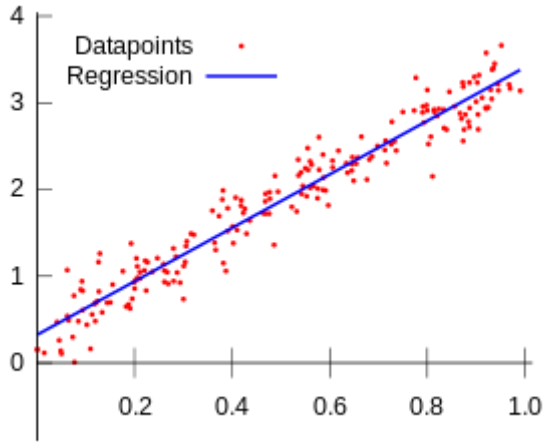
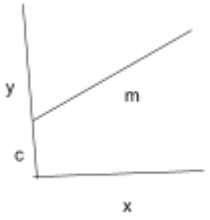
Linear regression in R

Intro video: https://www.youtube.com/watch?v=ZkjP5RJLQF4&t=788s
Intro doc: http://onlinestatbook.com/2/regression/intro.html

In statistics, linear regression is a linear approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X.



A linear regression line has an equation of the form Y = a + bX, where X is the explanatory variable and Y is the dependent variable. The slope of the line is b, and a is the intercept (the value of y when x = 0). Ordinary least squares linear regression is the most widely used type of regression for predicting the value of one dependent variable from



the value of one independent variable.[1]

Purpose of linear regression to bring down cost function:
Cost function = ordinary least squares/mean squared error = sum[sq(Yi - mXi + c)]
Steps to calculate minima of cost function:

- Find m and x using differentiation (differentiation gives us minimum value of cost function)

[1] http://web.csulb.edu/~msaintg/ppa696/696regmx.htm. Accessed 1 Sep. 2017.

- Perform T-test[2] for m and x
- Get best value of m and x by previous step

The t-test assesses whether the means of two groups are statistically different from each other. This analysis is appropriate whenever you want to.

Hypothesis Test for Regression Slope:
http://stattrek.com/regression/slope-test.aspx?Tutorial=AP

Note:
Use **T-test** to check effect of variables

Multi-linear regression:
We use partial differentiation to solve more than 1 unknown (ex: effect of age on height, keeping weight as constant)

Linear regression assumption:

More reference:
https://www.analyticsvidhya.com/blog/2016/07/deeper-regression-analysis-assumptions-plots-solutions/
https://www.analyticsvidhya.com/blog/2015/10/regression-python-beginners/

Pre - diagnosis assumptions:

1. Target value must follow normal distribution. Log, Square root and inverse (3 ways to bring input to normal distribution).
2. Linear relationship
3. Independence i.e., no multi-collinearity (use correlation matrix to check if inputs are related (>0.6))

Post - diagnosis assumptions:

1. **Model validation**
2. **Model diagnosis**
3. **Model assessment**

1. **Model validation:**

Coefficient of Determination (R Squared): Definition, Calculation[3]

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.

---

[2] "Student's t-test - Wikipedia." https://en.wikipedia.org/wiki/Student%27s_t-test. Accessed 1 Sep. 2017.
[3] "Calculating R-squared (video) | Khan Academy." https://www.khanacademy.org/math/statistics-probability/describing-relationships-quantitative-data/more-on-regression/v/calculating-r-squared. Accessed 4 Sep. 2017.

R^2 > 0.6 is good model
R^2 = 1 - RSS/TSS

RSS is residual sum of squares
TSS is sum of squares

Coefficient of Determination (R Squared)

The coefficient of determination, R2, is used to analyze how differences in one variable can be explained by a difference in a second variable. For example, *when* a person gets pregnant has a direct relation to when they give birth. The coefficient of determination is similar to the correlation coefficient, R. The correlation coefficient formula will tell you how strong of a linear relationship there is between two variables. R Squared is the square of the correlation coefficient, *r* (hence the term r squared). Watch this video for a short definition of r squared and how to find it:

Finding R Squared / The Coefficient of Determination

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[\,n\Sigma x^2 - (\Sigma x)^2\,]\,[\,n\Sigma y^2 - (\Sigma y)^2\,]}}$$

That's it!

