**STA 5703 FINAL – Spring 2024**

The dataset utilized for this final exam is referred to as "INSCharge". This dataset comprises 7 variables and consists of 1,338 observations. Among these variables, there exists a numerical target variable identified as "Charges," along with three categorical predictors identified as "Gender," "Smoker," and "Region," and three numerical predictors labeled as "Age," "BMI," and "Children".

The target variable "Charges" displays skewness; however, this is not a substantial concern as our approach involves utilizing tree-based algorithms to construct our model. Additionally, common data issues such as skewness in several numerical predictors and the potential presence of outliers need not overly concern us. These data challenges are unlikely to markedly impact the performance of the fitted models when employing tree-based algorithms. Consequently, we choose to forego the exhaustive data preparation process and proceed directly to the subsequent eight parts to evaluate your comprehension of tree-based regression algorithms and other pertinent topics covered this semester.

1) PART 1: Get Data (0 Points)

   Integrating data into the software system utilized for constructing this model can be accomplished through a variety of methods. Please place your code in Appendix 1 under the section titled "Data Entry."

**Solution:** Appendix 1

2) PART 2: Data Exploration on Categorical Predictors (15 Points)

Exploring data is pivotal in constructing a meaningful model. In this part, you can produce one table for each categorical predictor as follows.

| **Table 1 Exploration - Gender** | | | | |
|---|---|---|---|---|
| **Gender** | **Frequency** | **Average Charges** | **t Value** | **Significance** |
| Female | | | | |
| Male | | | | |
| All | 1,388 | | XXXX | XXXXXX |

Note:

1. t-value is the t-test of the hypothesis such as

$$H_0: \mu_{Female} = \mu_{All}\ H_a: \mu_{Female} \neq \mu_{All}$$

2. Significance is the test be significant or not at $\alpha = 0.05$

Put these tables created here and the code generating these tables in Appendix 2

**Solution:**

1. Gender

| Table 1 Exploration - Gender | | | | |
|---|---|---|---|---|
| **Gender** | **Frequency** | **Average Charges** | **t Value** | **Significance** |
| Female | 662 | 12569.58 | -2.1009 | 0.03584 |
| Male | 676 | 13956.75 | | |
| All | 1,388 | | XXXX | XXXXXX |

2. Smoker

| Table 1 Exploration - Smoker | | | | |
|---|---|---|---|---|
| **Smoker** | **Frequency** | **Average Charges** | **t Value** | **Significance** |
| Yes | 274 | 32050.232 | -32.752 | 2.2e-16 |
| No | 1064 | 8434.268 | | |
| All | 1,388 | | XXXX | XXXXXX |

3. Region

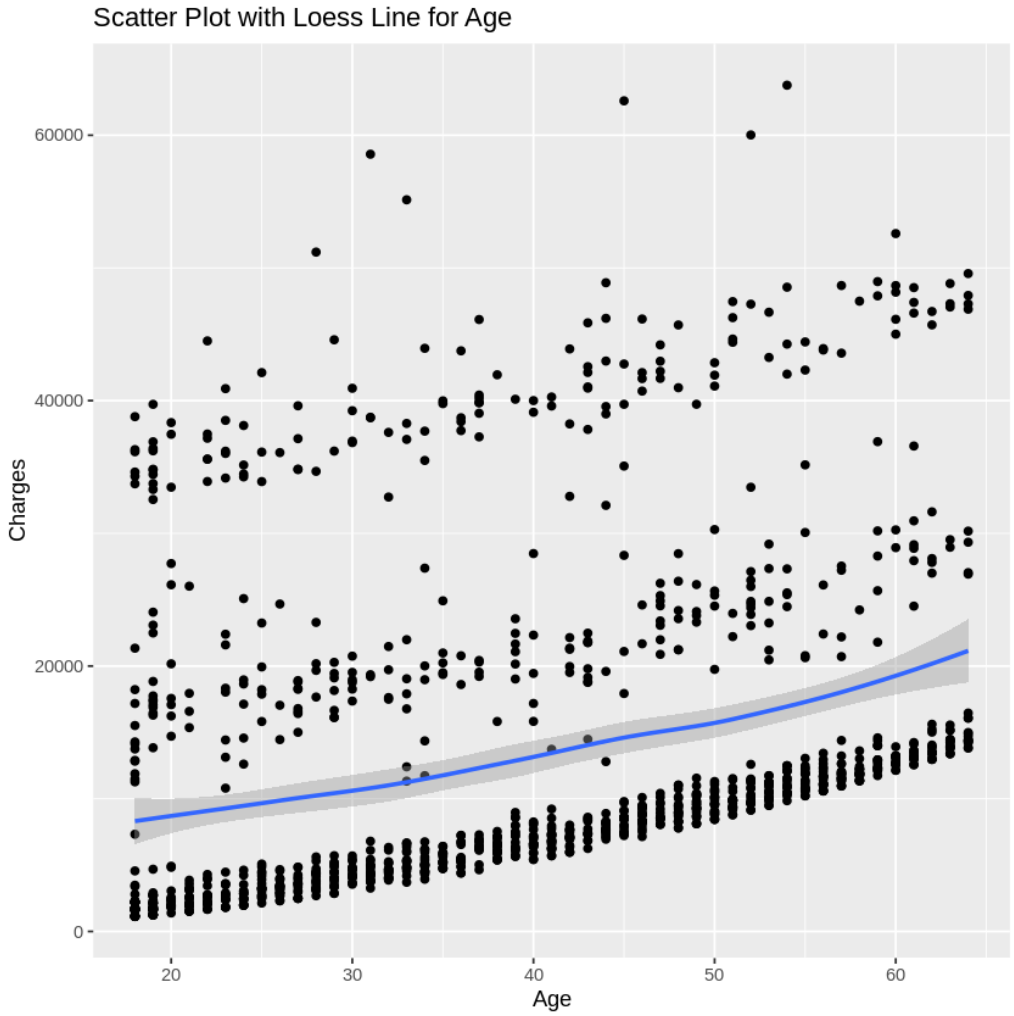
| Table 1 Exploration - Region | | | | |
|---|---|---|---|---|
| **Region** | **Frequency** | **Average Charges** | **F Value** | **Significance** |
| northeast | 324 | 13406.39 | 2.97 | 0.0309 |
| northwest | 325 | 12417.58 | | |
| southeast | 364 | 14735.41 | | |
| southwest | 325 | 12346.94 | | |
| All | 1,388 | | XXXX | XXXXXX |

