HC - Data Prepare

```
require(corrplot)
```

```
## Loading required package: corrplot
```

```
## Warning: package 'corrplot' was built under R version 4.0.2
```

```
## corrplot 0.84 loaded
```

```
require(ggplot2)    ## declaratively creating graphics - https://ggplot2.tidyverse.org/
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.0.2
```

```
require(gridExtra)          ## arrange visualizations using grid
```

```
## Loading required package: gridExtra
```

```
## Warning: package 'gridExtra' was built under R version 4.0.2
```

```
#Load the data and initialize necessary variables

medicalData = read.csv("http://www.datadescant.com/stat109/hospvisits.csv")

head(medicalData)
```

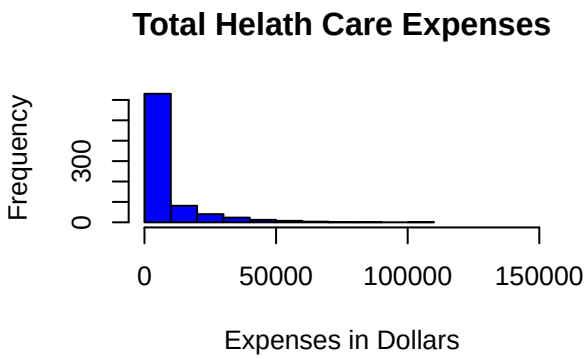
```
##   totalexp age marital educ income srhealth mntl_hlth phy_lim  bmi chd
## 1     5901  74       2   12  10326        4         3       0 17.4   0
## 2     3325  72       1   12  14814        2         2       1 31.3   0
## 3     1986  72       2   13  11054        3         1       0 26.6   0
## 4      550  66       1   10    938        3         3       0 30.0   1
## 5     4010  69       1   12  41100        2         2       0 28.5   0
## 6     5141  71       2   12    125        3         3       1 29.7   0
##   high_chol diabetes dr_visits msa race_grp smoker male high_bp hosp_vis
## 1         1        0        12   0        1      0    