Multiple Linear Regression Analysis of Insurance Charges and Other Significant Variables

Analysis of Insurance Charges

Cecelia Fu

Background Information and Analysis

insurance data set info/ mini story

Variable	Description
Age	How old the person who filed for insurance is.
Sex	Gender of the person.
BMI	The BMI index of the person that filed.
Children	The number of children the person has.
Smoker	A variable identifying if they smoked or not.
Region	The general geographic are where the person lives.

Read in the dataset.

Introduction

Insurance charges can be confusing, since there are lots of factors can contribute to charges. Information in this insurance charges dataset include individual's age, sex, bmi, number of children, region and insurance charges. The purpose of this analysis is to determine what factors would affect or predict the insurance charge, where these factors have positive or negative linear relationship with charge. The solution of the analysis would identify what factors can influence insurance charge, and also help consumers make better decision about their health.

Method

According to the data character, the multiple linear regression is applied. To start, I apply the shrinkage method to select which explanatory vairables can use to predict insurance charges; Then fit multiple linear regression model and check assumptions. When assumptions are not met, apply appropriate transformation to re-check assumptions.

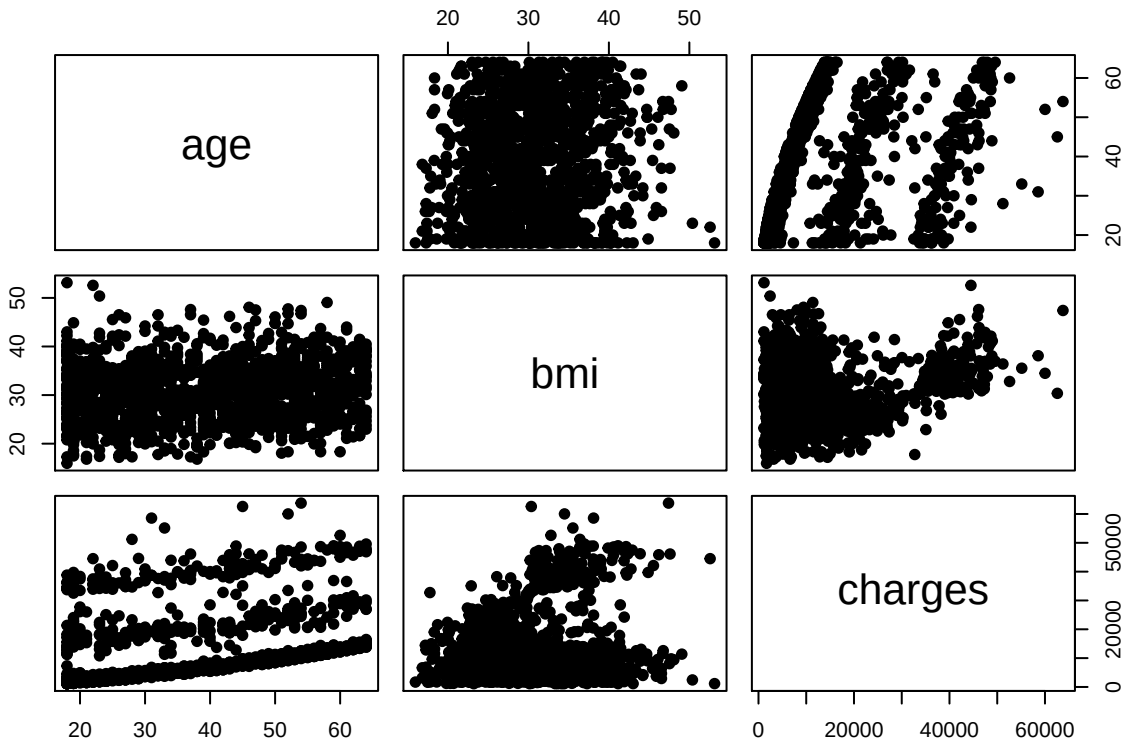
Exploratory analysis:

Start by looking at a summary of our data, making a scatter plot matrix of th continuous variables, and look at side by side boxplots for the categorical variables.

```
# Look at a summary of the data to make sure that it makes sense.
summary(insurance)
```

```
##       age            sex                 bmi           children
##  Min.   :18.00   Length:1338        Min.   :15.96   Min.   :0.000
##  1st Qu.:27.00   Class :character   1st Qu.:26.30   1st Qu.:0.000
##  Median :39.00   Mode  :character   Median :30.40   Median :1.000
##  Mean   :39.21                      Mean   :30.66   Mean   :1.095
##  3rd Qu.:51.00                      3rd Qu.:34.69   3rd Qu.:2.000
##  Max.   :64.00                      Max.   :53.13   Max.   :5.000
##     smoker             region             charges
##  Length:1338        Length:1338        Min.   : 1122
##  Class :character   Class :character   1st Qu.: 4740
##  Mode  :character   Mode  :character   Median : 9382
##                                        Mean   :13270
##                                        3rd Qu.:16640
##                                        Max.   :63770
```

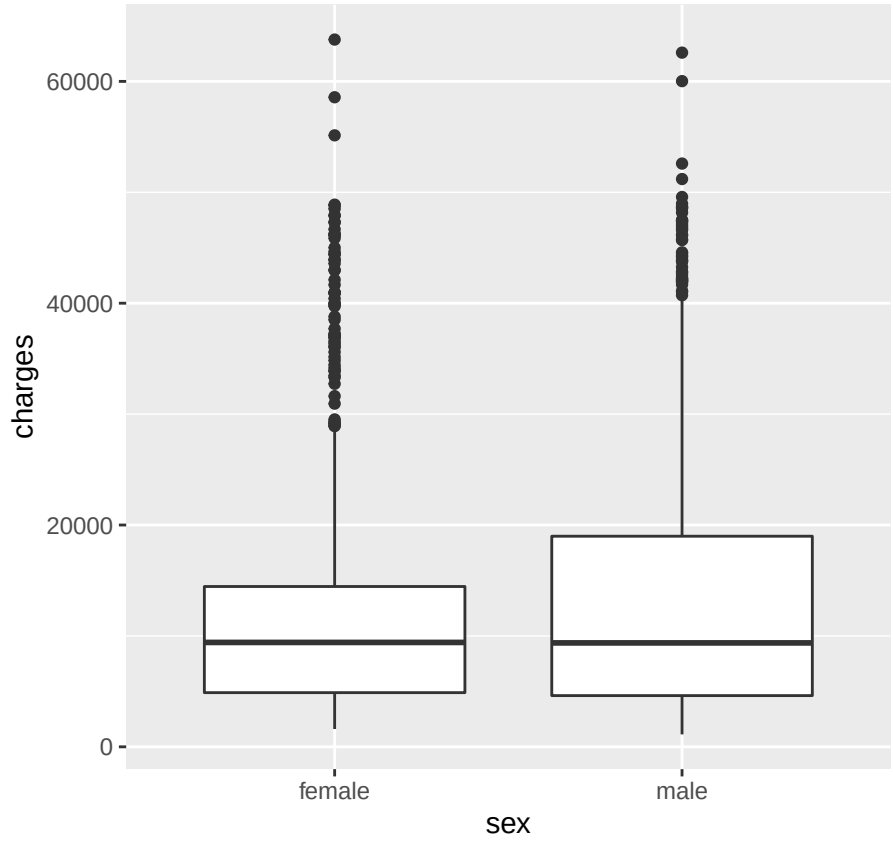
```
# Subset the data to make a scatter plot matrix of all the continuous variables
cont.insurance <- select(insurance, c("age", "bmi", "charges"))
plot(cont.insurance, pch = 19)
```



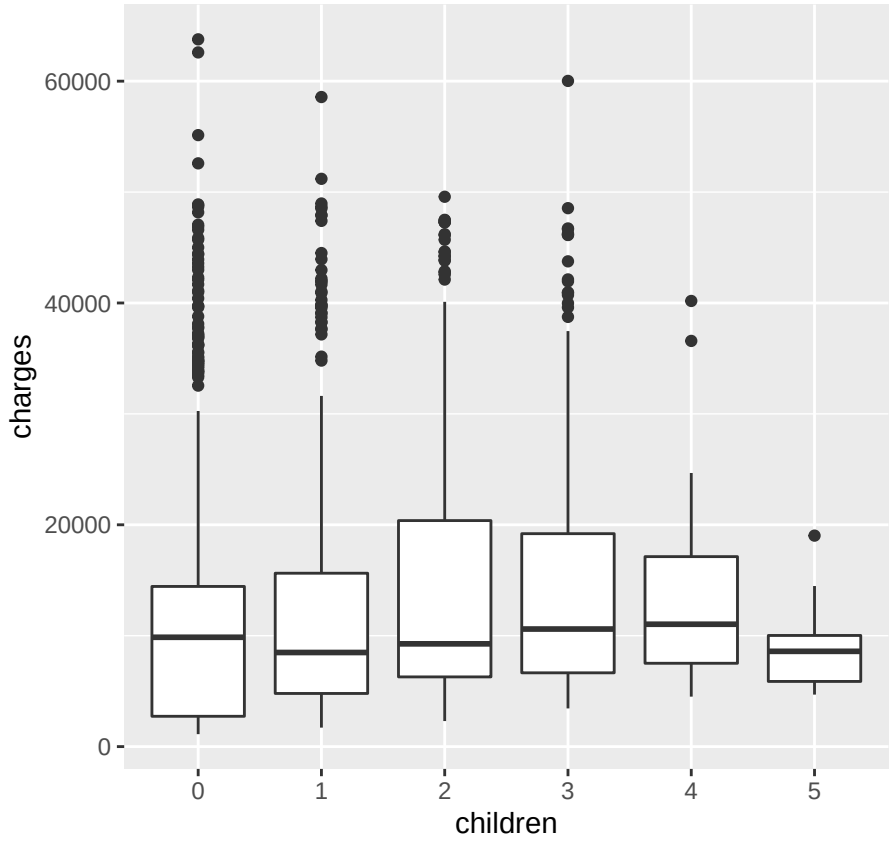
```r
# Make a correlation matrix of the continuous variables.
cor(cont.insurance)
```

```
##                age       bmi   charges
## age     1.0000000 0.1092719 0.2990082
## bmi     0.1092719 1.0000000 0.1983410
## charges 0.2990082 0.1983410 1.0000000
```

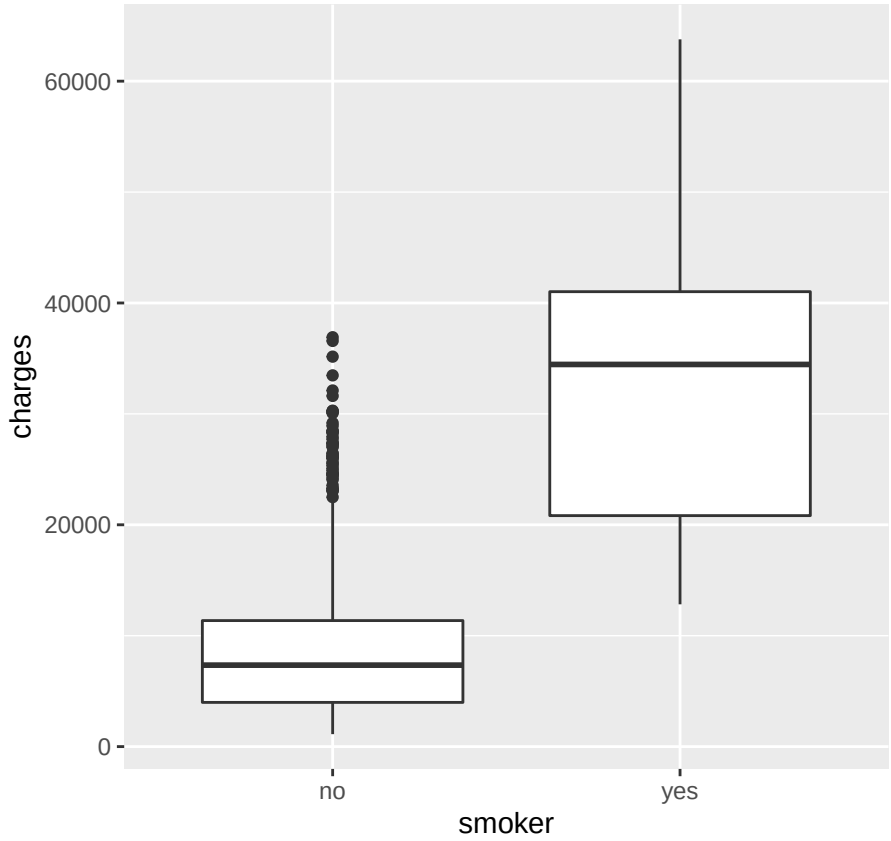
```r
# Boxplots of all other categorical variables to get an idea of what is going on.
box.sex <- ggplot(data = insurance,
                  mapping = aes(x = sex, y = charges)) +
    geom_boxplot() +
    theme(aspect.ratio = 1)
box.sex
```



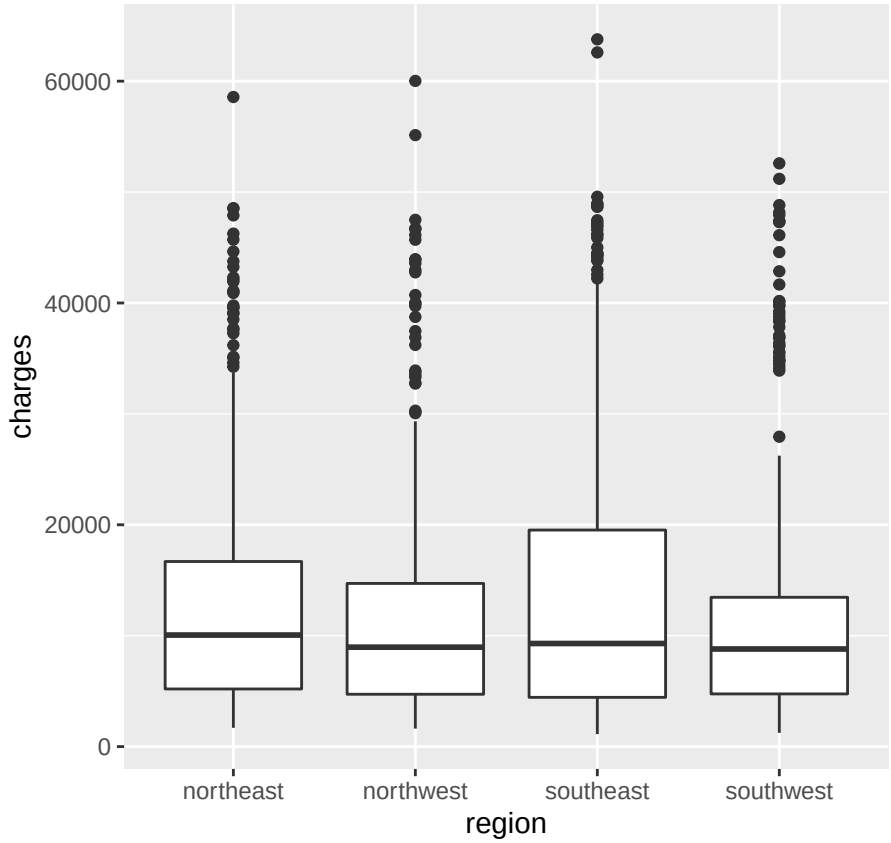

```r
insurance$children <- as.factor(insurance$children)
box.children <- ggplot(data = insurance,
                       mapping = aes(x = children, y = charges)) +
    geom_boxplot() +
    theme(aspect.ratio = 1)
box.children
```




```
box.smoker <- ggplot(data = insurance,
                     mapping = aes(x = smoker, y = charges)) +
    geom_boxplot() +
    theme(aspect.ratio = 1)
box.smoker
```




```
box.region <- ggplot(data = insurance,
                     mapping = aes(x = region, y = charges)) +
    geom_boxplot() +
    theme(aspect.ratio = 1)
box.region
```



Shrinkage Methods

Next apply shrinkage methods to the data to see if all of the categorical variables are relevant to this model.

```
#Convert all variables to factors or numerical.
insurance$children <- as.factor(insurance$children)
insurance$sex <- as.factor(insurance$sex)
insurance$smoker <- as.factor(insurance$smoker)
insurance$region <- as.factor(insurance$region)
```

