DATA

Dataset Description

This study draws on the "Medical Cost" dataset from the esteemed data science community, Kaggle.com. The dataset, curated from Brett Lantz's seminal text "Machine Learning with R," comprises the medical insurance expenses of 1338 individuals. Additionally, the dataset boasts a comprehensive range of features, including 3 categorical and 4 quantitative variables that will be elaborated in the ensuing table. The richness and complexity of this dataset ensure fertile ground for insightful analysis and interpretation. The dataset submitted from an open data source is available at the following website https://www.kaggle.com/mirichoi0218/insurance.

Loading Libraries

To achieve my objectives, I installed a selected set of packages and loaded several indispensable libraries. The code snippet below was instrumental in executing this process, propelling me closer to the end goal.

````
```{r}
install.packages("dplyr")
install.packages("ggplot2")
library(tidyverse)
library(corrplot)
library(class)
library(MASS)
library(ggplot2)
install.packages("leaps")
library(leaps)
```
````

Loading the Dataset

The following code was used to load the dataset, it was loaded from an Excel CSV file, and the dataset's correct loading was checked using a data frame.

````
```{r}
Insurance <- read.csv("C:/Users/Modupe Olayinka/OneDrive - University of Louisiana
Lafayette/Desktop/dataset/health insurance.csv", head=
TRUE)
is.data.frame(Insurance)
```
````

The first six rows of the dataset are displayed in the function head ()

````
```{r}
head(Insurance)
```
````

Description: df [6 x 7]

	age <int>	sex <chr>	bmi <dbl>	children <int>	smoker <chr>	region <chr>	charges <dbl>
1	19	female	27.900	0	yes	southwest	16884.924
2	18	male	33.770	1	no	southeast	1725.552
3	28	male	33.000	3	no	southeast	4449.462
4	33	male	22.705	0	no	northwest	21984.471
5	32	male	28.880	0	no	northwest	3866.855
6	31	female	25.740	0	no	southeast	3756.622

6 rows

To ascertain the dataset's dimensions, the function dim () was utilized. Our analysis revealed that the dataset comprises 7 columns and 1338 rows.

```
```{r}
dim(Insurance)
```
[1] 1338   7
```

The original structures

The output, shown below, showed that the original dataset included three modes: num, int, and Chr. I used the str () function to identify the variable modes in the dataset. It is important to note that before the study begins, the dataset should only receive a light cleaning.

```
```{r}
str(Insurance)
```
'data.frame':   1338 obs. of 7 variables:
 $ age   : int 19 18 28 33 32 31 46 37 37 60 ...
 $ sex   : chr "female" "male" "male" "male" ...
 $ bmi   : num 27.9 33.8 33 22.7 28.9 ...
 $ children: int 0 1 3 0 0 0 1 3 2 0 ...
 $ smoker : chr "yes" "no" "no" "no" ...
 $ region : chr "southwest" "southeast" "southeast" "northwest" ...
 $ charges : num 16885 1726 4449 21984 3867 ...
```

The colnames () method was employed to exhibit the list of columns in the dataset, as exemplified below.

```
```{r}
colnames(Insurance)
```
[1] "age"   "sex"   "bmi"   "children" "smoker"   "region"   "charges"
```

Data Cleaning

First, I looked at the dataset to check if each column had the same number of values. Then, I used the following code to search for any missing data, but there wasn't any missing data found.

```
```{r}
colSums(sapply(Insurance, is.na))
```
     age   sex   bmi children   smoker   region charges
       0   0   0   0   0   0   0
```

The variables "age" and "children" were recorded as whole numbers. To make it easier to analyze them, we changed the mode from "int" to "num" using the function called "as. numeric ()". The variables "sex," "smoker," and "region" were saved as text so we could easily change them during our investigation. We will use the function called "as. factor ()" to change them from being descriptive to being a measurable value.

```{r}
Insurance$sex <- as.factor(Insurance$sex)
is.factor(Insurance$sex)
Insurance$smoker <- as.factor(Insurance$smoker)
is.factor(Insurance$smoker)
Insurance$region <- as.factor(Insurance$region)
is.factor(Insurance$region)
Insurance$age <- as.numeric(Insurance$age)
is.numeric(Insurance$age)
Insurance$children<- as.numeric(Insurance$children)
is.numeric(Insurance$children)

[1] TRUE
[1] TRUE
[1] TRUE
[1] TRUE
[1] TRUE
```

Cleaned Data

This is the updated result of the "str()" command following data cleaning.

```{r}
str(Insurance)

'data.frame':	1338 obs. of  7 variables:
 $ age     : num  19 18 28 33 32 31 46 37 37 60 ...
 $ sex     : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...
 $ bmi     : num  27.9 33.8 33 22.7 28.9 ...
 $ children: num  0 1 3 0 0 0 1 3 2 0 ...
 $ smoker  : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
 $ region  : Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...
 $ charges : num  16885 1726 4449 21984 3867 ...
```

Variable Description

The cleaned data used to analyze this dataset is described in the table below. The descriptions were taken from the website's author description.

https://www.kaggle.com/mirichoi0218/insurance.

TABLE 1 - VARIABLE DESCRIPTION

Column Name	Independent/ Dependent	Mode	Description
age	Ind	Numeric	Age of primary beneficiary
sex	Ind	Factor	Insurance contractor gender: 2 levels (female, male)
bmi	Ind	Numeric	Body mass index
children	Ind	Numeric	Number of children/dependents covered by health insurance
smoker	Ind	Factor	Smoking: 2 levels (yes, no)
region	Ind	Factor	Beneficiary's residentaial area in the US: 4 levels (northeast, southeast, northwest, southwest)
Charges	Dep	Numeric	Individual medical costs billed by health insurance

Expectations

This project's primary goal is to forecast the medical costs that health insurance providers will charge. The cost of providing coverage to a person is estimated over a long period by insurance companies. The goal is to determine whether some people will need medical care based on an analysis of the data that is already available utilizing critical variables like BMI and smoking behaviors. Insurance providers can adjust their premiums using this information.

It is anticipated through data analysis that elements like BMI and smoking habits will have a substantial impact on insurance costs. Smokers and individuals with higher BMIs are likelier to have higher premiums than non-smokers. Several graphical methods, including bar graphs, plots, and heatmaps, will be employed to examine the dataset efficiently. Different methods, including linear regression, best subset, ridge, and lasso regressions, will be used to accurately estimate insurance costs. Techniques like K-fold cross-validation and validation set will also be employed.

Data Analysis

I first collected a data summary before I started to analyze the dataset.

```{r}
summary(Insurance)
```

```
      age            sex           bmi           children        smoker          region   
 Min.   :18.00   female:662   Min.   :15.96   Min.   :0.000   no :1064   northeast:324  
 1st Qu.:27.00   male  :676   1st Qu.:26.30   1st Qu.:0.000   yes: 274   northwest:325  
 Median :39.00                Median :30.40   Median :1.000              southeast:364  
 Mean   :39.21                Mean   :30.66   Mean   :1.095              southwest:325  
 3rd Qu.:51.00                3rd Qu.:34.69   3rd Qu.:2.000                             
 Max.   :64.00                Max.   :53.13   Max.   :5.000                             
    charges     
 Min.   : 1122  
 1st Qu.: 4740  
 Median : 9382  
 Mean   :13270  
 3rd Qu.:16640  
 Max.   :63770  
```

This summary gives a broad overview of how the data is split among the different features. It demonstrates that a primary beneficiary must be at least 18 years old. Additionally, the summary notes that there are equally as many males and female beneficiaries in the sample. We may also see that each primary beneficiary may have a maximum of five dependents.

