week1_R_data_prep_demo

Nate George

January 9, 2018

We are going to examine the auto mpg dataset from here:

as an example for how to prepare data.

First we'll load the libraries we will need (message=F will hide messages from the package loading):

```
library(data.table)
library(DMwR)
library(corrplot)
```

Next we'll load the data and check it out:

```
fn <- '/home/nate/Dropbox/MSDS/MSDS680_ncg_S8W1_18/week1/auto-mpg.data'

# as.is leaves characters as characters instead of converting to factors, so we can substitute NA for ?
df <- read.table(fn, as.is = T)
auto.dt <- as.data.table(df)
str(auto.dt)
```

```
## Classes 'data.table' and 'data.frame':   398 obs. of  9 variables:
##  $ V1: num  18 15 18 16 17 15 14 14 14 15 ...
##  $ V2: int  8 8 8 8 8 8 8 8 8 8 ...
##  $ V3: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ V4: chr  "130.0" "165.0" "150.0" "150.0" ...
##  $ V5: num  3504 3693 3436 3433 3449 ...
##  $ V6: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ V7: int  70 70 70 70 70 70 70 70 70 70 ...
##  $ V8: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ V9: chr  "chevrolet chevelle malibu" "buick skylark 320" "plymouth satellite" "amc rebel sst" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

V4 is the horsepower column, and is a string because there are some missing values represented as '?'. First we're going to replace the V-names with more accurate and easy-to-use names:

```
fn <- '/home/nate/Dropbox/MSDS/MSDS680_ncg_S8W1_18/week1/auto-mpg.names'

auto.names <- readLines(fn)
auto.names
```

```
## [1] "    1. mpg:           continuous"
## [2] "    2. cylinders:     multi-valued discrete"
## [3] "    3. displacement:  continuous"
## [4] "    4. horsepower:    continuous"
## [5] "    5. weight:        continuous"
## [6] "    6. acceleration:  continuous"
## [7] "    7. model year:    multi-valued discrete"
## [8] "    8. origin:        multi-valued discrete"
## [9] "    9. car name:      string (unique for each instance)"
```

```
# gets the name from the list of names using regular expressions
get_name <- function(x) gsub('\\s+\\d+\\.\\s+(.+):\\s+.+', '\\1', x)
```

```
short.names <- unlist(lapply(auto.names, FUN = get_name))
short.names <- unlist(lapply(short.names, FUN = function(x) gsub('\\s', '.', x)))
short.names
```

```
## [1] "mpg"          "cylinders"    "displacement" "horsepower"
## [5] "weight"       "acceleration" "model.year"   "origin"
## [9] "car.name"
```

```
names(auto.dt) <- short.names
```

Now we want to replace missing values with NA. We can see horsepower is the only colunm with missing values, and there are 6 of them.

```
auto.dt[auto.dt == '?'] <- NA
str(auto.dt)
```

```
## Classes 'data.table' and 'data.frame':   398 obs. of  9 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : int  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : chr  "130.0" "165.0" "150.0" "150.0" ...
##  $ weight      : num  3504 3693 3436 3433 3449 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ model.year  : int  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ car.name    : chr  "chevrolet chevelle malibu" "buick skylark 320" "plymouth satellite" "amc rebel
##  - attr(*, ".internal.selfref")=<externalptr>
```

There are a ton of car names, so we will drop that column, and we need to convert the horespower to numeric.

```
auto.dt[, horsepower:=as.numeric(horsepower)]
auto.dt[, car.name:=NULL]
str(auto.dt)
```

```
## Classes 'data.table' and 'data.frame':   398 obs. of  8 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : int  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : num  3504 3693 3436 3433 3449 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ model.year  : int  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : int  1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