Best Subsets:

```
best.subsets.bic <- bestglm(insurance,
                            IC = "BIC",
                            method = "exhaustive",
                            TopModels = 10)
```

```
## Morgan-Tatar search since factors present with more than 2 levels.
```

```
summary(best.subsets.bic$BestModel)
```

```
##
```

```
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12415.4  -2970.9   -980.5   1480.0  28971.8
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11676.83     937.57  -12.45   <2e-16 ***
## age            259.55      11.93   21.75   <2e-16 ***
## bmi            322.62      27.49   11.74   <2e-16 ***
## smokeryes    23823.68     412.87   57.70   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6092 on 1334 degrees of freedom
## Multiple R-squared:  0.7475, Adjusted R-squared:  0.7469
## F-statistic:  1316 on 3 and 1334 DF,  p-value: < 2.2e-16
```

Forward Selection:

```
forward.aic <- bestglm(insurance,
                           IC = "AIC",
                           method = "forward",
                           TopModels = 10)
```

```
## Morgan-Tatar search since factors present with more than 2 levels.
```

```
summary(forward.aic$BestModel)
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11620.3  -2883.5   -945.6   1513.0  29986.9
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -11977.26     984.79 -12.162  < 2e-16 ***
## age                257.30      11.91  21.609  < 2e-16 ***
## bmi                336.39      28.57  11.774  < 2e-16 ***
## children1          388.71     421.17   0.923 0.356211
## children2         1635.23     466.52   3.505 0.000471 ***
## children3          962.98     547.91   1.758 0.079055 .
## children4         2938.65    1238.56   2.373 0.017804 *
## children5         1106.45    1455.33   0.760 0.447227
## smokeryes        23824.24     412.80  57.714  < 2e-16 ***
```

```
## regionnorthwest   -379.44     476.40  -0.796 0.425908
## regionsoutheast  -1032.43     478.98  -2.155 0.031304 *
## regionsouthwest   -952.16     478.00  -1.992 0.046577 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6057 on 1326 degrees of freedom
## Multiple R-squared:  0.7519, Adjusted R-squared:  0.7498
## F-statistic: 365.3 on 11 and 1326 DF,  p-value: < 2.2e-16
```

Backward Selection:

```
backward.aic <- bestglm(insurance,
                        IC = "AIC",
                        method = "backward",
                        TopModels = 10)
```

```
## Morgan-Tatar search since factors present with more than 2 levels.
```

```
summary(backward.aic$BestModel)
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11620.3  -2883.5   -945.6   1513.0  29986.9
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -11977.26     984.79 -12.162  < 2e-16 ***
## age                257.30      11.91  21.609  < 2e-16 ***
## bmi                336.39      28.57  11.774  < 2e-16 ***
## children1          388.71     421.17   0.923 0.356211
## children2         1635.23     466.52   3.505 0.000471 ***
## children3          962.98     547.91   1.758 0.079055 .
## children4         2938.65    1238.56   2.373 0.017804 *
## children5         1106.45    1455.33   0.760 0.447227
## smokeryes        23824.24     412.80  57.714  < 2e-16 ***
## regionnorthwest   -379.44     476.40  -0.796 0.425908
## regionsoutheast  -1032.43     478.98  -2.155 0.031304 *
## regionsouthwest   -952.16     478.00  -1.992 0.046577 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6057 on 1326 degrees of freedom
## Multiple R-squared:  0.7519, Adjusted R-squared:  0.7498
## F-statistic: 365.3 on 11 and 1326 DF,  p-value: < 2.2e-16
```

Sequential Replacement:

```
seqrep.aic <- bestglm(insurance,
                      IC = "AIC",
                      method = "seqrep",
                      TopModels = 10,
                      t=100)
```

```
## Morgan-Tatar search since factors present with more than 2 levels.
```

```
summary(seqrep.aic$BestModel)
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11620.3  -2883.5   -945.6   1513.0  29986.9
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -11977.26     984.79 -12.162  < 2e-16 ***
## age                257.30      11.91  21.609  < 2e-16 ***
## bmi                336.39      28.57  11.774  < 2e-16 ***
## children1          388.71     421.17   0.923 0.356211
## children2         1635.23     466.52   3.505 0.000471 ***
## children3          962.98     547.91   1.758 0.079055 .
## children4         2938.65    1238.56   2.373 0.017804 *
## children5         1106.45    1455.33   0.760 0.447227
## smokeryes        23824.24     412.80  57.714  < 2e-16 ***
## regionnorthwest   -379.44     476.40  -0.796 0.425908
## regionsoutheast  -1032.43     478.98  -2.155 0.031304 *
## regionsouthwest   -952.16     478.00  -1.992 0.046577 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6057 on 1326 degrees of freedom
## Multiple R-squared:  0.7519, Adjusted R-squared:  0.7498
## F-statistic: 365.3 on 11 and 1326 DF,  p-value: < 2.2e-16
```

(This code alters the way the data is presented so lasso and elastic net can run accurately.)

```
#Convert elements to be represented as numbers and then change them to be factors
insurance <- read.csv("insurance.csv", header = TRUE)
insurance$smoker <- ifelse(insurance$smoker == "no", 0, 1)
insurance$sex <- ifelse(insurance$sex == "female", 0, 1)

for(i in 1:length(insurance$region)){
  if(insurance[i, 6] == "southwest"){
    insurance[i, 6] <- 1
```

```
  } else if(insurance[i, 6] == "northwest"){
    insurance[i, 6] <- 2
  } else if(insurance[i, 6] == "northeast"){
    insurance[i, 6] <- 3
  } else if(insurance[i, 6] == "southeast"){
    insurance[i, 6] <- 4
  }
}

#Make sure continuous variables are continuous
insurance$age <- as.numeric(insurance$age)
insurance$bmi <- as.numeric(insurance$bmi)

#Convert all categorical variables back to factors.
insurance$sex <- as.factor(insurance$sex)
insurance$children <- as.factor(insurance$children)
insurance$smoker <- as.factor(insurance$smoker)
insurance$region <- as.factor(insurance$region)
```

Lasso:

```
# make a matrix for our covariates and pull out response as its own variable
insurance.x <- data.matrix(insurance[, c(1:6)])
insurance.y <- insurance[, 7]

# Lasso (alpha = 1)
insurance.lasso <- glmnet(x = insurance.x, y = insurance.y, alpha = 1)

# use cross validation to pick the "best" lambda (based on MSE)
insurance.lasso.cv <- cv.glmnet(x = insurance.x, y = insurance.y,
                          type.measure = "mse", alpha = 1)

# lambda.min is the value of lambda that gives minimum mean cross-validated
# error
insurance.lasso.cv$lambda.min
```

```
## [1] 90.96904
```

```
# lambda.1se gives the most regularized model such that error is within one
# standard error of the minimum
insurance.lasso.cv$lambda.1se
```

```
## [1] 931.0964
```

```
# coefficients (betas) using a specific lambda penalty value
coef(insurance.lasso.cv, s = "lambda.min")
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept) -35332.2386
## age            252.0958
```

```
## sex              .
## bmi            308.6132
## children       402.3001
## smoker       23583.3455
## region           .
```

```
coef(insurance.lasso.cv, s = "lambda.1se")
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                        1
## (Intercept) -26063.2581
## age            198.0429
## sex              .
## bmi            185.9424
## children         .
## smoker       21470.5699
## region           .
```

Elastic Net:

```
# make a matrix for our covariates and pull out response as its own variable
insurance.x <- data.matrix(insurance[, 1:6])
insurance.y <- insurance[, 7]

# Elastic Net (alpha = .5)
insurance.elastic <- glmnet(x = insurance.x, y = insurance.y, alpha = .5)

# use cross validation to pick the "best" lambda (based on MSE)
insurance.elastic.cv <- cv.glmnet(x = insurance.x, y = insurance.y,
                          type.measure = "mse", alpha = .5)

# lambda.min is the value of lambda that gives minimum mean cross-validated
# error
insurance.elastic.cv$lambda.min
```

```
## [1] 104.1117
```

```
# lambda.1se gives the most regularized model such that error is within one
# standard error of the minimum
insurance.elastic.cv$lambda.1se
```

```
## [1] 1065.615
```

```
# coefficients (betas) using a specific lambda penalty value
coef(insurance.elastic.cv, s = "lambda.min")
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept) -35576.666568
## age            253.446519
## sex             -5.040773
```

```
## bmi            313.255543
## children       431.815164
## smoker       23579.092542
## region           .
```

```
coef(insurance.elastic.cv, s = "lambda.1se")
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## (Intercept) -28445.41299
## age            214.37553
## sex              .
## bmi            236.58693
## children        61.55346
## smoker       21520.30237
## region           .
```

Evaluation (Which variables are worthy of being included in the analysis.)

Variable	Best Subset	Forward	Backward	Sequential Replacement	LASSO	Elastic Net
age	X	X	X	X	X	X
sex						
bmi	X	X	X	X	X	X
children		X	X	X		
smoker	X	X	X	X		
region		X	X	X		

Based on the results from all of the shrinkage methods, I think it's best to take four of the original six variables which are age, bmi, smoker, and children. Even though many of them said that region could play somewhat of a factor, I removed it because it doesn't seem logical that the place that you live have as large an impact on your insurance charges as the number of children that you have. Moving forward I'll use these four variables and apply any necessary transformations later on.

Create a linear model of the data.

```
#Read in a fresh set of the data and convert the necessary data to factors
insurance <- read.csv("insurance.csv", header = TRUE)
insurance$children <- as.factor(insurance$children)
insurance$smoker <- as.factor(insurance$smoker)


sub.insurance <- insurance[, c(1,3,4,5,7)]
sub.insurance$children <- as.factor(sub.insurance$children)

sub.insurance.lm <- lm(charges ~ age + bmi + children + smoker,
                       data = sub.insurance)
summary(sub.insurance.lm)
```

```
##
```

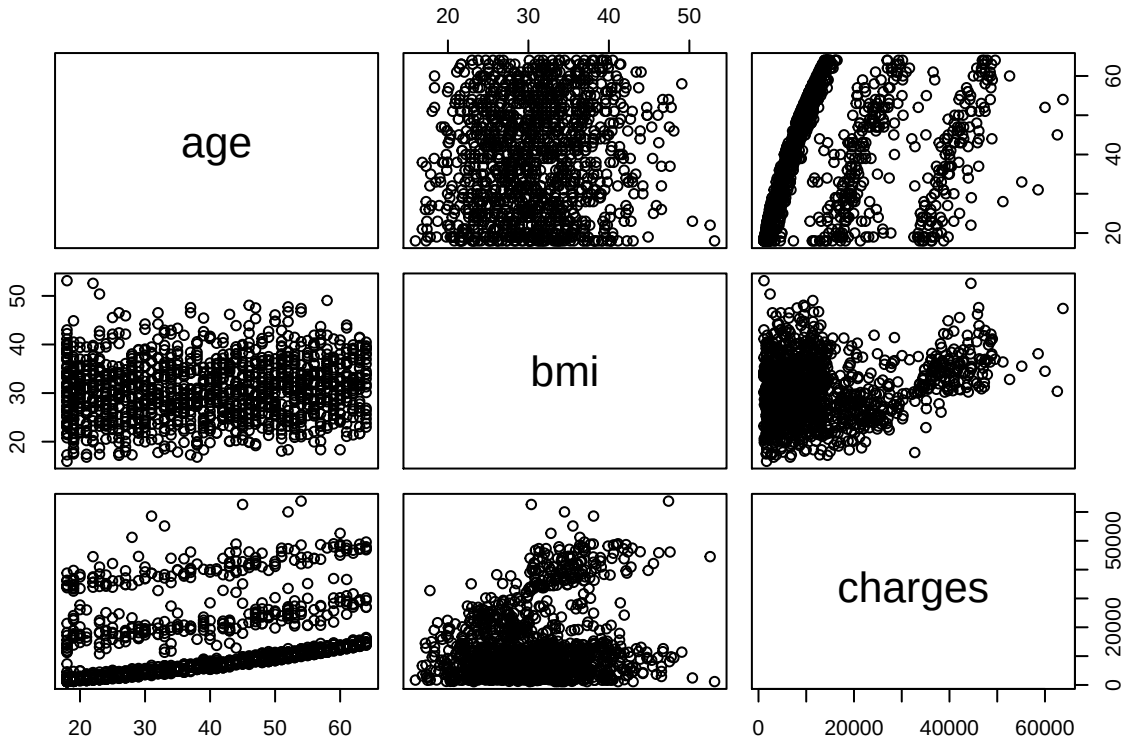
```
## Call:
## lm(formula = charges ~ age + bmi + children + smoker, data = sub.insurance)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12097.1  -2922.6   -950.7   1551.0  29566.8
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12093.32     947.78 -12.760  < 2e-16 ***
## age            258.08      11.91  21.665  < 2e-16 ***
## bmi            319.80      27.38  11.682  < 2e-16 ***
## children1      368.77     421.57   0.875 0.381868
## children2     1626.51     466.56   3.486 0.000506 ***
## children3      996.95     547.80   1.820 0.068997 .
## children4     2984.36    1239.60   2.408 0.016197 *
## children5      899.13    1453.36   0.619 0.536250
## smokeryes    23796.71     412.05  57.752  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6064 on 1329 degrees of freedom
## Multiple R-squared:  0.7508, Adjusted R-squared:  0.7493
## F-statistic: 500.4 on 8 and 1329 DF,  p-value: < 2.2e-16
```

```
#add residuals and fitted values to dataframe.
sub.insurance$residuals <- sub.insurance.lm$residuals
sub.insurance$fitted.values <- sub.insurance.lm$fitted.values
```

Check Assumptions

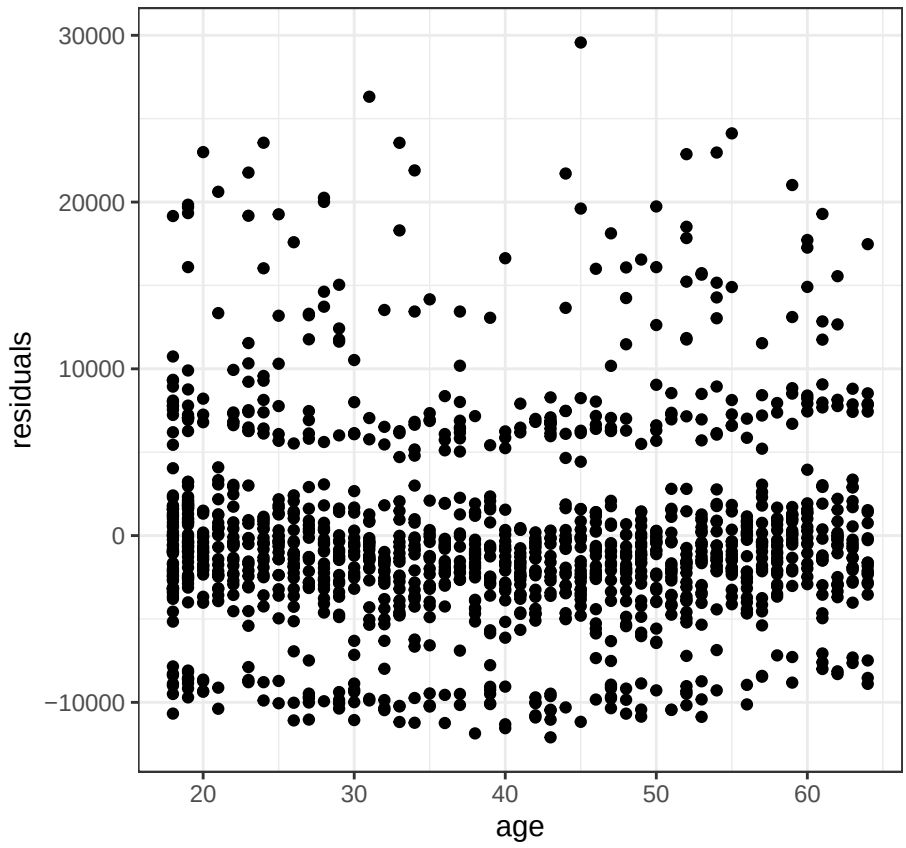
(L) Assumption 1: X vs. Y is linear (using scatter plots, partial regression plots, residuals vs. fitted values plots, and specific scatter plots that meet certain requirements).

```
#Scatter plot matrix of continuous variables.
plot(cont.insurance)
```