3) PART 3: Data Exploration on Numerical Predictors (15 Points)

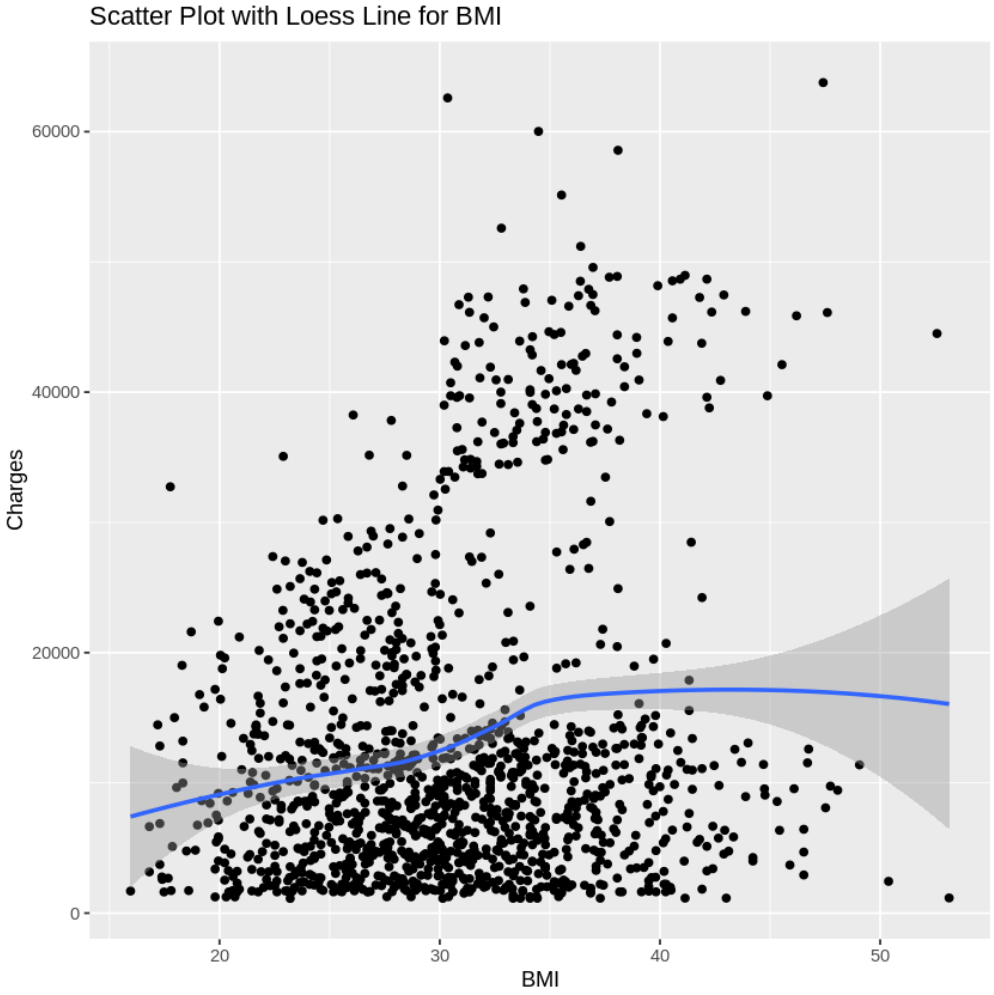
For each numerical predictor, generate a scatter plot with "Charges" on the Y-axis and each numerical predictor on the X-axis, accompanied by a Loess smoothing line overlaid on the plot. Examine each scatter plot to elucidate its relationship with the target variable "Charges". Please provide your scatter plots and detailed explanations here, while placing all code used to generate these plots in Appendix 3.