Meaning of the Coefficient of Determination

The coefficient of determination can be thought of as a percent. It gives you an idea of how many data points fall within the results of the line formed by the regression equation. The higher the coefficient, the higher percentage of points the line passes through when the data points and line are plotted. If the coefficient is 0.80, then 80% of the points should fall within the regression line. Values of 1 or 0 would indicate the regression line represents all or none of the data, respectively. A higher coefficient is an indicator of a better goodness of fit for the observations.

The CoD can be negative, although this usually means that your model is a poor fit for your data. It can also become negative if you didn't set an intercept.

Usefulness of R2

The usefulness of R2 is its ability to find the likelihood of future events falling within the predicted outcomes. The idea is that if more samples are added, the coefficient would show the probability of a new point falling on the line.

Even if there is a strong connection between the two variables, determination does not prove causality. For example, a study on birthdays may show a large number of birthdays happen within a

time frame of one or two months. This does not mean that the passage of time or the change of seasons causes pregnancy.

Adjusted R square:

Adjusted R squared is a corrected goodness-of-fit (model accuracy) measure for linear models. It identifies the percentage of variance in the target field that is explained by the input or inputs.

R2 tends to optimistically estimate the fit of the linear regression. It always increases as the number of effects are included in the model. Adjusted R2 attempts to correct for this overestimation. It might decrease if the effect does not improve the model.

Adjusted R squared is calculated by dividing the residual mean square error by the total mean square error (which is the sample variance of the target field). The result is then subtracted from 1.

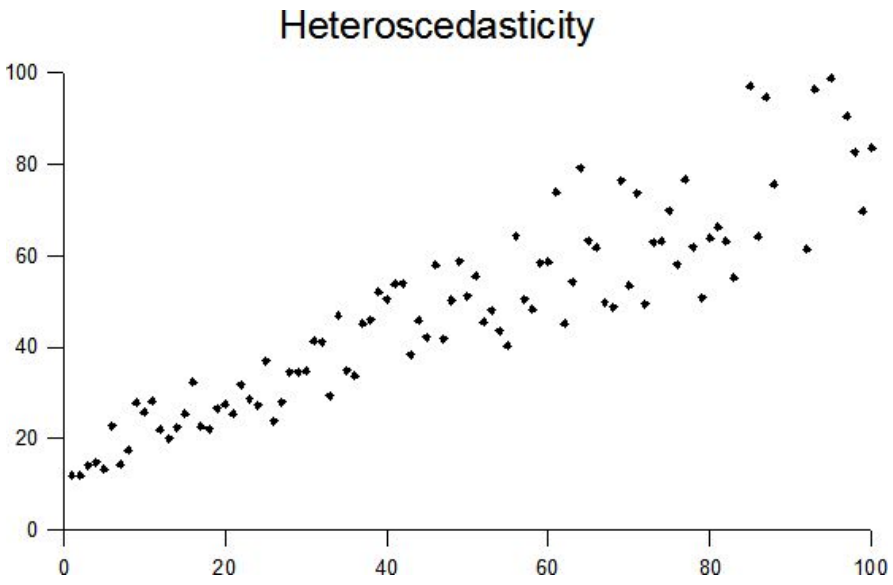
Adjusted R2 is always less than or equal to R2. A value of 1 indicates a model that perfectly predicts values in the target field. A value that is less than or equal to 0 indicates a model that has no predictive value. In the real world, adjusted R2 lies between these values.

2. Model diagnosis

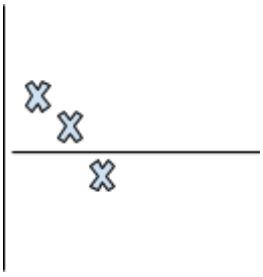
Assumptions:

- **1. Normality of errors:**
    - Use histogram or QQ plot to determine this
- **2. No heteroscedasticity[4]:**
    - Use scatter plot between residuals and predicted values. Model is good if errors are unbiased. [Ex: if predicted value is 60 Kg and error is 1 Kg and if predicted value is 90Kg and error is 10 kg then it is not a good model, see below graph. This is also known as non-constant variance - this problem is known as heteroscedasticity, opposite of this is known as homoscedasticity (which is good which means variance is error is constant)].