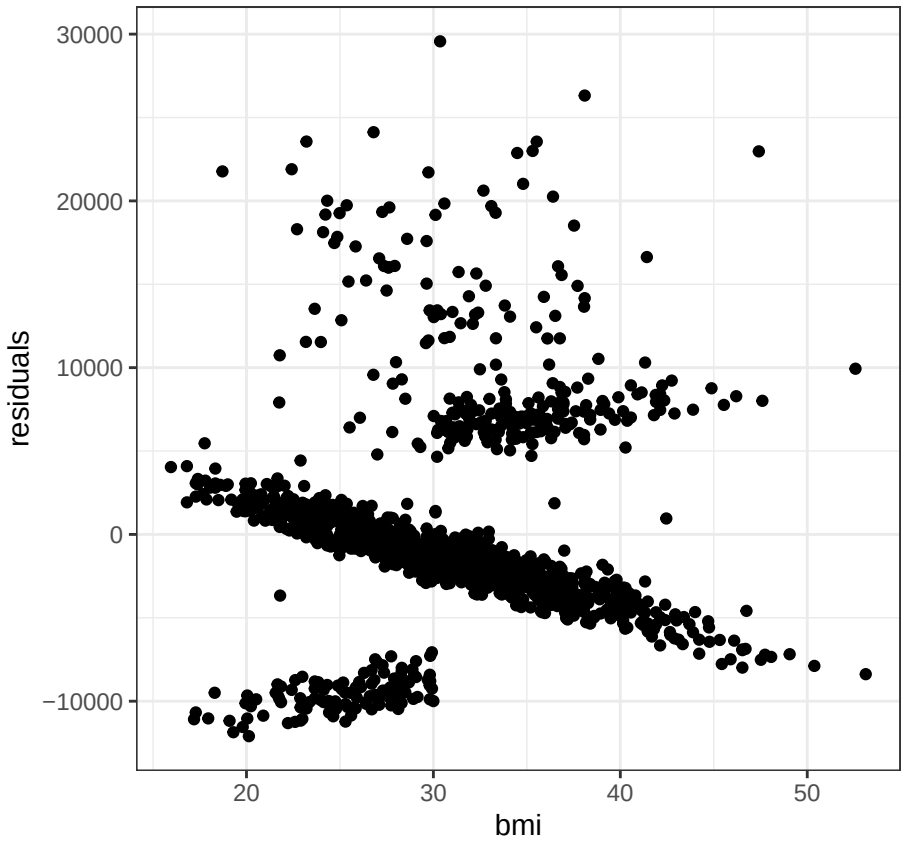



```
#Partial Regression plots
#Age plot code
plot.age <- ggplot(data = sub.insurance,
                   mapping = aes(x = age, y = residuals)) +
    geom_point() +
    theme_bw() +
    theme(aspect.ratio = 1)


#BMI plot code
plot.bmi <- ggplot(data = sub.insurance,
                   mapping = aes(x = bmi, y = residuals)) +
    geom_point() +
    theme_bw() +
    theme(aspect.ratio = 1)

#Children plot code
plot.children <- ggplot(data = sub.insurance,
                        mapping = aes(x = children, y = residuals)) +
    geom_point() +
    theme_bw() +
    theme(aspect.ratio = 1)

#Smoker plot code
plot.smoker <- ggplot(data = sub.insurance,
                      mapping = aes(x = smoker, y = residuals)) +
```

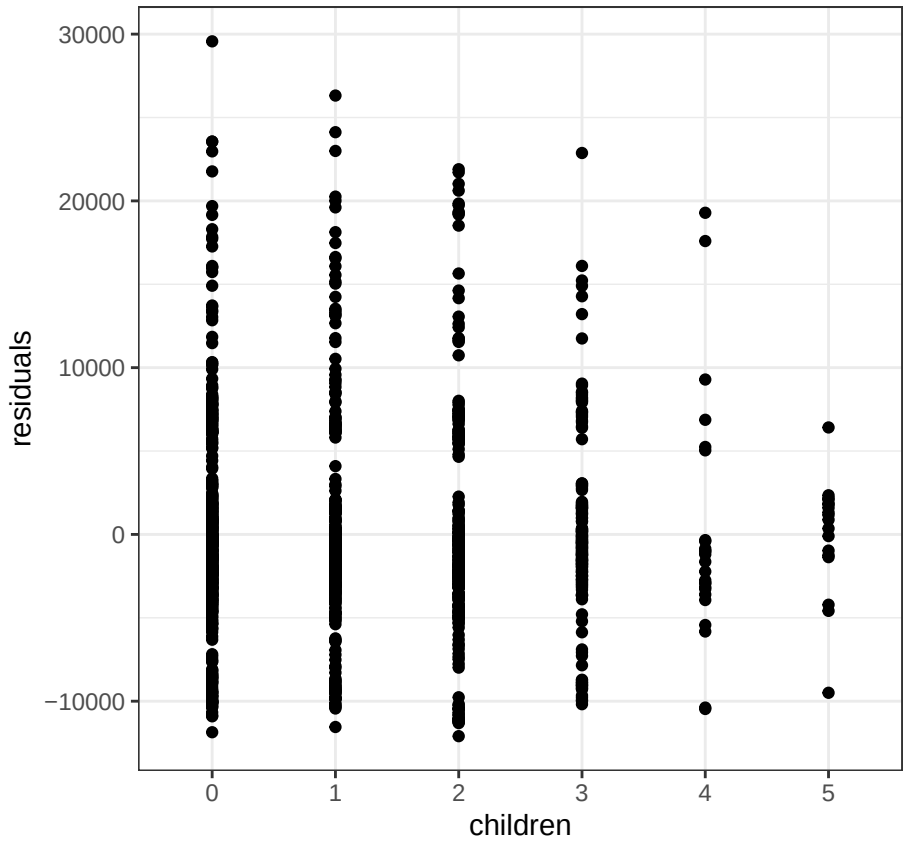
```
    geom_point() +
    theme_bw() +
    theme(aspect.ratio = 1)


plot.age
```



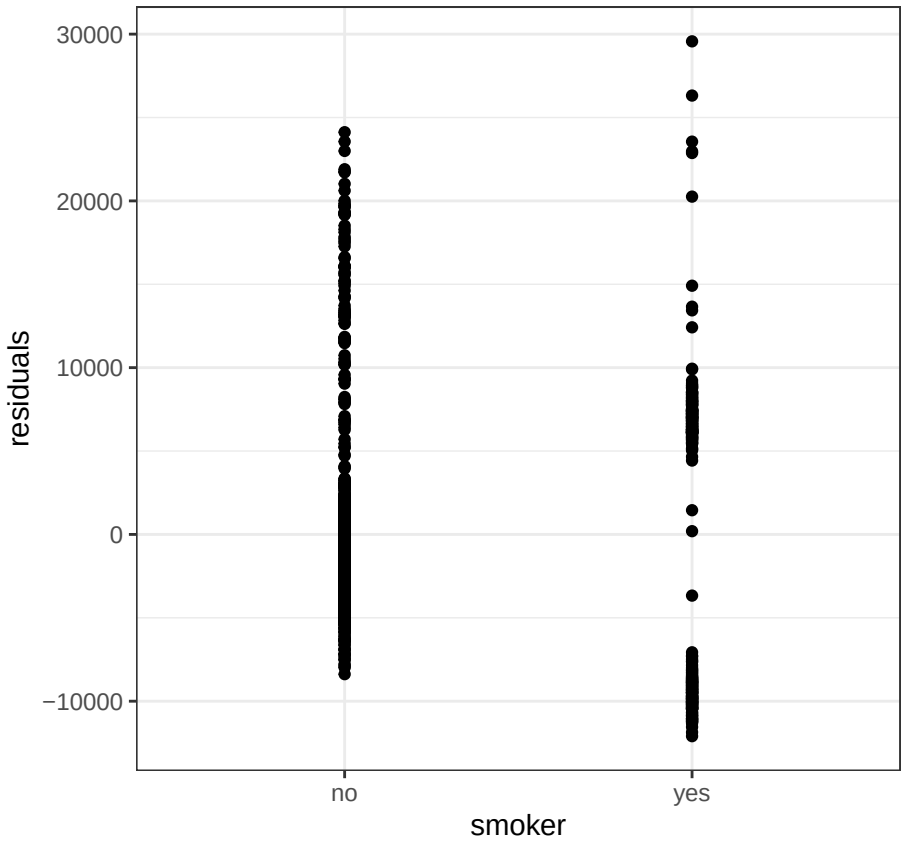

```
plot.bmi
```




```
plot.children
```



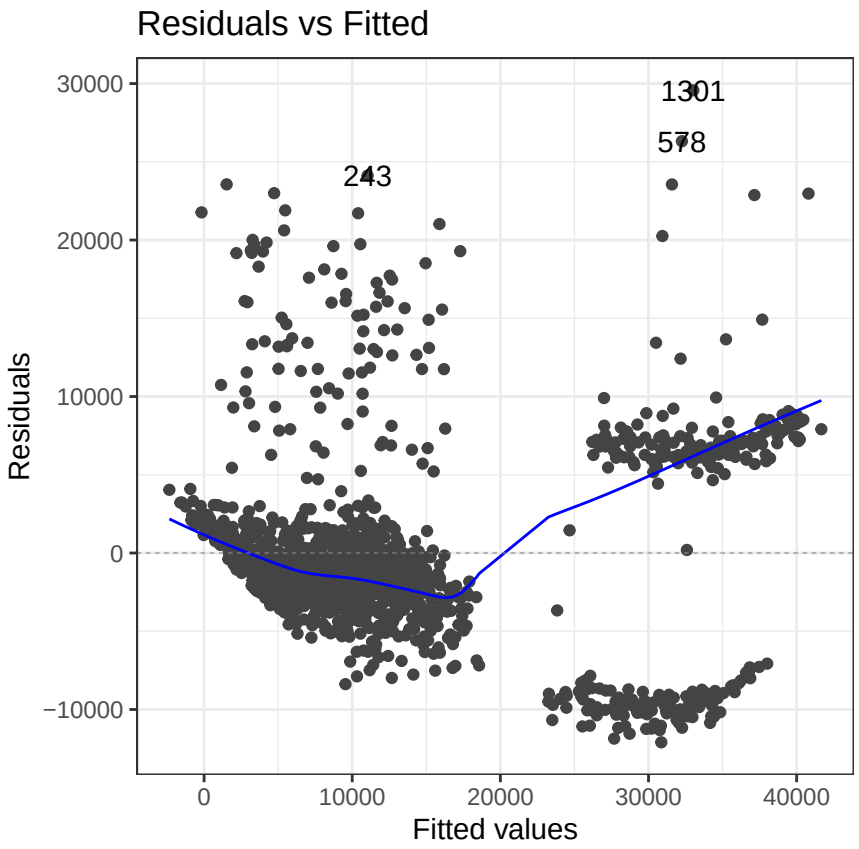

```
plot.smoker
```



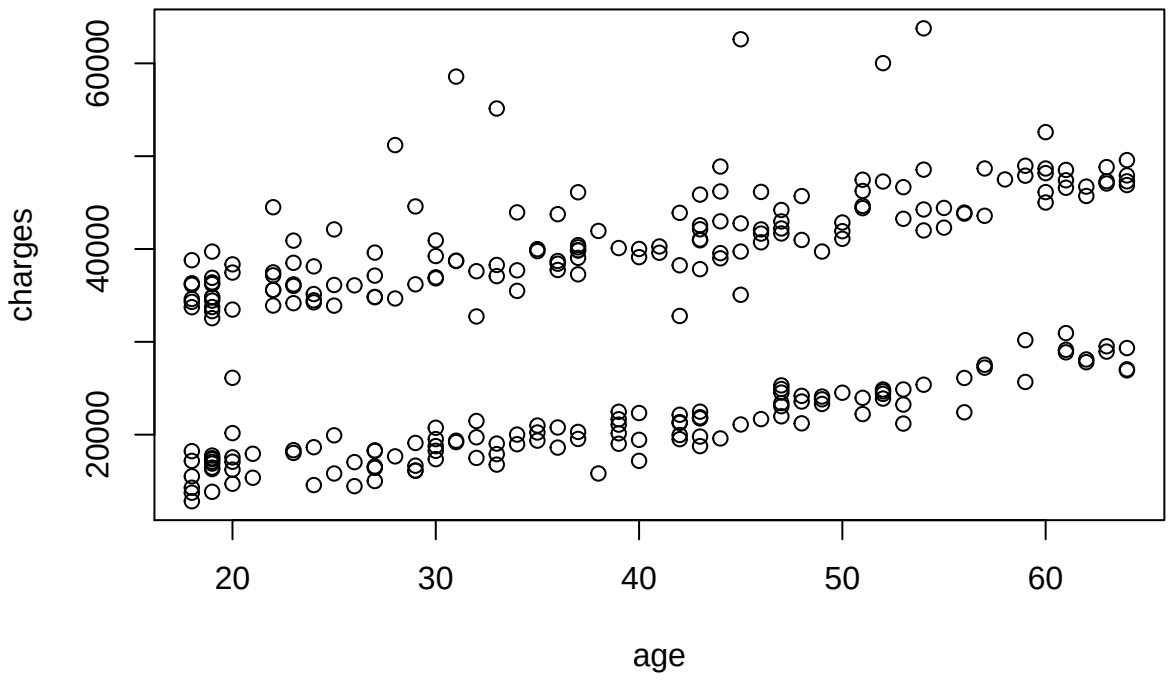

```
#Residuals vs. fitted values.
residuals.plot <- autoplot(sub.insurance.lm, which = 1, ncol = 1, nrow = 1) +
  theme_bw() +
  theme(aspect.ratio = 1)
```

```
## Warning: `arrange_()` was deprecated in dplyr 0.7.0.
## Please use `arrange()` instead.
## See vignette('programming') for more help
```

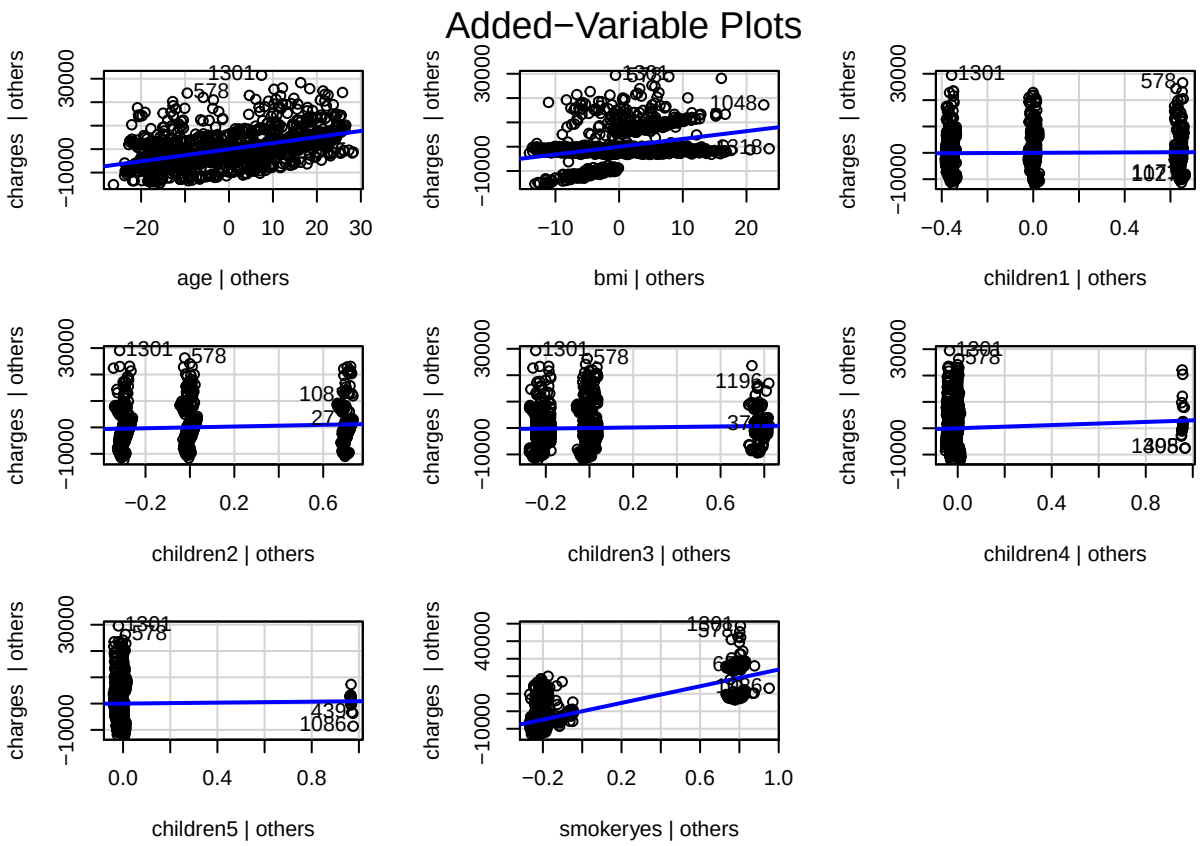
```
residuals.plot
```




```
#Specific scatter plots
plot(charges ~ age, data = subset(sub.insurance, smoker == "yes"))
```




```
#AV Plots
avPlots(sub.insurance.lm)
```



