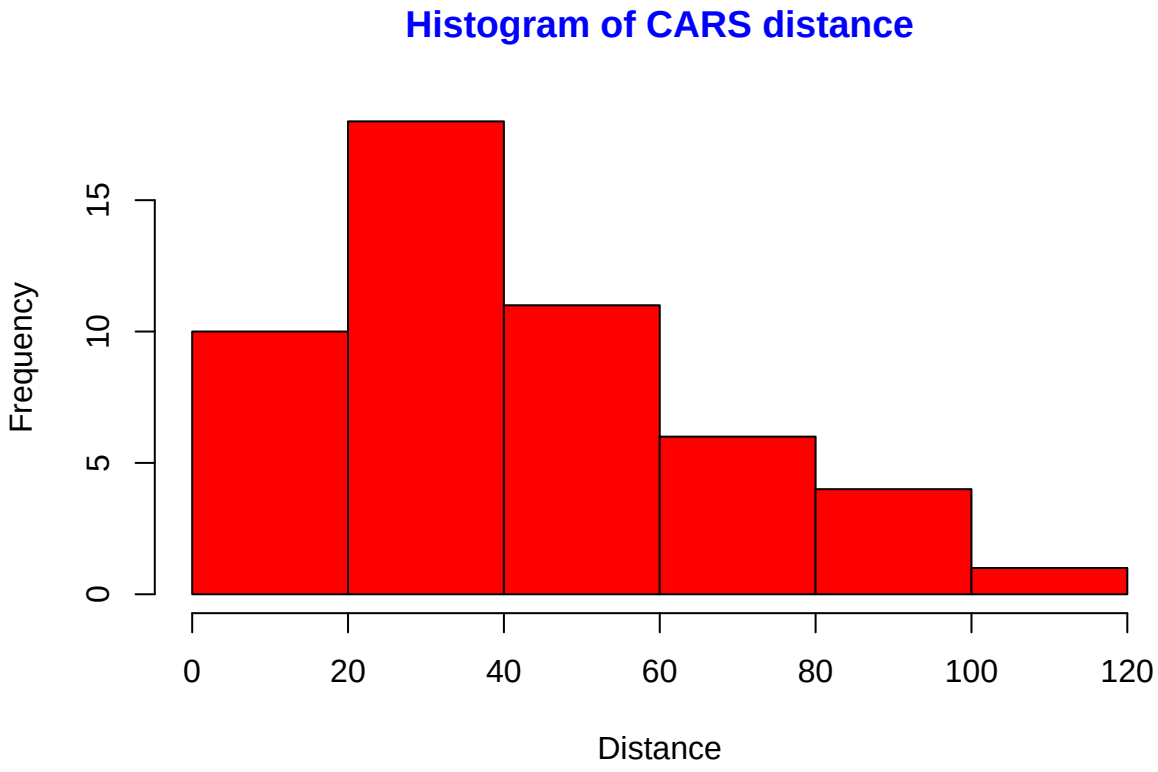
2.1 Make a plot of the distance field in terms of the speed field (use the $ syntax).

```
plot(cars$dist, cars$speed, main="CARS", col.main = "blue", xlab="Distance", ylab="Speed")
```