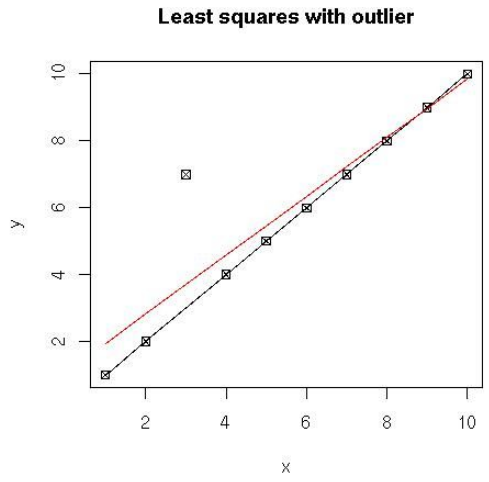
[4] "Confusing Stats Terms Explained: Heteroscedasticity ...." 22 Apr. 2013, http://www.statsmakemecry.com/smmctheblog/confusing-stats-terms-explained-heteroscedasticity-heteroske.html. Accessed 4 Sep. 2017.



- **3. No auto correlation:**
    - Errors should not be correlated with each other. Can be analyzed using scatter plot of residuals and predicted values. Perform derbin-watson test. Ex:



- **4. No Outliers:**
    - See below graph (red line is caused by single outlier)



- **5. No Leverage:**
  - Similar to outlier, example: a car whose odometer is 1.5 L and price is 60K when rest of cars with 60K price has odometer reading around 0.5 L. Leverage can be measured using cook's distance, cutoff for cook's distance is
  - 4/(n-k-1)
  - (n is number of training observations, k is number of variables in the model)

- **6. No multicollinearity:**
  - **Can be measured using VIF (variance inflation factor)**

Okay, now that we know the effects that multicollinearity can have on our regression analyses and subsequent conclusions, how do we tell when it exists? That is, how can we tell if multicollinearity is present in our data?

Some of the common methods used for detecting multicollinearity include:

- *The analysis exhibits the signs of multicollinearity — such as, estimates of the coefficients vary from model to model.*
- *The t-tests for each of the individual slopes are non-significant (P > 0.05), but the overall F-test for testing all of the slopes are simultaneously 0 is significant (P < 0.05).*
- *The correlations among pairs of predictor variables are large.*

Looking at correlations only among pairs of predictors, however, is limiting. It is possible that the pairwise correlations are small, and yet a linear dependence exists among three or even more variables, for example, if $X_3 = 2X_1 + 5X_2$ + error, say. That's why many regression analysts often rely on what are called variance inflation factors (VIF) to help detect multicollinearity.

What is a Variation Inflation Factor?

As the name suggests, a variance inflation factor (VIF) quantifies how much the variance is inflated. But what variance? Recall that we learned previously that the standard errors — and hence the variances — of the estimated coefficients are inflated when multicollinearity exists. So, the variance

inflation factor for the estimated coefficient b_k—denoted VIF_k—is just the factor by which the variance is inflated.

R-formula for linear regression:

Develop the model:

```
lin_model = lm(log_charges ~ ., data=train) #'.' means include all the variables
```

Typical output:

```
Call:
lm(formula = log_charges ~ ., data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-1.1818 -0.1228  0.0228  0.1185  1.0685

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      7.665e+00  5.909e-02 129.708  < 2e-16 ***
age              2.102e-02  7.692e-04  27.326  < 2e-16 ***
sexmale         -6.377e-02  1.863e-02  -3.423 0.000643 ***
bmi             -4.517e-03  1.717e-03  -2.631 0.008641 **
children         8.121e-02  7.757e-03  10.469  < 2e-16 ***
smokeryes        2.994e-01  4.331e-02   6.912 8.21e-12 ***
regionnorthwest -4.680e-02  2.670e-02  -1.753 0.079940 .
regionsoutheast -1.201e-01  2.708e-02  -4.435 1.02e-05 ***
regionsouthwest -8.375e-02  2.688e-02  -3.115 0.001887 **
charges          5.248e-05  1.533e-06  34.226  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3031 on 1060 degrees of freedom
Multiple R-squared:  0.8915, Adjusted R-squared:  0.8906
F-statistic:   968 on 9 and 1060 DF,  p-value: < 2.2e-16
```

Predict the output:

```
test$pred = predict(lin_model, newdata=test)
```

Useful commands for diagnosis

```
fit = lm(log_charges ~., data = train)
summary (fit)
```

Check if errors are following normal distribution:

The residual data of the simple linear regression model is the difference between the observed data of the dependent variable y and the fitted values ŷ.