Assumption 1 Conclusions:

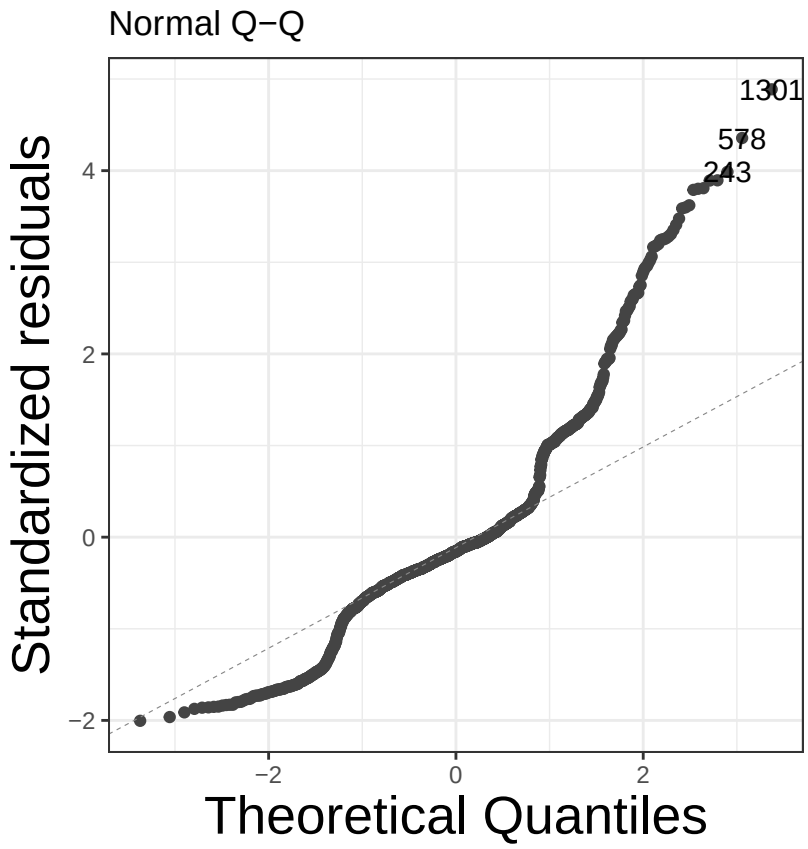
This assumption is still met. There are 3 distinct lines that are being affected by a number of other variables, but that hasn't changed the linear nature of our data. I do think that it will become more strongly true through some sort of transformation.

(I) Assumption 2: Residuals are independent.

Do not know how the data was collected, so I cannot conclusively state that the residuals are independent. My assumption is that the residuals will be independent since the medical conditions and needs of a particular person can't influence the medical conditions and needs of another.

(N) Assumption 3: Residuals are normally distributed and centered at 0.

```
prob.plot <- autoplot(sub.insurance.lm, which = 2, ncol = 1, nrow = 1) +
  theme_bw() +
  theme(aspect.ratio = 1,
        axis.title.x = element_text(size = sz),
        axis.title.y = element_text(size = sz),
        axis.title = element_text(size = sz))

prob.plot
```




```
shapiro.test(sub.insurance.lm$residuals)
```

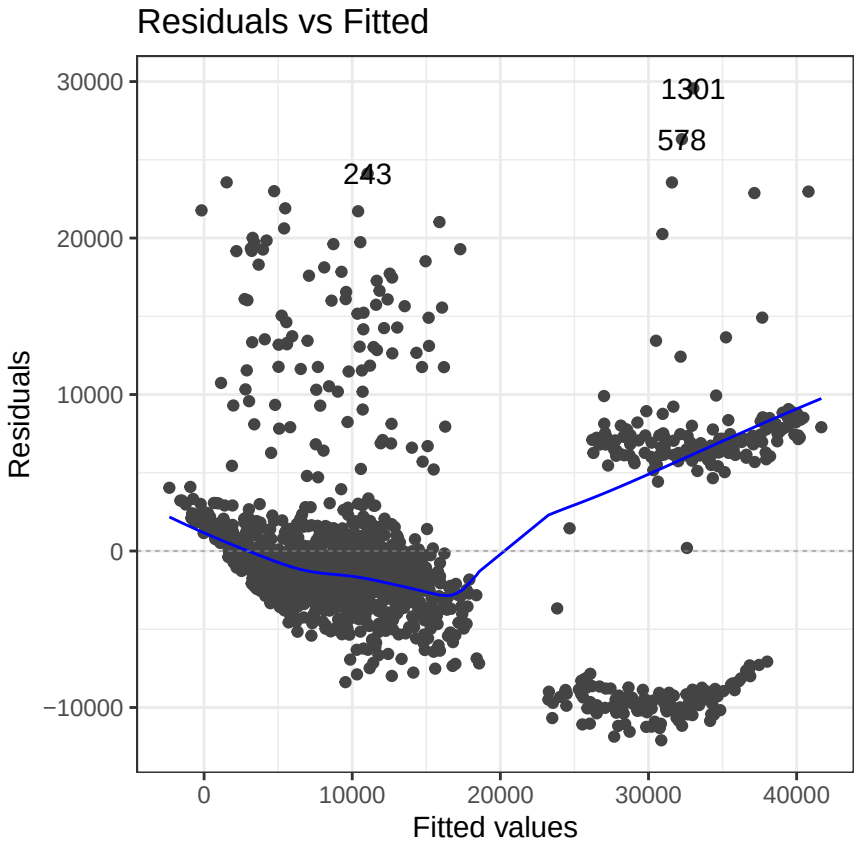
```
##
##  Shapiro-Wilk normality test
##
## data:  sub.insurance.lm$residuals
## W = 0.90418, p-value < 2.2e-16
```

Assumption 3 Conclusions:

Normality is not met. The normal probability plot doesn't follow a straight line, and the Shapiro-Wilk test gave us a p value of basically 0. This assumption can do better. I will make a transformation of the data and see if it helps the situation.

(E) Assumption 4: Residuals have equal variance.

```
residuals.plot
```




```
grp <- as.factor(c(rep("lower", floor(dim(insurance)[1] / 2)),
                   rep("upper", ceiling(dim(insurance)[1] / 2))))
leveneTest(sub.insurance[order(sub.insurance$age),
                         "residuals"] ~ grp, center = median)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    1  0.0895 0.7649
##       1336
```

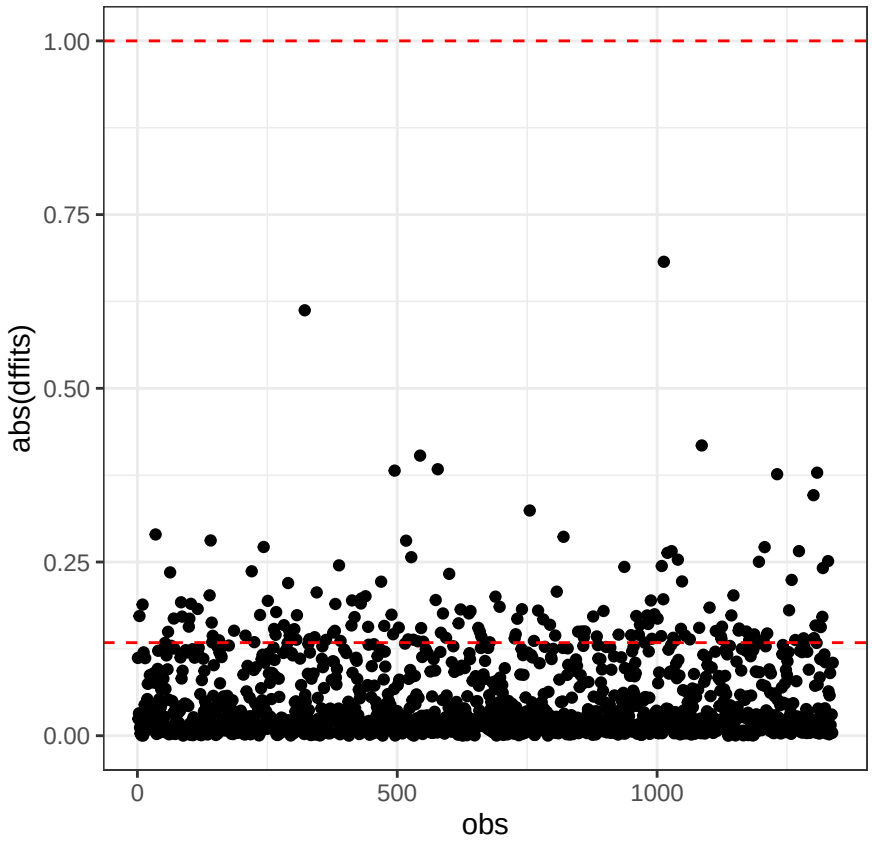
Assumption 4 Conclusions:

This assumption is not met. There is no even spread in the data currently and it looks to be clumped into large groups. This suggests that there are other predictor variables that we currently don't have access to. I'll still apply a transformation to see if I can get the model to look a little better.

(A) Assumption 5: Model describes ALL observations.

```
#DFFits
insurance.dffits <- data.frame ("dffits" = dffits(sub.insurance.lm))
insurance.dffits$obs <- 1:length(sub.insurance$age)

ggplot(data = insurance.dffits) +
  geom_point(mapping = aes(x = obs, y = abs(dffits))) +
```

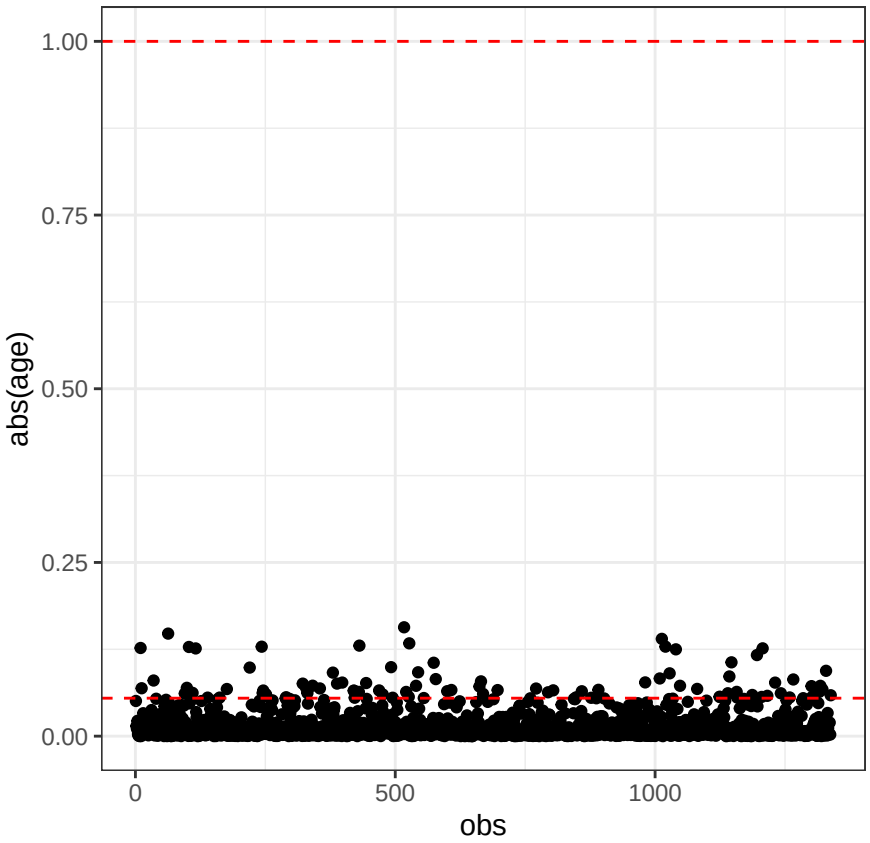
```
  geom_hline(mapping = aes(yintercept = 1),
             color = "red", linetype = 2) +
  geom_hline(mapping = aes(yintercept = 2 * sqrt(6 / length(obs))),
             color = "red", linetype = 2) +
  theme_bw() +
  theme(aspect.ratio = 1)
```



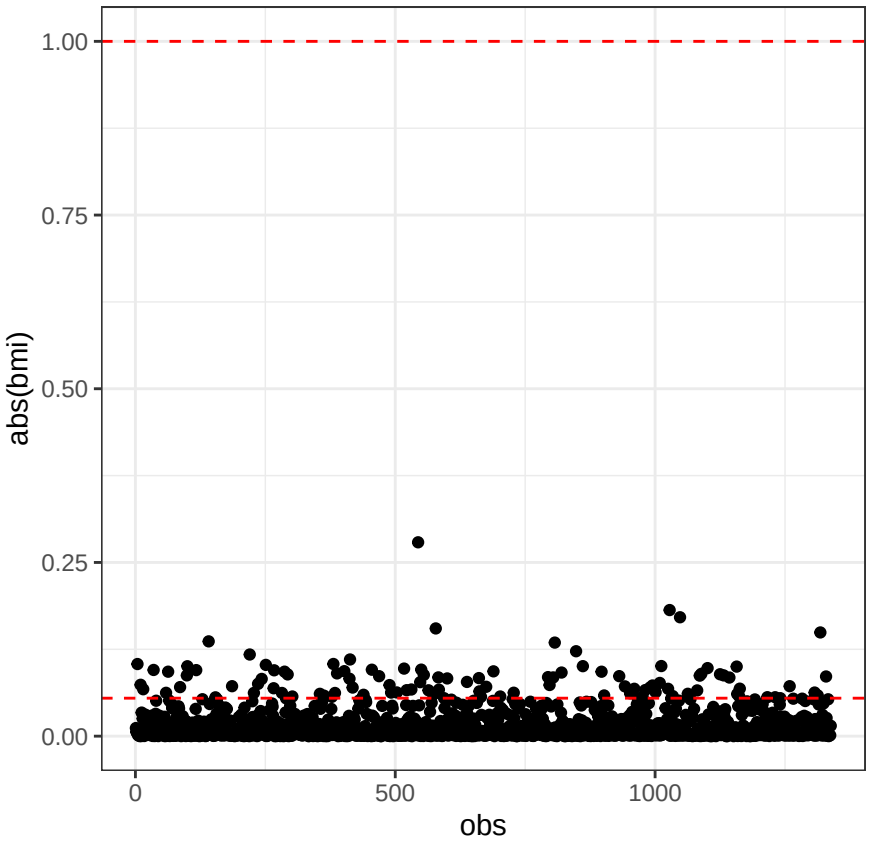

```
insurance.dffits[abs(insurance.dffits$dffits) > 1, ]
```

```
## [1] dffits obs
## <0 rows> (or 0-length row.names)
```

```
#DFBetas
insurance.dfbetas <- as.data.frame(dfbetas(sub.insurance.lm))
insurance.dfbetas$obs <- 1:length(sub.insurance$age)


ggplot(data = insurance.dfbetas) +
  geom_point(mapping = aes(x = obs, y = abs(age))) +
  geom_hline(mapping = aes(yintercept = 1),
             color = "red", linetype = 2) +
  geom_hline(mapping = aes(yintercept = 2 / sqrt(length(obs))),
             color = "red", linetype = 2) +
  theme_bw() +
  theme(aspect.ratio = 1)
```




```
ggplot(data = insurance.dfbetas) +
  geom_point(mapping = aes(x = obs, y = abs(bmi))) +
  geom_hline(mapping = aes(yintercept = 1),
             color = "red", linetype = 2) +
  geom_hline(mapping = aes(yintercept = 2 / sqrt(length(obs))),
             color = "red", linetype = 2) +
  theme_bw() +
  theme(aspect.ratio = 1)
```



Assumption 5 Conclusions:

The DFFits and DFBetas show that there are no observations marked as influential and that all data points are described in the model.