**Solution:**

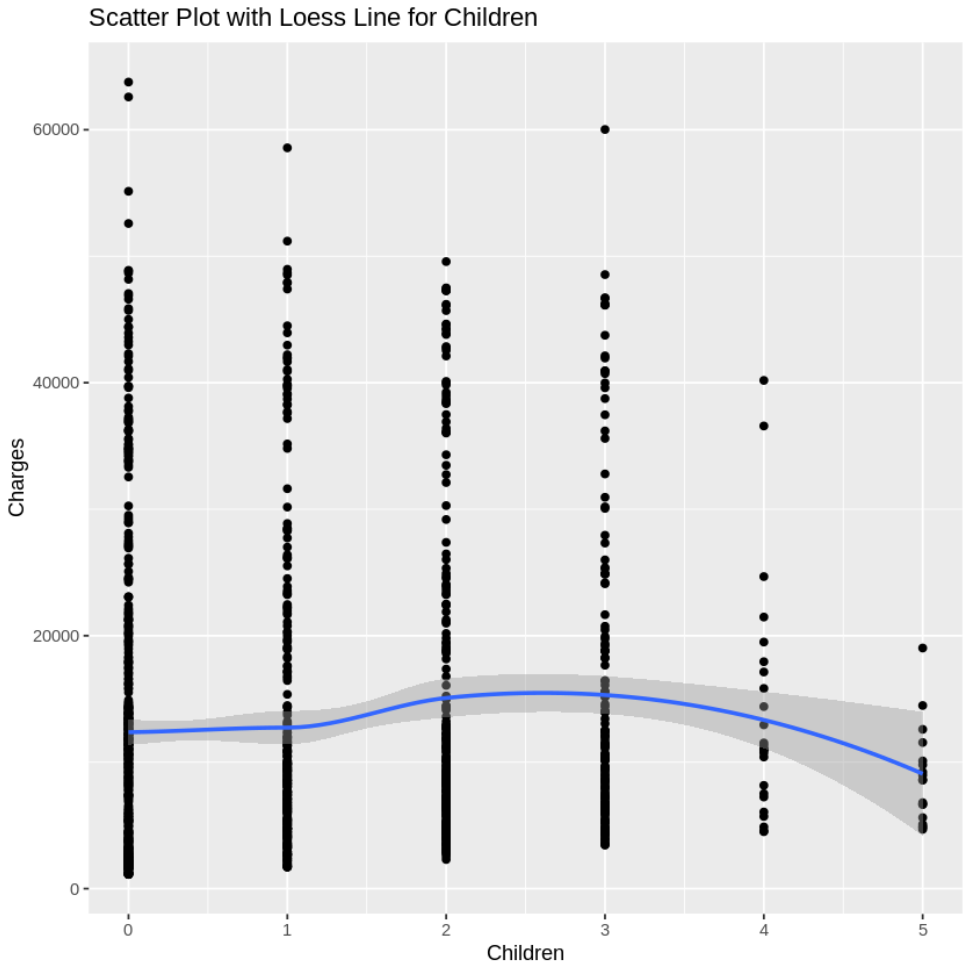
1. Age



2. BMI


Scatter Plot with Loess Line for BMI
60000
40000
20000
0
Charges
20
30
40
50
BMI

3. Children




4) PART 4: Data Splitting – 5-Fold Validation (30 Points)

Given the dataset's relatively modest size of 1,338 cases, we choose a five-fold cross-validation approach for this exam. The complete dataset, denoted as B, is randomly partitioned (random sampling without replacement) into five distinct subsets: $B_1, B_2, B_3, B_4, and\ B_5$, where $B = \bigcup_{i=1}^{5} B_i$. Implementing this part in Python is straightforward, and you can find the code in Appendix 4.

**Solution:** Appendix 4

5) PART 5: Modeling (60 Points)

We adopt a four-piece approach to construct the model, utilizing four out of the five partitions. The remaining partition is reserved for estimating the performance of this model as the following table.

| Model | Training Data | Testing Data |
|---|---|---|
| I | $B_2 \cup B_3 \cup B_4 \cup B_5$ | $B_1$ |
| II | $B_1 \cup B_3 \cup B_4 \cup B_5$ | $B_2$ |

| | | |
|---|---|---|
| III | $B_2 \cup B_1 \cup B_4 \cup B_5$ | $B_3$ |
| IV | $B_2 \cup B_3 \cup B_1 \cup B_5$ | $B_4$ |
| V | $B_2 \cup B_3 \cup B_4 \cup B_1$ | $B_5$ |

This process is repeated five times to build a total of five models and compute five distinct performance metrics for each algorithm used.

Step 5A: Fit "LASSO" Regression Models: Fit five models using "LASSO" regression.

Step 5B: Fit "Random Forest" Models: Fit five random forest models using Python or software of your choice

Step 5C: Fit "Gradient Boosting" Models: Fit five Gradient Boosting models using Python or software of your choice

Step 5D: Fit "XGBoost" Models: Fit five eXtreme Gradient Boosting models using Python or software of your choice

After completing this step, please proceed to fill in the following tables:

**Solution:**

| **Model** | **Training ASE** | | | |
|---|---|---|---|---|
| | **LASSO** | **Random Forest** | **GB** | **XGB** |
| **1** | 36494123 | 15948708 | 22679578 | 17057597 |
| **2** | 36813021 | 16222932 | 22670785 | 17199002 |
| **3** | 36396289 | 14707718 | 22186798 | 16622476 |
| **4** | 37552153 | 16300747 | 24465382 | 19102394 |
| **5** | 35062382 | 15649464 | 21909662 | 15720682 |

| **Model** | **Testing ASE** | | | |
|---|---|---|---|---|
| | **LASSO** | **Random Forest** | **GB** | **XGB** |
| **1** | 37429951 | 26247893 | 23426994 | 24529266 |
| **2** | 35970504 | 27458775 | 25743649 | 23831697 |
| **3** | 37093965 | 26516593 | 25560021 | 24866192 |
| **4** | 32891283 | 22767918 | 18815860 | 14258119 |
| **5** | 42529671 | 31409319 | 26286930 | 26841840 |

Put your code in Appendix 5

6) PART 6: Ensemble (20 Points)

The ultimate model is an ensemble formed by combining these five models, and the performance of this ensemble model is estimated by averaging the five individual performance metrics. For example, the predicted value for the entire data set for each of the five model are as follows: $\hat{y}_{ij}$, where i = 1, 2, 3, 4, 5 and j = 1, 2, …, n. The ensemble model prediction for observation j is $\hat{\hat{y}}_j = \frac{\sum_{i=1}^{5} \hat{y}_{ij}}{5}$, where j= 1, 2, …, n. Ensemble the five models for the training algorithms perform the best in PART 5 and score the ensemble models on the entire data sample.