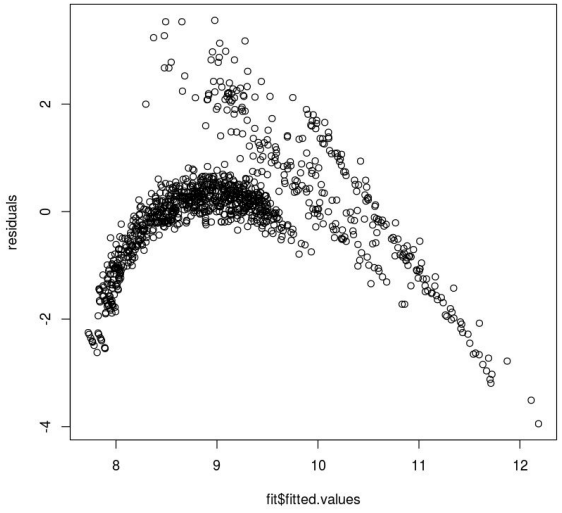
```
hist(fit$residuals)
fit$coefficients
```

Calculate standard residuals:

```
library(MASS) #need to calculate standard residuals
residuals = stdres(fit) #standardized residuals
summary (residuals)
```

Check for autocorrelation and heteroscedasticity

```
# predicted value vs fitted value
plot(fit$fitted.values,residuals)
#See below there is a problem w.r.t autocorrelation and heteroscedasticity (even
though model is good)
```



Check for autocorrelation

```
durbinWatsonTest (fit)
```

if p-value is less than 0.05 reject null hypothesis

Outlier test

```
outlierTest(fit) #remove outliers and re-train
```

Note:

Find outliers in EDA (this for pre-model building stats):

- Using box plot
- Anything more or less than 3 standard deviations

Removing outliers and leverages, example:

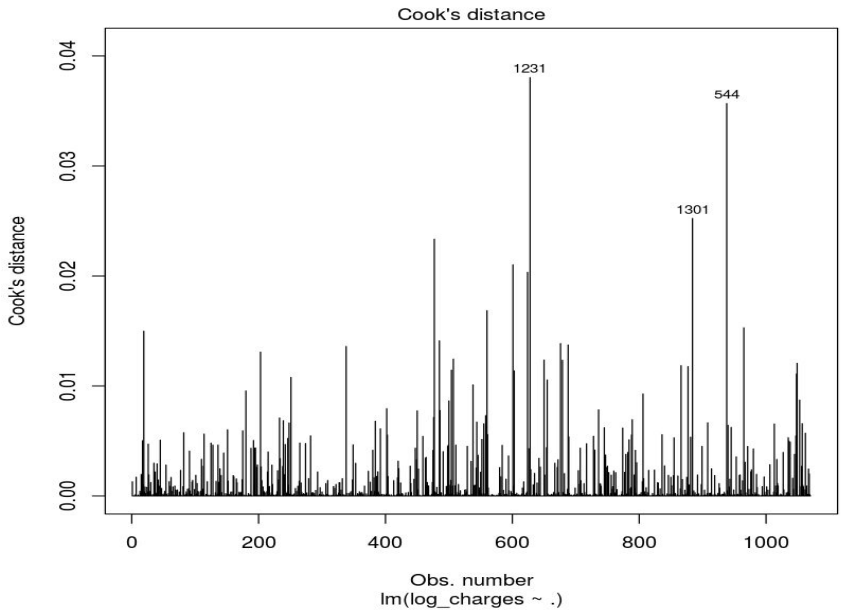
```
train = train [-c(431,220,1028,1040,103,527,1020),]
```

Leverages:

```
cd = cooks.distance(fit)
cutoff = 4 /(nrow(train) - length(fit$coefficients))
```

-- Any value more than cutoff value (0.00377 - in below case) should not be taken into consideration (see below graph; note: remove these and retrain)

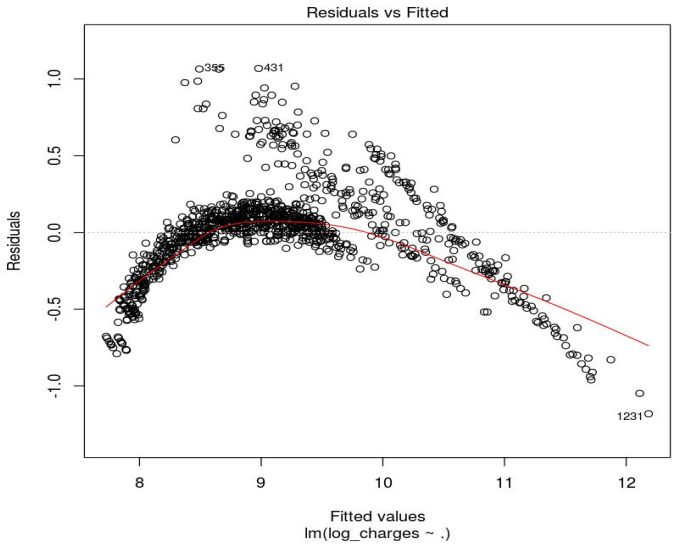
```
plot(fit,which=4,cook.levels = cutoff)
```



