Plopper

Alexandra Vajda

22/05/2019

mydata <- read.csv(file = "3.1.txt", sep = ",", header = TRUE)  
dim(mydata)

## [1] 33988 5

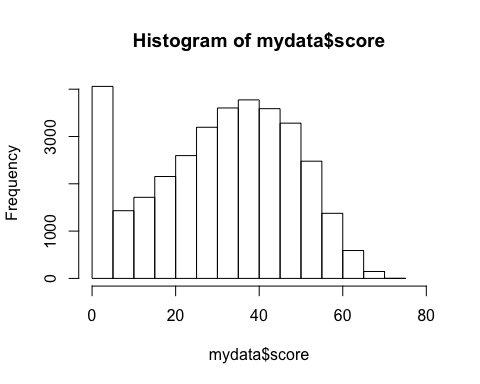
str(mydata)

## 'data.frame': 33988 obs. of 5 variables:  
## $ caseid : int 339 340 345 346 352 353 354 361 362 363 ...  
## $ score : int 49 18 46 43 17 29 15 19 45 12 ...  
## $ cohort90: int -6 -6 -6 -6 -6 -6 -6 -6 -6 -6 ...  
## $ female : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ sclass : int 2 3 4 3 3 2 3 2 3 1 ...

mydata [1:20, ]

## caseid score cohort90 female sclass  
## 1 339 49 -6 0 2  
## 2 340 18 -6 0 3  
## 3 345 46 -6 0 4  
## 4 346 43 -6 0 3  
## 5 352 17 -6 0 3  
## 6 353 29 -6 0 2  
## 7 354 15 -6 0 3  
## 8 361 19 -6 0 2  
## 9 362 45 -6 0 3  
## 10 363 12 -6 0 1  
## 11 6824 0 -4 0 1  
## 12 6826 0 -4 0 3  
## 13 6827 20 -4 0 2  
## 14 6828 32 -4 0 1  
## 15 6829 0 -4 0 2  
## 16 6834 24 -4 0 3  
## 17 6836 23 -4 0 2  
## 18 13206 7 -2 0 3  
## 19 13209 38 -2 0 3  
## 20 13215 46 -2 0 1

hist(mydata$score, xlim = c(0,80))



summary(mydata)

## caseid score cohort90 female   
## Min. : 1 Min. : 0.00 Min. :-6.0000 Min. :0.0000   
## 1st Qu.: 8532 1st Qu.:19.00 1st Qu.:-4.0000 1st Qu.:0.0000   
## Median :17318 Median :33.00 Median :-2.0000 Median :1.0000   
## Mean :18466 Mean :31.09 Mean : 0.2767 Mean :0.5276   
## 3rd Qu.:29428 3rd Qu.:45.00 3rd Qu.: 6.0000 3rd Qu.:1.0000   
## Max. :38192 Max. :75.00 Max. : 8.0000 Max. :1.0000   
## sclass   
## Min. :1.000   
## 1st Qu.:1.000   
## Median :2.000   
## Mean :2.147   
## 3rd Qu.:3.000   
## Max. :4.000

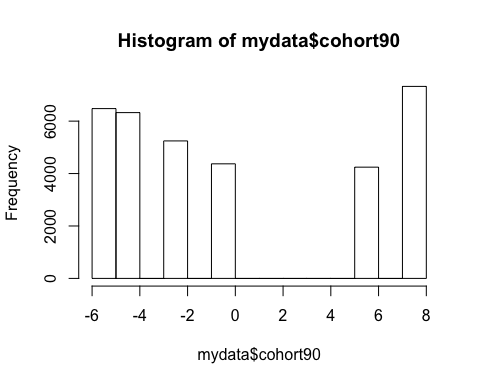
summary(mydata$score)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 19.00 33.00 31.09 45.00 75.00

sd(mydata$score)

## [1] 17.31437

hist(mydata$cohort90, xlim = c(-6, 8))



mydata <- read.csv("3.1.txt", sep = ",", header = TRUE)  
  
mytable <- table(mydata$cohort90)  
mytable

##   
## -6 -4 -2 0 6 8   
## 6478 6325 5245 4371 4244 7325

prop.table(mytable)

##   
## -6 -4 -2 0 6 8   
## 0.1905967 0.1860951 0.1543192 0.1286042 0.1248676 0.2155172

mytablecomb <- cbind(mytable, prop.table(mytable), cumsum(prop.table(mytable)))  
mytablecomb

## mytable   
## -6 6478 0.1905967 0.1905967  
## -4 6325 0.1860951 0.3766918  
## -2 5245 0.1543192 0.5310109  
## 0 4371 0.1286042 0.6596152  
## 6 4244 0.1248676 0.7844828  
## 8 7325 0.2155172 1.0000000