0       1        0
## 2         0        0         8   1        1      0    0       1        0
## 3         0        0         3   1        1      0    0       0        0
## 4         0        0         1   0        1      0    1       1        0
## 5         1        0         2   1        1      0    0       1        0
## 6         1        0         3   1        1      0    0       1        0
```

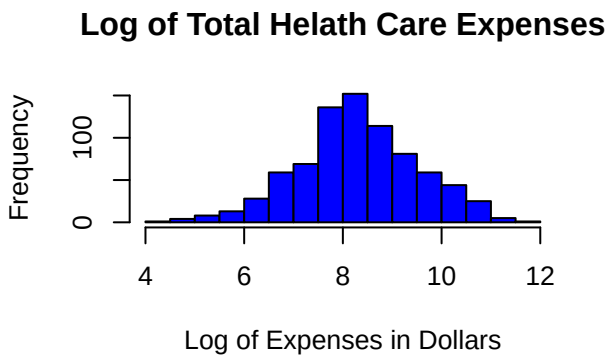
```
response_df = medicalData['totalexp']  # Y variable
predictors_df = medicalData[, !names(medicalData) %in% "totalexp" ]  # X variables

# Data frame to store the results of the various models
modelResults = setNames(data.frame(matrix(ncol = 7, nrow = 0)), c("Name", "Model", "RMSE", "R2", "MAE",

summary(medicalData)
```

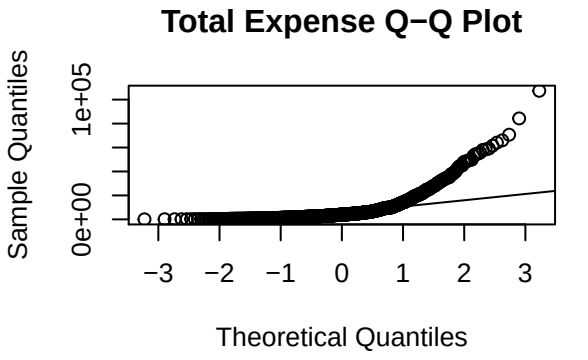
```
##     totalexp           age           marital           educ
##  Min.   :    58   Min.   :65.00   Min.   :1.000   Min.   : 0.00
##  1st Qu.:  1951   1st Qu.:69.00   1st Qu.:1.000   1st Qu.:10.00
##  Median :  3974   Median :74.00   Median :1.000   Median :12.00
##  Mean   :  8164   Mean   :74.22   Mean   :1.672   Mean   :11.66
##  3rd Qu.:  9018   3rd Qu.:79.00   3rd Qu.:2.000   3rd Qu.:14.00
##  Max.   :107355   Max.   :85.00   Max.   :4.000   Max.   :17.00
##      income          srhealth       mntl_hlth       phy_lim
##  Min.   :   125   Min.   :1.000   Min.   :1.00   Min.   :0.0000
##  1st Qu.:  9585   1st Qu.:2.000   1st Qu.:2.00   1st Qu.:0.0000
##  Median : 16381   Median :3.000   Median :2.00   Median :0.0000
##  Mean   : 24105   Mean   :2.767   Mean   :2.34   Mean   :0.3742
##  3rd Qu.: 31982   3rd Qu.:4.000   3rd Qu.:3.00   3rd Qu.:1.0000
##  Max.   :176839   Max.   :5.000   Max.   :5.00   Max.   :1.0000
##       bmi              chd           high_chol         diabetes
##  Min.   :16.10   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:23.70   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :26.30   Median :0.0000   Median :1.0000   Median :0.0000
##  Mean   :27.22   Mean   :0.1377   Mean   :0.5369   Mean   :0.2065
##  3rd Qu.:30.00   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :58.50   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##    dr_visits           msa           race_grp         smoker
##  Min.   :  0.00   Min.   :0.000   Min.   :1.000   Min.   :0.0000
##  1st Qu.:  4.00   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.0000
##  Median :  7.00   Median :1.000   Median :1.000   Median :0.0000
##  Mean   : 10.62   Mean   :0.806   Mean   :1.574   Mean   :0.1076
##  3rd Qu.: 14.00   3rd Qu.:1.000   3rd Qu.:2.000   3rd Qu.:0.0000
##  Max.   :159.00   Max.   :1.000   Max.   :4.000   Max.   :1.0000
##       male          high_bp         hosp_vis
##  Min.   :0.