Exploring Data

Chirantan Ganguly

26/04/2020

Checking the dimensions of your data

The first value returned by dim() is the number of cases (rows) and the second value is the number of variables (columns).

```
dim(mtcars)
```

```
## [1] 32 11
```

Data Structures

Using the str() function we can look at the structure of a dataset. str() takes the name of the data set as its first argument. The output shows the variable names, their type, and the values of the first observations.

```
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

Recoding Variables

Change mpg(miles per gallon) from a continuous variable to a categorical variable indicating “high” if mpg>=20 otherwise “low” and add this as another variable to mtcars

```
mpgcategory<-mtcars$mpg
mpgcategory[mpgcategory>=20]<-"high"
mpgcategory[mpgcategory<20]<-"low"
mtcars$mpgfactor<-as.factor(mpgcategory)
mtcars
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
##                     mpgfactor
## Mazda RX4                high
## Mazda RX4 Wag            high
## Datsun 710               high
## Hornet 4 Drive           high
## Hornet Sportabout         low
## Valiant                   low
## Duster 360                low
## Merc 240D                high
## Merc 230                 high
## Merc 280                  low
## Merc 280C                 low
## Merc 450SE                low
## Merc 450SL                low
## Merc 450SLC               low
## Cadillac Fleetwood        low
## Lincoln Continental       low
## Chrysler Imperial         low
## Fiat 128                 high
## Honda Civic              high
## Toyota Corolla           high
## Toyota Corona            high
## Dodge Challenger          low
## AMC Javelin               low
## Camaro Z28                low
## Pontiac Firebird          low
## Fiat X1-9                high
## Porsche 914-2            high
## Lotus Europa             high
## Ford Pantera L            low
## Ferrari Dino              low
## Maserati Bora             low
## Volvo 142E               high
```

Examining Frequencies

Let us now find the frequencies of number of cars with auto transmission and manual transmission

```
table<-table(mtcars$am)
table
```

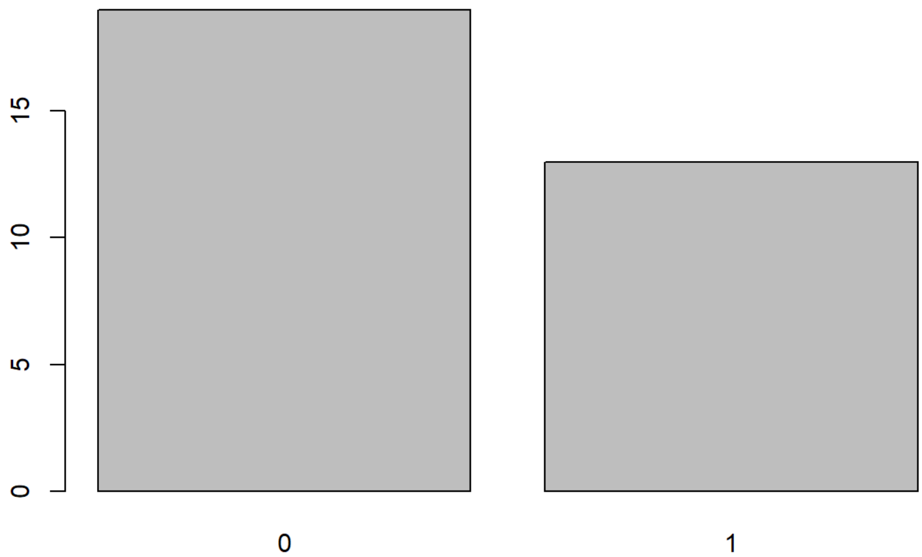
```
##
##  0  1
## 19 13
```

We get the no of O’s and the number of 1’s

Making Bar Graphs

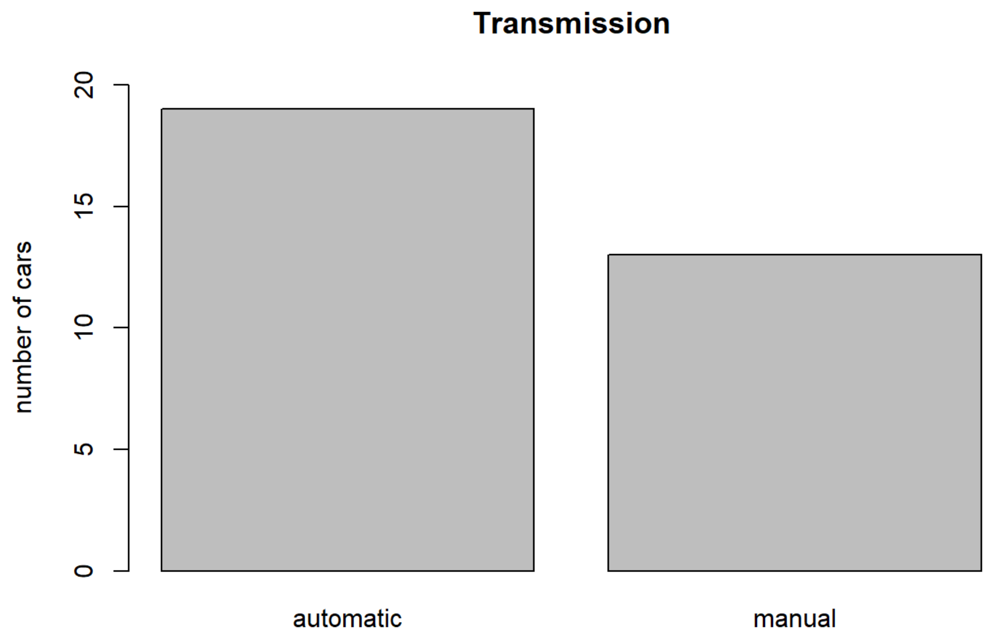
We easily can make graphs to visualize our data. Let’s visualize the number of manual and automatic transmissions in our car sample through a bar graph, using the function barplot().