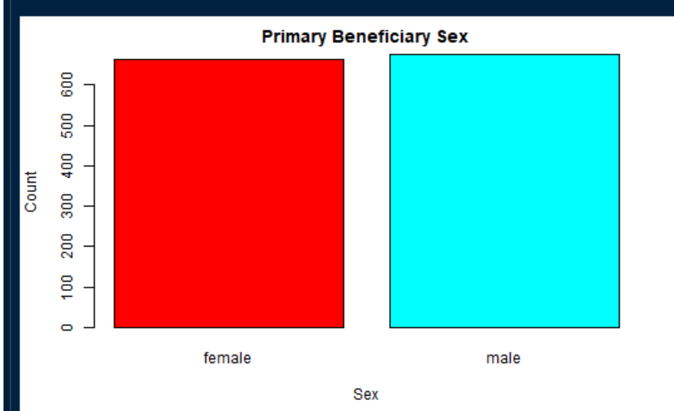
Regarding smoking, the summary reveals that there are significantly more non-smokers than smokers in the dataset.

Categorical Variables

I utilized barplots to investigate the categorical variables, Sex, Smoker, and Region, because I thought they would more accurately depict these factors.

Beneficiaries Count by Sex

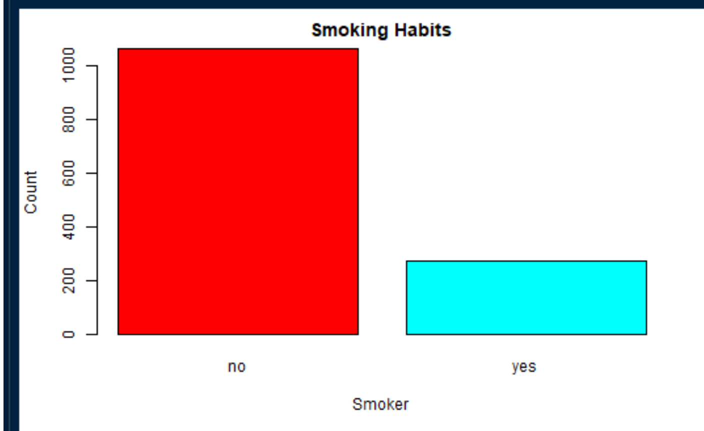
```{r}
bartable <- table(Insurance$sex)
barplot(bartable, main = "Primary Beneficiary Sex", xlab = "Sex", ylab = "Count", col =
rainbow(nlevels(Insurance$sex)))
```



As seen in the summary command, the plot indicates that the dataset has an equal number of females and males.

Smoking Habits

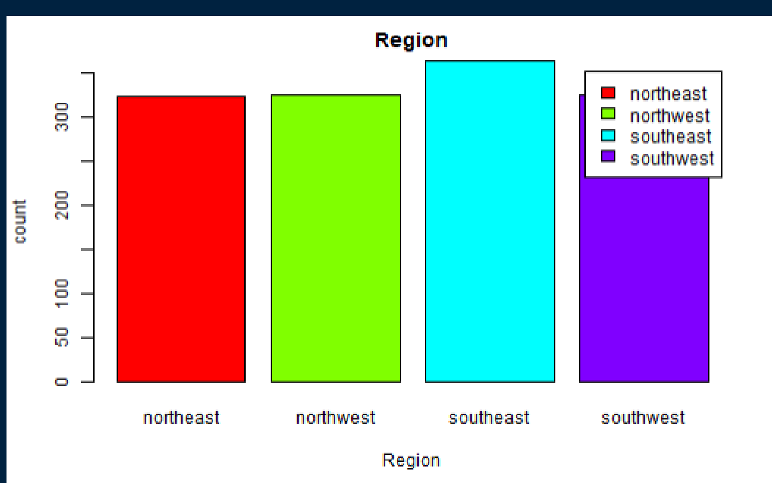
````
```{r}
bartable <- table(Insurance$smoker)
barplot(bartable, main = "Smoking Habits", xlab = "Smoker", ylab = "Count", col =
rainbow(nlevels(Insurance$smoker)))
```
````



The barplot shows that in the dataset, there are noticeably more non-smokers than smokers.

Region

````
```{r}
bartable = table(Insurance$region)
barplot(bartable, legend =
levels(unique(Insurance$region)),main="Region",xlab="Region",ylab="count",col =
rainbow(nlevels(Insurance$region)))
```
````



We can see from the region barplot that the beneficiaries are roughly evenly divided throughout the various areas, with somewhat more people in the southeast.

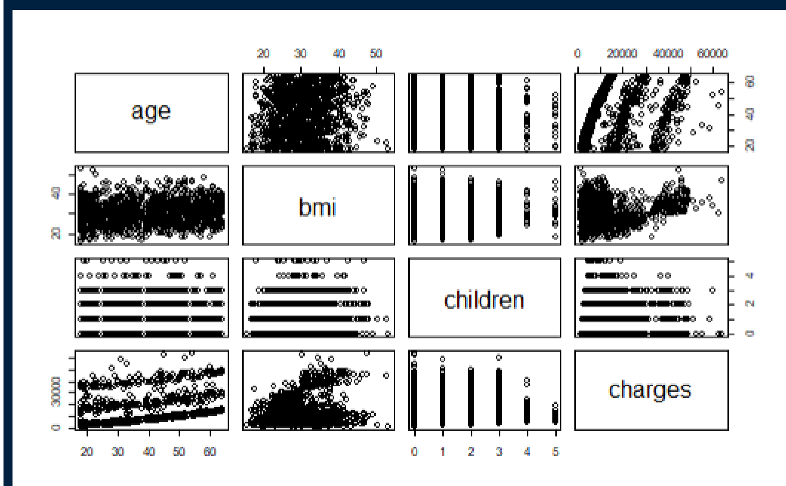
Continuous Variables

```{r}
cor(Insurance[ , c(1,3,4,7)])
```

```
                age       bmi   children    charges
age       1.0000000 0.1092719 0.04246900 0.29900819
bmi       0.1092719 1.0000000 0.01275890 0.19834097
children  0.0424690 0.0127589 1.00000000 0.06799823
charges   0.2990082 0.1983410 0.06799823 1.00000000
```

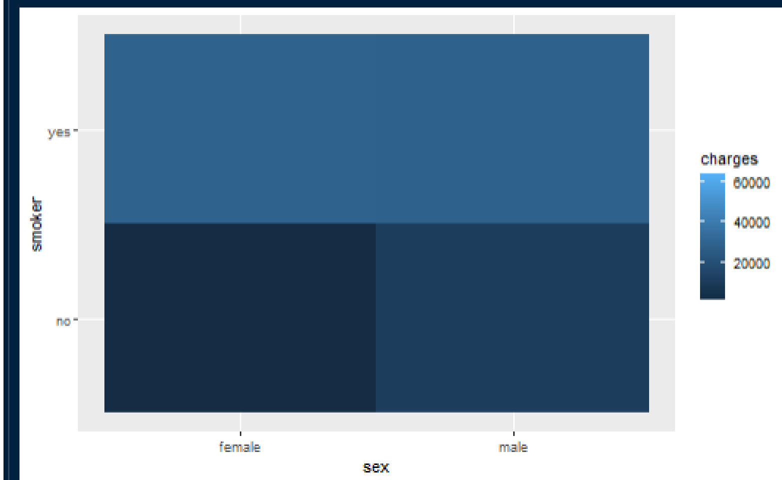
To find the correlation coefficient for each of my continuous variables, I used the cor command. The results showed that no two variables have a meaningful association. To provide a visual depiction, I also plotted the variables using the pairs command. The pairs command verified the output from the cor command.

```{r}
pairs(Insurance[ , c(1,3,4,7) ])
```



I used the pairs () function to create a scatter plot matrix for the numerical variables.