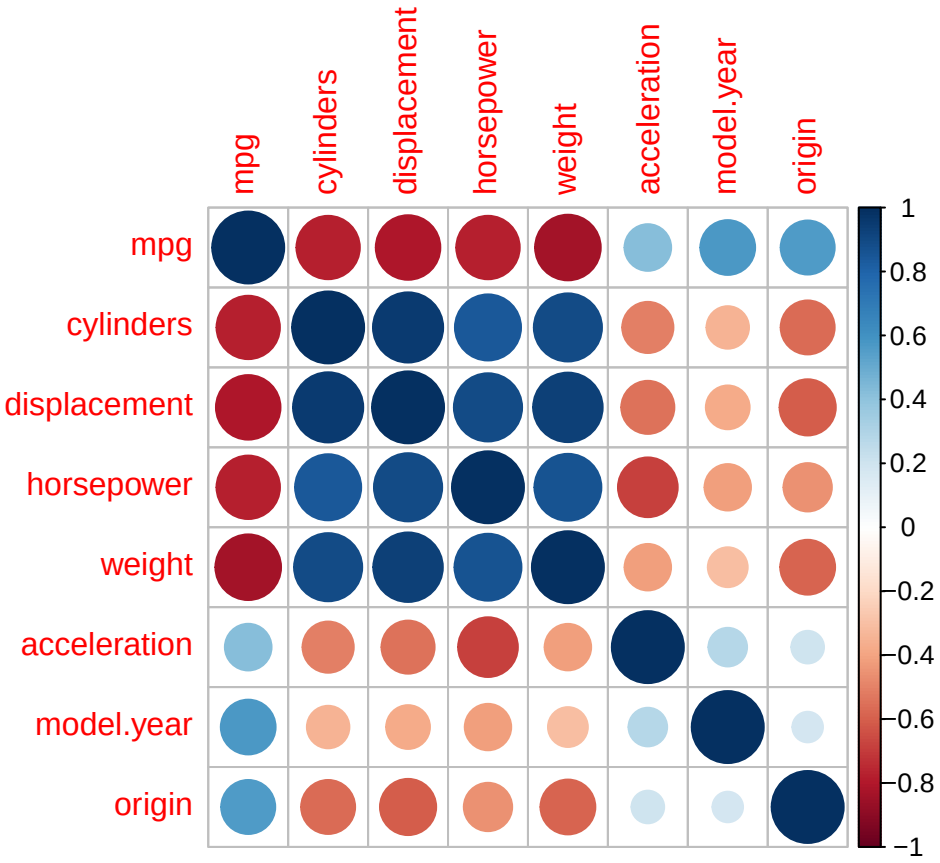
Now we will replace missing values by imputing them with K-nearest neighbors.

```
auto.dt.nona <- knnImputation(auto.dt)
# if you're curious, see what the nas were replaced with
auto.dt.nona[is.na(auto.dt$horsepower)]$horsepower
```

```
## [1] 77.33912 94.68976 69.41971 89.74111 73.99194 85.42545
```

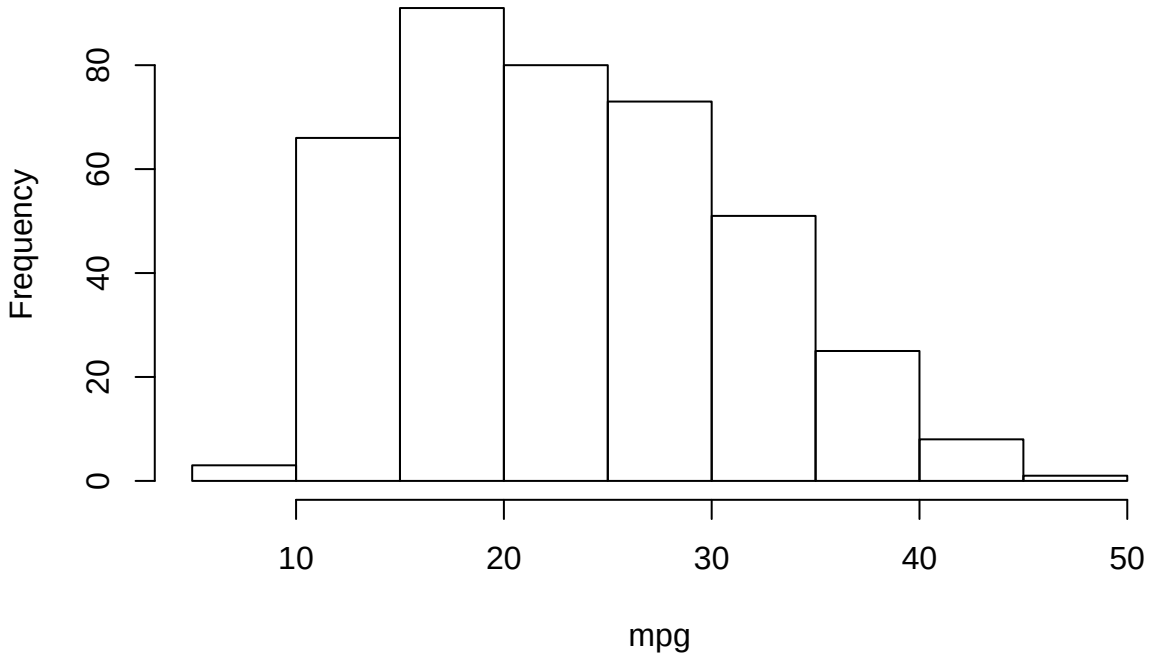
Finally let's do some EDA on the data. First a correlation plot, which shows all the variables to be highly correlated to mpg.