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.000   Median :1.0000   Median :0.0000
##  Mean   :0.418   Mean   :0.6683   Mean   :0.2365
##  3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :1.000   Max.   :1.0000   Max.   :7.0000
```

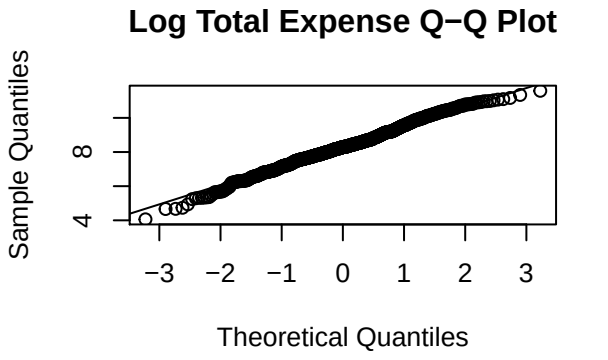
```
# Create train & test data set

set.seed(123)
sample = sample.int(n = nrow(medicalData),
                    size = floor(.80*nrow(medicalData)), replace = F)
medicalData.Train = medicalData[sample,]
medicalData.Test = medicalData[-sample,]
```

```r
# Data exploration

numericVars = which(sapply(medicalData, is.numeric)) #index vector numeric variables
numericVarNames = names(numericVars) #saving names vector for use later on
##cat('There are', length(numericVars), 'numeric variables')

par(mfrow = c(2, 2))

hist(medicalData$totalexp,
main="Total Helath Care Expenses",
xlab="Expenses in Dollars",
xlim=c(0,150000),
col="blue",
freq=TRUE
)

hist(log(medicalData$totalexp),
main="Log of Total Helath Care Expenses",
xlab="Log of Expenses in Dollars",
col="blue",
freq=TRUE
)

qqnorm(medicalData$totalexp, main = "Total Expense Q-Q Plot", xlab = "Theoretical Quantiles",
       ylab = "Sample Quantiles", plot.it = TRUE, datax = FALSE)
qqline(medicalData$totalexp, datax = FALSE, distribution = qnorm,
       probs = c(0.25, 0.75), qtype = 7)


qqnorm(log(medicalData$totalexp), main = "Log Total Expense Q-Q Plot", xlab = "Theoretical Quantiles",