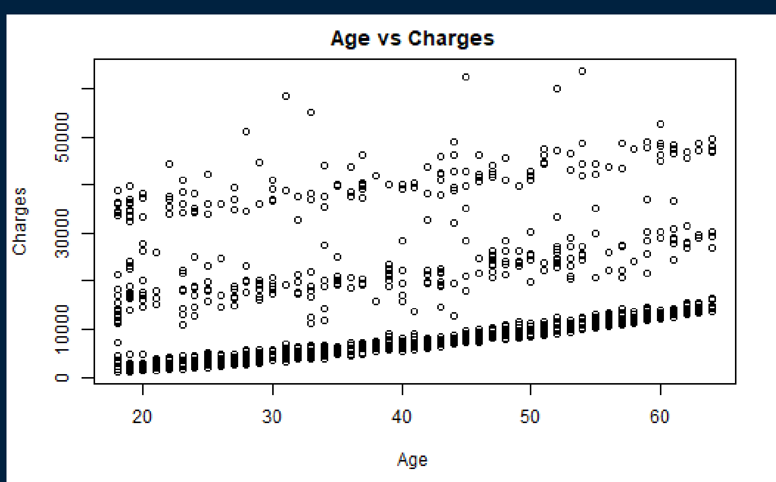
```{r}
count.table <- table(Insurance$sex, Insurance$smoker)
ggplot(data = Insurance, aes(x = sex, y = smoker, fill = charges)) + geom_tile()
```



We can see from the heatmap that smokers, whether they are male or female, typically pay more for insurance than nonsmokers do. Additionally, females appear to pay less for insurance than males do for nonsmokers. This heatmap, which only applies to nonsmokers, supports the findings from the sex vs. charges barplot.

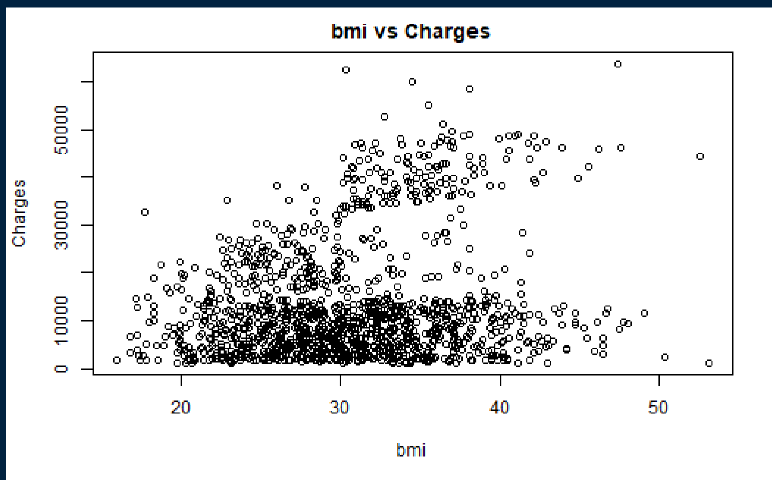
Age vs. Charges

```{r}
plot(Insurance$age, Insurance$charges, main = "Age vs Charges", xlab = "Age", ylab =
"Charges")

```



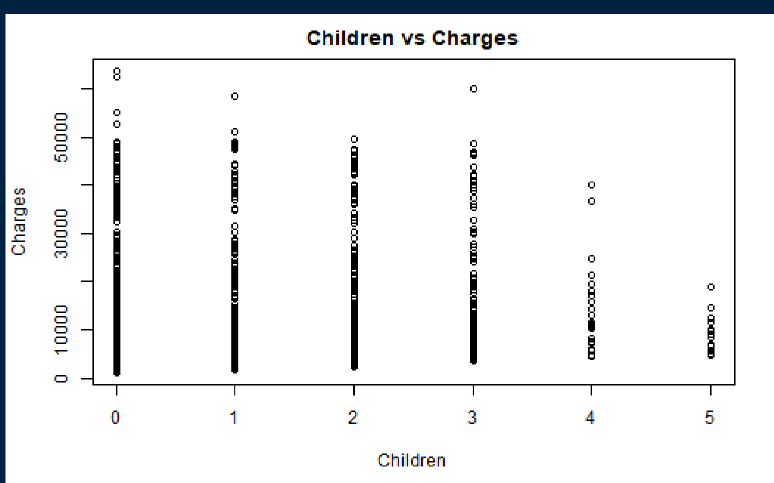
bmi vs Charges

```{r}
plot(Insurance$bmi, Insurance$charges, main = "bmi vs Charges", xlab = "bmi", ylab =
"Charges")
```



Children vs Charges

```{r}
plot(Insurance$children, Insurance$charges, main = "Children vs Charges", xlab = "Children",
ylab = "Charges")
```

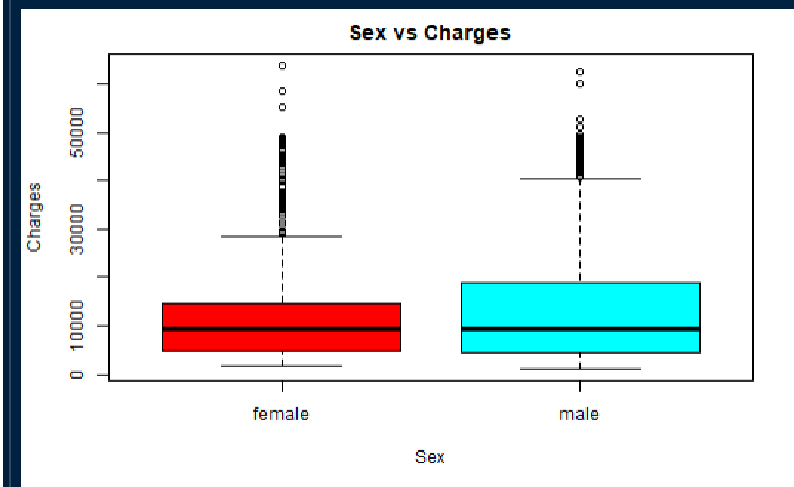


This graph demonstrates that when the number of dependents increases to four or more, charges tend to decrease.