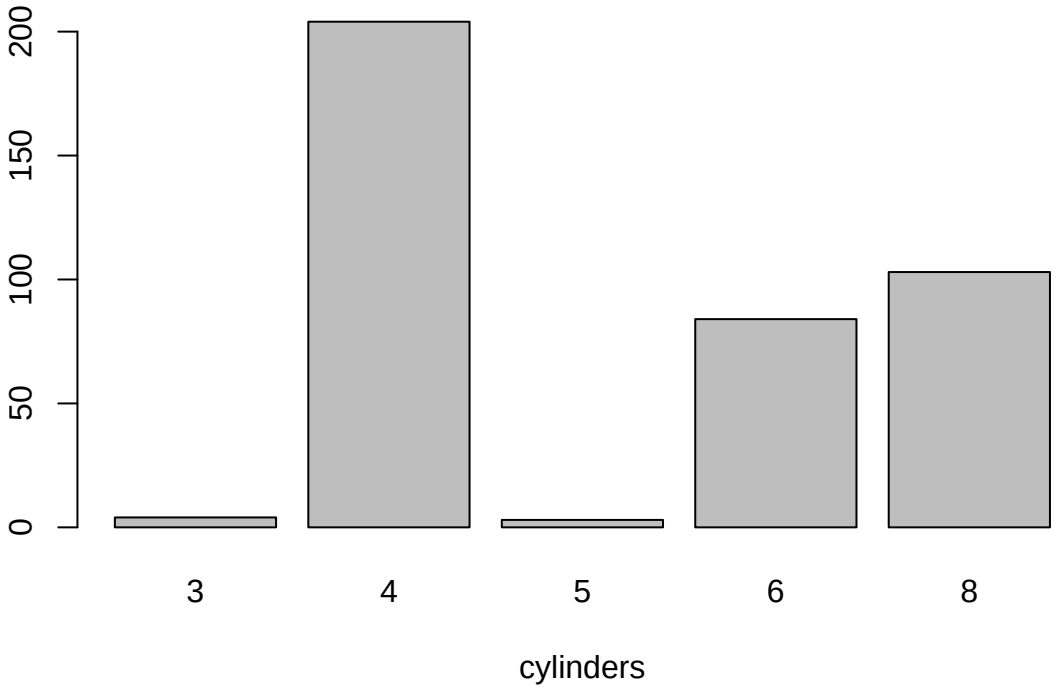
```
corrplot(cor(auto.dt.nona))
```

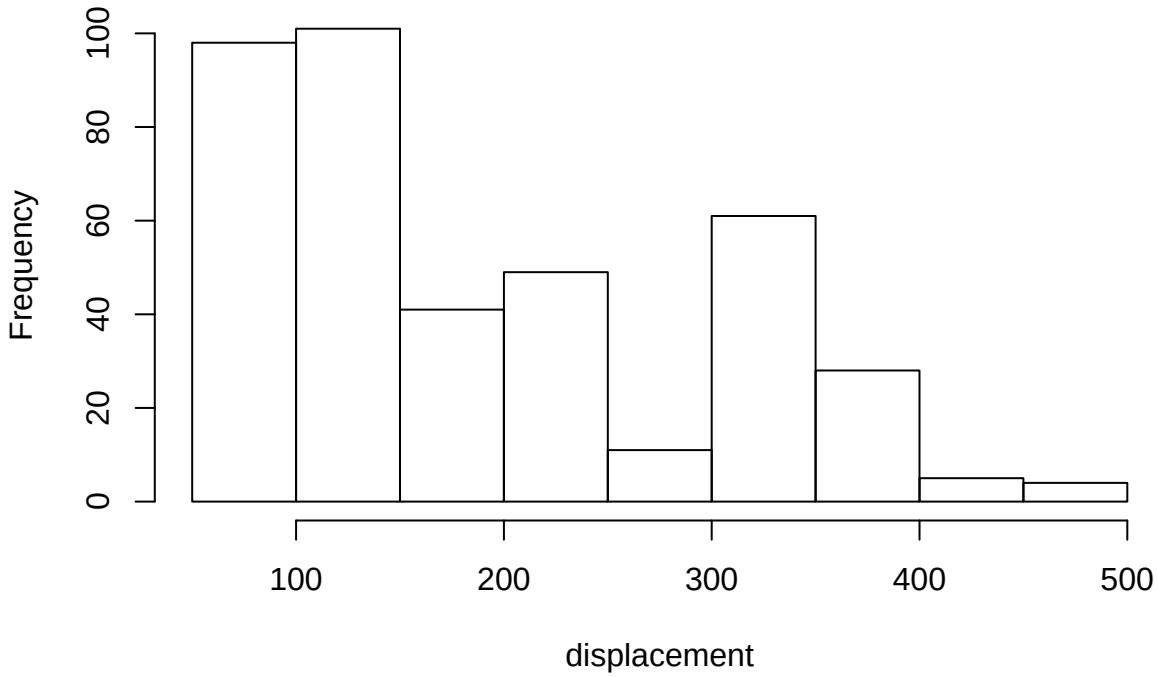


There are not many 3- and 5-cyliner engines in the dataset, so we may want to throw those out.

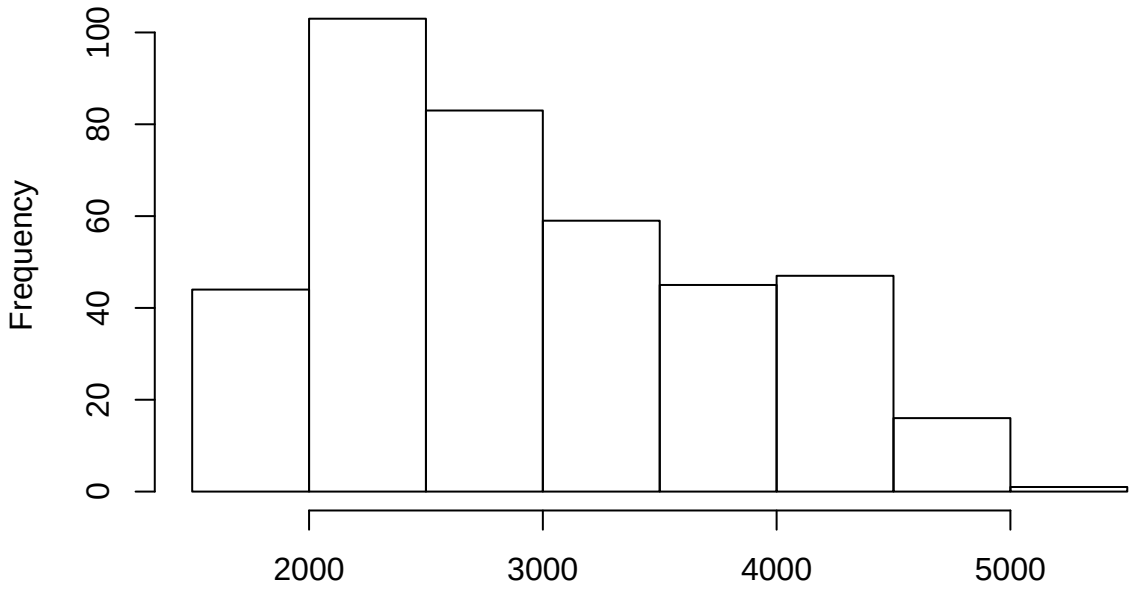
```
dt.names <- names(auto.dt.nona)
for (i in seq(dim(auto.dt)[2])) {
  coldata <- auto.dt.nona[, get(dt.names[i])]
  n.levels <- nlevels(as.factor(coldata))
  if (n.levels <= 10) {
    barplot(table(coldata), xlab = dt.names[i])
  } else {
    hist(coldata, main = NULL, xlab = dt.names[i])
  }
  # add newlines so the plots all show up
  cat('\n\n')
}
```