Put your code in Appendix 6

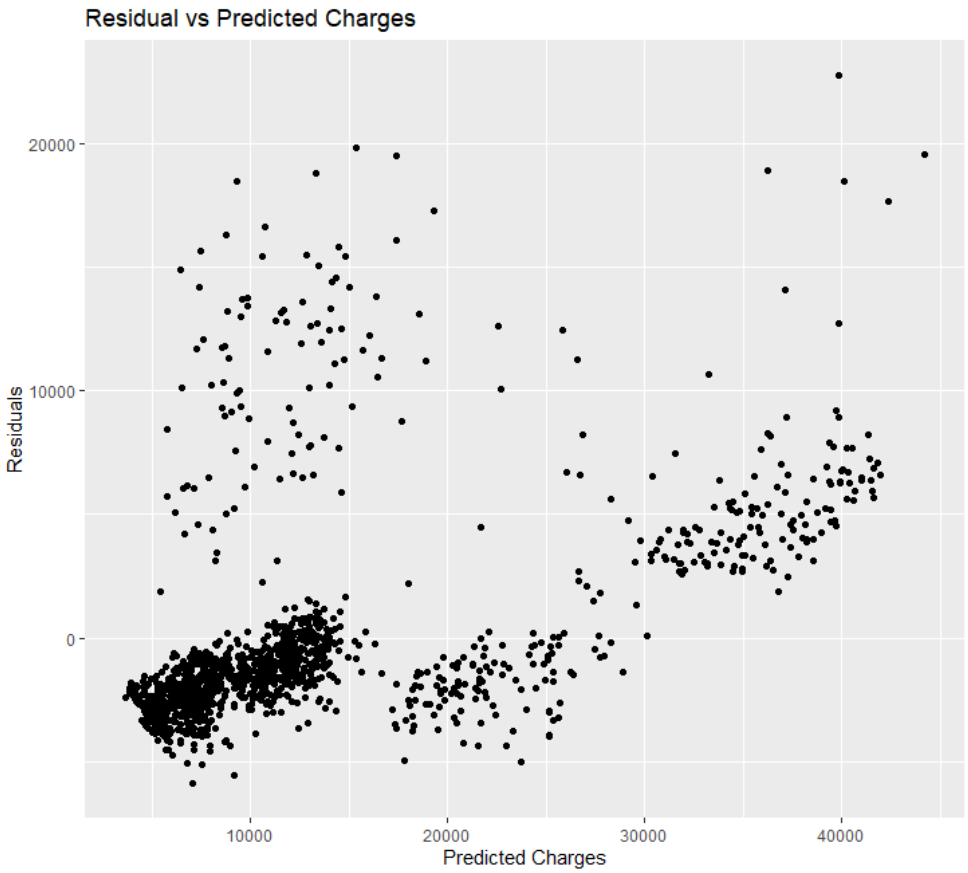
**Solution:**

The algorithm that performs the best in each of the cross-validation training sets was the Random Forest. The **ASE** for the ensemble model in the entire dataset is: **18202044**

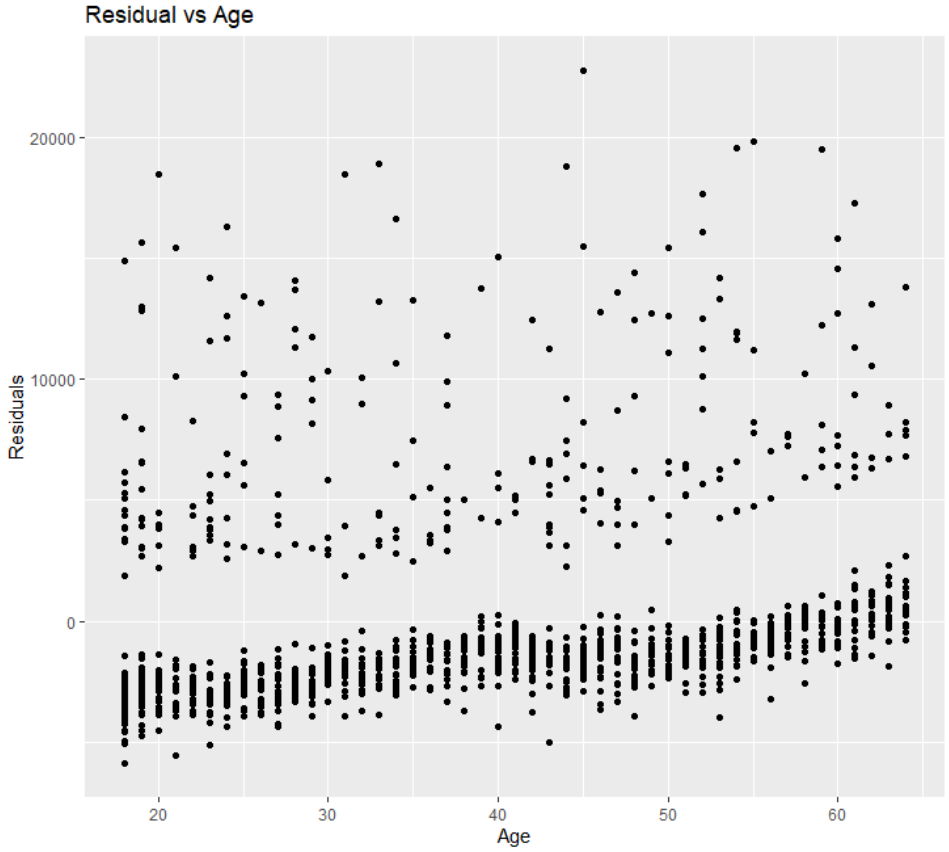
7) PART 7: Residual Plots (40 Points)

Produce residual plots for the ensemble model in PART 6

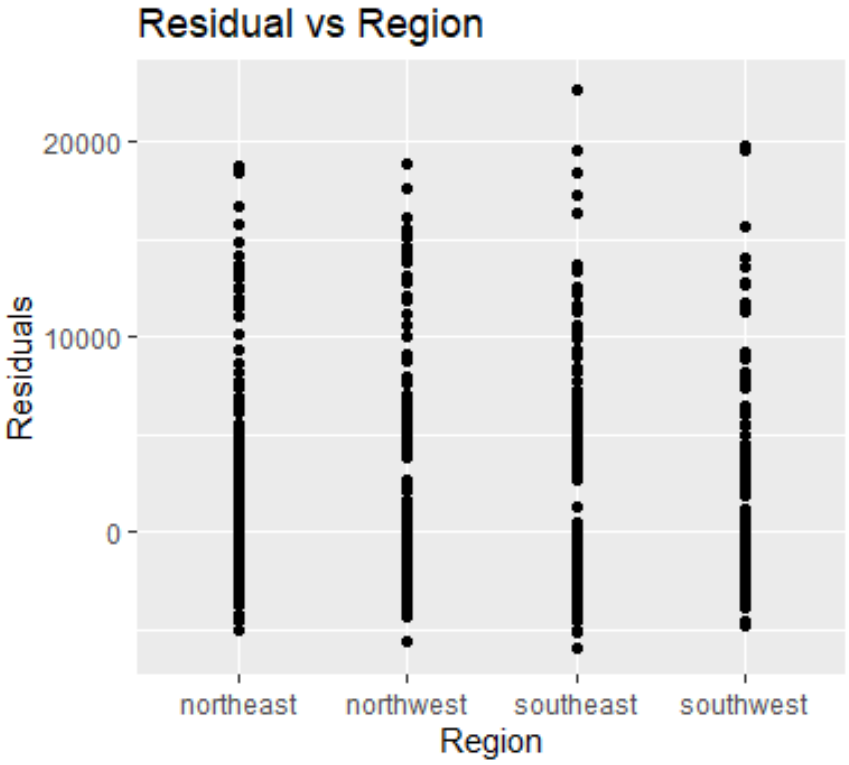
1.Plot #1: Residual against Predicted Value of "Charges"