```
barplot(table)
```



Labelling a Bar Graph

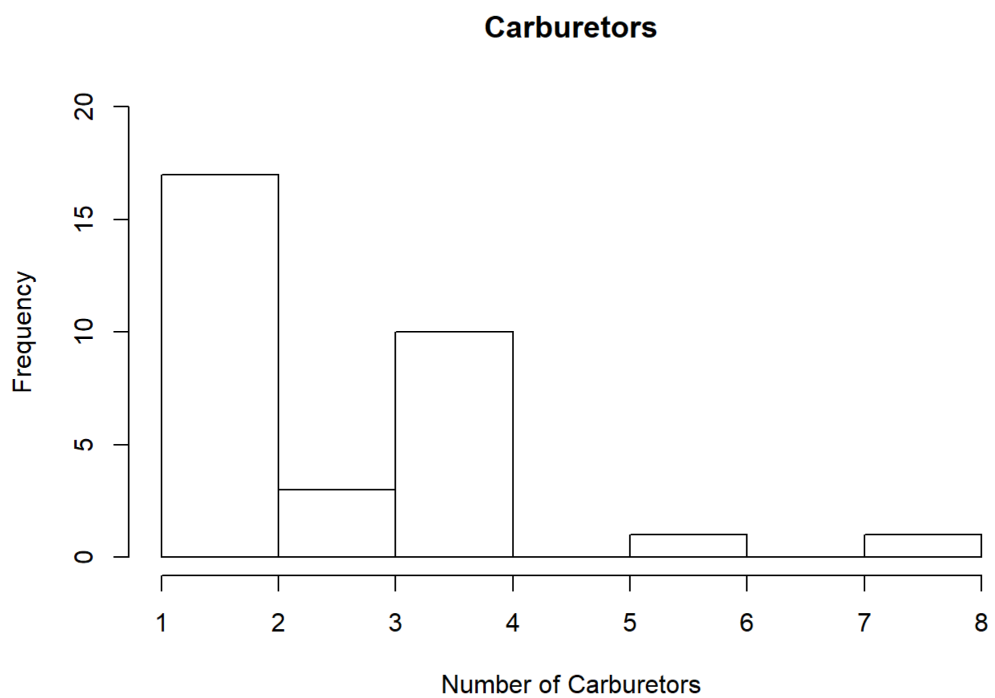
```
barnames <- c("automatic", "manual")
barplot(table, ylab = "number of cars", names.arg = barnames, main="Transmission", ylim=c(0,20))
```



Histograms

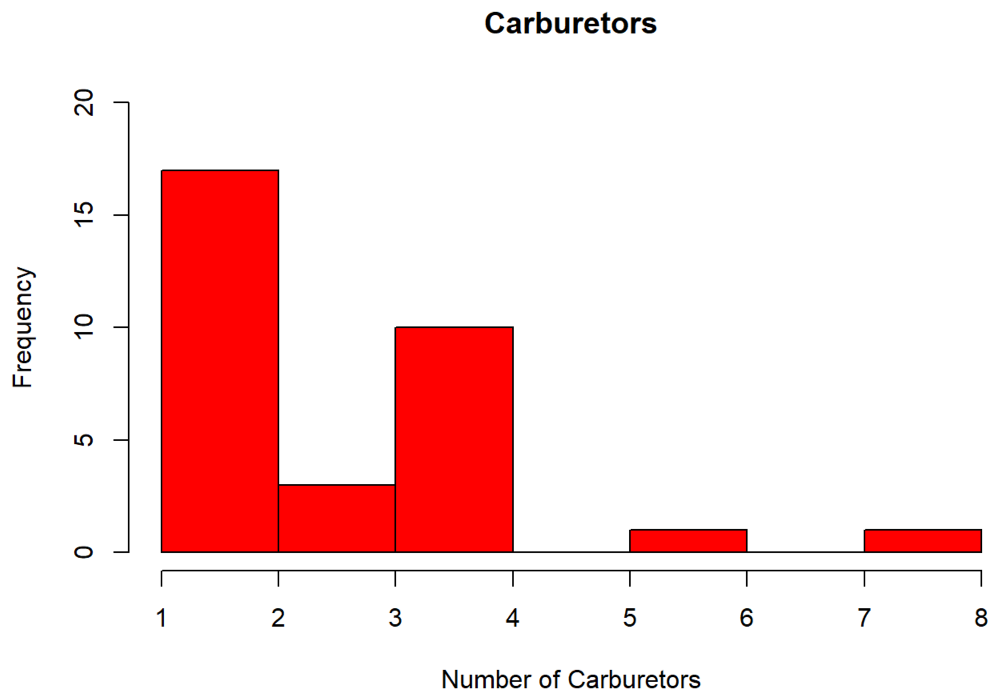
It can be useful to plot frequencies as histograms to visualize the spread of our data. Lets plot a Histogram of the Number of Carburetors: -