(R) Assumption 6: No other predictor variables are required.

It looks better of the assumption. The model is more accurately describing the data, and many of assumptions are looking much stronger than when first did this analysis just basing model off of age alone. However, some of the assumptions are showing trends that suggest that there are further variables that may need to be added in order for the model to reach maximum accuracy. This assumption is met for now, and I understand that other variables can be worth adding in the future.

Assumption 7: Test for Multicolinearity.

```
#vif
insurance.vif <- vif(sub.insurance.lm)
insurance.vif
```

```
##              GVIF Df GVIF^(1/(2*Df))
## age      1.018469  1        1.009192
## bmi      1.013263  1        1.006609
## children 1.012171  5        1.001210
## smoker   1.006048  1        1.003020
```

Assumption 7 Conclusions:

This assumption is met. The VIF test showed each variable as being within .01 of 1. This is awesome because values of 1 show that there is no multicollinearity between variables.

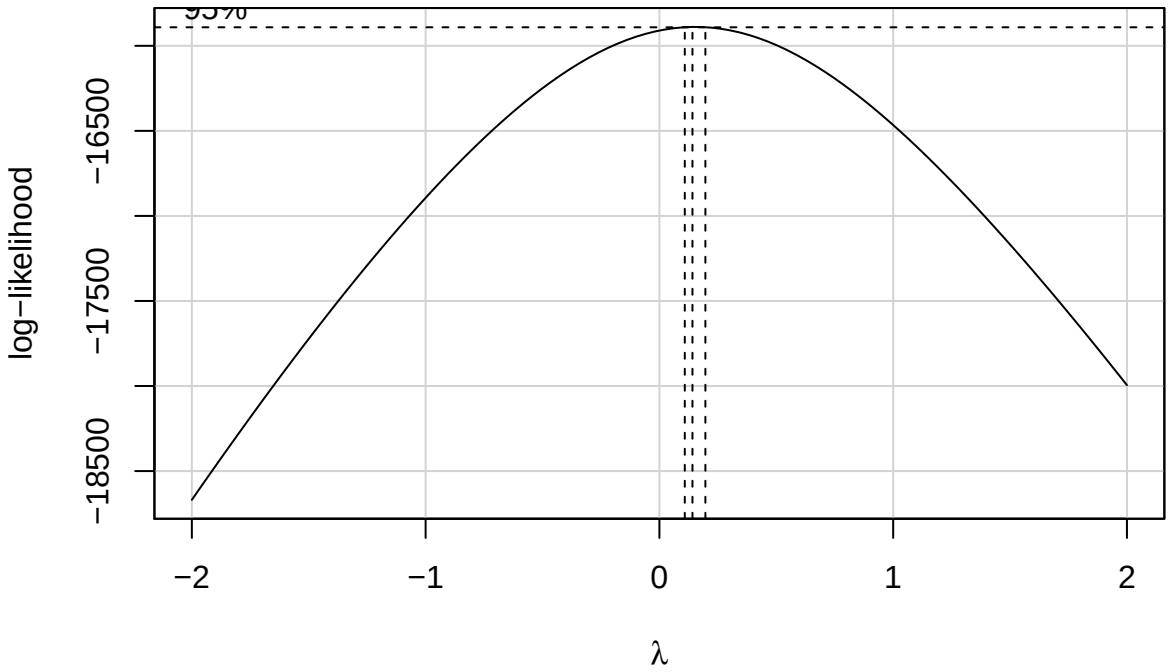
Overall Conclusions:

Additional methods need to apply to potential met the multiple linear regression assumptions. I'll start by applying a transformation to y and seeing if that helps reduce the non-normal patterns and discrepancies in the residuals.

Transformations:

Box Cox Analysis

```
bc <- boxCox(sub.insurance.lm)
```




```
bc$x[which.max(bc$y)]
```

```
## [1] 0.1414141
```

```
#Apply log() transformation to charges
sub.insurance$log.charges <- log(sub.insurance$charges)
sub.insurance.lm.trans <- lm(log.charges ~ age + bmi + children + smoker,
```

```
                              data = sub.insurance)
summary(sub.insurance.lm.trans)
```

```
##
## Call:
## lm(formula = log.charges ~ age + bmi + children + smoker, data = sub.insurance)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.09016 -0.21154 -0.04662  0.08017  2.08333
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.9692685  0.0699693  99.605  < 2e-16 ***
## age         0.0348147  0.0008794  39.588  < 2e-16 ***
## bmi         0.0104155  0.0020210   5.154 2.94e-07 ***
## children1   0.1423780  0.0311224   4.575 5.21e-06 ***
## children2   0.2792741  0.0344436   8.108 1.16e-15 ***
## children3   0.2494404  0.0404410   6.168 9.16e-10 ***
## children4   0.5197718  0.0915124   5.680 1.65e-08 ***
## children5   0.3979236  0.1072935   3.709 0.000217 ***
## smokeryes   1.5437176  0.0304196  50.748  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4477 on 1329 degrees of freedom
## Multiple R-squared:  0.7644, Adjusted R-squared:  0.763
## F-statistic:   539 on 8 and 1329 DF,  p-value: < 2.2e-16
```

```
#Add new residuals to .
sub.insurance$residuals.trans <- sub.insurance.lm.trans$residuals
sub.insurance$fitted.values.trans <- sub.insurance.lm.trans$fitted.values

#Create predictor values.
pred.vals <- seq(min(sub.insurance$age), max(sub.insurance$age), length = 1338)
preds.trans <- sub.insurance.lm.trans$coefficients[1] +
  sub.insurance.lm.trans$coefficients[2] * pred.vals
preds.orig <- exp(preds.trans)
preds <- data.frame("pred.vals" = pred.vals, "pred_orig" = preds.orig)
```

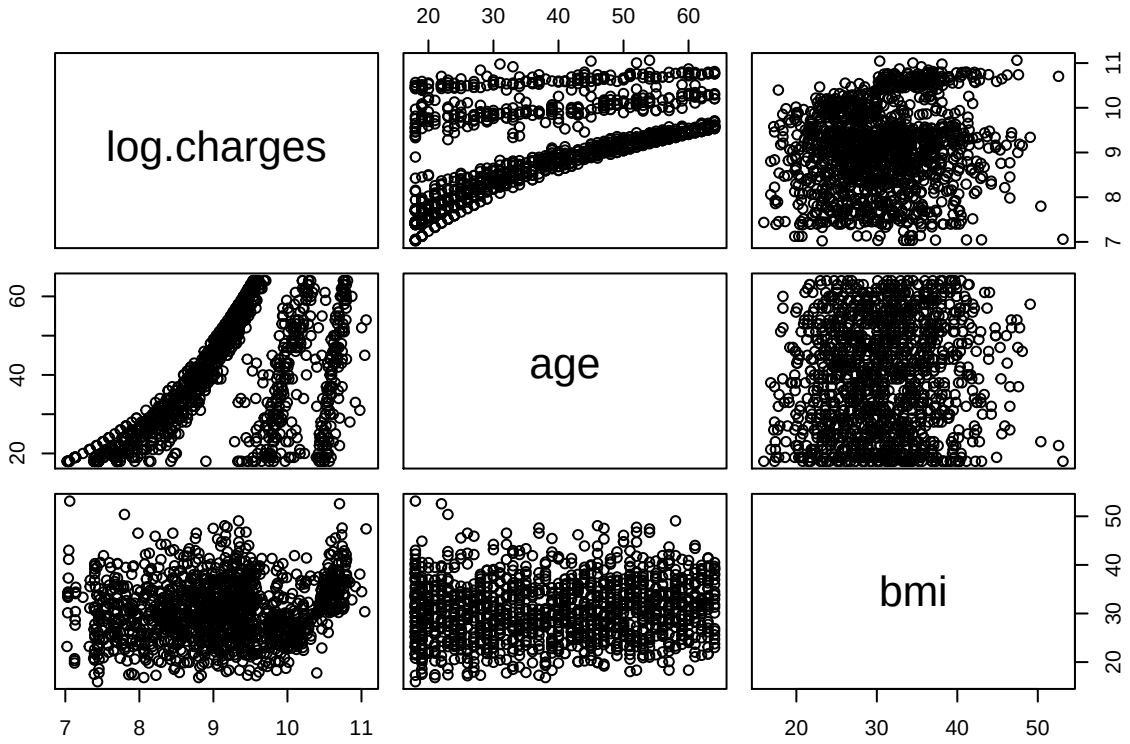
Transformation Summary:

The log transformation is applied to repsonse variable.

Re-Check Assumptions

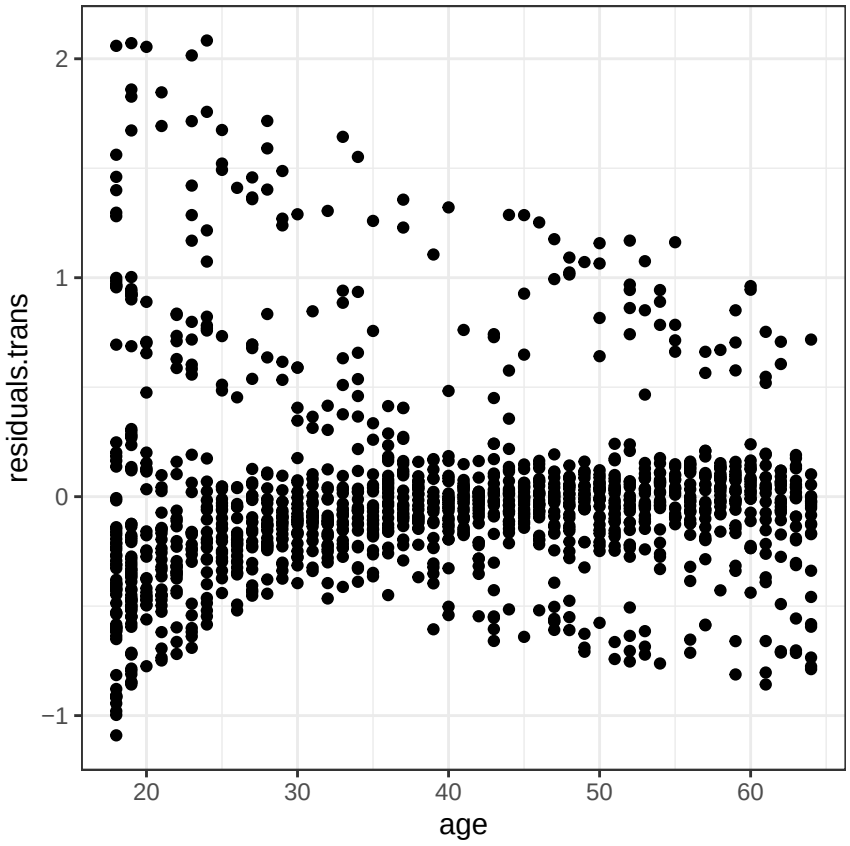
(L) Assumption 1: X vs. Y is linear (using scatter plots, partial regression plots, residuals vs. fitted values plots, and specific scatter plots that meet certain requirements).

```
#Scatter plot matrix of continuous variables.
cont.insurance.trans <- sub.insurance[, c(8, 1, 2)]
plot(cont.insurance.trans)
```

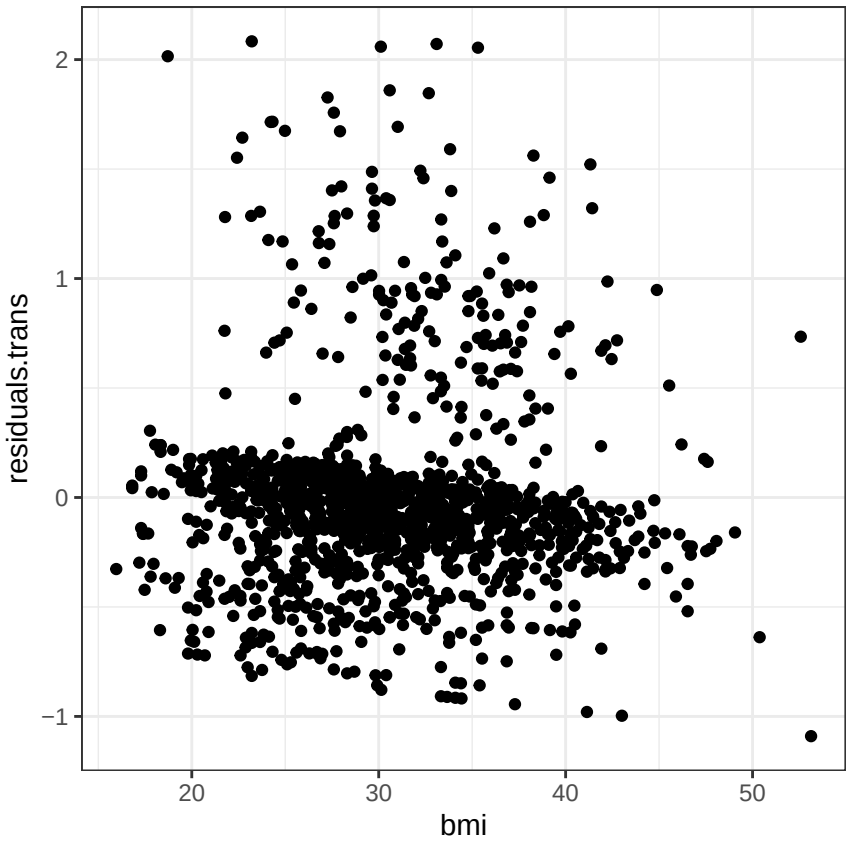



```
#Predictors vs. Residuals
#Age plot code
plot.age <- ggplot(data = sub.insurance,
                   mapping = aes(x = age, y = residuals.trans)) +
    geom_point() +
    theme_bw() +
    theme(aspect.ratio = 1)


#BMI plot code
plot.bmi <- ggplot(data = sub.insurance,
                   mapping = aes(x = bmi, y = residuals.trans)) +
    geom_point() +
    theme_bw() +
    theme(aspect.ratio = 1)

#Children plot code
plot.children <- ggplot(data = sub.insurance,
                        mapping = aes(x = children, y = residuals.trans)) +
    geom_point() +
    theme_bw() +
    theme(aspect.ratio = 1)

#Smoker plot code
plot.smoker <- ggplot(data = sub.insurance,
                      mapping = aes(x = smoker, y = residuals.trans)) +
```

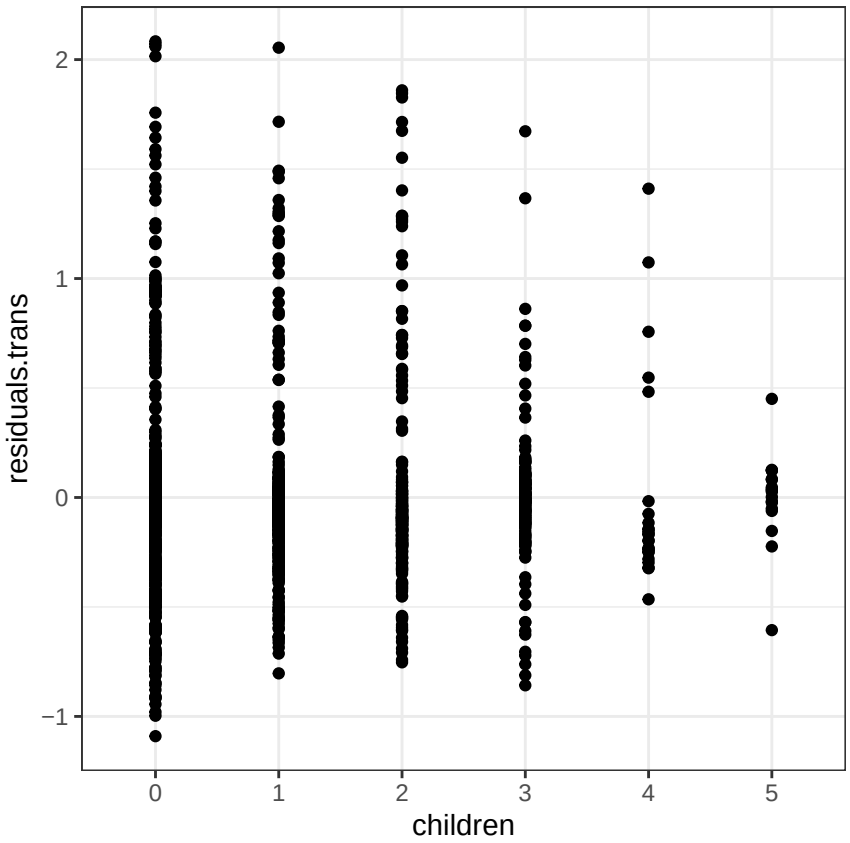
```
    geom_point() +
    theme_bw() +
    theme(aspect.ratio = 1)


plot.age
```