Plots:

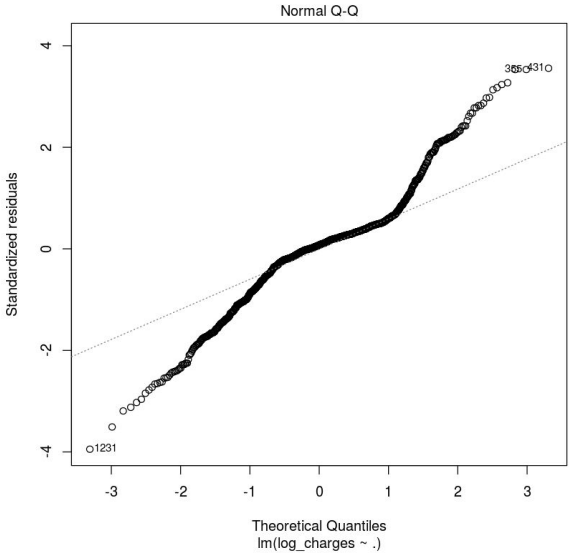
1. **autocorrelation and heteroscedasticity**

```
plot(fit, which=1) # see it shows both issue with autocorrelation and
heteroscedasticity
```



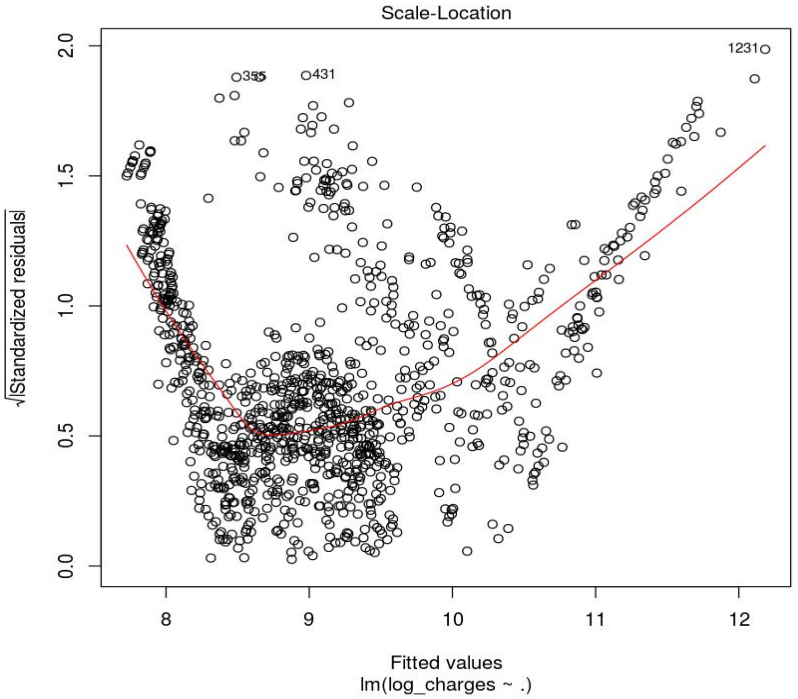
2. QQ plot

```
# if errors are following normal distribution then all circles should be on dotted line
plot(fit, which=2)
```



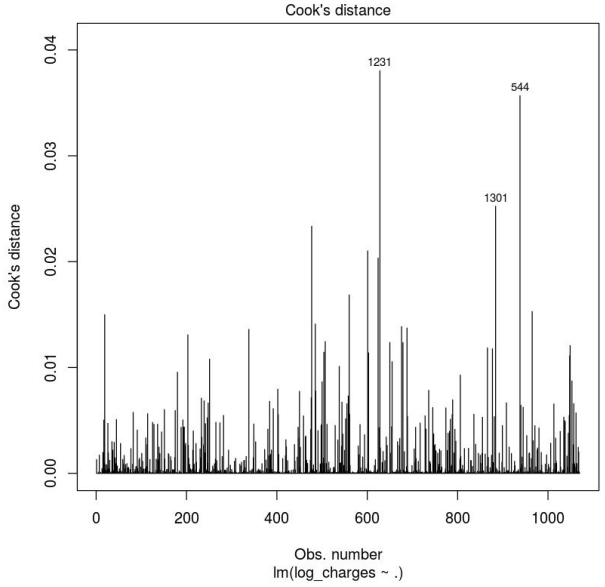
3. Autocorrelation with standard residuals

```
plot(fit, which=3) # autocorrelation plot with standard residuals
```