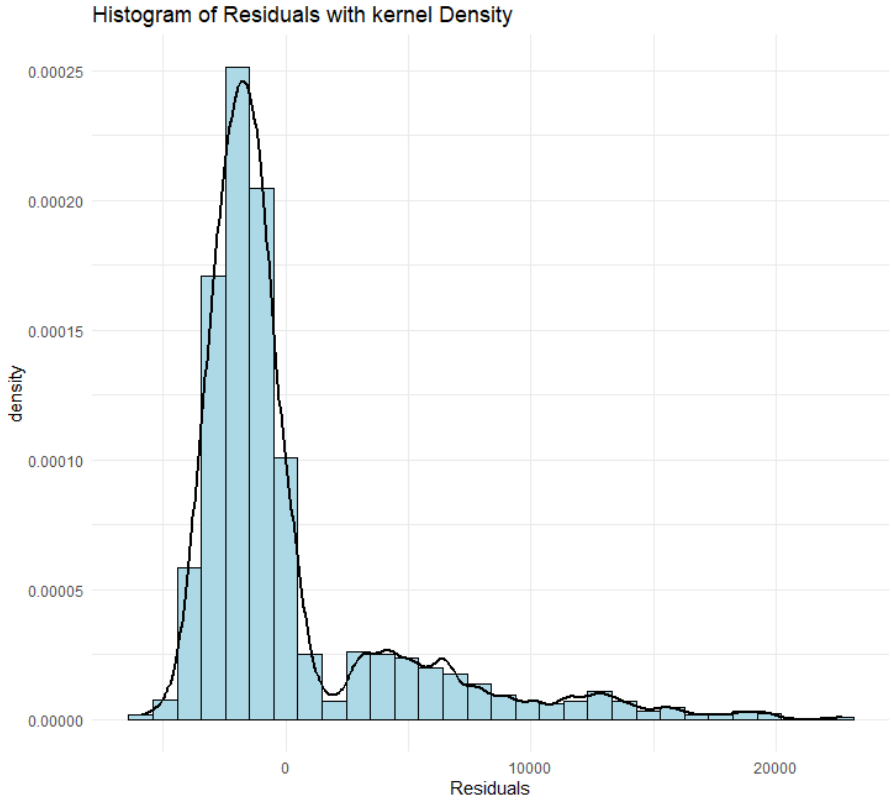
2. Plot #2: Residual against Age

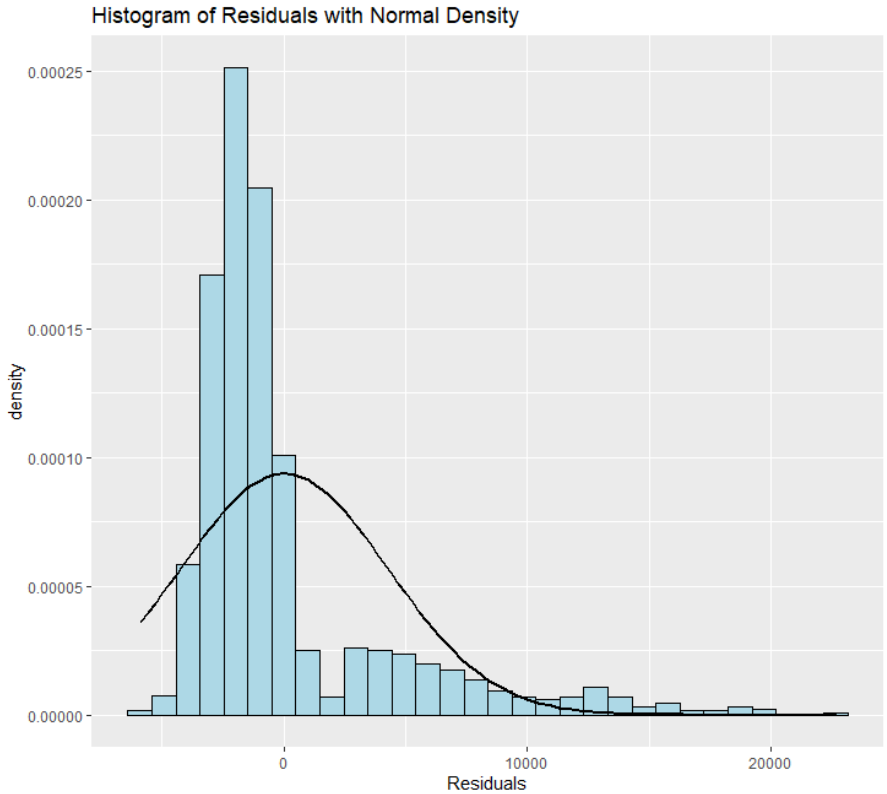



3. Plot #3: Residual against Region

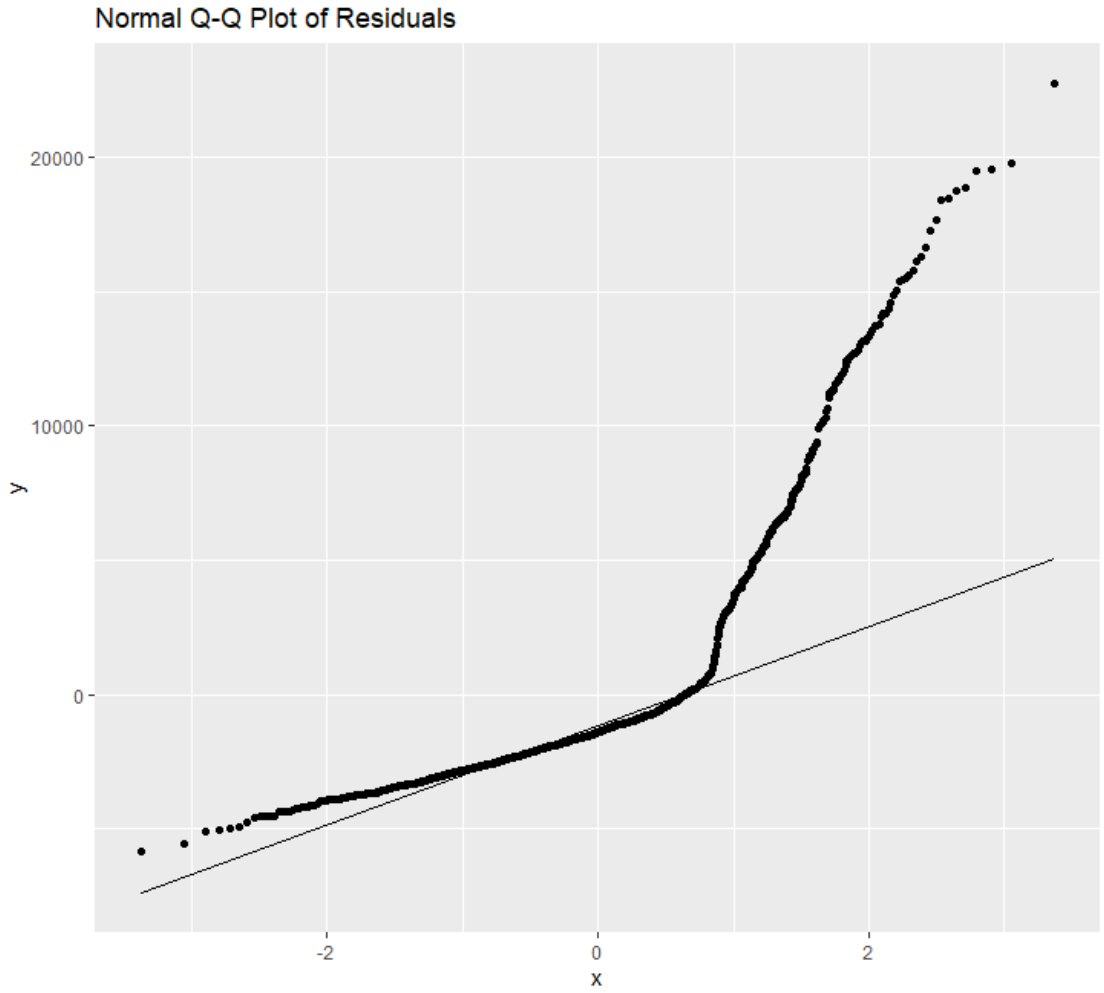


4. Plot #4: Histogram of residuals with normal density and Kernel density



5. Plot #5: Normal quantile plot of residuals




8) Step 8: Conclusions (20 Points)

State your overall findings in one page double spacing.

**Conclusions:**

In **PART 2** we can observe that the categorical predictors Gender, Smoker, and Region have p-values lower than 0.05, this means that the means for the categories in Gender and Smoker are different and at least one of the means for the categories in region is different. The lowest p-value corresponds to the variable Smoker, and we can observe that the mean of charges for people who smoke are 4 times higher compared with people who don't.

In **PART 3** for the numerical predictors, it is observed that people with different ages can have a wide range of charges, and the loess indicates the following trend: younger people have lower chargers compared to older people, for BMI we can observe that the variance of charges is higher when BMI>30 (heteroscedasticity), and for children we can observe that there is a small decrease for charges when the number of children is higher but this could be related with the number of samples (there is less people with 5 and 4 children).

In **PART 5** the model that performs the best in the training sets is the random forest but in the testing sets XGB and GB have better performance, LASSO constantly has higher ASE for both Training and Testing indicating that this data doesn't meet linearity assumptions. In **PART 6**, The Ensemble model had a lower ASE compared with Training ASE for GB and LASSO and the ASE is higher compared with Random Forest and XGB Training ASE.

In **PART 7**, We notice that the residuals exhibit non-normality and are right-skewed in the histogram and Q-Q plot. In the first plot (Residuals vs Predicted), negative residuals are closer to 0 compared with positive residuals, and there is a higher frequency for negative residuals, this indicates that the ensemble model tends to predict values higher than those observed. There is also a similarity between the plots Residual vs Age and Charges vs Age, indicating that age plays a significant role in the accuracy of the predictions.

# APPENDIX

**Programing Language: R**
**Note:** The program was executed multiple times without a specified seed. A seed was then selected to replicate the outcomes, and resembles the outcomes generated in most runs without a seed.