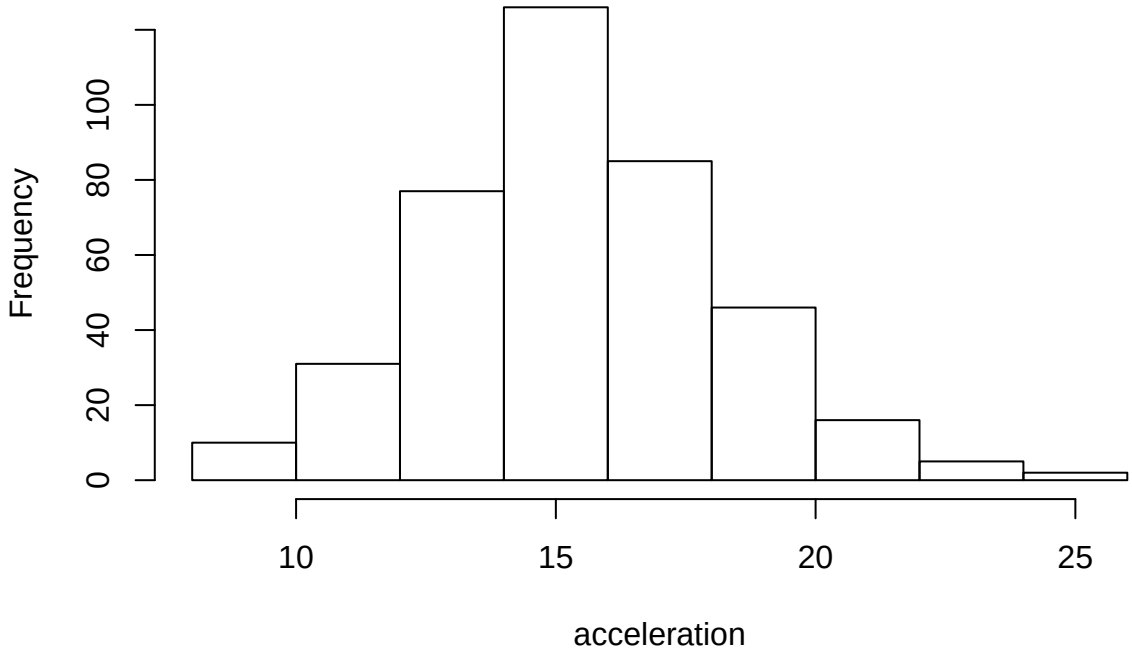

Frequency
0
20
40
60
80
10
20
30
40
50
mpg

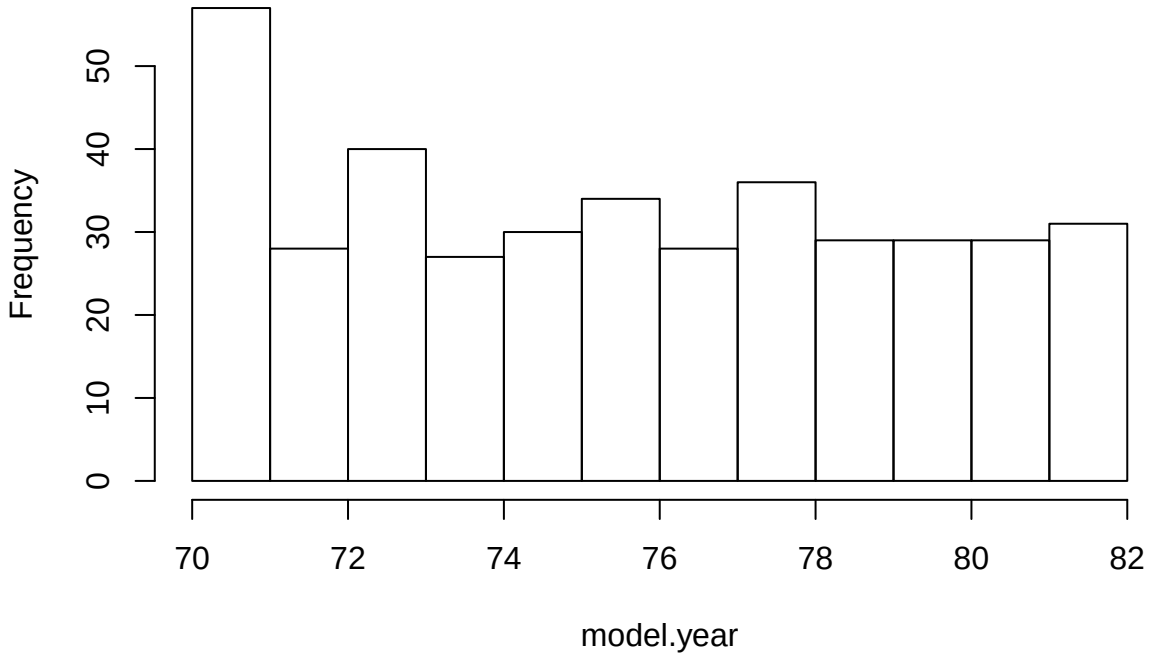

0
50
100
150
200
3
4
5
6
8
cylinders

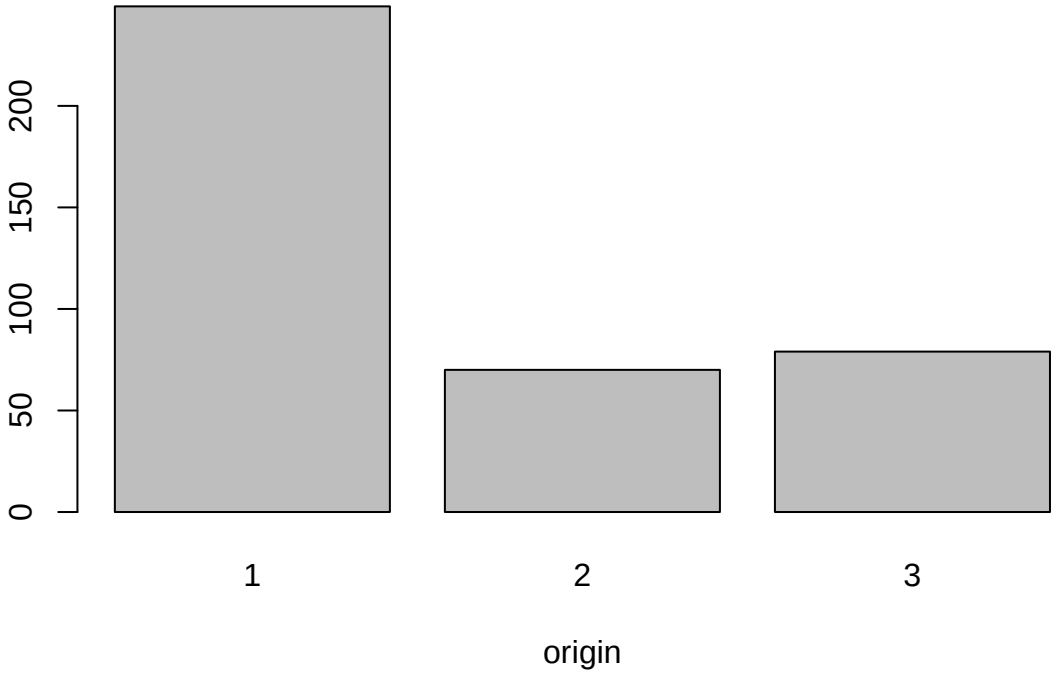

Frequency
0
20
40
60
80
100
100
200
300
400
500
displacement


Frequency
0
20
40
60
80
100
50
100
150
200
horsepower


Frequency
0
20
40
60
80
100
2000
3000
4000
5000
weight


Frequency
0
20
40
60
80
100
10
15
20
25
acceleration


Frequency
0
10
20
30
40
50
70
72
74
76
78
80
82
model.year


0
50
100
150
200
1
2
3
origin