ds\_r\_proj\_solution\_code\_ntsua.R

2020-08-15

# This document is generated using knitr, File->Knit Document. Knitr then runs my r-code and knit the output into this document.  
# Warning: If there is a line in the code that installs a package , knitr expect the repository to be explicitly specified: install.packages(..., repos="....")  
  
  
# Package installation and loading  
  
# Package names  
packages <- c("Hmisc","corrplot","PerformanceAnalytics","correlation","car","caTools")  
  
# Install packages not yet installed  
installed\_packages <- packages %in% rownames(installed.packages())  
if (any(installed\_packages == FALSE)) {  
 install.packages(packages[!installed\_packages], repos = "http://cran.us.r-project.org" )  
}  
  
# Packages loading  
invisible(lapply(packages, library, character.only = TRUE))

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':  
##   
## format.pval, units

## corrplot 0.84 loaded

## Loading required package: xts

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

##   
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':  
##   
## legend

## Loading required package: carData

# Importing data  
  
library(readr)  
Hcost <- read\_csv("HospitalCosts.csv", TRUE)

## Parsed with column specification:  
## cols(  
## AGE = col\_double(),  
## FEMALE = col\_double(),  
## LOS = col\_double(),  
## RACE = col\_double(),  
## TOTCHG = col\_double(),  
## APRDRG = col\_double()  
## )

names(Hcost)

## [1] "AGE" "FEMALE" "LOS" "RACE" "TOTCHG" "APRDRG"

dim(Hcost)

## [1] 500 6

ls.str(Hcost)

## AGE : num [1:500] 17 17 17 17 17 17 17 16 16 17 ...  
## APRDRG : num [1:500] 560 753 930 758 754 347 754 754 753 758 ...  
## FEMALE : num [1:500] 1 0 1 1 1 0 1 1 1 1 ...  
## LOS : num [1:500] 2 2 7 1 1 0 4 2 1 2 ...  
## RACE : num [1:500] 1 1 1 1 1 1 1 1 1 1 ...  
## TOTCHG : num [1:500] 2660 1689 20060 736 1194 ...

# The structure of the data shows Gender and Race are numeric.  
# We need to convert to factor so that calculation is not done on them.  
# For example, if they remain numeric as opposed to factors,  
# the summary function will produce the mean of Race, with is meaningless.  
# For analysis reasons, I am going to create factor variable of these two variables:  
  
Hcost$FAC\_FEM <- as.factor(Hcost$FEMALE)  
Hcost$FAC\_RACE <- as.factor(Hcost$RACE)  
  
  
# Descriptive Statistics of variables  
  
#library(dplyr)  
sapply(Hcost, summary)

## $AGE  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 0.000 0.000 5.086 13.000 17.000   
##   
## $FEMALE  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 0.000 1.000 0.512 1.000 1.000   
##   
## $LOS  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 2.000 2.000 2.828 3.000 41.000   
##   
## $RACE  
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 1.000 1.000 1.000 1.078 1.000 6.000 1   
##   
## $TOTCHG  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 532 1216 1536 2774 2530 48388   
##   
## $APRDRG  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 21.0 640.0 640.0 616.4 751.0 952.0   
##   
## $FAC\_FEM  
## 0 1   
## 244 256   
##   
## $FAC\_RACE  
## 1 2 3 4 5 6 NA's   
## 484 6 1 3 3 2 1

# Checking on missingness  
sum(is.na(Hcost))

## [1] 2

# Checking on correlation between cost and the rest of the variables: correlation matrix.  
# We want to display the correlation coefficients as well as the p-values, so we need to install the packages: "Hmisc","corrplot","PerformanceAnalytics and "correlation"  
  
  
library(Hmisc); library(corrplot); library(PerformanceAnalytics) ; library(correlation)  
  
# Correlation matrix  
corM <- rcorr(as.matrix(Hcost), type="pearson")  
  
#Instead of displaying the entire corM, let's look at the various parts we can extract  
  
names(corM)