4. **Leverages using cook's formula**

```
plot(fit, which=4,levels=cutoff) # Leverages using cooks
```



General plot

```
ggplot(train, aes(bmi, log_charges)) + geom_point()
```

Complete linear regression code

```
setwd("D:/AP/linear")

insurance = read.csv("insurance.csv")
head(insurance)

### pre checks

### target variable should follow normal distribution
hist(insurance$charges)

### target variable is right skewed
## apply transformation
insurance$log_charges = log( insurance$charges)
hist(insurance$log_charges)

### linear relationship b/w input and target
plot(insurance$age, insurance$log_charges)
plot(insurance$age , insurance$charges)

### correlation b/w input and target variable
cor( insurance$age, insurance$log_charges) ## Correlation with taarget
variabel


### multicollinearity ( input variables are correlated with each other)
cor( insurance$age, insurance$bmi) ### cor with input variable ( not
desirable)
ggplot( insurance, aes( smoker, log_charges)) + geom_boxplot()

### Model building
insurance$charges = NULL

## train and test set split
set.seed(675)
ids = sample( nrow(insurance), nrow(insurance)*0.8)
train = insurance[ids,]
test = insurance[-ids,]

## model
lin_model = lm( log_charges ~ . , data=train )
summary(lin_model)

## Test the model
test$pred = predict(lin_model, newdata=test )
```

```r
summary(lin_model)

### RMSE
test$error = test$log_charges - test$pred
test$error_sq = test$error ** 2
rmse = sqrt(mean(test$error_sq))
rmse
summary(test$log_charges)
0.43/9.13

###### Diagnosis ####################

### select only a few variable
fit = lm(log_charges  ~  . , data=train)

#### correlation check or Multicollinearity
summary(fit)
names(fit)
fit$coefficients
head(fit$residuals)

### check for normality of errors
hist(fit$residuals)

### check for autocorrelation and heteroscedasticity
library(MASS) ## need to calculate standardised residuals
residuals = stdres(fit) ## standardised residuals
summary(residuals)

### predicted values vs. fitted values
plot(fit$fitted.values, residuals )

### statistical test for autocorrelation
durbinWatsonTest(fit)  ## to check autocorrelation

##### Outliers test
outlierTest(fit)

### leverage statistics ( cooks.distance)
 cd = cooks.distance(fit)
 cutoff = 4/( nrow(train) - length(fit$coefficients) )

 ### plot for finding the obs which has high leverage(using cd)
 plot(fit, which=4, cook.levels=cutoff)
```

```r
#Outlier and leverage observations
 431, 220, 1028, 1040, 103,527, 1020,
### Variance inflation factor to check multicollinearity
vif(fit)

 ## rebuild the model by removing outlier observations
 train = train[ -c( 431, 220, 1028, 1040, 103,527, 1020), ]

 ## fit the model
 fit = lm(log_charges  ~  . -bmi , data=train)
 summary(fit)

 ## check if Autocorrelation or Heteroscedasticity has improved
 plot(fit, which=3)

 ##  explore the relationship b/w target and input
 ggplot( train, aes( bmi, log_charges)) + geom_point()
 ggplot( train, aes( age, log_charges)) + geom_point()

### combine bmi and age into a new variable
 train$age_bmi = sqrt(train$bmi/train$age)
 ggplot( train, aes( sqrt(age_bmi), log_charges)) + geom_point()

#### final model with ratio of age and bmi
fit = lm( log_charges ~ . -bmi -age,  data=train)
summary(fit)
train$bmi.age = log( train$bmi*train$age)
ggplot( train, aes( sqrt(bmi.age), log_charges)) + geom_point()
fit = lm( log_charges ~ . -bmi -age -age_bmi, data=train)
summary(fit)
plot(fit, which = 1)
```

Ref:
http://tutorials.iq.harvard.edu/R/Rstatistics/Rstatistics.html
http://tutorials.iq.harvard.edu/R/RProgramming/Rprogramming.html#final_england_and_wales_data_cleanup
http://tutorials.iq.harvard.edu/R/RProgramming/Short-refcard.pdf

Interview questions on regression:
https://www.analyticsvidhya.com/blog/2016/12/45-questions-to-test-a-data-scientist-on-regression-skill-test-regression-solution/