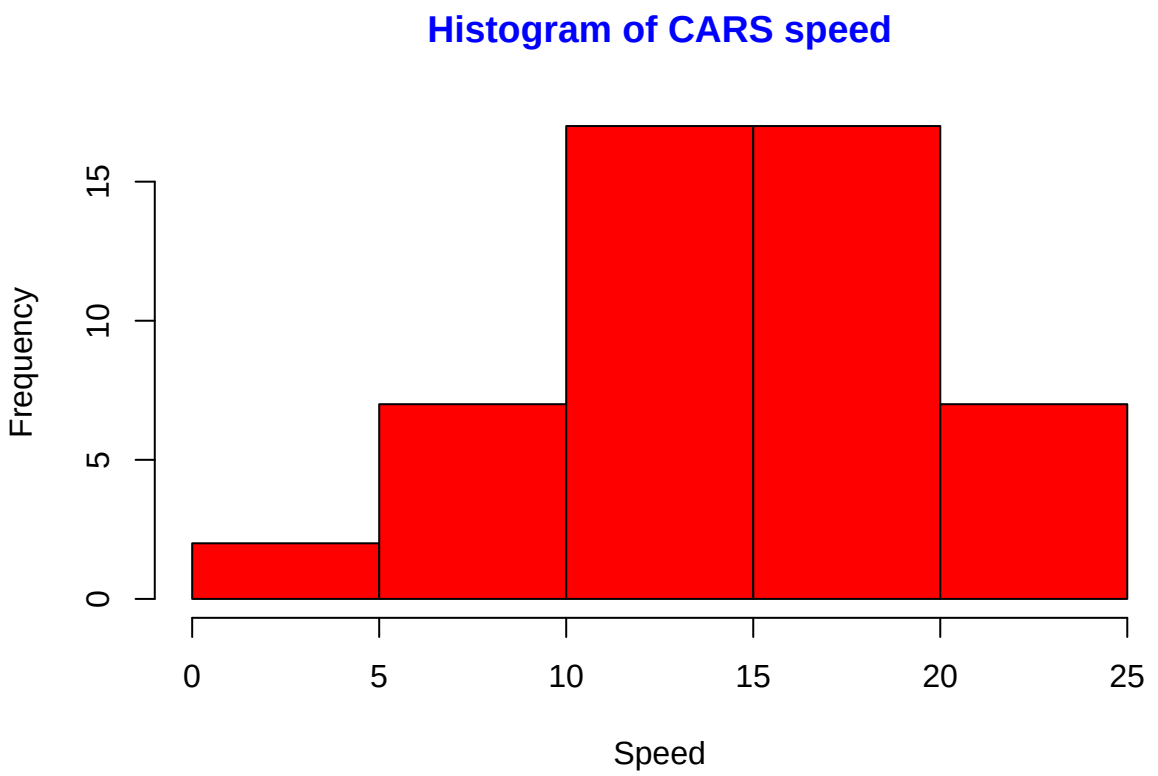
2.2 Make a histogram of the distance variable.

```
hist(cars$dist, main="Histogram of CARS distance", col.main = "blue", xlab="Distance", col="red")
```



2.3 Make a histogram of the speed variable.

```
hist(cars$speed, main="Histogram of CARS speed", col.main = "blue", xlab="Speed", col="red")
```



The modifications of section 2.4 have been made in the previous sections.

Exercise 3: Transformations of variables and datasets.

Now, assume that data from two more cars are made available:

TODO

3.1 Construct a new data frame with the above data.

```
new_cars <- data.frame(speed=c(21, 34), dist=c(47, 87))
new_cars
```

```
##   speed dist
## 1    21   47
## 2    34   87
```

3.2 Add the constructed data frame to the cars data frame.

```
tail(cars)
```

```
##    speed dist
## 45    23   54
## 46    24   70
## 47    24   92
## 48    24   93
## 49    24  120
## 50    25   85
```

```
cars <- rbind(cars, new_cars)
tail(cars)
```

```
##    speed dist
## 47    24   92
## 48    24   93
## 49    24  120
## 50    25   85
## 51    21   47
## 52    34   87
```