```
hist(mtcars$carb, main="Carburetors", xlab="Number of Carburetors", ylim = c(0,20))
```



Formating your histogram

```
hist(mtcars$carb, main = "Carburetors", ylim=c(0,20), col="red", xlab="Number of Carburetors")
```



Mean and Median

Calculating the mean and median are very important for understanding the central tendancy of a given data. Lets measure the mean() and the median() of mpg

```
mean(mtcars$mpg)
```

```
## [1] 20.09062
```

```
median(mtcars$mpg)
```

```
## [1] 19.2
```

Mode

Lets use sort() and table() to find the mode of the carb variable of mtcars.

```
sort(table(mtcars$carb), decreasing = T)
```

```
##
##  2  4  1  3  6  8
## 10 10  7  3  1  1
```

Range

The range of various values in a dataset is particularly important because it tells us about the dispersion of the data we use max() and min() to find the range: -

```
x <-min(mtcars$mpg)
y <-max(mtcars$mpg)
y-x
```

```
## [1] 23.5
```

Quartiles

We can measure the quartiles in our dataset using the function quantile: -

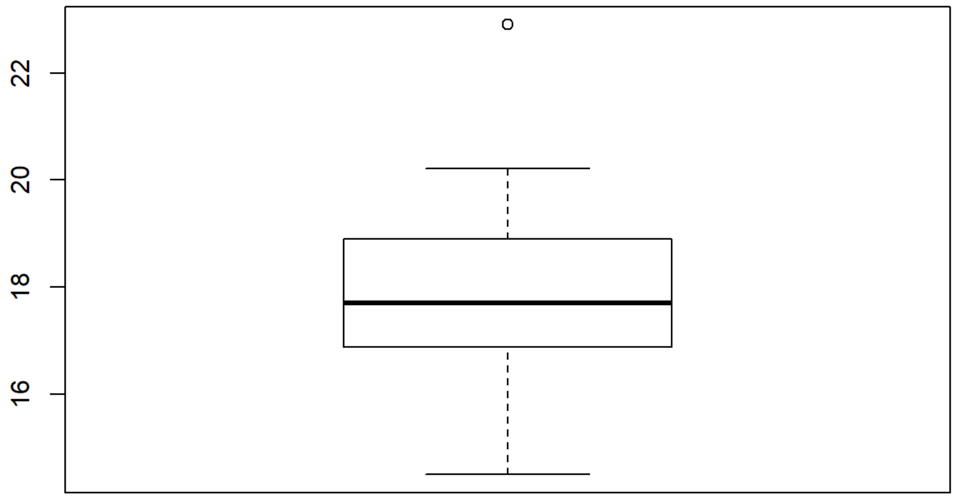
```
quantile(mtcars$mpg)
```

```
##     0%    25%    50%    75%   100% 
## 10.400 15.425 19.200 22.800 33.900
```

IQR and Boxplot

To better visualise your data’s quartiles you can create a boxplot using the function boxplot(). Similarly, you can calculate the interquartile range manually by subtracting the value of the third quartile from the value of the first quartile, or we can use the function IQR()

```
boxplot(mtcars$qsec)
```




```
IQR(mtcars$qsec)
```

```
## [1] 2.0075
```

IQR outliers

We can find outliers by first finding the first and the third quartile using the function quantile() and there after we can calculate the threshold value for outliers below the first Quartile and above the second Quartile as follow: -

```
quantile(mtcars$qsec)
```

```
##      0%     25%     50%     75%    100% 
## 14.5000 16.8925 17.7100 18.9000 22.9000
```

```
#The threshold value of outliers below the first quartile is
16.8925-1.5*IQR(mtcars$qsec)
```

```
## [1] 13.88125
```

```
#The threshold value of outliers above the third quartile is
18.9+1.5*IQR(mtcars$qsec)
```

```
## [1] 21.91125
```

Standard Deviation

We can also measure the spread of data through the standard deviation. We can calculate these using the function sd()

```
IQR(mtcars$hp)
```

```
## [1] 83.5
```

```
sd(mtcars$hp)
```

```
## [1] 68.56287
```

```
IQR(mtcars$mpg)
```

```
## [1] 7.375
```

```
sd(mtcars$mpg)
```

```
## [1] 6.026948
```

Calculating Z Scores

We can calculate the z-score for a given value (X) as (X - mean) / standard deviation.

```
(mtcars$mpg-mean(mtcars$mpg))/sd(mtcars$mpg)
```

```
##  [1]  0.15088482  0.15088482  0.44954345  0.21725341 -0.23073453 -0.33028740
##  [7] -0.96078893  0.71501778  0.44954345 -0.14777380 -0.38006384 -0.61235388
## [13] -0.46302456 -0.81145962 -1.60788262 -1.60788262 -0.89442035  2.04238943
## [19]  1.71054652  2.29127162  0.23384555 -0.76168319 -0.81145962 -1.12671039
## [25] -0.14777380  1.19619000  0.98049211  1.71054652 -0.71190675 -0.06481307
## [31] -0.84464392  0.21725341
```