Continuous Variable and Categorical Variables

Sex vs. Charges

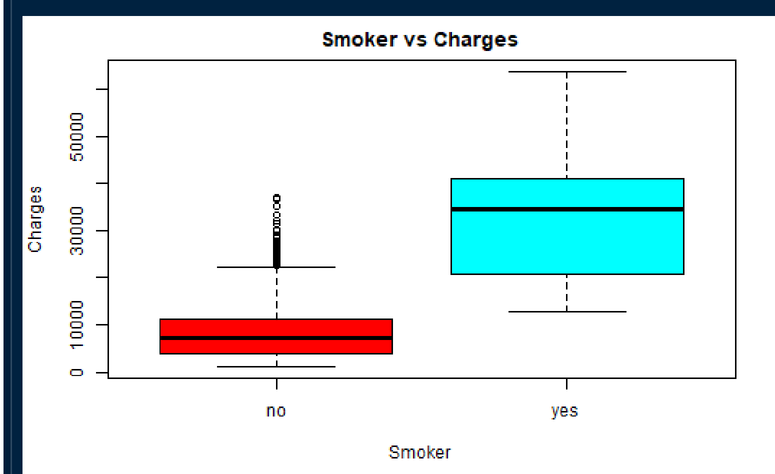
````
```{r}
plot(Insurance$sex, Insurance$charges, main = "Sex vs Charges", xlab= "Sex",
ylab= "Charges",col = rainbow(nlevels(Insurance$sex)) )
```
````



This graphic demonstrates that men typically have higher insurance than women do. To validate it, we will need to check into it more thoroughly.

Smoker vs. Charges

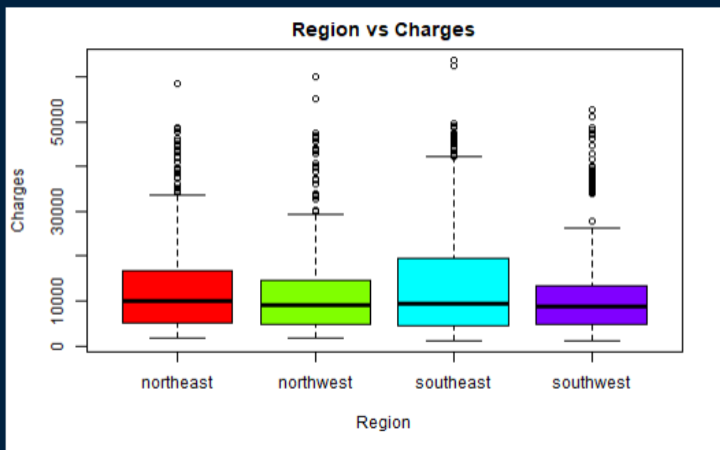
````
```{r}
plot(Insurance$smoker, Insurance$charges, main = "Smoker vs Charges",
xlab= "Smoker", ylab= "Charges",col = rainbow(nlevels(Insurance$smoker)) )

```
````



Smokers pay much more for insurance than non-smokers, as would be expected. This seems reasonable, given that smoking can cause various major health problems

Region vs. Charges

```{r}
plot(Insurance$region, Insurance$charges, main = "Region vs Charges", xlab=
"Region", ylab= "Charges",col = rainbow(nlevels(Insurance$region)) )
```



The insurance payment methods are roughly equivalent. However, it appears that southeasters are charged more than persons from other parts of the country.

Models

1. Multiple Linear Regression

I will begin by using all variables in multiple linear regression before determining which ones are statistically significant based on their corresponding p-values.

Model 1

```{r}
insurance.lm <- lm(charges~., data = Insurance)
summary(insurance.lm)
```

```
Call:
lm(formula = charges ~ ., data = Insurance)

Residuals:
     Min       1Q   Median       3Q      Max
-11304.9  -2848.1   -982.1   1393.9  29992.8

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     -11938.5      987.8 -12.086  < 2e-16 ***
age                256.9       11.9  21.587  < 2e-16 ***
sexmale           -131.3      332.9  -0.394 0.693348
bmi                339.2       28.6  11.860  < 2e-16 ***
children           475.5      137.8   3.451 0.000577 ***
smokeryes        23848.5      413.1  57.723  < 2e-16 ***
regionnorthwest   -353.0      476.3  -0.741 0.458769
regionsoutheast  -1035.0      478.7  -2.162 0.030782 *
regionsouthwest   -960.0      477.9  -2.009 0.044765 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6062 on 1329 degrees of freedom
Multiple R-squared:  0.7509,    Adjusted R-squared:  0.7494
F-statistic: 500.8 on 8 and 1329 DF,  p-value: < 2.2e-16
```

Age, BMI, children, and smokers are statistically significant factors according to this model. We will thus solely use those variables to fit another model. Despite having p-values below 0.05, the regions in the southeast and southwest will not be used at this time.

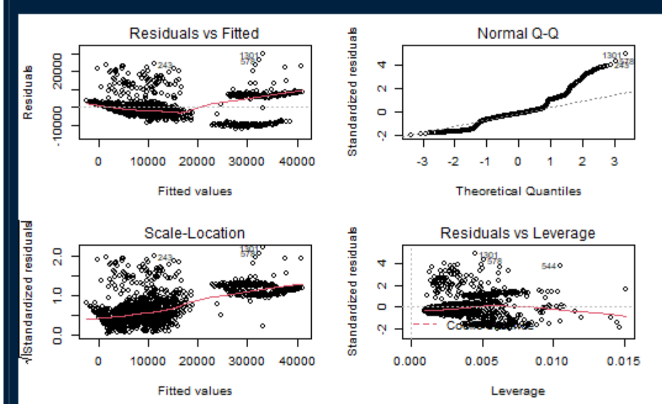
Model 2

```{r}
insurance.lm <- lm(charges~.-sex-region, data = Insurance)
summary(insurance.lm)
```

```
Call:
lm(formula = charges ~ . - sex - region, data = Insurance)

Residuals:
     Min       1Q   Median       3Q      Max 
-11897.9  -2920.8   -986.6   1392.2  29509.6 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -12102.77     941.98 -12.848  < 2e-16 ***
age            257.85      11.90  21.675  < 2e-16 ***
bmi            321.85      27.38  11.756  < 2e-16 ***
children       473.50     137.79   3.436 0.000608 ***
smokeryes    23811.40     411.22  57.904  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6068 on 1333 degrees of freedom
Multiple R-squared:  0.7497,	Adjusted R-squared:  0.7489 
F-statistic: 998.1 on 4 and 1333 DF,  p-value: < 2.2e-16
```

Despite having a slightly lower R-squared than the previous model, this model reveals that all the variables included are statistically significant.

```{r}
par(mfrow=c(2,2))
plot(insurance.lm)
```




```{r}
insurance.lm1 <- lm(charges~.-sex-region + I(bmi^2) + I(age^2) + I(children^2), data =
Insurance)
summary(insurance.lm1)
```

The residual vs. fitted graphic demonstrates a pattern that supports the data's nonlinearity. The residual vs. leverage plot and the scale-location plot show evidence of high high-leverages and outliers, respectively.

To see if the model can be enhanced, we will then alter our variables.

