Georgia Tech | H. Milton Stewart School of Industrial and Systems Engineering

College of Engineering

ISyE 6414 | *Regression Analysis*

Computer Project

Vidyut Rao | 903511348 | Nov 2nd, 2019

Regression Analysis of Systolic Blood Pressure

Languages used: *Python* and *R*

Overview:

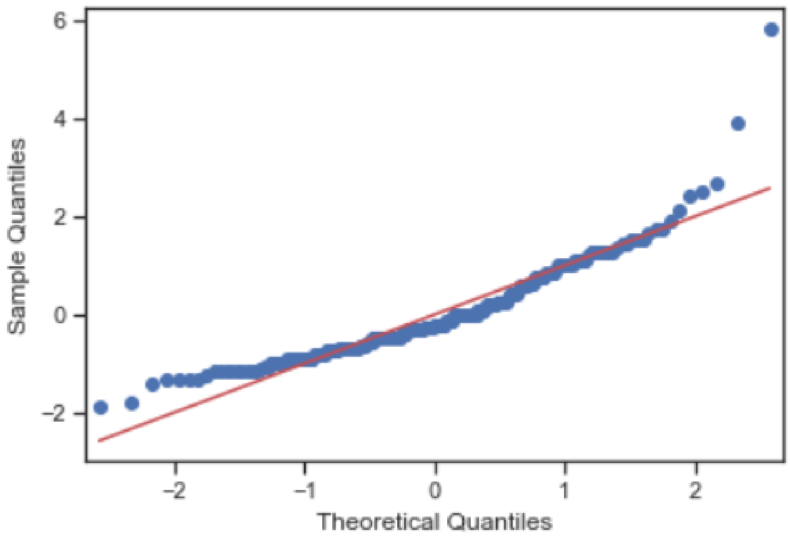
In this problem, we perform data analysis on a dataset containing the blood pressure data of 199 individuals. We begin with some explanatory data analysis, succeeded by building regression models based on stepwise selection and all subsets selection. The objective is to predict Systolic Blood Pressure from the 8 given explanatory variables. We use Python for exploratory data analysis and feature engineering and move on to R for variable selection.

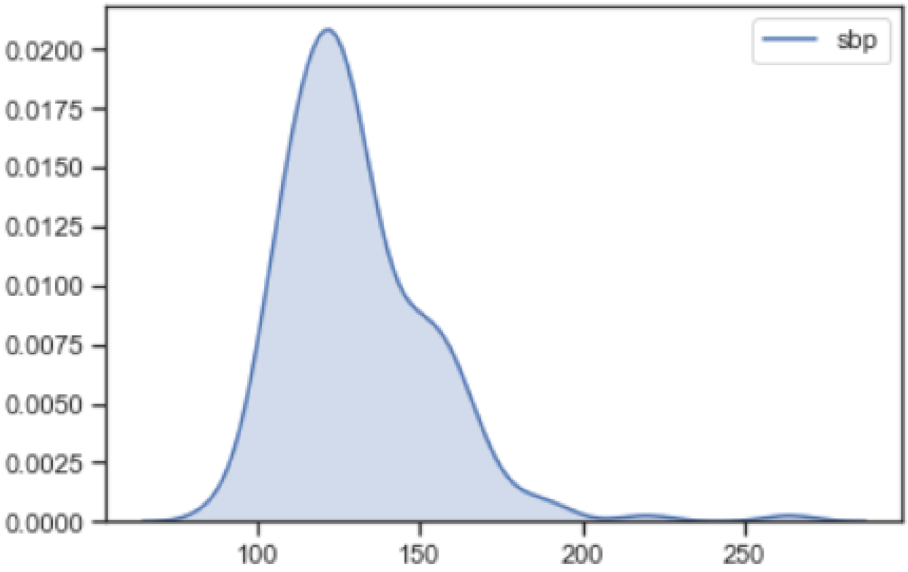
Variable	Description
sex	Gender of patient. "1" is "man"
sbp (target)	Systolic Blood Pressure
dbp	Diastolic Blood Pressure
scl	Serum Cholesterol
chdfate	Coronary Heart Disease ("1" if patient has it)
followup	Follow up in days
age	Age in years
bmi	Body Mass Index
month	Study Month of Baseline Exam

Exploratory Data Analysis:

Firstly, we'll check to see if the response variable *sbp* is normally distributed. The figure below shows the density and Normal QQ plot of the response variable. The plots show the data is skewed to the right with a thin tail. A Box-Cox transformation is necessary to transform the data into a near normal distribution.