3.3 Sort the data in the resulting dataset by column speed (ascending).

```
cars <- cars[order(cars$speed), ]
cars
```

```
##    speed dist
## 1      4    2
## 2      4   10
## 3      7    4
## 4      7   22
## 5      8   16
## 6      9   10
## 7     10   18
## 8     10   26
## 9     10   34
## 10    11   17
## 11    11   28
## 12    12   14
## 13    12   20
## 14    12   24
```

```
## 15    12   28
## 16    13   26
## 17    13   34
## 18    13   34
## 19    13   46
## 20    14   26
## 21    14   36
## 22    14   60
## 23    14   80
## 24    15   20
## 25    15   26
## 26    15   54
## 27    16   32
## 28    16   40
## 29    17   32
## 30    17   40
## 31    17   50
## 32    18   42
## 33    18   56
## 34    18   76
## 35    18   84
## 36    19   36
## 37    19   46
## 38    19   68
## 39    20   32
## 40    20   48
## 41    20   52
## 42    20   56
## 43    20   64
## 51    21   47
## 44    22   66
## 45    23   54
## 46    24   70
## 47    24   92
## 48    24   93
## 49    24  120
## 50    25   85
## 52    34   87
```

Exercise 4: Data manipulation.

Download the file "airquality.csv" from the platform. This dataset contains some New York air quality measurements.

```
airquality <- read.csv("airquality.csv",header=TRUE, sep=",")
```

4.1 Extract the first 2 rows of the data frame and print them to the console. What does the output look like?

```
head(airquality, 2)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
```

4.2 How many observations (i.e. rows) are in this data frame?

```r
nrow(airquality)
```

```
## [1] 153
```

4.3 What is the value of Ozone in the 40th row?

```r
airquality[40,1]
```

```
## [1] 71
```

4.4 How many missing values are in the Ozone column of this data frame?

```r
sum(is.na(airquality$Ozone))
```

```
## [1] 37
```

4.5 What is the mean of the Ozone column in this dataset? Exclude missing values (coded as NA) from this calculation.

```r
airquality_tmp <- airquality[complete.cases(airquality$Ozone), ]
sum(is.na(airquality_tmp$Ozone))
```

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

```r
nrow(airquality_tmp)
```

```
## [1] 116
```

```r
mean(airquality_tmp$Ozone)
```

```
## [1] 42.12931
```

4.6 Extract the subset of rows of the data frame where Ozone values are above 31 and Temp values are above 90. What is the mean of Solar.R in this subset?

```r
solarSubset <- subset(airquality_tmp , airquality_tmp$Ozone > 31 | airquality_tmp$Temp > 90)
solarSubset_tmp <- solarSubset[complete.cases(solarSubset$Solar.R), ]
mean(solarSubset_tmp$Solar.R)
```

```
## [1] 215.0545
```

Exercise 5: Data transformation (2).

5.1 Discretise the Ozone column into five bins ('bin1', 'bin2', ...) of equal width and a sixth bin ('binNA') for NA. 5.2 Discretise the Solar column into four bins of equal size and a fifth bin for NA. 5.3 Create a new column AbsDay from the columns Month and Day such that counts the number of days passed from Month=5 and Day=1.

Exercise 6: Data transformation (3).

Exercise 7: Data selection.

7.1 Calculate a correlation matrix for the air dataset. Do you see a pair of attributes that are redundant?

```r
cor(airquality)
```

```
##         Ozone Solar.R       Wind       Temp        Month          Day
## Ozone       1      NA         NA         NA           NA           NA
## Solar.R    NA       1         NA         NA           NA           NA
## Wind       NA      NA  1.0000000 -0.4579879 -0.178292579  0.027180903
## Temp       NA      NA -0.4579879  1.0000000  0.420947252 -0.130593175
```

```
## Month      NA      NA -0.1782926  0.4209473  1.000000000 -0.007961763
## Day        NA      NA  0.0271809 -0.1305932 -0.007961763  1.000000000
```

7.2 Calculate a correlation matrix for the cars dataset. Do you see a pair of attributes that are redundant?

```
cor(cars)
```

```
##           speed      dist
## speed 1.0000000 0.8025411
## dist  0.8025411 1.0000000
```