       ylab = "Sample Quantiles", plot.it = TRUE, datax = FALSE)
qqline(log(medicalData$totalexp), datax = FALSE, distribution = qnorm,
       probs = c(0.25, 0.75), qtype = 7)
```

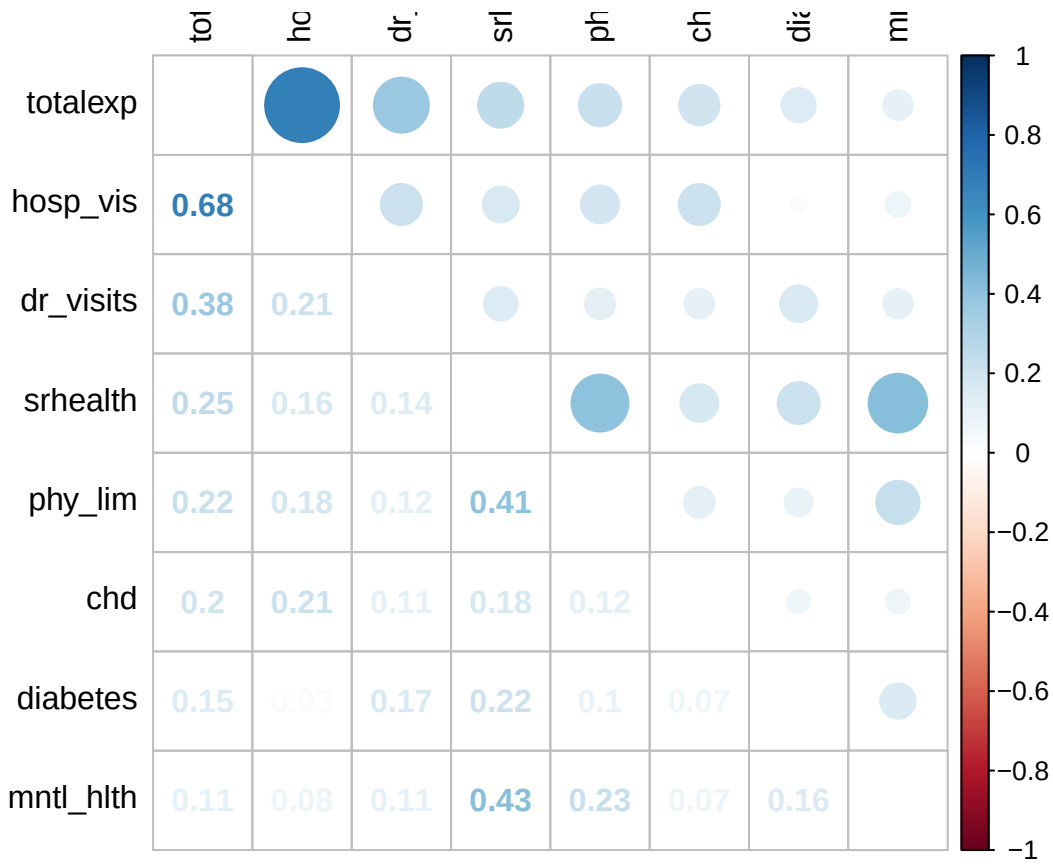




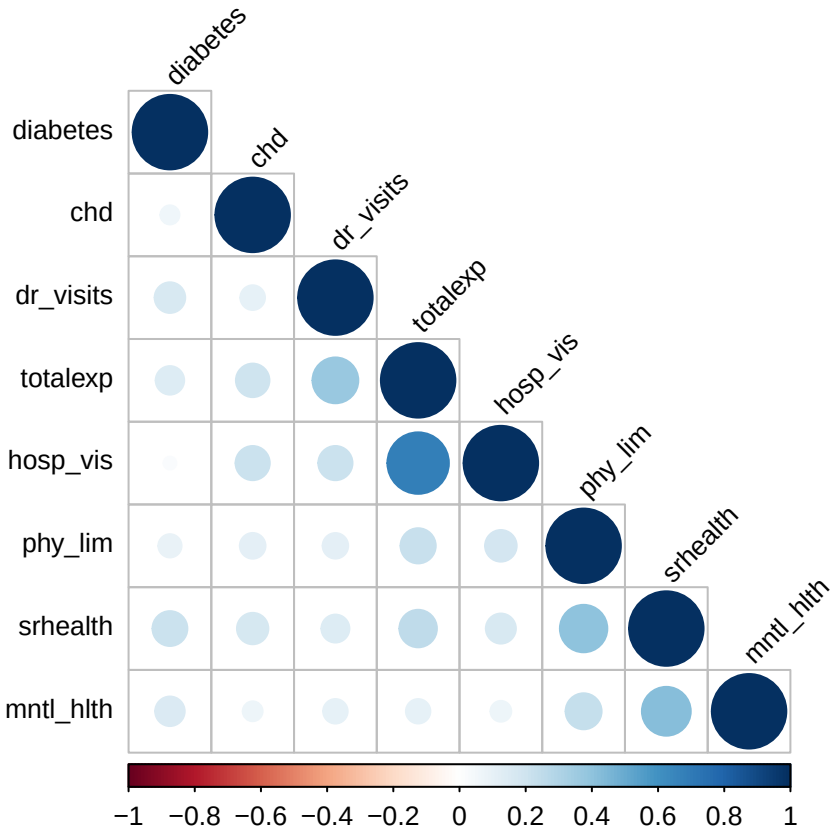





```r
# Check for correlation among the variables

cor_numVar = cor(medicalData, use="pairwise.complete.obs") #correlations of all numeric variables
cor_sorted = as.matrix(sort(cor_numVar[,'totalexp'], decreasing = TRUE))

#select only high corelations
CorHigh = names(which(apply(cor_sorted, 1, function(x) abs(x)>0.1)))
cor_numVar = cor_numVar[CorHigh, CorHigh]

corrplot.mixed(cor_numVar, tl.col="black", tl.pos = "lt")
```

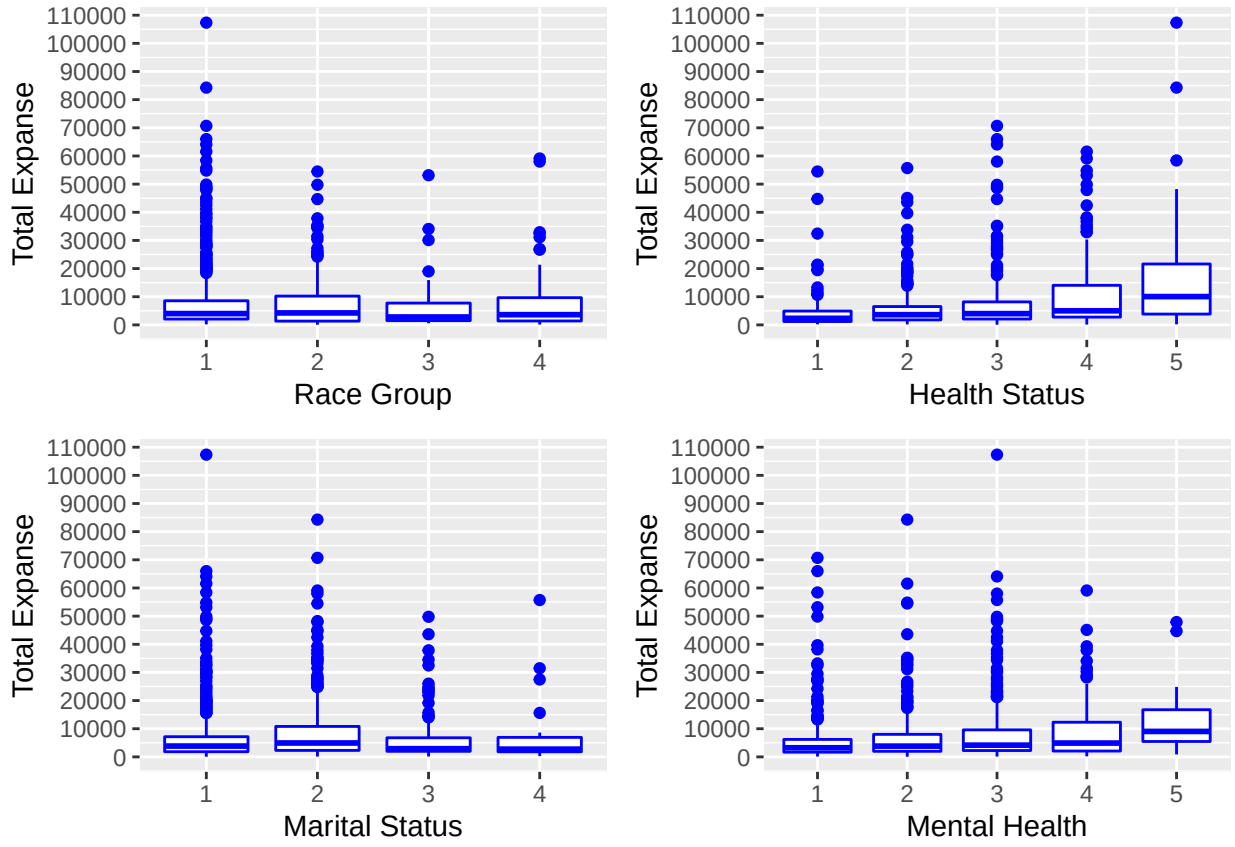



```
corrplot(cor_numVar, title = "",
             type = "lower",
             order = "hclust",
             hclust.method = "centroid",
             tl.cex = 0.8,
             tl.col = "black",
             tl.srt = 45)
```



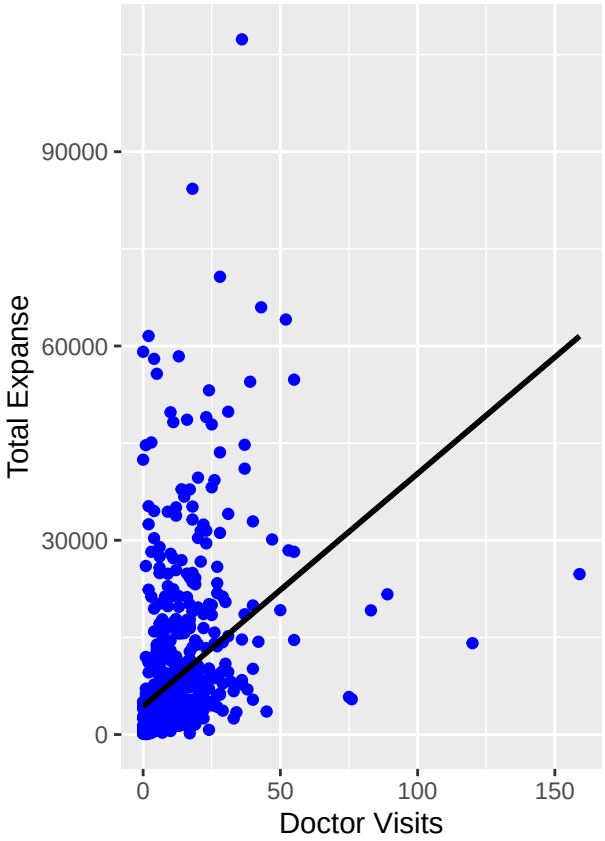

```
# More data exploration

# Race Group
p1 = ggplot(data=medicalData[!is.na(medicalData$totalexp),], aes(x=factor(race_grp), y=totalexp))+
        geom_boxplot(col='blue') + labs(x='Race Group',y='Total Expanse') +
        scale_y_continuous(breaks= seq(0, 200000, by=10000))

# Health Status
p2 = ggplot(data=medicalData[!is.na(medicalData$totalexp),], aes(x=factor(srhealth), y=totalexp))+