Model 3

```{r}
insurance.lm1 <- lm(charges~.-sex-region + I(bmi^2) + I(age^2) + I(children^2), data =
Insurance)
summary(insurance.lm1)
```

```
Call:
lm(formula = charges ~ . - sex - region + I(bmi^2) + I(age^2) +
    I(children^2), data = Insurance)

Residuals:
   Min     1Q Median     3Q    Max
-10551  -3114  -1196   1702  30359

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   -13518.329   3498.607  -3.864 0.000117 ***
age              -87.357     82.479  -1.059 0.289726
bmi              792.804    206.940   3.831 0.000134 ***
children        1272.677    371.985   3.421 0.000642 ***
smokeryes      23813.533    408.529  58.291  < 2e-16 ***
I(bmi^2)          -7.542      3.251  -2.320 0.020496 *
I(age^2)           4.322      1.028   4.204  2.8e-05 ***
I(children^2)   -185.366    100.799  -1.839 0.066142 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6021 on 1330 degrees of freedom
Multiple R-squared:  0.7541,	Adjusted R-squared:  0.7528
F-statistic: 582.7 on 7 and 1330 DF,  p-value: < 2.2e-16
```

This transformation produced the highest R-squared and adjusted R-squared value after experimenting with other transformations. However, I(children2)'s p-value is higher than 0.05, indicating that this variable is not statistically significant. Therefore, we shall fit a model without it.

Model 4

```{r}
insurance.lm1 <- lm(charges~.-sex-region + I(bmi^2) + I(age^2) , data = Insurance)
summary(insurance.lm1)
```

```
Call:
lm(formula = charges ~ . - sex - region + I(bmi^2) + I(age^2),
    data = Insurance)

Residuals:
   Min     1Q Median     3Q    Max
-10532  -3085  -1211   1671  30071

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -13808.067   3498.184  -3.947 8.32e-05 ***
age            -57.539     80.942  -0.711 0.477289
bmi            788.095    207.109   3.805 0.000148 ***
children       641.361    143.373   4.473 8.36e-06 ***
smokeryes    23845.198    408.531  58.368  < 2e-16 ***
I(bmi^2)        -7.449      3.253  -2.289 0.022210 *
I(age^2)         3.957      1.010   3.920 9.32e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6026 on 1331 degrees of freedom
Multiple R-squared:  0.7535,	Adjusted R-squared:  0.7524
F-statistic:   678 on 6 and 1331 DF,  p-value: < 2.2e-16
```

Although this model has a lower R-squared, we will keep it for now and investigate other models to improve how well they fit our dataset.

2. Best Subset Regression

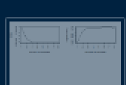
```{r}
regfit.full <- regsubsets(charges~., Insurance)
summary(regfit.full)
```

```
Subset selection object
Call: regsubsets.formula(charges ~ ., Insurance)
8 Variables  (and intercept)
                Forced in Forced out
age                 FALSE      FALSE
sexmale             FALSE      FALSE
bmi                 FALSE      FALSE
children            FALSE      FALSE
smokeryes           FALSE      FALSE
regionnorthwest     FALSE      FALSE
regionsoutheast     FALSE      FALSE
regionsouthwest     FALSE      FALSE
1 subsets of each size up to 8
Selection Algorithm: exhaustive
         age sexmale bmi children smokeryes regionnorthwest regionsoutheast regionsouthwest
1  ( 1 ) " " " "     " " " "      "*"       " "             " "             " "
2  ( 1 ) "*" " "     " " " "      "*"       " "             " "             " "
3  ( 1 ) "*" " "     "*" " "      "*"       " "             " "             " "
4  ( 1 ) "*" " "     "*" "*"      "*"       " "             " "             " "
5  ( 1 ) "*" " "     "*" "*"      "*"       " "             "*"             " "
6  ( 1 ) "*" " "     "*" "*"      "*"       " "             "*"             "*"
7  ( 1 ) "*" " "     "*" "*"      "*"       "*"             "*"             "*"
8  ( 1 ) "*" "*"     "*" "*"      "*"       "*"             "*"             "*"
```

```{r}
reg.summary <- summary(regfit.full)
reg.summary$rsq

```

```
[1] 0.6197648 0.7214008 0.7474772 0.7496945 0.7501113 0.7507814 0.7508839 0.7509130
```

```{r}
par(mfrow = c(2,2))
plot(reg.summary$rss ,xlab="Number of Variables ",ylab="RSS", type="l")
plot(reg.summary$adjr2 ,xlab="Number of Variables ", ylab="Adjusted RSq",type="l")
which.max(reg.summary$adjr2)
```



R Console

```
[1] 6
```

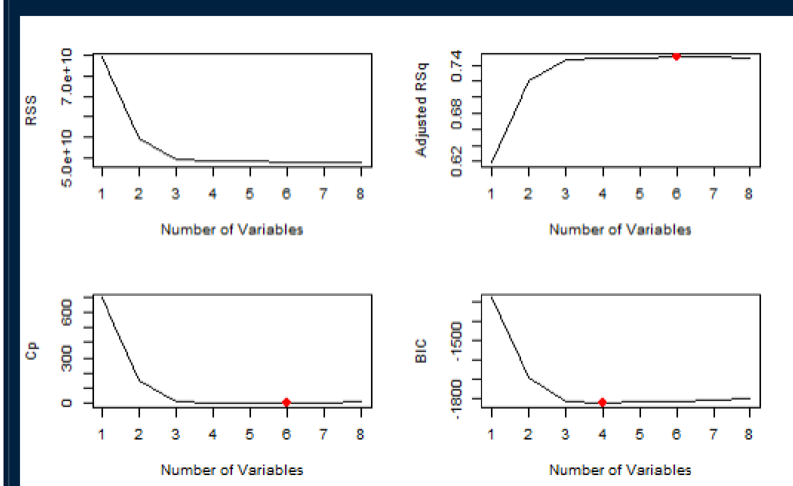
```{r}
par(mfrow = c(2,2))
plot.new()
points(6, reg.summary$adjr2[6], col = "red", cex = 2, pch = 20)
plot(reg.summary$cp, xlab = "Number of Variables", ylab = "Cp", type = "l")
which.min(reg.summary$cp)
```

```
[1] 6
```

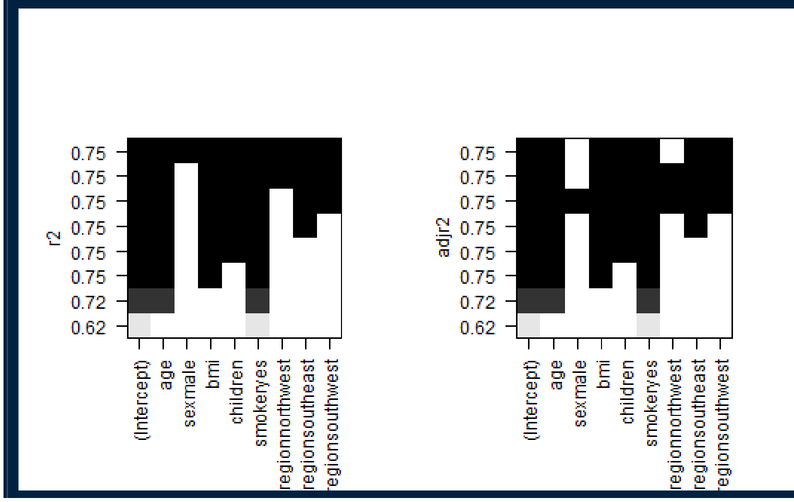
```{r}
par(mfrow = c(2,2))
plot.new()
points(6,reg.summary$cp [6],col="red",cex=2,pch=20)
plot(reg.summary$bic ,xlab="Number of Variables ",ylab="BIC", type="l")
which.min(reg.summary$bic )
```