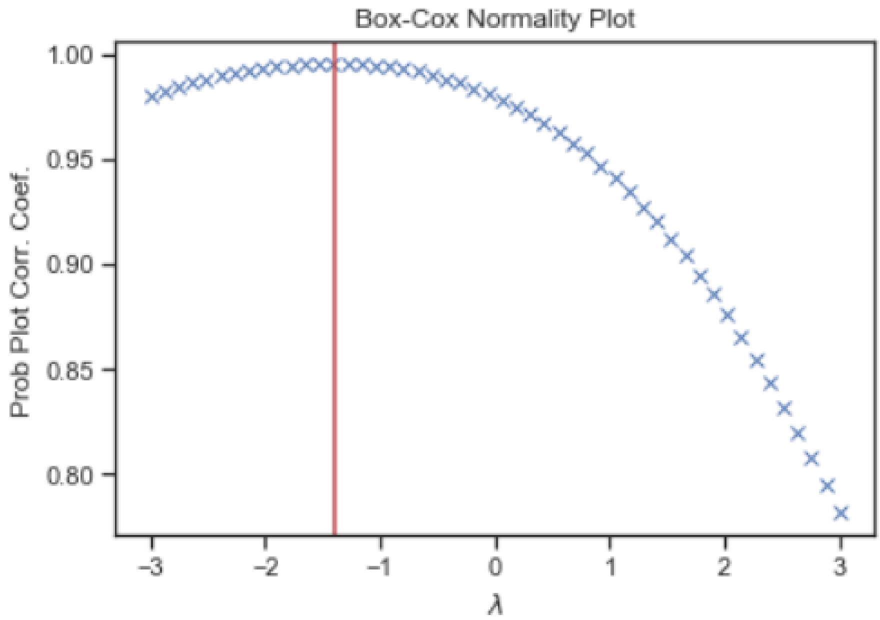
```
#Normality Checks and Box-Cox Transformation
sns.kdeplot(df['sbp'], shade = True,)
sm.qqplot(df['sbp'], line = 's', fit = True)
```





The KDE plot of the target variable. A skew to the right can be observed.

To obtain the value of lambda necessary to transform the target variable, we plot the Box-Cox Plot for Normality Transformation. The plot shows the log likelihood value as a function of different lambda values.

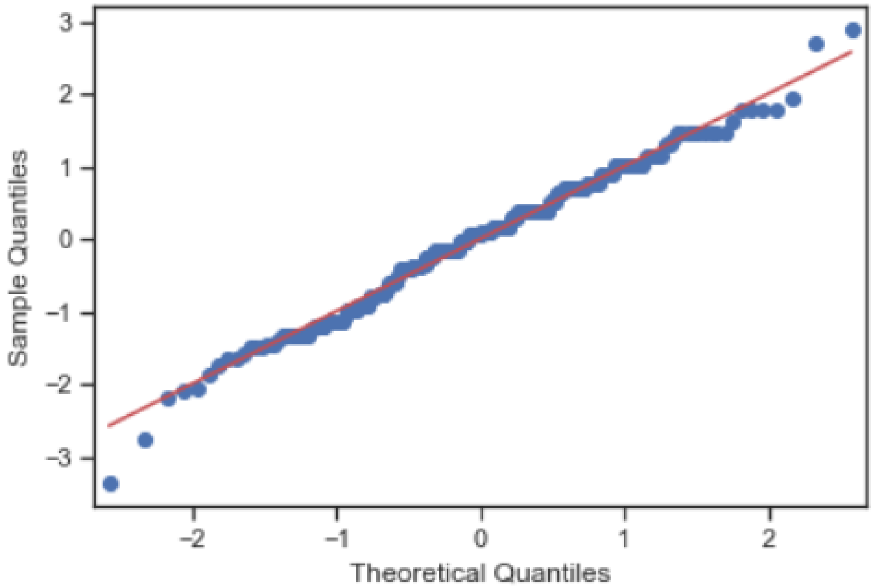



```
from scipy.stats import boxcox
a,b,(c1,c2) = boxcox(df['sbp'], alpha = 0.05)
print('Lambda value that maximizes log-likelihood function: ', b)
print('95% confidence interval for lambda: ',(c1,c2))
```