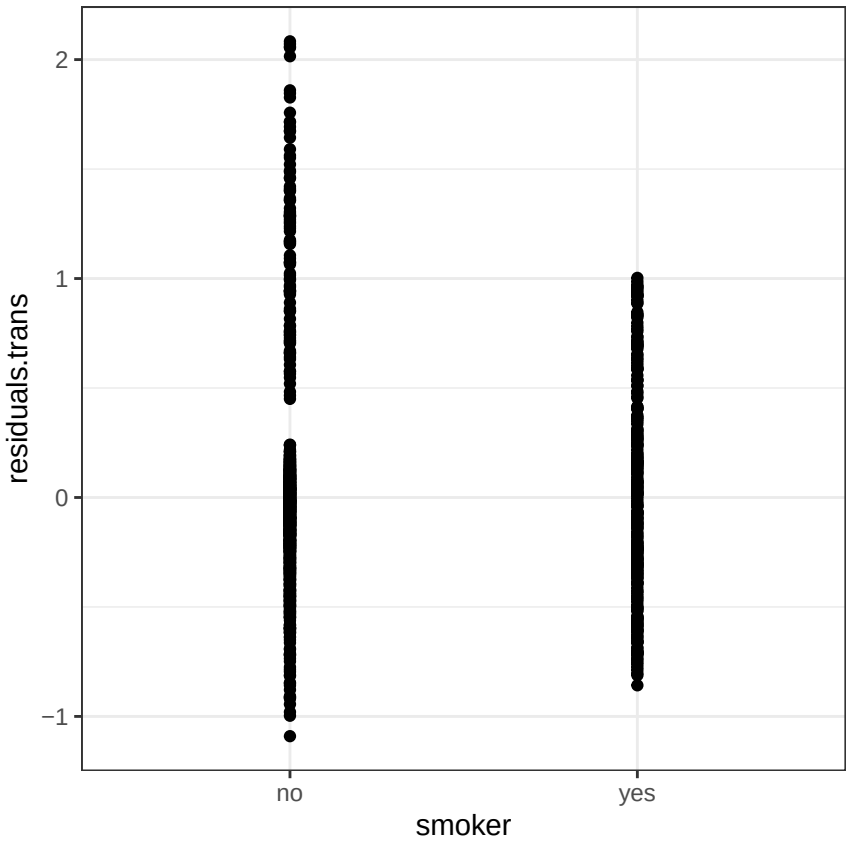

```
plot.bmi
```



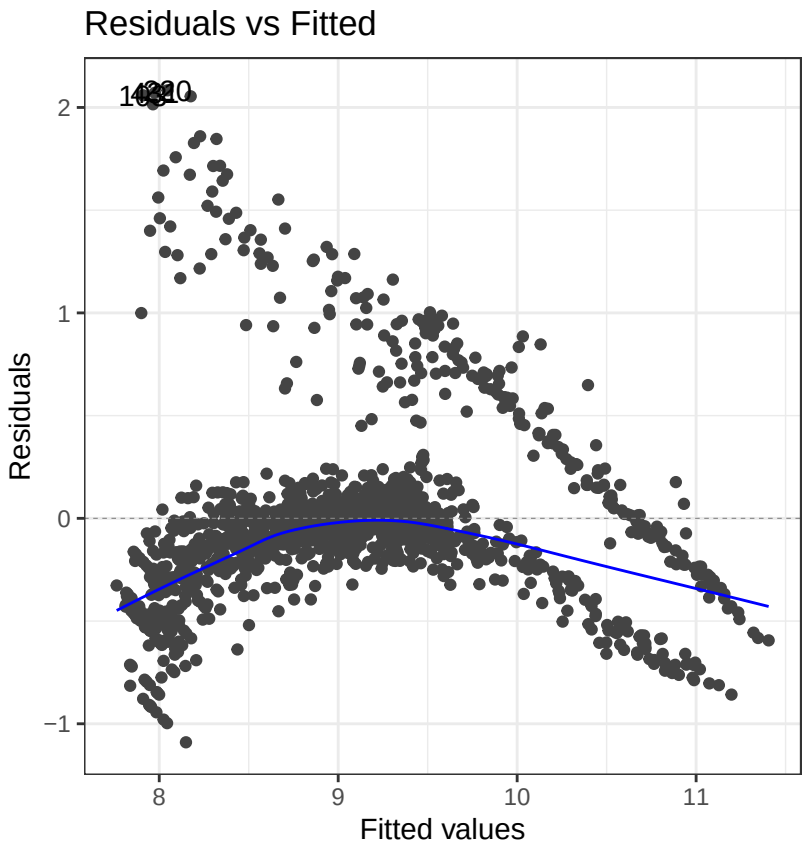

```
plot.children
```



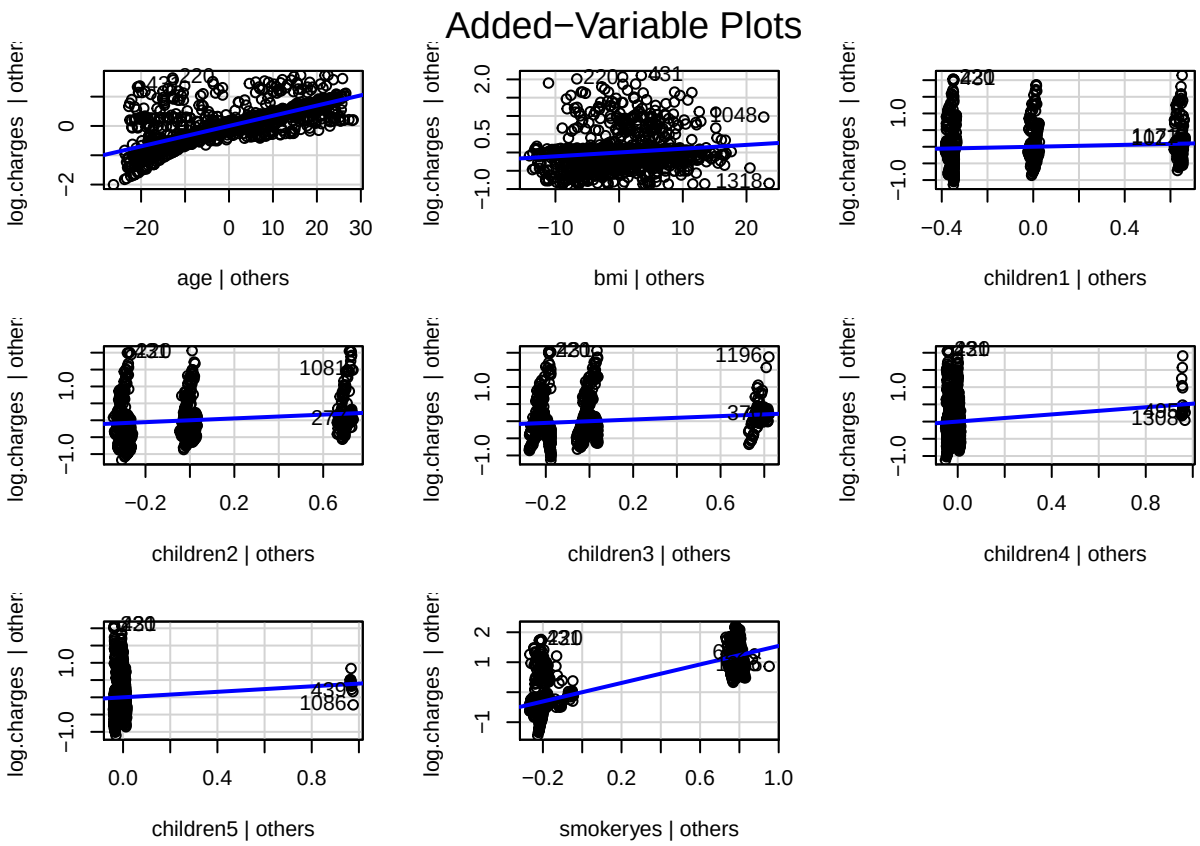

```
plot.smoker
```




```
#Residuals vs. fitted values.
residuals.plot <- autoplot(sub.insurance.lm.trans,
                           which = 1,
                           ncol = 1,
                           nrow = 1) +
  theme_bw() +
  theme(aspect.ratio = 1)

residuals.plot
```




```
#AvPlots
avPlots(sub.insurance.lm.trans)
```



Assumption 1 Conclusions:

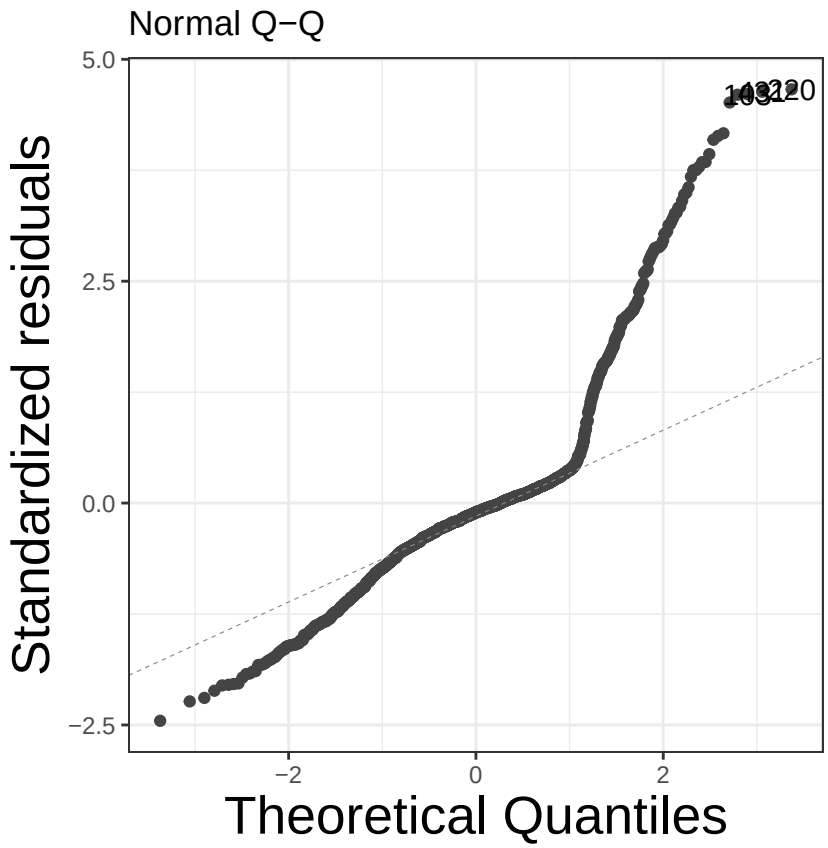
The blue lines on the partial regression plots look linear. Definately see they are seperated into groups, but they are still linear. In the bmi and children plots, there are some random points but overall the lines are linear. Overall, I conclude that the linearity assumption is met.

(I) Assumption 2: Residuals are independent.

Do not know how the data was collected, so cannot conclusively state that the residuals are independent. However, my assumption is that the residuals will be independent since the medical conditions and needs of a particular person can't influence the medical conditions and needs of another.

(N) Assumption 3: Residuals are normally distributed and centered at 0.

```
prob.plot <- autoplot(sub.insurance.lm.trans, which = 2, ncol = 1, nrow = 1) +
  theme_bw() +
  theme(aspect.ratio = 1,
        axis.title.x = element_text(size = sz),
        axis.title.y = element_text(size = sz),
        axis.title = element_text(size = sz))

prob.plot
```




```
shapiro.test(sub.insurance.lm.trans$residuals)
```

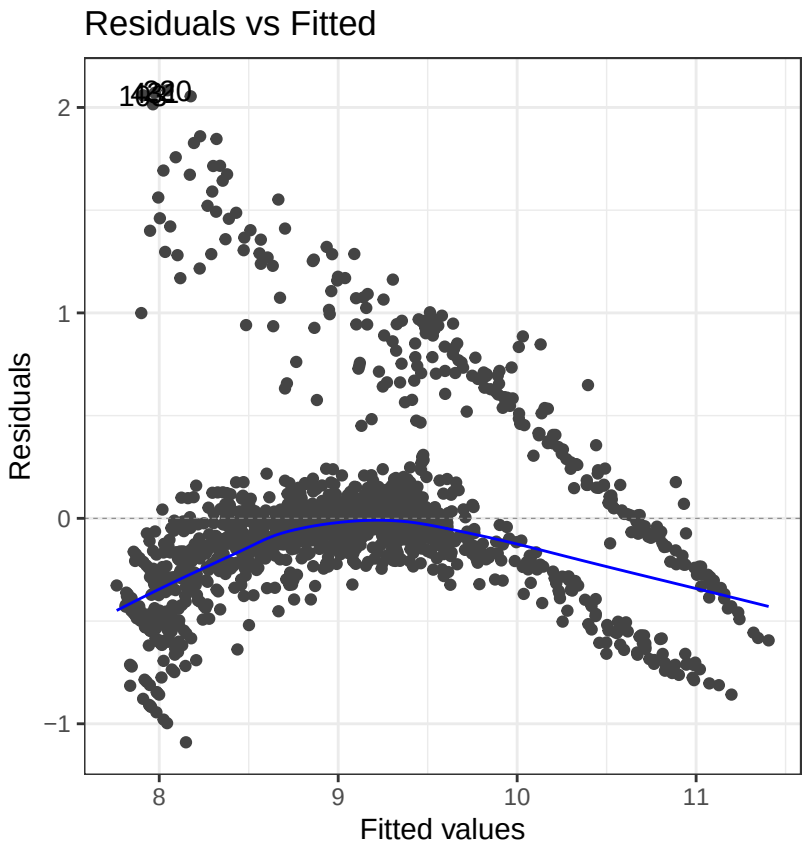
```
##
##  Shapiro-Wilk normality test
##
## data:  sub.insurance.lm.trans$residuals
## W = 0.85164, p-value < 2.2e-16
```

Assumption 3 Conclusions:

The residuals are behaving better that they were before. There are still other factors influencing the behavior of our residuals that aren't included in this model. According to the result on the Shapiro-Wilk test, the p-value is less than 0.05, so reject the null hypothesis and conclude that the residuals are not normally distributed. In the QQ plot, half of the data points are way beyond the line, and the top points can be influential points.

(E) Assumption 4: Residuals have equal variance.

```
residuals.plot
```




```
grp <- as.factor(c(rep("lower", floor(dim(insurance)[1] / 2)),
                   rep("upper", ceiling(dim(insurance)[1] / 2))))
leveneTest(sub.insurance[order(sub.insurance$age),
                         "residuals.trans"]~ grp, center = median)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value   Pr(>F)
## group    1  61.867 7.53e-15 ***
##       1336
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

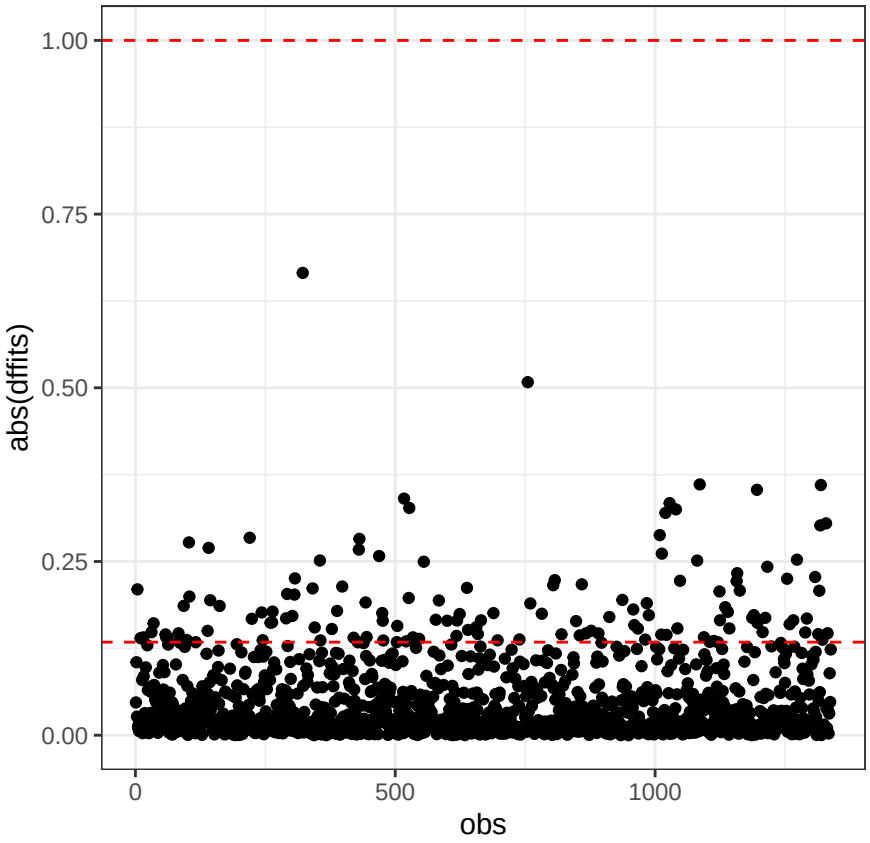
Assumption 4 Conclusions:

The result of Levene's test is significant, that means the residuals does not have equal variance. The line in residuals vs fitted values plot is skewed, and the points around the blue line are spread apart.

(A) Assumption 5: Model describes ALL observations.