**Appendix 1:**

```
####LIBRARIES

#install.packages('car')
library(car)
library(ggplot2)

#install.packages('caret')
library(caret)

#install.packages('randomForest')
library(randomForest)

#install.packages('glmnet')
library(glmnet)

#install.packages('rpart')
library(rpart)

#install.packages('rpart.plot')
library(rpart.plot)

#install.packages('xgboost')
library(xgboost)

#install.packages("fastDummies")
library(fastDummies)

library(stats)
library(dplyr)

#install.packages('gbm')
library(gbm)

library(tidyverse)
```

```
library(rstatix)
library(ggpubr)

#Avoid Scientific Notation:
options(scipen = 999)


#### PART 1
## Read data

original_data <- read.csv("C:/Users/MS-XUserPC/Desktop/Project Data
Mining/inscharge.csv",header=TRUE)
data<-original_data
data$Gender <- factor(data$Gender)
data$Smoker <- factor(data$Smoker)
data$Rgion <- factor(data$Rgion)

head(data)

dim(data)
summary(data)
```

**Appendix 2:**

```
#### PART 2
##Data exploration, categorical predictors

## 2.2 Gender

table(data$Gender)

t.test(data$Charges ~ data$Gender)

##2.2 Smoker
table(data$Smoker)

t.test(data$Charges ~ data$Smoker)

## 2.3 Region
table(data$Rgion)
data %>%
  group_by(Rgion) %>%
  get_summary_stats(Charges, type = "mean_sd")

res_aov <- aov(Charges ~ Rgion, data = data)
```

```
summary(res_aov)

par(mfrow = c(1, 2)) # combine plots

# histogram
hist(res_aov$residuals)

# QQ-plot
qqPlot(res_aov$residuals,
       id = FALSE # id = FALSE to remove point identification
)

## The residuals are skewed
```

**Appendix 3:**

```
#### PART 3:

## Age
ggplot(data, aes(x = Age, y = Charges)) +
  geom_point() +  # Scatter plot
  geom_smooth(method = "loess") +  # Loess line
  labs(x = "Age", y = "Charges", title = "Scatter Plot with Loess Line for
Age")

## BMI
ggplot(data, aes(x = BMI, y = Charges)) +
  geom_point() +  # Scatter plot
  geom_smooth(method = "loess") +  # Loess line
  labs(x = "BMI", y = "Charges", title = "Scatter Plot with Loess Line for
BMI")

## Children
ggplot(data, aes(x = children, y = Charges)) +
  geom_point() +  # Scatter plot
  geom_smooth(method = "loess") +  # Loess line
  labs(x = "Children", y = "Charges", title = "Scatter Plot with Loess
Line for Children")
```

**Appendix 4:**

```
#### PART 4
```

```
# Create dummy variable
data <- dummy_cols(data, select_columns = "Rgion", remove_first_dummy =
TRUE)
data$Gendernum <- ifelse(data$Gender == 'female',1 ,0)
data$Smokernum <- ifelse(data$Smoker == 'no',1 ,0)
data <- subset(data, select =-c(Gender,Smoker,Rgion))
head(data)

## folds
set.seed(42)

fold<-createFolds(data$Charges, k = 5, list = FALSE, returnTrain = FALSE)
B_1 <- data[fold==1,]
B_2 <- data[fold==2,]
B_3 <- data[fold==3,]
B_4 <- data[fold==4,]
B_5 <- data[fold==5,]

TEST=list(B_1,B_2,B_3,B_4,B_5)


head(B_1)
```

**Appendix 5:**

```
#### Part 5

## ASE function:
ASE <- function(y_obs,y_pred) {
  ase <- mean((y_obs-y_pred)^2)
  return(ase)
}

## Train data
train1 <- data[fold != 1,]
train2 <- data[fold != 2,]
train3 <- data[fold != 3,]
train4 <- data[fold != 4,]
train5 <- data[fold != 5,]

TRAIN=list(train1,train2,train3,train4,train5)

## X and y - train

X1_train <- subset(train1, select = -c(Charges))
```

```
y1_train <- train1$Charges

X2_train <- subset(train2, select = -c(Charges))
y2_train <- train2$Charges

X3_train <- subset(train3, select = -c(Charges))
y3_train <- train3$Charges

X4_train <- subset(train4, select = -c(Charges))
y4_train <- train4$Charges

X5_train <- subset(train5, select = -c(Charges))
y5_train <- train5$Charges

X_TRAIN=list(X1_train,X2_train,X3_train,X4_train,X5_train)
Y_TRAIN=list(y1_train,y2_train,y3_train,y4_train,y5_train)


## X and y - test

X1_test <- subset(B_1, select = -c(Charges))
y1_test <- B_1$Charges

X2_test <- subset(B_2, select = -c(Charges))
y2_test <- B_2$Charges

X3_test <- subset(B_3, select = -c(Charges))
y3_test <- B_3$Charges

X4_test <- subset(B_4, select = -c(Charges))
y4_test <- B_4$Charges

X5_test <- subset(B_5, select = -c(Charges))
y5_test <- B_5$Charges

X_TEST=list(X1_test,X2_test,X3_test,X4_test,X5_test)
Y_TEST=list(y1_test,y2_test,y3_test,y4_test,y5_test)


#### LASSO

training_ASE_lasso <- numeric()
testing_ASE_lasso <- numeric()
lambda_seq <- 10^seq(-3,3,by=0.1)
```

```
for (i in 1:5) {
  lasso_model_ini <-cv.glmnet(as.matrix(X_TRAIN[[i]]), Y_TRAIN[[i]], alpha
= 1,lambda = lambda_seq)
  best_lambda<-lasso_model_ini$lambda.min
  print(best_lambda)
  lasso_model <-glmnet(X_TRAIN[[i]], Y_TRAIN[[i]], alpha = 1,lambda =
best_lambda )
  lasso_pred_train<-predict(lasso_model,newx=as.matrix(X_TRAIN[[i]]))
  training_ASE_lasso <- c(training_ASE_lasso,
ASE(Y_TRAIN[[i]],lasso_pred_train))
  lasso_pred_test<-predict(lasso_model,newx=as.matrix(X_TEST[[i]]))
  testing_ASE_lasso <-
c(testing_ASE_lasso,ASE(Y_TEST[[i]],lasso_pred_test))
}

#### Random Forest

training_ASE_rf <- numeric()
testing_ASE_rf <- numeric()

for (i in 1:5) {
  set.seed(123)
  rf_model = randomForest(Charges ~ ., data = TRAIN[[i]], ntree=100)
  rf_pred_train <- predict(rf_model,X_TRAIN[[i]])
  training_ASE_rf <- c(training_ASE_rf, ASE(Y_TRAIN[[i]],rf_pred_train))
  rf_pred_test<-predict(rf_model,X_TEST[[i]])
  testing_ASE_rf<- c(testing_ASE_rf, ASE(Y_TEST[[i]],rf_pred_test))

}

#### Gradient boosting

training_ASE_gb <- numeric()
testing_ASE_gb <- numeric()

for (i in 1:5){
  gbm_model <- gbm(Charges ~ ., data = TRAIN[[i]], distribution
=  "gaussian", n.trees = 200, interaction.depth =  4, shrinkage = 0.01)
  pred<- predict(gbm_model,X_TRAIN[[i]],n.trees=200)
  training_ASE_gb <-c(training_ASE_gb, ASE(Y_TRAIN[[i]],pred))
  pred2<- predict(gbm_model,X_TEST[[i]],n.trees=200)
  testing_ASE_gb<-c(testing_ASE_gb,ASE(Y_TEST[[i]],pred2))
}
```

```
#### XGB

training_ASE_XGB <- numeric()
testing_ASE_XGB <- numeric()

params <- list(
  objective = "reg:squarederror",  # Objective function for regression
  eta = 0.1,  # Learning rate
  max_depth = 5  # Maximum depth of trees
)

for (i in 1:5) {
  xgb_model = xgboost(data = as.matrix(X_TRAIN[[i]]), label =
Y_TRAIN[[i]], params = params, nthread = 1, nrounds = 25)
  xgb_pred_train <-predict(xgb_model,newdata = as.matrix(X_TRAIN[[i]]),
type='response')
  training_ASE_XGB<- c(training_ASE_XGB,ASE(Y_TRAIN[[i]],xgb_pred_train))
  xgb_pred_test <-predict(xgb_model,newdata = as.matrix(X_TEST[[i]]),
type='response')
  testing_ASE_XGB<- c(testing_ASE_XGB,ASE(Y_TEST[[i]],xgb_pred_test))
}



##
Testing_ASE <-
data.frame(testing_ASE_lasso,testing_ASE_rf,testing_ASE_gb,testing_ASE_XGB
)
names(Testing_ASE)<-c('LASSO','Random Forest', 'GB', 'XGB')
print(Testing_ASE)
Training_ASE <-
data.frame(training_ASE_lasso,training_ASE_rf,training_ASE_gb,training_ASE
_XGB)
names(Training_ASE)<-c('LASSO','Random Forest', 'GB', 'XGB')
print(Training_ASE)
```