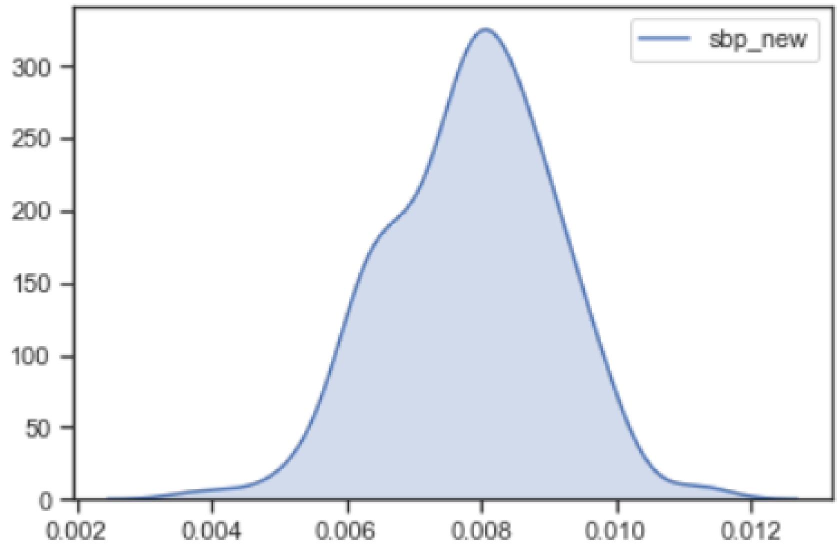
```
Lambda value that maximizes log-likelihood function:  -1.3989153525347984
95% confidence interval for lambda:  (-2.1097285739272937, -0.715577480148888)
```

As it can be seen, the global maximum of the likelihood function is attained very close to λ value of **-1**. Hence, we use the inverse transformation on our Y variable to obtain a normal distribution.

After the transformation, we validate our results by plotting the density and Normal QQ plots of the response variable.



The Normal QQ-Plot follows a near straight line.



The KDE Plot exhibits far lesser skew.

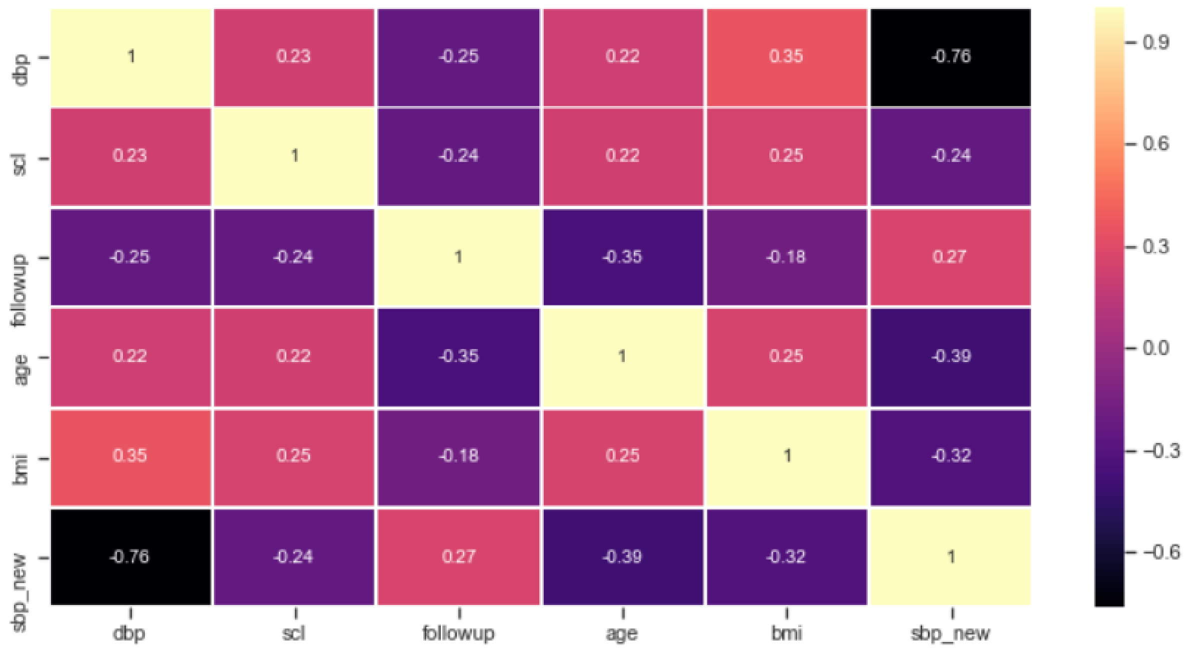
Based on the results of the transformation above, our target variable is now near normally distributed. With this, we can proceed to the next steps of our exploratory analysis.