## [1] "r" "n" "P"

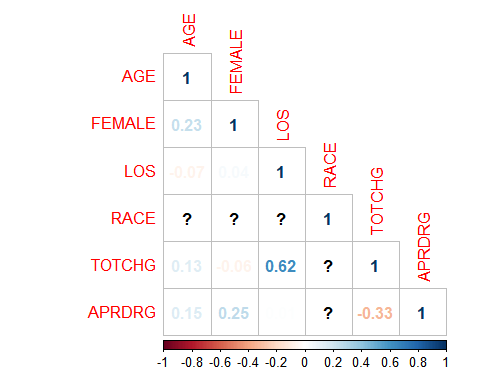
# We see that we can choose $r (correlation matrix), $n (case count) and $P (the p-values)  
# Let's round these numbers to 2 digits after decimal.  
round(corM$r,2)

## AGE FEMALE LOS RACE TOTCHG APRDRG FAC\_FEM FAC\_RACE  
## AGE 1.00 0.23 -0.07 0.02 0.13 0.15 0.23 0.02  
## FEMALE 0.23 1.00 0.04 -0.04 -0.06 0.25 1.00 -0.04  
## LOS -0.07 0.04 1.00 -0.02 0.62 0.01 0.04 -0.02  
## RACE 0.02 -0.04 -0.02 1.00 -0.02 -0.04 -0.04 1.00  
## TOTCHG 0.13 -0.06 0.62 -0.02 1.00 -0.33 -0.06 -0.02  
## APRDRG 0.15 0.25 0.01 -0.04 -0.33 1.00 0.25 -0.04  
## FAC\_FEM 0.23 1.00 0.04 -0.04 -0.06 0.25 1.00 -0.04  
## FAC\_RACE 0.02 -0.04 -0.02 1.00 -0.02 -0.04 -0.04 1.00

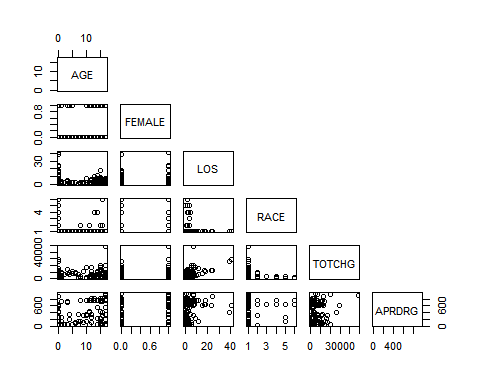
round(corM$P,2)

## AGE FEMALE LOS RACE TOTCHG APRDRG FAC\_FEM FAC\_RACE  
## AGE NA 0.00 0.13 0.72 0.00 0.00 0.00 0.72  
## FEMALE 0.00 NA 0.42 0.39 0.18 0.00 0.00 0.39  
## LOS 0.13 0.42 NA 0.69 0.00 0.91 0.42 0.69  
## RACE 0.72 0.39 0.69 NA 0.69 0.33 0.39 0.00  
## TOTCHG 0.00 0.18 0.00 0.69 NA 0.00 0.18 0.69  
## APRDRG 0.00 0.00 0.91 0.33 0.00 NA 0.00 0.33  
## FAC\_FEM 0.00 0.00 0.42 0.39 0.18 0.00 NA 0.39  
## FAC\_RACE 0.72 0.39 0.69 0.00 0.69 0.33 0.39 NA

#To easily see positive and negative correlation among variables, let's use corrplot to visualize the matrix. Especially, method="ellipse" is very useful in seeing the sign of the correlation.  
  
corrplot(cor(Hcost[1:6]), method = "number" , type ="lower")



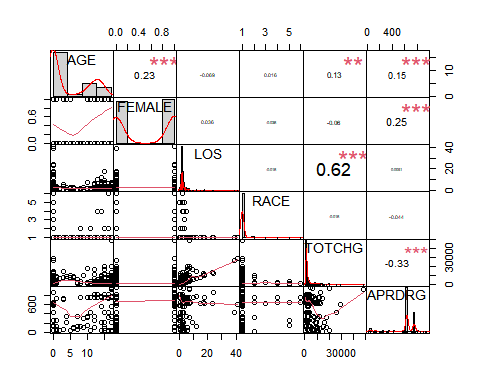
# Plotting the correlation, 3 variables at a time, so we don't get a cluttered graph  
  
library(ggplot2)  
  
pairs(Hcost[1:6],upper.panel = NULL)