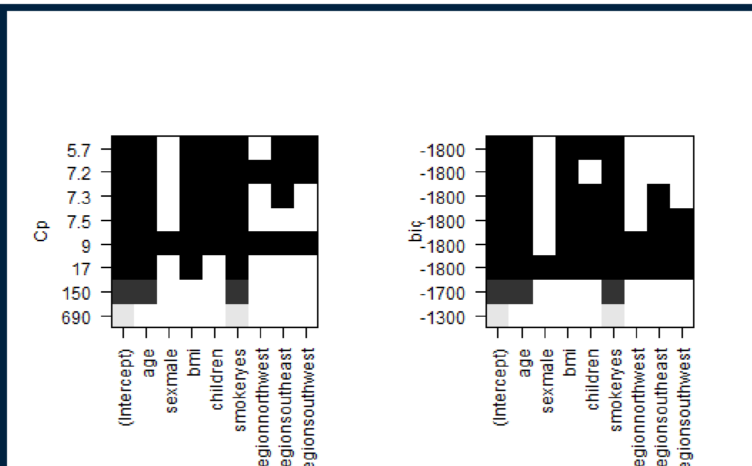
```
[1] 4
```

```{r}
par(mfrow = c(2,2))

# Plot RSS versus Number of Variables
plot(reg.summary$rss, xlab="Number of Variables", ylab="RSS", type="l")
# Plot Adjusted R-squared versus Number of Variables
plot(reg.summary$adjr2, xlab="Number of Variables", ylab="Adjusted RSq", type="l")
points(6, reg.summary$adjr2[6], col="red", cex=2, pch=20)
# Plot Cp versus Number of Variables
plot(reg.summary$cp, xlab="Number of Variables", ylab="Cp", type="l")
# Add a red point at position 6
points(6, reg.summary$cp[6], col="red", cex=2, pch=20)

# Plot BIC versus Number of Variables
plot(reg.summary$bic, xlab="Number of Variables", ylab="BIC", type="l")
# Add a red point at position 6
points(4, reg.summary$bic[4], col="red", cex=2, pch=20)
```



```{r}
par(mfrow = c(1,2))
plot(regfit.full,scale="r2")
plot(regfit.full,scale="adjr2")
plot(regfit.full,scale="Cp")
plot(regfit.full,scale="bic")
```





After considering all the available information, I think that the most appropriate model comprises of four variables, which include age, BMI, number of children, and smoking status.

3. Forward and Backward Stepwise Selection

```{r}
regfit.fwd<- regsubsets(charges~., data = Insurance, method = "forward")
summary(regfit.fwd)
```

```
Subset selection object
Call: regsubsets.formula(charges ~ ., data = Insurance, method = "forward")
8 Variables  (and intercept)
                Forced in Forced out
age                 FALSE      FALSE
sexmale             FALSE      FALSE
bmi                 FALSE      FALSE
children            FALSE      FALSE
smokeryes           FALSE      FALSE
regionnorthwest     FALSE      FALSE
regionsoutheast     FALSE      FALSE
regionsouthwest     FALSE      FALSE
1 subsets of each size up to 8
Selection Algorithm: forward
         age sexmale bmi children smokeryes regionnorthwest regionsoutheast regionsouthwest
1  ( 1 ) " " " "     " " " "      "*"       " "             " "             " "
2  ( 1 ) "*" " "     " " " "      "*"       " "             " "             " "
3  ( 1 ) "*" " "     "*" " "      "*"       " "             " "             " "
4  ( 1 ) "*" " "     "*" "*"      "*"       " "             " "             " "
5  ( 1 ) "*" " "     "*" "*"      "*"       " "             "*"             " "
6  ( 1 ) "*" " "     "*" "*"      "*"       " "             "*"             "*"
7  ( 1 ) "*" " "     "*" "*"      "*"       "*"             "*"             "*"
8  ( 1 ) "*" "*"     "*" "*"      "*"       "*"             "*"             "*"
```

```{r}
regfit.bwd <- regsubsets(charges~., data = Insurance, method = "backward")
summary(regfit.bwd)
```

```
Subset selection object
Call: regsubsets.formula(charges ~ ., data = Insurance, method = "backward")
8 Variables  (and intercept)
                Forced in Forced out
age                 FALSE      FALSE
sexmale             FALSE      FALSE
bmi                 FALSE      FALSE
children            FALSE      FALSE
smokeryes           FALSE      FALSE
regionnorthwest     FALSE      FALSE
regionsoutheast     FALSE      FALSE
regionsouthwest     FALSE      FALSE
1 subsets of each size up to 8
Selection Algorithm: backward
         age sexmale bmi children smokeryes regionnorthwest regionsoutheast regionsouthwest
1  ( 1 ) " " " "     " " " "      "*"       " "             " "             " "
2  ( 1 ) "*" " "     " " " "      "*"       " "             " "             " "
3  ( 1 ) "*" " "     "*" " "      "*"       " "             " "             " "
4  ( 1 ) "*" " "     "*" "*"      "*"       " "             " "             " "
5  ( 1 ) "*" " "     "*" "*"      "*"       " "             "*"             " "
6  ( 1 ) "*" " "     "*" "*"      "*"       " "             "*"             "*"
7  ( 1 ) "*" " "     "*" "*"      "*"       "*"             "*"             "*"
8  ( 1 ) "*" "*"     "*" "*"      "*"       "*"             "*"             "*"
```

Both forward and backward stepwise selection methods yielded identical variable selection for each model.

Validation Set Approach

```r
```{r}
set.seed(1)
train <- sample(c(TRUE, FALSE), nrow(Insurance), rep = TRUE)
test<- (!train)
regfit.best <- regsubsets(charges~., data = Insurance[train, ], )
test.mat <- model.matrix(charges~., data = Insurance[test, ])
val.errors=rep(NA,8)
for(i in 1:8){
 coefi <- coef(regfit.best, id=i)
 pred<- test.mat[ , names(coefi)]%*%coefi
 val.errors[i]=mean((Insurance$charges[test]-pred)^2)
 }
val.errors

```

[1] 60092451 43450714 38869494 38860514 38853079 38426858 38471229 38489704
```

```r
```{r}
which.min(val.errors)
```

[1] 6
```

```r
```{r}
coef(regfit.best, 6)
```

    (Intercept)   age   bmi   children   smokeryes regionsoutheast
regionsouthwest
    -10490.6818   248.8687   305.4671   275.0695   23132.9314   -1183.8361
-980.2071
```