7.3 Using the data frame airquality, perform a simple random sampling of 50 examples.

```
sample_n(airquality, 50)
```

```
##    Ozone Solar.R Wind Temp Month Day
## 1     NA      98 11.5   80     6  28
## 2     28     273 11.5   82     8  13
## 3     NA     135  8.0   75     6  25
## 4     31     244 10.9   78     8  19
## 5     NA      47 10.3   73     6  27
## 6     21     230 10.9   75     9   9
## 7     18     313 11.5   62     5   4
## 8    122     255  4.0   89     8   7
## 9      7      49 10.3   69     9  24
## 10    44     192 11.5   86     8  12
## 11    NA     242 16.1   67     6   3
## 12    45     252 14.9   81     5  29
## 13    13     112 11.5   71     9  15
## 14    34     307 12.0   66     5  17
## 15    NA     138  8.0   83     6  30
## 16    18     224 13.8   67     9  17
## 17    29     127  9.7   82     6   7
## 18    61     285  6.3   84     7  18
## 19    20     252 10.9   80     9   7
## 20    78     197  5.1   92     9   2
## 21    47      95  7.4   87     9   5
## 22    97     272  5.7   92     7   9
## 23    27     175 14.9   81     7  13
## 24     9      24 10.9   71     9  14
## 25    16     256  9.7   69     5  12
## 26    21     191 14.9   77     6  16
## 27    14     334 11.5   64     5  16
## 28    30     193  6.9   70     9  26
## 29    85     188  6.3   94     8  31
## 30     7      NA  6.9   74     5  11
## 31    49     248  9.2   85     7   2
## 32    32     236  9.2   81     7   3
## 33    NA      59  1.7   76     6  22
## 34   110     207  8.0   90     8   9
## 35    36     118  8.0   72     5   2
## 36    79     187  5.1   87     7  19
## 37    35     274 10.3   82     7  17
## 38    21     259 15.5   76     9  12
## 39   118     225  2.3   94     8  29
## 40    96     167  6.9   91     9   1
```

```
## 41    NA     259 10.9   93     6  11
## 42    NA     258  9.7   81     7  22
## 43    14     191 14.3   75     9  28
## 44    18     131  8.0   76     9  29
## 45    10     264 14.3   73     7  12
## 46    66      NA  4.6   87     8   6
## 47    NA     222  8.6   92     8  10
## 48    64     253  7.4   83     7  30
## 49     9      36 14.3   72     8  22
## 50    78      NA  6.9   86     8   4
```

7.4 Using the data frame airquality, perform a stratified random sampling of 5 examples of each month.

```
set.seed(1)
airquality %>%
    group_by (Month) %>%
    sample_n(., 5)
```

```
## # A tibble: 25 x 6
## # Groups:   Month [5]
##    Ozone Solar.R  Wind  Temp Month   Day
##    <int>   <int> <dbl> <int> <int> <int>
##  1    NA      66  16.6    57     5    25
##  2    18     313  11.5    62     5     4
##  3    23     299   8.6    65     5     7
##  4    41     190   7.4    67     5     1
##  5    36     118   8      72     5     2
##  6    NA      31  14.9    77     6    29
##  7    NA      91   4.6    76     6    23
##  8    NA     259  10.9    93     6    11
##  9    NA     332  13.8    80     6    14
## 10    20      37   9.2    65     6    18
## 11    52      82  12      86     7    27
## 12    79     187   5.1    87     7    19
## 13   135     269   4.1    84     7     1
## 14    16       7   6.9    74     7    21
## 15    82     213   7.4    88     7    28
## 16    NA     222   8.6    92     8    10
## 17     9      36  14.3    72     8    22
## 18    65     157   9.7    80     8    14
## 19    85     188   6.3    94     8    31
## 20   122     255   4      89     8     7
## 21    21     230  10.9    75     9     9
## 22    13     112  11.5    71     9    15
## 23    13     238  12.6    64     9    21
## 24    47      95   7.4    87     9     5
## 25    20     223  11.5    68     9    30
```