```
#DFFits
insurance.dffits <- data.frame ("dffits" = dffits(sub.insurance.lm.trans))
insurance.dffits$obs <- 1:length(sub.insurance$age)

ggplot(data = insurance.dffits) +
```

```
  geom_point(mapping = aes(x = obs, y = abs(dffits))) +
  geom_hline(mapping = aes(yintercept = 1),
             color = "red", linetype = 2) +
  geom_hline(mapping = aes(yintercept = 2 * sqrt(6 / length(obs))),
             color = "red", linetype = 2) +
  theme_bw() +
  theme(aspect.ratio = 1)
```

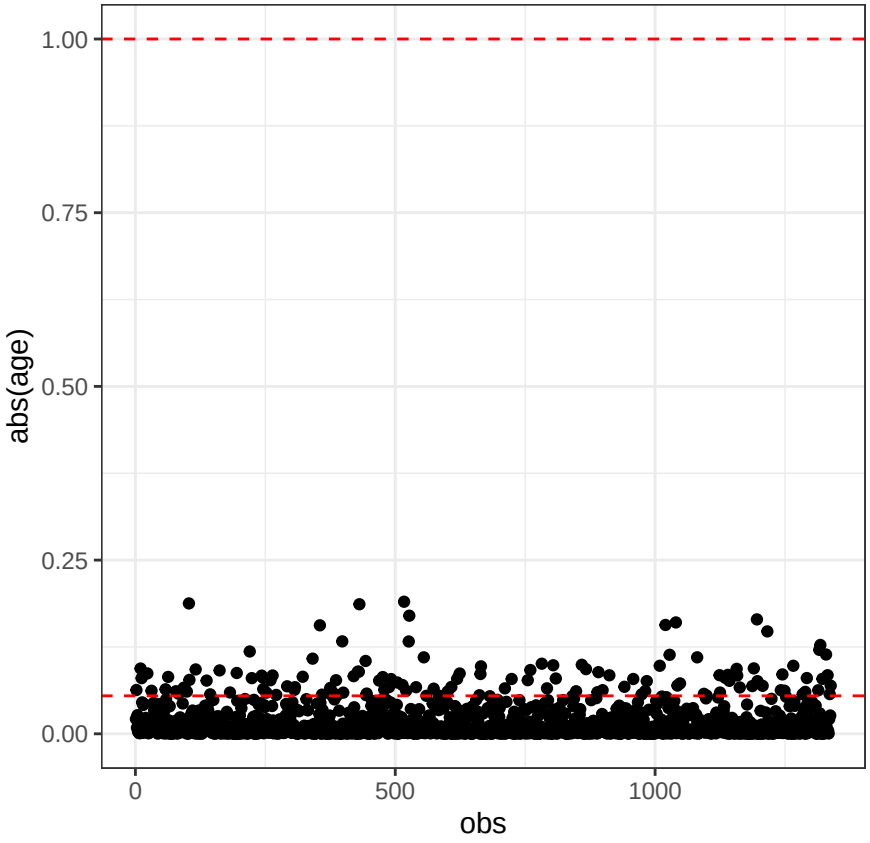



```
insurance.dffits[abs(insurance.dffits$dffits) > 1, ]
```

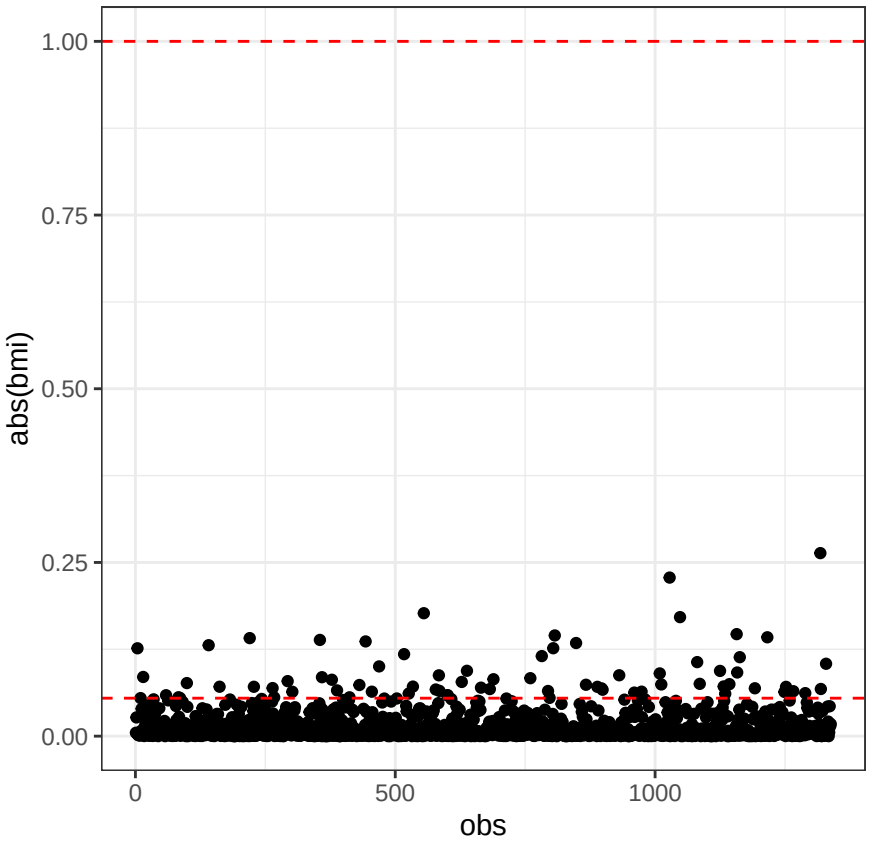
```
## [1] dffits obs
## <0 rows> (or 0-length row.names)
```

```
#DFBetas
insurance.dfbetas <- as.data.frame(dfbetas(sub.insurance.lm.trans))
insurance.dfbetas$obs <- 1:length(sub.insurance$age)


ggplot(data = insurance.dfbetas) +
  geom_point(mapping = aes(x = obs, y = abs(age))) +
  geom_hline(mapping = aes(yintercept = 1),
             color = "red", linetype = 2) +
  geom_hline(mapping = aes(yintercept = 2 / sqrt(length(obs))),
             color = "red", linetype = 2) +
```

```
  theme_bw() +
  theme(aspect.ratio = 1)
```




```
ggplot(data = insurance.dfbetas) +
  geom_point(mapping = aes(x = obs, y = abs(bmi))) +
  geom_hline(mapping = aes(yintercept = 1),
             color = "red", linetype = 2) +
  geom_hline(mapping = aes(yintercept = 2 / sqrt(length(obs))),
             color = "red", linetype = 2) +
  theme_bw() +
  theme(aspect.ratio = 1)
```



Assumption 5 Conclusions:

Similiar with before the transformation. The DFFits and DFBetas show that there are no observations marked as influential and that all data points are described in our model. This is much better where we ran an analysis and saw a couple hundred different points that were marked as influential, because there were predictor variables that were not included.

(R) Assumption 6: No other predictor variables are required.

This time there are no other predictor variables that aren't accounted for. There are some anomallies that need to be handled through a transformation, so I'll work on transforming the model to improve how well it describes our data.

Assumption 7: Test for Multicolinearity.

```
#vif
insurance.vif <- vif(sub.insurance.lm.trans)
insurance.vif
```

```
##              GVIF Df GVIF^(1/(2*Df))
## age      1.018469  1        1.009192
## bmi      1.013263  1        1.006609
## children 1.012171  5        1.001210
## smoker   1.006048  1        1.003020
```

Assumption 7 Conclusions:

This assumption is still met. The VIF test showed each variable as being within .01 of 1. This is awesome because values of 1 show that there is no multicolinearity between variables.

Overall Conclusions:

With the transformation, the assumptions are better met, but still are not as good as they could be. The residuals are not normally distributed and they do not have equal variance. The rest of the assumptions are met.

Model Evaluations

Confidence Interval and Hypothesis Test for the Slope

```
confint(sub.insurance.lm, level = .95)
```

```
##                     2.5 %      97.5 %
## (Intercept) -13952.63420 -10234.0156
## age            234.70712    281.4449
## bmi            266.10150    373.5078
## children1     -458.24985   1195.7919
## children2      711.23308   2541.7860
## children3      -77.69683   2071.5991
## children4      552.58186   5416.1353
## children5    -1952.00322   3750.2619
## smokeryes    22988.36518  24605.0553
```

Confidence Interval of insurance cost for someone who is 40 years old, has a bmi of 32, with 2 children, and smokes.

```
predict(sub.insurance.lm, newdata = data.frame(age = 40, bmi = 32, children = "2", smoker = "yes"),
        interval = "confidence", level = 0.95)
```

```
##        fit      lwr      upr
## 1 33886.69 32896.22 34877.15
```

Preiction Interval for someone who is 40 years old, has a bmi of 32, with 2 children, and smokes.

```
predict(sub.insurance.lm, newdata = data.frame(age = 40,
                                               bmi = 32,
                                               children = "2",
                                               smoker = "yes"),
        interval = "prediction", level = 0.95)
```

```
##        fit     lwr      upr
## 1 33886.69 21949.4 45823.97
```

MSE

```r
anova <- aov(sub.insurance.lm.trans)
mse <- summary(anova)[[1]][2, 2] / summary(anova)[[1]][2,1]
mse
```

```
## [1] 6.434382
```

RMSE

```r
sqrt(mse)
```

```
## [1] 2.536608
```

MAE

```r
#fill in variable with own still
sum(abs(sub.insurance$residuals.trans)) / (length(sub.insurance$charges) -2)
```

```
## [1] 0.2852043
```

R-Squared

```r
summary(sub.insurance.lm.trans)$r.squared
```

```
## [1] 0.7643933
```

Adjusted R-Squared

```r
summary(sub.insurance.lm.trans)$adj.r.squared
```

```
## [1] 0.762975
```

F-Statistic

```r
summary(sub.insurance.lm.trans)
```

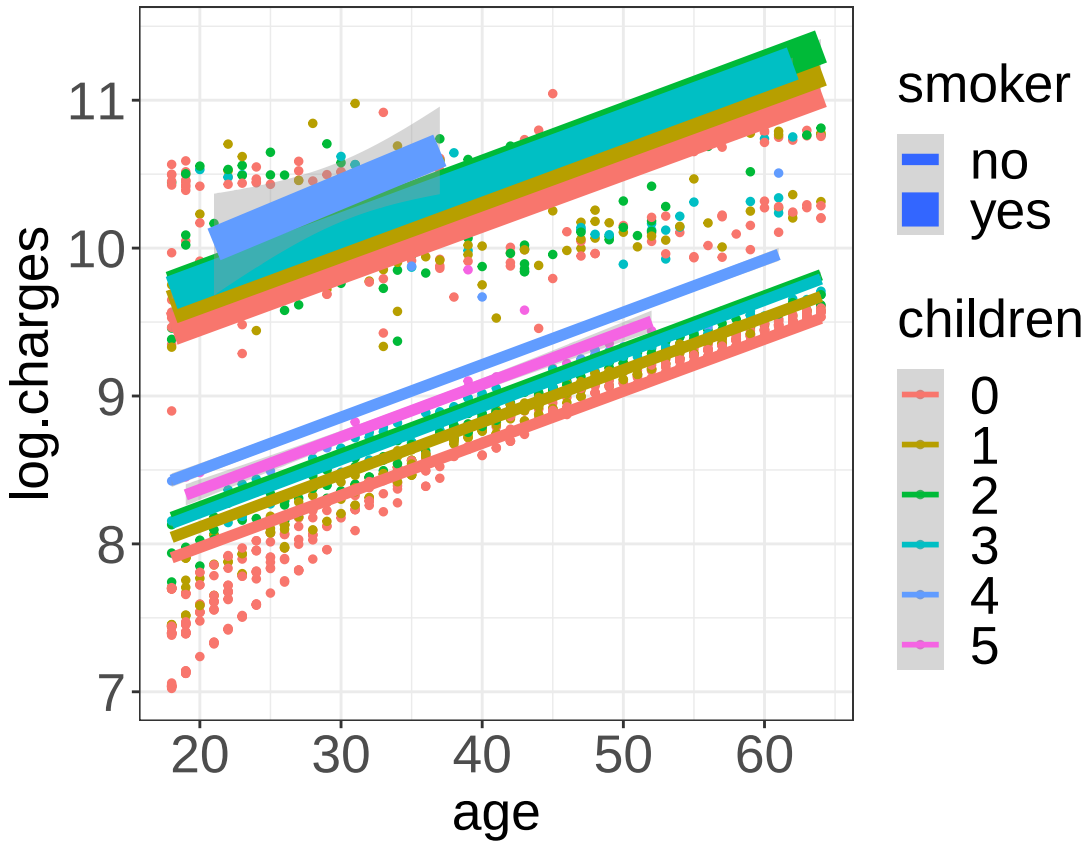
```
##
## Call:
## lm(formula = log.charges ~ age + bmi + children + smoker, data = sub.insurance)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.09016 -0.21154 -0.04662  0.08017  2.08333
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.9692685  0.0699693  99.605  < 2e-16 ***
## age         0.0348147  0.0008794  39.588  < 2e-16 ***
```

```
## bmi         0.0104155  0.0020210   5.154 2.94e-07 ***
## children1   0.1423780  0.0311224   4.575 5.21e-06 ***
## children2   0.2792741  0.0344436   8.108 1.16e-15 ***
## children3   0.2494404  0.0404410   6.168 9.16e-10 ***
## children4   0.5197718  0.0915124   5.680 1.65e-08 ***
## children5   0.3979236  0.1072935   3.709 0.000217 ***
## smokeryes   1.5437176  0.0304196  50.748  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4477 on 1329 degrees of freedom
## Multiple R-squared:  0.7644, Adjusted R-squared:  0.763
## F-statistic:   539 on 8 and 1329 DF,  p-value: < 2.2e-16
```

```
fitted.plot <- ggplot(data = sub.insurance, aes(x = age,
                                                y = log.charges,
                                                color = children,
                                                size = smoker)) +
  geom_point(size = 1) +
  theme_bw() +
  geom_smooth(method = "lm",
              mapping = aes(y = predict(sub.insurance.lm.trans))) +
  theme(axis.title.x = element_text(size = sz),
        axis.title.y = element_text(size = sz),
        axis.text = element_text(size = sz),
        legend.text = element_text(size = sz),
        legend.title = element_text(size = sz),
        aspect.ratio = 1)
fitted.plot
```

```
## Warning: Using size for a discrete variable is not advised.
```

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