The model with six variables is the best one, according to the validation set approach. Since these six variables might not be the same as the ones picked for the training batch, I will use the entire model to identify them.

```r
```{r}
regfit.best <- regsubsets(charges~., data = Insurance)
coef(regfit.best, 6)
```

    (Intercept)   age   bmi   children   smokeryes regionsoutheast
regionsouthwest
    -12165.3824   257.0064   338.6413   471.5441   23843.8749   -858.4696
-782.7452
```

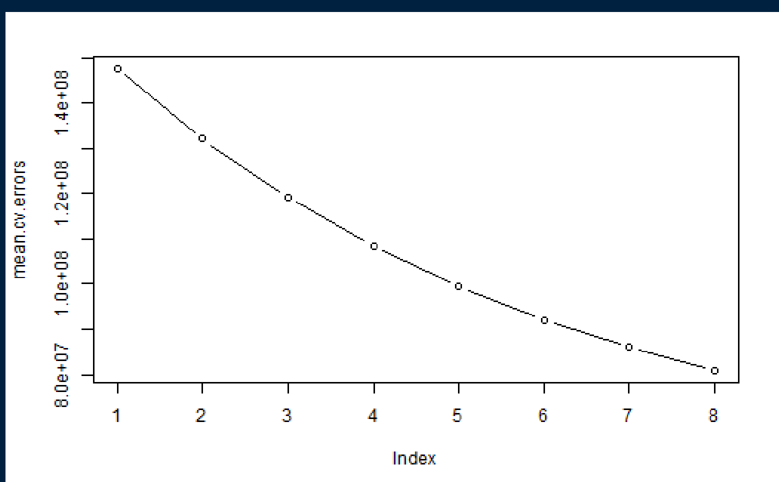
In this instance, the training set and the entire dataset both chose the six previously indicated variables.

Cross Validation

```{r}
set.seed(1)
folds <- sample(1:k, nrow(Insurance), replace = TRUE)
cv.errors <- matrix(NA, k, 8, dimnames = list(NULL, paste(1:8)))

for(j in 1:k) {
  # Convert categorical variables to numeric format
  x_train <- model.matrix(charges ~ ., data = Insurance[folds != j, ])

  # Fit cross-validated elastic net model
  cv.fit <- cv.glmnet(x = x_train, y = Insurance$charges[folds != j], alpha = 1)

  for(i in 1:8) {
    # Predict on test fold and calculate CV error
    x_test <- model.matrix(charges ~ ., data = Insurance[folds == j, ])
    pred <- predict(cv.fit, newx = x_test, s = cv.fit$lambda[i], type = "response")
    cv.errors[j, i] = mean((Insurance$charges[folds == j] - pred)^2)
  }
}

mean.cv.errors <- apply(cv.errors, 2, mean)
mean.cv.errors
```

```
        1         2         3         4         5         6         7         8 
147632577 132079572 119164511 108439756  99533649  92137614  85995446  80894426 
```

```{r}
par(mfrow=c(1,1))
plot(mean.cv.errors ,type="b")
```




```{r}
reg.best <- regsubsets(charges~., data = Insurance)
coef(reg.best,4)
```

```
(Intercept)         age         bmi    children   smokeryes 
-12102.7694    257.8495    321.8514    473.5023  23811.3998 
```

The plot indicates that the cross-validation method resulted in a four-variable model. To obtain these four variables, we can apply the best subset selection technique to the full model.

4. Ridge Regression

```{r}
set.seed(1)
x <- model.matrix(charges~., Insurance)[,-1]
y <- Insurance$charges
grid <- 10^seq(10, -2, length=100)
train<- sample(1:nrow(x), nrow(x)/1.3)
test<- (-train)
y.test <- y[test]
cv.out <- cv.glmnet(x[train, ], y[train], alpha = 0)
best.lam <- cv.out$lambda.min
glm.mod <- glmnet(x[train, ], y[train], alpha = 0, lambda = grid, thresh = 1e-12)
glm.pred <- predict(glm.mod, s=best.lam, newx = x[test, ])
mean((glm.pred - y.test)^2)
```

```
[1] 47024809
```

```{r}
glm.coef <- predict(glm.mod, type="coefficients", s=best.lam)[1:9, ]
glm.coef
```

```
    (Intercept)             age         sexmale             bmi        children       smokeryes 
regionnorthwest regionsoutheast regionsouthwest 
    -10235.7545        250.1924        139.2664        291.0564        432.4727      21633.0029 
-307.2834        -654.8357        -656.0909
```

Ridge regression produced the optimal model that includes all variables, which is unsurprising.

5. Lasso Regression

```{r}
set.seed(1)
cv.out <- cv.glmnet(x[train, ], y[train], alpha = 1)
best.lam <- cv.out$lambda.min
lasso.mod <- glmnet(x[train, ], y[train], alpha = 1, lambda = grid)
lasso.pred<- predict(lasso.mod, s=best.lam,newx = x[test, ] )
mean((lasso.pred - y.test)^2)
```

```
[1] 45671404
```

```{r}
lasso.coef <- predict(lasso.mod, type="coefficients", s=best.lam)[1:9, ]
lasso.coef
```

```
    (Intercept)             age         sexmale             bmi        children       smokeryes 
regionnorthwest regionsoutheast regionsouthwest 
  -11651.568001      266.630591        0.000000      303.235459      394.635541    23245.603786 
-1.513378       -510.147078       -371.040819
```

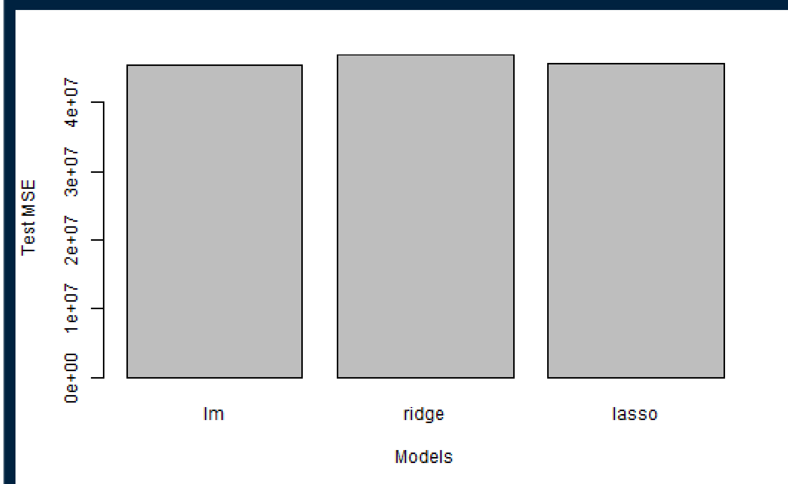
Lasso regression resulted in a lower test MSE than ridge regression. Additionally, the lasso model has one less variable (specifically, 'sex male') than the ridge model.