**Appendix 6:**

```
#### PART 6

#This function get's the name of the column with lowest ASE
get_column <- function(row){
  column_idx<- which.min(row)
  column_name <-names(row)[column_idx]
```

```
  return(column_name)
}

##ENSEMBLE MODEL

min_ASE_training <- apply(Training_ASE,1,get_column)
print(min_ASE_training)
min_ASE_testing <- apply(Testing_ASE,1,get_column)
print(min_ASE_testing)

X<-subset(data, select = -c(Charges)) #Predictors
y <- data$Charges# Target

y_pred_df <- list() #Predicted values

for (i in 1:5) {
  set.seed(42)
  rf_model = randomForest(Charges ~ ., data = TRAIN[[i]], ntree=200)
  rf_pred_train <- predict(rf_model,X)
  cn <- paste0("Y_pred_", i)
  y_pred_df[[cn]] <- rf_pred_train
} #Random forest models for ensemble


y_pred_df <- as.data.frame(y_pred_df)

y_pred_df$Ensemble_prediction  <- rowMeans(y_pred_df) #Ensemble model
prediction

ASE(y,y_pred_df$Ensemble_prediction )

y_pred_df$Res_ensemble<- y-y_pred_df$Ensemble_prediction #Res = observed-
predicted
```

**Appendix 7:**

```
#### PART 7
final_df<- cbind(original_data,y_pred_df)

# Plot #1: Residual against Predicted Value of "Charges"
ggplot(final_df,aes(x = Ensemble_prediction, y = Res_ensemble)) +
  geom_point() +
  labs(title = "Residual vs Predicted Charges",
       x = "Predicted Charges",
```

```
       y = "Residuals")

ggplot(final_df, aes(x = Age, y = Res_ensemble)) +
  geom_point() +
  labs(title = "Residual vs Age",
       x = "Age",
       y = "Residuals")

ggplot(final_df, aes(x = Rgion, y = Res_ensemble)) +
  geom_point() +
  labs(title = "Residual vs Region",
       x = "Region",
       y = "Residuals")

ggplot(final_df, aes(x = Res_ensemble)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "lightblue",
color = "black") +
  geom_density() +
  theme_minimal()+
  labs(title = "Histogram of Residuals with kernel Density",
       x = "Residuals")

ggplot(final_df, aes(x = Res_ensemble)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "lightblue",
color = "black") +
  geom_density() +
  theme_minimal()+
  labs(title = "Histogram of Residuals with kernel Density",
       x = "Residuals")

ggplot(final_df, aes(x = Res_ensemble)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "lightblue",
color = "black") +
  stat_function(fun=dnorm,args=list(mean=mean(final_df$Res_ensemble),sd=sd
(final_df$Res_ensemble)))+
  labs(title = "Histogram of Residuals with Normal Density",
       x = "Residuals")


ggplot(final_df, aes(sample = Res_ensemble)) +
  stat_qq() +
  stat_qq_line() +
  labs(title = "Normal Q-Q Plot of Residuals")

boxplot(y_pred_df$Res_ensemble)
```

```
boxplot(y)
boxplot(y_pred_df$Ensemble_prediction)

RSS<-sum(y_pred_df$Res_ensemble^2)
TSS<-sum((y-mean(y))^2)
R_squared <- 1-(RSS/TSS)
print(R_squared)
```