# Correlation table  
  
correl <- correlation(Hcost)  
summary(correl)

## Parameter | APRDRG | TOTCHG | RACE | LOS | FEMALE  
## --------------------------------------------------------  
## AGE | 0.15\* | 0.13\* | 0.02 | -0.07 | 0.23\*\*\*  
## FEMALE | 0.25\*\*\* | -0.06 | -0.04 | 0.04 |   
## LOS | 0.01 | 0.62\*\*\* | -0.02 | |   
## RACE | -0.04 | -0.02 | | |   
## TOTCHG | -0.33\*\*\* | | | |

chart.Correlation(Hcost[1:6], histogram=TRUE, pch=19)



# Slitting data into training and testing, using caTools package  
  
library(caTools); library(car)  
  
# To be able to produce the same training and testing data in the future, we set a seed to a randomly picked value.  
set.seed(10)  
# Splitting the data into 70/30  
splitsample <- sample.split(Hcost$TOTCHG,SplitRatio = .7)  
  
train\_hcost <- subset(Hcost,splitsample==TRUE)  
test\_hcost <- subset(Hcost,splitsample==FALSE)  
  
# Building the model with all variables  
train\_model\_all <- lm(train\_hcost$TOTCHG ~ ., data = train\_hcost)  
  
summary(train\_model\_all)

##   
## Call:  
## lm(formula = train\_hcost$TOTCHG ~ ., data = train\_hcost)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6864.5 -538.4 -202.5 69.0 15973.1   
##   
## Coefficients: (2 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6084.128 526.891 11.547 < 2e-16 \*\*\*  
## AGE 116.670 14.884 7.839 5.91e-14 \*\*\*  
## FEMALE -400.046 213.579 -1.873 0.0619 .   
## LOS 708.675 26.161 27.089 < 2e-16 \*\*\*  
## RACE -174.838 374.722 -0.467 0.6411   
## APRDRG -8.996 0.597 -15.070 < 2e-16 \*\*\*  
## FAC\_FEM1 NA NA NA NA   
## FAC\_RACE2 644.662 856.620 0.753 0.4522   
## FAC\_RACE3 678.339 2014.622 0.337 0.7365   
## FAC\_RACE4 252.570 1552.491 0.163 0.8709   
## FAC\_RACE5 -1085.959 2007.956 -0.541 0.5890   
## FAC\_RACE6 NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1863 on 340 degrees of freedom  
## Multiple R-squared: 0.7479, Adjusted R-squared: 0.7412   
## F-statistic: 112.1 on 9 and 340 DF, p-value: < 2.2e-16

#The summary statistics of train\_model show that the coefficients of AGE, LOS and APRDRG are significantly different from zero, with a confidence level of 0.05, therefore, we will include them in our final linear model.  
  
# Building the model with significant variables only  
  
train\_model\_sig <- lm(TOTCHG ~ AGE+LOS+APRDRG , data = train\_hcost)  
  
summary(train\_model\_sig)

##   
## Call:  
## lm(formula = TOTCHG ~ AGE + LOS + APRDRG, data = train\_hcost)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7159.2 -538.5 -196.9 73.8 15915.1   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5900.9997 375.3714 15.72 < 2e-16 \*\*\*  
## AGE 111.9048 14.3651 7.79 7.89e-14 \*\*\*  
## LOS 706.3185 26.1254 27.04 < 2e-16 \*\*\*  
## APRDRG -9.2503 0.5738 -16.12 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1863 on 346 degrees of freedom  
## Multiple R-squared: 0.7435, Adjusted R-squared: 0.7412   
## F-statistic: 334.3 on 3 and 346 DF, p-value: < 2.2e-16

names(train\_model\_sig)

## [1] "coefficients" "residuals" "effects" "rank"   
## [5] "fitted.values" "assign" "qr" "df.residual"   
## [9] "xlevels" "call" "terms" "model"

# We check if the number of fitted values is equal to number of observations in the training data  
length(train\_model\_sig$fitted.values)

## [1] 350

dim(train\_hcost)