6. Least Square Regression

```r
```{r}
train.df <- data.frame(Insurance[train, ])
test.df <- data.frame(Insurance[test, ])
lm.fit <- lm(charges~., data=train.df)
lm.pred <- predict(lm.fit, test.df, type = c("response"))
mean((lm.pred - test.df$charges)^2)
```
```

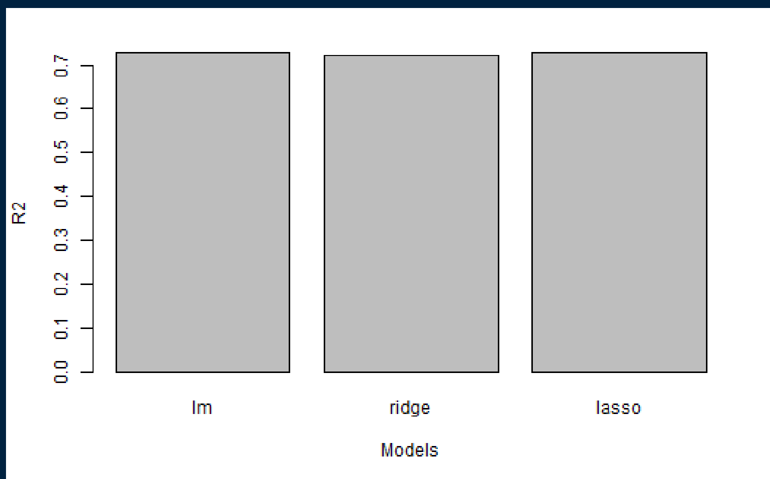
```
[1] 45478363
```

```r
```{r}
err.lm <- mean((lm.pred - test.df$charges)^2)
err.ridge <- mean((glm.pred - y.test)^2)
err.lasso <- mean((lasso.pred - y.test)^2)
err.all <- c(err.lm, err.ridge, err.lasso)
barplot(err.all, xlab = "Models", ylab = "Test MSE", names= c("lm", "ridge", "lasso"))
```
```



When I examine all three data sets, I can see that they all produce a similar test MSE ("mean squared error"). However, the lasso and least square sets produce a slightly lower test MSE, and the least square set produces the smallest value.

```{r}
test.avg <- mean(y.test)
lm.r2 = 1 - mean((lm.pred - y.test)^2) / mean((test.avg - y.test)^2)
ridge.r2 = 1 - mean((glm.pred - y.test)^2) / mean((test.avg - y.test)^2)
lasso.r2 = 1 - mean((lasso.pred - y.test)^2) / mean((test.avg - y.test)^2)
barplot(c(lm.r2, ridge.r2, lasso.r2), xlab="Models", ylab="R2",names=c("lm", "ridge",
"lasso"))
```



R-squared values for all three models are greater than 0.7, however, the ridge model has the lowest value. In general, I am confident in the projections' accuracy.

7. Qualitative Analysis Using BMI as a Categorical Variable

In this section, I'll test my ability to predict outcomes using the body mass index (BMI) as a categorical variable. I'll classify someone as "obese" if their BMI is 30 or higher, and I'll give them a BMI score of 1 for being obese and 0 for not being. The data set will then be subjected to four different analyses (LDA, QDA, Logistic Regression, and KNN) to determine how well each method predicts outcomes and to determine the test error.

```{r}
Insurance$bmi1 <- ifelse(Insurance$bmi > 30, 1,0)
set.seed(1)
subset <- sample(nrow(Insurance), nrow(Insurance)*0.7)
datatrain <- Insurance[subset,]
datatest <- Insurance[-subset, ]
dim(datatest)
```

```
[1] 402   8
```

```{r}
dim(datatrain)
```

```
[1] 936   8
```

a. Linear Discriminant Analysis

```{r}
attach(Insurance)
lda.fit <- lda(bmi1~., data = datatrain)
lda.predict <- predict(lda.fit, datatest)
predictions <- lda.predict$class
actual <- datatest$bmi1
table(predictions, actual)
```

```
           actual
predictions   0   1
          0 181  13
          1   3 205
```

(13+3)/(402) = 0.0398 is the test error.

b. Quadratic Discriminant Analysis

```{r}
qda.fit <- qda(bmi1~., data = datatrain)
qda.predict <- predict(qda.fit, datatest)
predictions <- qda.predict$class
table(predictions, actual)
```

```
           actual
predictions   0   1
          0 176  14
          1   8 204
```

(14+8)/402 = 0.05547 is the test error.

c. Logistic Regression

```{r}
logistic.fit <- glm(bmi1~., data = datatrain, family = binomial)
logistic.probs <- predict(logistic.fit, datatest, type = "response")
logistic.pred <- rep(0, length(datatest$bmi1))
logistic.pred[logistic.probs>0.5]=1
table(logistic.pred, actual)
```

```
Warning: glm.fit: algorithm did not convergeWarning: glm.fit: fitted probabilities numerically 0
or 1 occurred             actual
logistic.pred   0   1
            0 183   2
            1   1 216
```

(2+1)/402 = 0.00746 is the test error.

d. K Nearest Neighbor

```{r}
library(class)
train.x <- data.matrix(datatrain)
test.x <- data.matrix(datatest)
train.y <- data.matrix(datatrain$bmi1)
test.y <- data.matrix(datatest$bmi1)
knn.predict <- knn(train.x, test.x, train.y, k=1)
table(knn.predict, test.y)
```

```
           test.y
knn.predict   0   1
          0 122  79
          1  62 139
```

(79+62)/402 = 0.35 is the test error for K=1

```{r}
knn.predict2 <- knn(train.x, test.x, train.y, k=5)
table(knn.predict2, test.y)
```

```
            test.y
knn.predict2   0   1
           0 110  91
           1  74 127
```

(91+74)/402 = 0.41is the test error for K=5

```{r}
knn.predict3 <- knn(train.x, test.x, train.y, k=10)
table(knn.predict3, test.y)
```

```
            test.y
knn.predict3   0   1
           0 108  95
           1  76 123
```

(95+76)/402 = 0.425 is the test error for K=10

Given that KNN's test error is significantly larger than that of QDA, LDA, and logistic regression, it doesn't appear to be an appropriate method for this dataset.

The minimal test error (0.00746) is provided by logistic regression compared to the other three classification techniques.

Summary

Implementing various graphical techniques and regression methods facilitated in-depth analysis of the Insurance dataset aimed at uncovering the key factors that impact insurance charges. The primary objective was to assist insurance companies in establishing an appropriate premium price. The analysis revealed that smokers were subject to significantly higher charges than non-smokers. However, an unexpected finding emerged, with non-smoking males incurring higher charges than their female counterparts. Furthermore, the graphical analysis highlighted that those individuals with four or five children had lower charges than those with fewer children.

Multiple linear regression was employed to develop the most accurate predictive model, resulting in an impressive R-squared value of 0.7535 and an adjusted R-squared of 0.7525. To further optimize the model, I utilized the best subset selection, which identified four critical variables, namely age, BMI, children, and smoking status. The validation procedures, including validation set and cross-validation, revealed divergent models with six and four variables, respectively. Moreover, the forward and backward stepwise selection methods yielded the same models, adding further robustness to the findings.

Additionally, I evaluated the dataset using least square, lasso, and ridge regressions. The analysis demonstrated that least square and lasso regression outperformed ridge regression, exhibiting smaller test mean squared error (MSE) and higher R-squared. Finally, four classification algorithms were applied, namely LDA, QDA, logistic regression, and KNN, utilizing BMI as a categorical variable. My findings indicated that logistic regression had the smallest test error (0.00746), with LDA and QDA closely following suit. However, KNN exhibited relatively high test errors, indicating its unsuitability for my predictive model.