```
### If you decide the plot looks a little crowded, delete "size = smoker" and replace "color = children
```

Evaluation Metrics Conclusions:

After all the transformation, the R-squared and adjusted R-square values are all close to 1, which means that a better model is fitted. About 76% of variability in medical charges are explained by the variables after accounting for predictors in the model. The confidence intervals are also very informative. For example, we are 95% confidence that the average medical charges increase between 234.71 and 281.44 dollars, for every additional year in age. According to the F-statistics p-value, I can conclude that at least one variable is useful at predicting overall charges.

Overall Summary & Conclusions:

Understanding how age, gender, bmi, number of children and smoking status characteristics contribute to the person's medical charges can be critical to understand their medial costs and making good decisions about a person's health. I conducted an analysis to determine which of these types of variable significantly affect medical charges. After fitting a multiple linear regression model, that many of predictor variables, do, indeed, have a significant negative impact on medical charges. To meet more assumptions, a transformation is applied to multiple linear regression model, and this has made assumptions better met. The Adjusted R-squared value suggested that about 76% of variability in medical charges are explained by the variables after accounting for predictors in the model. I can definately do more transformation to make model better, but it is better than what I started with.

Apendix: Code

```
library(tidyverse)
library(car)
library(ggfortify)
library(bestglm)
library(glmnet)
library(corrplot)
library(dplyr)
sz <- 20
insurance <- read.csv("insurance.csv", header = TRUE)

# Look at a summary of the data to make sure that it makes sense.
summary(insurance)

# Subset the data to make a scatter plot matrix of all the continuous variables
cont.insurance <- select(insurance, c("age", "bmi", "charges"))
plot(cont.insurance, pch = 19)

# Make a correlation matrix of the continuous variables.
cor(cont.insurance)

# Boxplots of all other categorical variables to get an idea of what is going on.
box.sex <- ggplot(data = insurance,
                  mapping = aes(x = sex, y = charges)) +
    geom_boxplot() +
    theme(aspect.ratio = 1)
box.sex

insurance$children <- as.factor(insurance$children)
box.children <- ggplot(data = insurance,
                       mapping = aes(x = children, y = charges)) +
    geom_boxplot() +
    theme(aspect.ratio = 1)
box.children


box.smoker <- ggplot(data = insurance,
                     mapping = aes(x = smoker, y = charges)) +
    geom_boxplot() +
    theme(aspect.ratio = 1)
box.smoker


box.region <- ggplot(data = insurance,
                     mapping = aes(x = region, y = charges)) +
    geom_boxplot() +
    theme(aspect.ratio = 1)
box.region



#Convert all variables to factors or numerical.
```

```r
insurance$children <- as.factor(insurance$children)
insurance$sex <- as.factor(insurance$sex)
insurance$smoker <- as.factor(insurance$smoker)
insurance$region <- as.factor(insurance$region)
best.subsets.bic <- bestglm(insurance,
                            IC = "BIC",
                            method = "exhaustive",
                            TopModels = 10)

summary(best.subsets.bic$BestModel)

forward.aic <- bestglm(insurance,
                            IC = "AIC",
                            method = "forward",
                            TopModels = 10)

summary(forward.aic$BestModel)
backward.aic <- bestglm(insurance,
                        IC = "AIC",
                        method = "backward",
                        TopModels = 10)

summary(backward.aic$BestModel)
seqrep.aic <- bestglm(insurance,
                        IC = "AIC",
                        method = "seqrep",
                        TopModels = 10,
                        t=100)
summary(seqrep.aic$BestModel)
#Convert elements to be represented as numbers and then change them to be factors
insurance <- read.csv("insurance.csv", header = TRUE)
insurance$smoker <- ifelse(insurance$smoker == "no", 0, 1)
insurance$sex <- ifelse(insurance$sex == "female", 0, 1)

for(i in 1:length(insurance$region)){
  if(insurance[i, 6] == "southwest"){
    insurance[i, 6] <- 1
  } else if(insurance[i, 6] == "northwest"){
    insurance[i, 6] <- 2
  } else if(insurance[i, 6] == "northeast"){
    insurance[i, 6] <- 3
  } else if(insurance[i, 6] == "southeast"){
    insurance[i, 6] <- 4
  }
}

#Make sure continuous variables are continuous
insurance$age <- as.numeric(insurance$age)
insurance$bmi <- as.numeric(insurance$bmi)

#Convert all categorical variables back to factors.
insurance$sex <- as.factor(insurance$sex)
insurance$children <- as.factor(insurance$children)
```

```
insurance$smoker <- as.factor(insurance$smoker)
insurance$region <- as.factor(insurance$region)
# make a matrix for our covariates and pull out response as its own variable
insurance.x <- data.matrix(insurance[, c(1:6)])
insurance.y <- insurance[, 7]

# Lasso (alpha = 1)
insurance.lasso <- glmnet(x = insurance.x, y = insurance.y, alpha = 1)

# use cross validation to pick the "best" lambda (based on MSE)
insurance.lasso.cv <- cv.glmnet(x = insurance.x, y = insurance.y,
                          type.measure = "mse", alpha = 1)

# lambda.min is the value of lambda that gives minimum mean cross-validated
# error
insurance.lasso.cv$lambda.min
# lambda.1se gives the most regularized model such that error is within one
# standard error of the minimum
insurance.lasso.cv$lambda.1se

# coefficients (betas) using a specific lambda penalty value
coef(insurance.lasso.cv, s = "lambda.min")
coef(insurance.lasso.cv, s = "lambda.1se")
# make a matrix for our covariates and pull out response as its own variable
insurance.x <- data.matrix(insurance[, 1:6])
insurance.y <- insurance[, 7]

# Elastic Net (alpha = .5)
insurance.elastic <- glmnet(x = insurance.x, y = insurance.y, alpha = .5)

# use cross validation to pick the "best" lambda (based on MSE)
insurance.elastic.cv <- cv.glmnet(x = insurance.x, y = insurance.y,
                          type.measure = "mse", alpha = .5)

# lambda.min is the value of lambda that gives minimum mean cross-validated
# error
insurance.elastic.cv$lambda.min
# lambda.1se gives the most regularized model such that error is within one
# standard error of the minimum
insurance.elastic.cv$lambda.1se

# coefficients (betas) using a specific lambda penalty value
coef(insurance.elastic.cv, s = "lambda.min")
coef(insurance.elastic.cv, s = "lambda.1se")
#Read in a fresh set of the data and convert the necessary data to factors
insurance <- read.csv("insurance.csv", header = TRUE)
insurance$children <- as.factor(insurance$children)
insurance$smoker <- as.factor(insurance$smoker)


sub.insurance <- insurance[, c(1,3,4,5,7)]
sub.insurance$children <- as.factor(sub.insurance$children)
```

```r
sub.insurance.lm <- lm(charges ~ age + bmi + children + smoker,
                       data = sub.insurance)
summary(sub.insurance.lm)

#add residuals and fitted values to dataframe.
sub.insurance$residuals <- sub.insurance.lm$residuals
sub.insurance$fitted.values <- sub.insurance.lm$fitted.values
#Scatter plot matrix of continuous variables.
plot(cont.insurance)


#Partial Regression plots
#Age plot code
plot.age <- ggplot(data = sub.insurance,
                   mapping = aes(x = age, y = residuals)) +
    geom_point() +
    theme_bw() +
    theme(aspect.ratio = 1)


#BMI plot code
plot.bmi <- ggplot(data = sub.insurance,
                   mapping = aes(x = bmi, y = residuals)) +
    geom_point() +
    theme_bw() +
    theme(aspect.ratio = 1)

#Children plot code
plot.children <- ggplot(data = sub.insurance,
                        mapping = aes(x = children, y = residuals)) +
    geom_point() +
    theme_bw() +
    theme(aspect.ratio = 1)

#Smoker plot code
plot.smoker <- ggplot(data = sub.insurance,
                      mapping = aes(x = smoker, y = residuals)) +
    geom_point() +
    theme_bw() +
    theme(aspect.ratio = 1)


plot.age
plot.bmi
plot.children
plot.smoker

#Residuals vs. fitted values.
residuals.plot <- autoplot(sub.insurance.lm, which = 1, ncol = 1, nrow = 1) +
  theme_bw() +
  theme(aspect.ratio = 1)

residuals.plot
```

```r
#Specific scatter plots
plot(charges ~ age, data = subset(sub.insurance, smoker == "yes"))

#AV Plots
avPlots(sub.insurance.lm)
prob.plot <- autoplot(sub.insurance.lm, which = 2, ncol = 1, nrow = 1) +
  theme_bw() +
  theme(aspect.ratio = 1,
        axis.title.x = element_text(size = sz),
        axis.title.y = element_text(size = sz),
        axis.title = element_text(size = sz))

prob.plot

shapiro.test(sub.insurance.lm$residuals)
residuals.plot

grp <- as.factor(c(rep("lower", floor(dim(insurance)[1] / 2)),
                   rep("upper", ceiling(dim(insurance)[1] / 2))))
leveneTest(sub.insurance[order(sub.insurance$age),
                         "residuals"] ~ grp, center = median)
#DFFits
insurance.dffits <- data.frame ("dffits" = dffits(sub.insurance.lm))
insurance.dffits$obs <- 1:length(sub.insurance$age)

ggplot(data = insurance.dffits) +
  geom_point(mapping = aes(x = obs, y = abs(dffits))) +
  geom_hline(mapping = aes(yintercept = 1),
             color = "red", linetype = 2) +
  geom_hline(mapping = aes(yintercept = 2 * sqrt(6 / length(obs))),
             color = "red", linetype = 2) +
  theme_bw() +
  theme(aspect.ratio = 1)

insurance.dffits[abs(insurance.dffits$dffits) > 1, ]


#DFBetas
insurance.dfbetas <- as.data.frame(dfbetas(sub.insurance.lm))
insurance.dfbetas$obs <- 1:length(sub.insurance$age)


ggplot(data = insurance.dfbetas) +
  geom_point(mapping = aes(x = obs, y = abs(age))) +
  geom_hline(mapping = aes(yintercept = 1),
             color = "red", linetype = 2) +
  geom_hline(mapping = aes(yintercept = 2 / sqrt(length(obs))),
             color = "red", linetype = 2) +
  theme_bw() +
  theme(aspect.ratio = 1)

ggplot(data = insurance.dfbetas) +
  geom_point(mapping = aes(x = obs, y = abs(bmi))) +
```

```
  geom_hline(mapping = aes(yintercept = 1),
             color = "red", linetype = 2) +
  geom_hline(mapping = aes(yintercept = 2 / sqrt(length(obs))),
             color = "red", linetype = 2) +
  theme_bw() +
  theme(aspect.ratio = 1)

#vif
insurance.vif <- vif(sub.insurance.lm)
insurance.vif
bc <- boxCox(sub.insurance.lm)
bc$x[which.max(bc$y)]

#Apply log() transformation to charges
sub.insurance$log.charges <- log(sub.insurance$charges)
sub.insurance.lm.trans <- lm(log.charges ~ age + bmi + children + smoker,
                             data = sub.insurance)
summary(sub.insurance.lm.trans)

#Add new residuals to .
sub.insurance$residuals.trans <- sub.insurance.lm.trans$residuals
sub.insurance$fitted.values.trans <- sub.insurance.lm.trans$fitted.values

#Create predictor values.
pred.vals <- seq(min(sub.insurance$age), max(sub.insurance$age), length = 1338)
preds.trans <- sub.insurance.lm.trans$coefficients[1] +
  sub.insurance.lm.trans$coefficients[2] * pred.vals
preds.orig <- exp(preds.trans)
preds <- data.frame("pred.vals" = pred.vals, "pred_orig" = preds.orig)



#Scatter plot matrix of continuous variables.
cont.insurance.trans <- sub.insurance[, c(8, 1, 2)]
plot(cont.insurance.trans)

#Predictors vs. Residuals
#Age plot code
plot.age <- ggplot(data = sub.insurance,
                   mapping = aes(x = age, y = residuals.trans)) +
    geom_point() +
    theme_bw() +
    theme(aspect.ratio = 1)


#BMI plot code
plot.bmi <- ggplot(data = sub.insurance,
                   mapping = aes(x = bmi, y = residuals.trans)) +
    geom_point() +
    theme_bw() +
    theme(aspect.ratio = 1)

#Children plot code
```

```
plot.children <- ggplot(data = sub.insurance,
                        mapping = aes(x = children, y = residuals.trans)) +
    geom_point() +
    theme_bw() +
    theme(aspect.ratio = 1)

#Smoker plot code
plot.smoker <- ggplot(data = sub.insurance,
                      mapping = aes(x = smoker, y = residuals.trans)) +
    geom_point() +
    theme_bw() +
    theme(aspect.ratio = 1)


plot.age
plot.bmi
plot.children
plot.smoker

#Residuals vs. fitted values.
residuals.plot <- autoplot(sub.insurance.lm.trans,
                           which = 1,
                           ncol = 1,
                           nrow = 1) +
  theme_bw() +
  theme(aspect.ratio = 1)

residuals.plot

#AvPlots
avPlots(sub.insurance.lm.trans)

prob.plot <- autoplot(sub.insurance.lm.trans, which = 2, ncol = 1, nrow = 1) +
  theme_bw() +
  theme(aspect.ratio = 1,
        axis.title.x = element_text(size = sz),
        axis.title.y = element_text(size = sz),
        axis.title = element_text(size = sz))

prob.plot

shapiro.test(sub.insurance.lm.trans$residuals)
residuals.plot

grp <- as.factor(c(rep("lower", floor(dim(insurance)[1] / 2)),
                   rep("upper", ceiling(dim(insurance)[1] / 2))))
leveneTest(sub.insurance[order(sub.insurance$age),
                         "residuals.trans"]~ grp, center = median)
#DFFits
insurance.dffits <- data.frame ("dffits" = dffits(sub.insurance.lm.trans))
insurance.dffits$obs <- 1:length(sub.insurance$age)

ggplot(data = insurance.dffits) +
```

```
  geom_point(mapping = aes(x = obs, y = abs(dffits))) +
  geom_hline(mapping = aes(yintercept = 1),
             color = "red", linetype = 2) +
  geom_hline(mapping = aes(yintercept = 2 * sqrt(6 / length(obs))),
             color = "red", linetype = 2) +
  theme_bw() +
  theme(aspect.ratio = 1)

insurance.dffits[abs(insurance.dffits$dffits) > 1, ]


#DFBetas
insurance.dfbetas <- as.data.frame(dfbetas(sub.insurance.lm.trans))
insurance.dfbetas$obs <- 1:length(sub.insurance$age)


ggplot(data = insurance.dfbetas) +
  geom_point(mapping = aes(x = obs, y = abs(age))) +
  geom_hline(mapping = aes(yintercept = 1),
             color = "red", linetype = 2) +
  geom_hline(mapping = aes(yintercept = 2 / sqrt(length(obs))),
             color = "red", linetype = 2) +
  theme_bw() +
  theme(aspect.ratio = 1)

ggplot(data = insurance.dfbetas) +
  geom_point(mapping = aes(x = obs, y = abs(bmi))) +
  geom_hline(mapping = aes(yintercept = 1),
             color = "red", linetype = 2) +
  geom_hline(mapping = aes(yintercept = 2 / sqrt(length(obs))),
             color = "red", linetype = 2) +
  theme_bw() +
  theme(aspect.ratio = 1)

#vif
insurance.vif <- vif(sub.insurance.lm.trans)
insurance.vif
confint(sub.insurance.lm, level = .95)

predict(sub.insurance.lm, newdata = data.frame(age = 40, bmi = 32, children = "2", smoker = "yes"),
        interval = "confidence", level = 0.95)

predict(sub.insurance.lm, newdata = data.frame(age = 40,
                                               bmi = 32,
                                               children = "2",
                                               smoker = "yes"),
        interval = "prediction", level = 0.95)
anova <- aov(sub.insurance.lm.trans)
mse <- summary(anova)[[1]][2, 2] / summary(anova)[[1]][2,1]
mse
sqrt(mse)
#fill in variable with own still
sum(abs(sub.insurance$residuals.trans)) / (length(sub.insurance$charges) -2)
```

```r
summary(sub.insurance.lm.trans)$r.squared

summary(sub.insurance.lm.trans)$adj.r.squared
summary(sub.insurance.lm.trans)
fitted.plot <- ggplot(data = sub.insurance, aes(x = age,
                                                y = log.charges,
                                                color = children,
                                                size = smoker)) +
  geom_point(size = 1) +
  theme_bw() +
  geom_smooth(method = "lm",
              mapping = aes(y = predict(sub.insurance.lm.trans))) +
  theme(axis.title.x = element_text(size = sz),
        axis.title.y = element_text(size = sz),
        axis.text = element_text(size = sz),
        legend.text = element_text(size = sz),
        legend.title = element_text(size = sz),
        aspect.ratio = 1)
fitted.plot

### If you decide the plot looks a little crowded, delete "size = smoker" and replace "color = children
```