Exploring the relationships between the predictors and the target variable:

We begin with drawing a heatmap of correlations to obtain a general understanding of the underlying relationships in our data.

```
plt.figure(figsize=(12, 6))
sns.heatmap(df.corr(),cmap='magma', annot = True, linecolor = 'white', linewidth = 1)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x15ab38035c0>
```

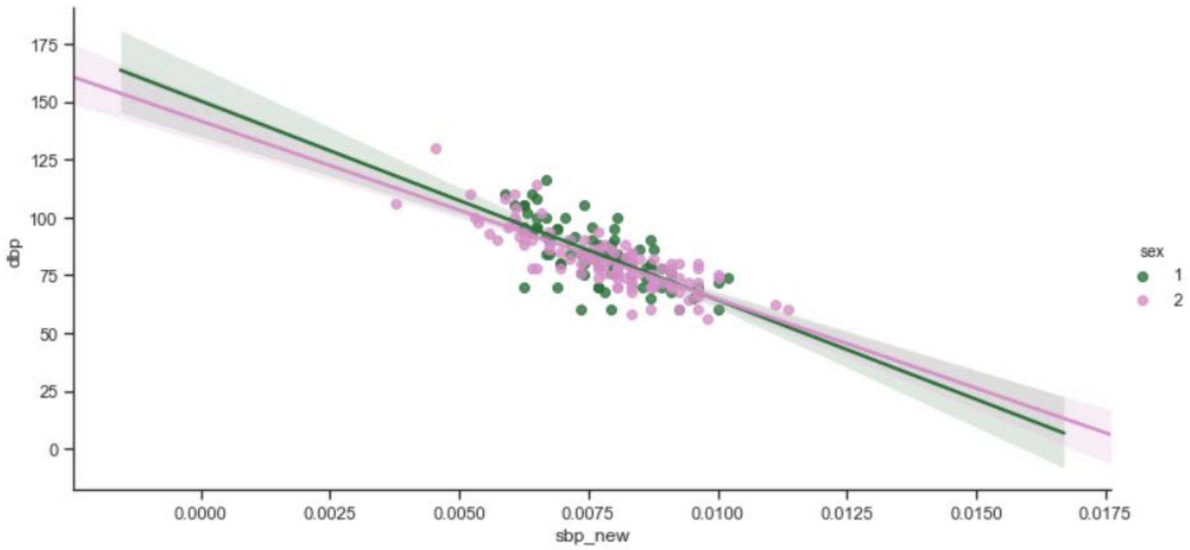


Correlation Heatmap

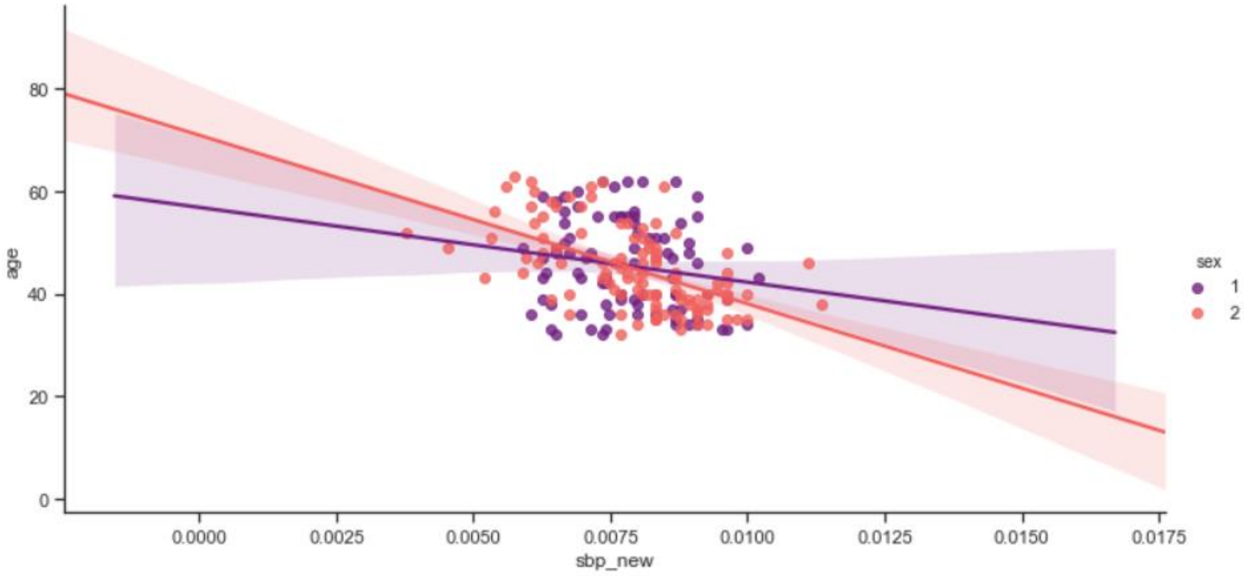
Clearly , the variable *dbp* is the most correlated with the target variable (it is most negatively correlated), followed by *age*. We proceed by examining the pairwise regression plots.

```
sns.lmplot(x = 'sbp_new', y = 'dbp', data = df,hue = 'sex',height = 5,aspect = 2, palette = 'cubehelix')
sns.lmplot(x = 'sbp_new', y = 'age', data = df,hue = 'sex',height = 5,aspect = 2, palette = 'magma')
```