        geom_boxplot(col='blue') + labs(x='Health Status',y='Total Expanse') +
        scale_y_continuous(breaks= seq(0, 200000, by=10000))

# Marital Status
p3 = ggplot(data=medicalData[!is.na(medicalData$totalexp),], aes(x=factor(marital), y=totalexp))+
        geom_boxplot(col='blue') + labs(x='Marital Status',y='Total Expanse') +
        scale_y_continuous(breaks= seq(0, 200000, by=10000))

# Mental Health
p4 = ggplot(data=medicalData[!is.na(medicalData$totalexp),], aes(x=factor(mntl_hlth), y=totalexp))+
        geom_boxplot(col='blue') + labs(x='Mental Health',y='Total Expanse') +
        scale_y_continuous(breaks= seq(0, 200000, by=10000))

# Doctor Visits
p5 = ggplot(data=medicalData[!is.na(medicalData$totalexp),], aes(x=dr_visits, y=totalexp)) +
  geom_point(col='blue') +
  geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
```

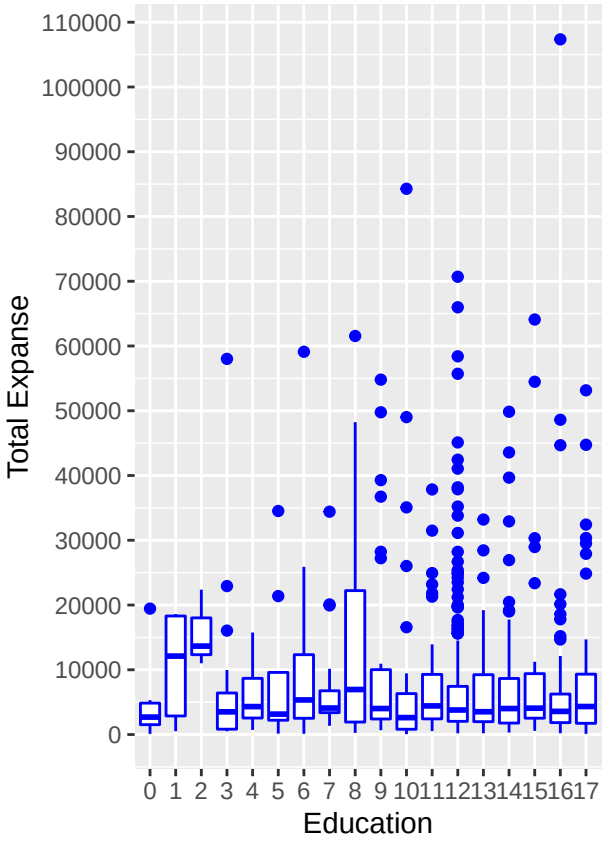
```
  labs(x='Doctor Visits',y='Total Expanse')

# Education
p6 = ggplot(data=medicalData[!is.na(medicalData$totalexp),], aes(x=factor(educ), y=totalexp))+
        geom_boxplot(col='blue') + labs(x='Education',y='Total Expanse') +
        scale_y_continuous(breaks= seq(0, 200000, by=10000))

grid.arrange(p1,p2,p3,p4,nrow = 2)
```




```
grid.arrange(p5,p6,nrow = 1)
```

```
## `geom_smooth()` using formula 'y ~ x'
```





Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.