## [1] 350 8

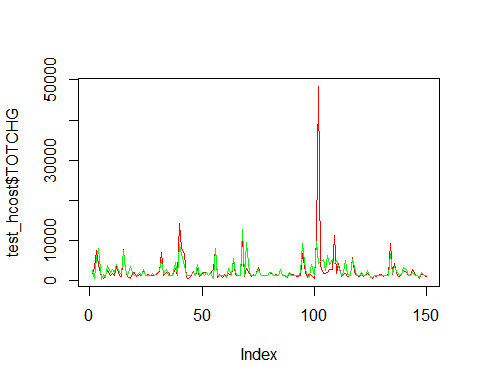
# They are, so we can predict.  
# Create data frames for predict and residuals of the training data.  
  
predict\_train\_df <- data.frame(train\_model\_sig$fitted.values)  
resid\_train\_df <- data.frame(train\_model\_sig$residuals)  
str(resid\_train\_df)

## 'data.frame': 350 obs. of 1 variable:  
## $ train\_model\_sig.residuals: num -1376 -562 15915 -762 -341 ...

# Perform prediction using testing data  
  
predict\_test\_df <- predict(train\_model\_sig,newdata = test\_hcost)  
predict\_test\_df <- data.frame(predict\_test\_df)  
str(predict\_test\_df)

## 'data.frame': 150 obs. of 1 variable:  
## $ predict\_test\_df: num 2575 551 5647 8082 167 ...

# Plotting testing data versus predicted testing data  
  
plot(test\_hcost$TOTCHG,col="red",type = "l")  
lines(predict\_test\_df,col="green", type = "l")



# Linear regression model assumptions verification: let's diagnose our linear model to see if our data does not violate the assumptions  
  
  
# Test for auto-correlation in residuals  
  
durbinWatsonTest(train\_model\_sig)

## lag Autocorrelation D-W Statistic p-value  
## 1 0.02421334 1.946236 0.512  
## Alternative hypothesis: rho != 0

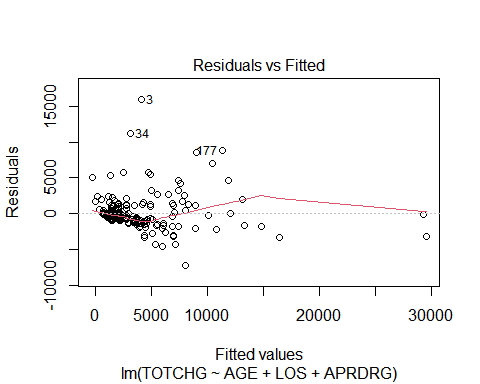
#==>> p-values =0.512 >> 0.001, so there is no first order correlation in residuals.  
  
# Test for collinearity using VIF  
vif(train\_model\_sig)

## AGE LOS APRDRG   
## 1.032630 1.008145 1.024618

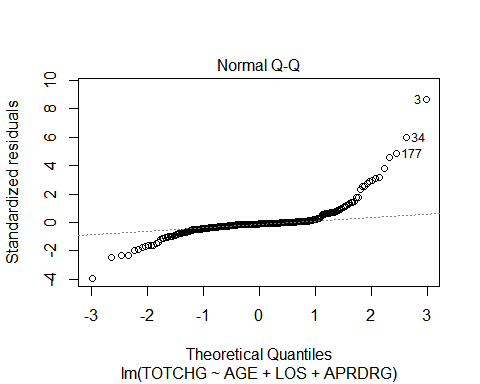
sqrt(vif(train\_model\_sig))>5

## AGE LOS APRDRG   
## FALSE FALSE FALSE

#==> All VIF factors are less than 5, we can conclude that there is no collinearity among the variables retained for the final model.  
  
# Test for linearity and homoscedasticity using plot of Residual versus fitted data  
plot(train\_model\_sig,which=1)



#==>> Except from three data points with large residuals (3,34 and 177), the data is fairly spread around the x-axis (y=0)  
  
# Test for normality of the data using quantile-quantile plot  
plot(train\_model\_sig,which=2)



## The data in most cases aligns with the 45 degree line, meaning the data is fairly normally distributed  
  
# Conlusion  
## Even thought the data seems not to be race-balanced, the tests of the model assumptions indicate linear regression model is appropriate for the analysis.  
## The analysis shows that the model can accurately predict the cost of hospitalization using patient's age, length of stay and diagnosis group.