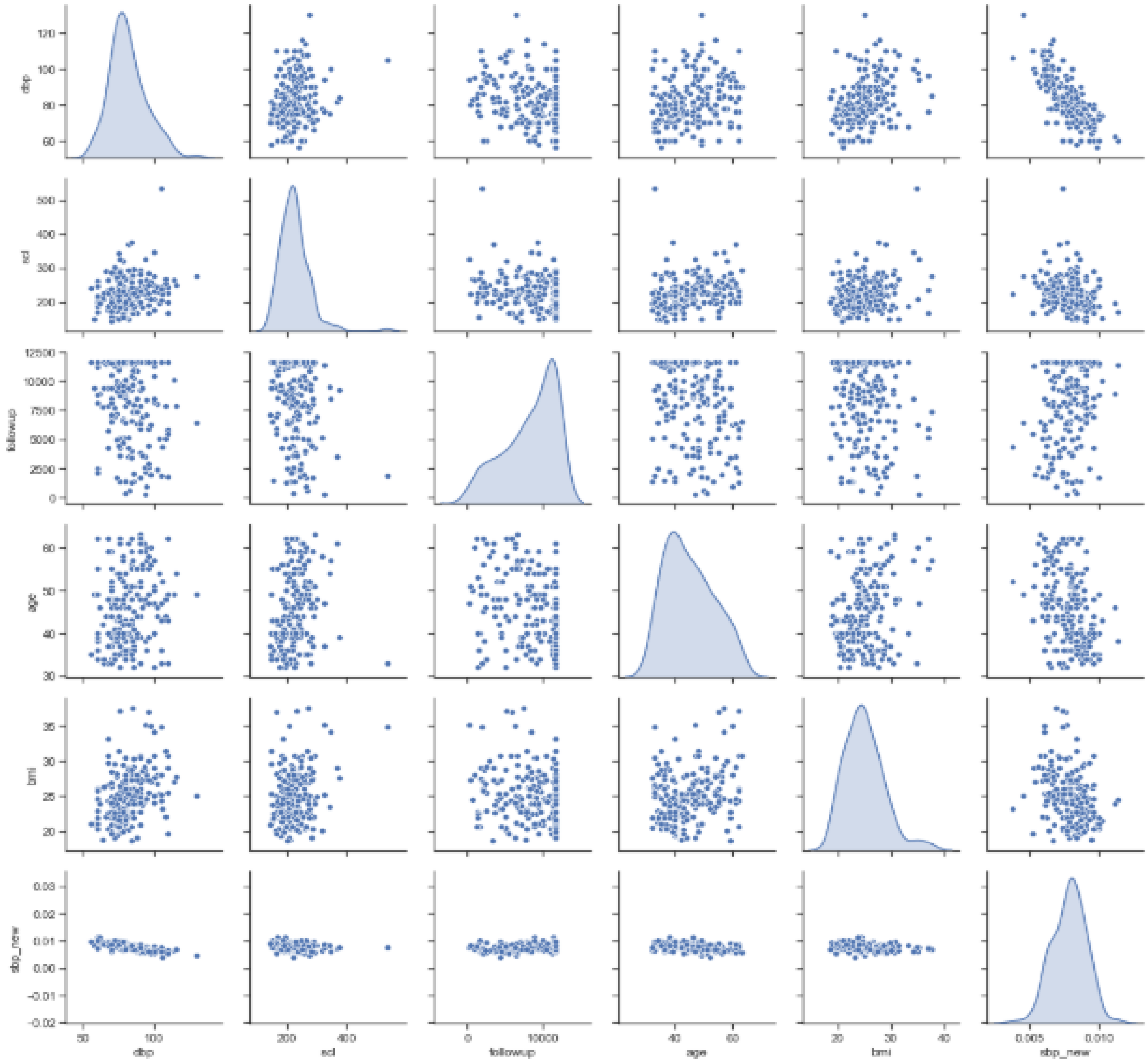
```
<seaborn.axisgrid.FacetGrid at 0x20f0d2c7550>
```



DBP and the target are negatively correlated to a significant degree. Both genders follow a similar pattern.



A weaker correlation exists between Age and the target.



Building an initial multiple linear regression model using the main effects:
We use the Ordinary Least Squares method available in the *statsmodels* library in Python using the main effects alone and study the regression output obtained.

```
#Linear Regression with statsmodels
SModel = sm.OLS(endog = y, exog = X).fit()
print(SModel.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.649
Model:                            OLS   Adj. R-squared:                  0.634
Method:                 Least Squares   F-statistic:                     43.91
Date:                Fri, 01 Nov 2019   Prob (F-statistic):           2.70e-39
Time:                        09:40:26   Log-Likelihood:                 1159.0
No. Observations:                 199   AIC:                            -2300.
Df Residuals:                     190   BIC:                            -2270.
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0147      0.001     24.490      0.000       0.013       0.016
x1         -6.611e-05   4.55e-06    -14.516      0.000   -7.51e-05   -5.71e-05
x2         -4.396e-07   1.19e-06     -0.370      0.712   -2.79e-06    1.91e-06
x3         -8.328e-09   1.77e-08     -0.471      0.638   -4.32e-08    2.65e-08
x4         -3.496e-05   6.93e-06     -5.044      0.000   -4.86e-05   -2.13e-05
x5          1.366e-06   1.57e-05      0.087      0.931   -2.96e-05    3.23e-05
x6          2.237e-05   1.45e-05      1.542      0.125   -6.25e-06     5.1e-05
x7           9.42e-05      0.000      0.893      0.373      -0.000       0.000
x8             0.0002      0.000      1.670      0.097   -3.79e-05       0.000
==============================================================================
Omnibus:                        1.387   Durbin-Watson:                   2.056
Prob(Omnibus):                  0.500   Jarque-Bera (JB):                1.077
Skew:                           0.009   Prob(JB):                        0.584
Kurtosis:                       3.360   Cond. No.                     1.05e+05
==============================================================================
```

Similar values of R-square and Adjusted R-square tells us that this model does not have significant variable interactions. The VIF analysis will be done later. The table also tells us the most significant variables in estimating the target are *dbp*, *age*, *month* and *chdfate* (all p values are < 0.15).