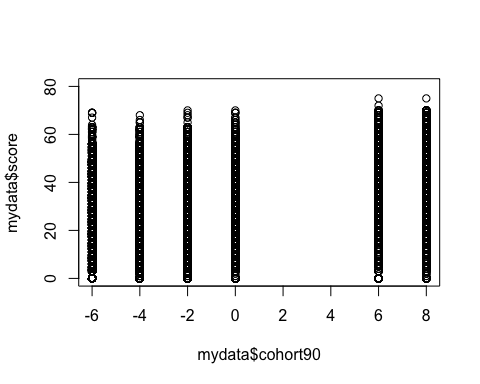
colnames(mytablecomb) <- c("Freq", "Perc", "Cum")  
mytablecomb

## Freq Perc Cum  
## -6 6478 0.1905967 0.1905967  
## -4 6325 0.1860951 0.3766918  
## -2 5245 0.1543192 0.5310109  
## 0 4371 0.1286042 0.6596152  
## 6 4244 0.1248676 0.7844828  
## 8 7325 0.2155172 1.0000000

mytablecomb

## Freq Perc Cum  
## -6 6478 0.1905967 0.1905967  
## -4 6325 0.1860951 0.3766918  
## -2 5245 0.1543192 0.5310109  
## 0 4371 0.1286042 0.6596152  
## 6 4244 0.1248676 0.7844828  
## 8 7325 0.2155172 1.0000000

plot(mydata$cohort90, mydata$score, ylim = c(0,80))

 #Tapply command applies a function to each group of values given by the levels of the specified factor. #The first line calculates the lenght of score for each value of cohort90, providing the number of observations for each level of cohort90. #The second line calculates the mean value of score separately for each value of cohort90. #The third line calculates the standard deviation of score for each cohort.

l <- tapply(mydata$score, factor(mydata$cohort90), length)  
m <- tapply(mydata$score, factor(mydata$cohort90), mean)  
s <- tapply(mydata$score, factor(mydata$cohort90), sd)  
tableScore <- cbind("Freq" = l, "mean(score)"= m, "sd(score)" = s)  
tableScore

## Freq mean(score) sd(score)  
## -6 6478 23.65545 18.07995  
## -4 6325 24.77265 17.37533  
## -2 5245 28.52450 15.93629  
## 0 4371 29.10043 15.76355  
## 6 4244 39.43473 13.55147  
## 8 7325 41.33065 13.00926

cor(mydata$score, mydata$cohort90)

## [1] 0.4088625

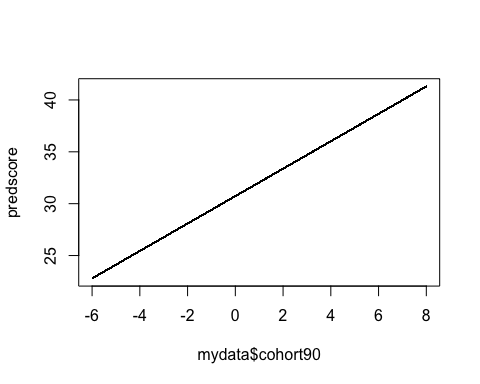
# Linear regression model

fit <- lm(score ~ cohort90, data = mydata)  
summary(fit)

##   
## Call:  
## lm(formula = score ~ cohort90, data = mydata)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -41.31 -11.73 0.56 12.20 46.20   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 30.72873 0.08582 358.04 <2e-16 \*\*\*  
## cohort90 1.32214 0.01601 82.59 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 15.8 on 33986 degrees of freedom  
## Multiple R-squared: 0.1672, Adjusted R-squared: 0.1671   
## F-statistic: 6822 on 1 and 33986 DF, p-value: < 2.2e-16

# Adding the linear plot

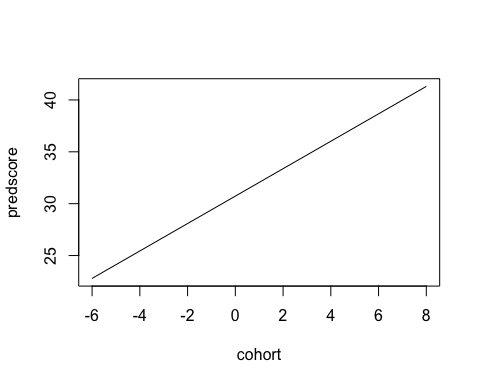
predscore <- predict(fit)  
plot(mydata$cohort90, predscore, type = "l")

 #The previous method was not efficient computationally. We simply need to run the command for a subset of data that gives us the combinations of the two variables used to plot the graph. First, we need to make a matrix with the two objects and then reduce it with the unique command.

uniquedata <- cbind(cohort90 = mydata$cohort90, predscore = predscore)  
uniquedata <- unique(uniquedata)  
uniquedata

## cohort90 predscore  
## 1 -6 22.79586  
## 11 -4 25.44015  
## 18 -2 28.08444  
## 23 0 30.72873  
## 27 6 38.66159  
## 31 8 41.30588

plot(uniquedata[, 1], uniquedata[,2], xlab = "cohort", ylab = "predscore", type = "l")

 ##Model checking: checking assumptions. The score value has a high proportion of zeros. We have to check the distribution of the residuals. We will look at two plot: 1. standardised residuals vs. normal scores (normal plot), and 2. standardised residuals vs fixed part prediction.