Next, we build 2 variable interactions based around *dbp* and construct a Multiple Regression Model on it. The two factor interactions considered are :

```
df['dscl'] = df['dbp']*df['scl']
df['dfol'] = df['dbp']*df['followup']
df['dage'] = df['dbp']*df['age']
df['dbmi'] = df['dbp']*df['bmi']
df['dmon'] = df['dbp']*df['month']
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.667
Model:                            OLS   Adj. R-squared:                  0.644
Method:                 Least Squares   F-statistic:                     28.50
Date:                Fri, 01 Nov 2019   Prob (F-statistic):           2.04e-37
Time:                        12:52:06   Log-Likelihood:                 1164.2
No. Observations:                 199   AIC:                            -2300.
Df Residuals:                     185   BIC:                            -2254.
Df Model:                          13
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0254      0.004      6.752      0.000       0.018       0.033
x1            -0.0002   4.51e-05     -4.354      0.000      -0.000      -0.000
x2            -8.2e-06   7.52e-06    -1.090      0.277    -2.3e-05    6.64e-06
x3         -7.695e-08   1.23e-07     -0.627      0.532   -3.19e-07    1.65e-07
x4            -0.0001   4.76e-05     -2.572      0.011      -0.000   -2.85e-05
x5            -0.0002      0.000     -1.416      0.158      -0.000    6.15e-05
x6         -5.993e-05      0.000     -0.588      0.557      -0.000       0.000
x7             0.0001      0.000      1.171      0.243   -8.57e-05       0.000
x8             0.0002      0.000      1.738      0.084   -2.94e-05       0.000
x9          9.067e-08   8.82e-08      1.029      0.305   -8.32e-08    2.65e-07
x10         8.655e-10   1.46e-09      0.594      0.553   -2.01e-09    3.74e-09
x11         1.091e-06    5.7e-07      1.914      0.057   -3.33e-08    2.21e-06
x12         1.834e-06   1.33e-06      1.382      0.169   -7.85e-07    4.45e-06
x13         1.015e-06   1.25e-06      0.810      0.419   -1.46e-06    3.49e-06
==============================================================================
Omnibus:                        1.144   Durbin-Watson:                   2.016
Prob(Omnibus):                  0.564   Jarque-Bera (JB):                0.806
Skew:                          -0.050   Prob(JB):                        0.668
Kurtosis:                       3.295   Cond. No.                     5.37e+07
==============================================================================
```

Regression summary with 2 factor interactions considered.

This model gives us marginally better results with an R-squared value of **0.667**. The only significant two factor interaction is *dage* but adding the 2nd order terms has increased the p value of *month* (>0.15).

VIF Analysis:

For the model created considering the main effects alone, we get the following results on running the *vif()* method available in R.

```
> vif(MLR1)
     dbp      scl followup      age      bmi    month      sex  chdfate
1.238425 1.194747 1.333201 1.227791 1.231565 1.044694 1.024188 1.279322
```

No variable has a significant VIF value (>5).

For the second order model, we get the following VIF results:

```
> #VIF Analysis
> vif(MLR2)
       dbp        scl    followup        age        bmi      month        sex    chdfate       dscl       dfol       dage
124.590020  49.012490  66.020630  59.488044  62.617097  52.932214   1.078251   1.309445  91.985821  63.219185 122.149919
      dbmi        dmo
181.949408  53.776605
```

As expected, adding the 2 factor interactions has increased multicollinearity greatly. All variables have a significantly high VIF (> 5) with the exception of *sex* and *chdfate* only because they are categorical variables and higher order terms weren't created with them.

Variable Selection

1. Stepwise Selection:

Here we'll implement forward variable selection to create a model. Broadly, we'll start with the variable that is the most correlated with the response and add variables sequentially, checking the p-value at every step to check if the new variable's coefficient is statistically significant.

Step 1:

With our previous data analysis, we know that the most highly correlated variable is *dbp*. We create a Simple Linear Regression with this variable alone.

```
#One Variable
SModel = sm.OLS(endog = y, exog = X[:,:2]).fit()
print(SModel.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.584
Model:                            OLS   Adj. R-squared:                  0.581
Method:                 Least Squares   F-statistic:                     276.0
Date:                Sat, 02 Nov 2019   Prob (F-statistic):           2.51e-39
Time:                        11:58:36   Log-Likelihood:                 1142.0
No. Observations:                 199   AIC:                            -2280.
Df Residuals:                     197   BIC:                            -2273.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0138      0.000     38.104      0.000       0.013       0.015
x1         -7.273e-05   4.38e-06    -16.613      0.000    -8.14e-05   -6.41e-05
==============================================================================
Omnibus:                        4.249   Durbin-Watson:                   2.024
Prob(Omnibus):                  0.119   Jarque-Bera (JB):                3.984
Skew:                          -0.256   Prob(JB):                        0.136
Kurtosis:                       3.467   Cond. No.                         541.
==============================================================================
```

The p value of this variable is ~ **0**. We move on to step 2.

Step 2:

A model is then built by adding the second highest correlated variable , which is *age*. The output for this is given below:

```
#Two Variables
SModel2 = sm.OLS(endog = y, exog = X[:,[0,1,4]]).fit()
print(SModel2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.638
Model:                            OLS   Adj. R-squared:                  0.634
Method:                 Least Squares   F-statistic:                     172.7
Date:                Sat, 02 Nov 2019   Prob (F-statistic):           5.62e-44
Time:                        12:05:55   Log-Likelihood:                 1155.9
No. Observations:                 199   AIC:                            -2306.
Df Residuals:                     196   BIC:                            -2296.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0150      0.000     37.347      0.000       0.014       0.016
x1         -6.778e-05   4.19e-06    -16.171      0.000   -7.61e-05   -5.95e-05
x2         -3.481e-05   6.41e-06     -5.433      0.000   -4.74e-05   -2.22e-05
==============================================================================
Omnibus:                        0.512   Durbin-Watson:                   2.085
Prob(Omnibus):                  0.774   Jarque-Bera (JB):                0.239
Skew:                          -0.005   Prob(JB):                        0.887
Kurtosis:                       3.169   Cond. No.                         730.
==============================================================================
```

Introducing the second variable has increased the R-sqr value by 5 points

On running an ANOVA test between the first and second models we get:

```
anova_tab = sm.stats.anova_lm(SModel,SModel2)
anova_tab
```

	df_resid	ssr	df_diff	ss_diff	F	Pr(>F)
0	197.0	0.000121	0.0	NaN	NaN	NaN
1	196.0	0.000105	1.0	0.000016	29.520648	1.629702e-07

This p value is less than the α to enter = **0.15**. We can proceed to adding another variable.

Step 3:

The next most correlated/significant variable is *chdfate.* The summary for the model created by adding this is given below.

```
SModel3 = sm.OLS(endog = y, exog = X[:,[0,1,4,8]]).fit()
print(SModel3.summary())
anova_tab = sm.stats.anova_lm(SModel2,SModel3)
anova_tab
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.643
Model:                            OLS   Adj. R-squared:                  0.637
Method:                 Least Squares   F-statistic:                     117.0
Date:                Sat, 02 Nov 2019   Prob (F-statistic):           2.32e-43
Time:                        14:59:27   Log-Likelihood:                 1157.3
No. Observations:                 199   AIC:                            -2307.
Df Residuals:                     195   BIC:                            -2293.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0147      0.000     33.106      0.000       0.014       0.016
x1         -6.637e-05   4.27e-06    -15.559      0.000    -7.48e-05    -5.8e-05
x2         -3.328e-05   6.45e-06     -5.159      0.000     -4.6e-05   -2.06e-05
x3             0.0002      0.000      1.610      0.109    -4.15e-05       0.000
==============================================================================
Omnibus:                        0.750   Durbin-Watson:                   2.064
Prob(Omnibus):                  0.687   Jarque-Bera (JB):                0.439
Skew:                          -0.046   Prob(JB):                        0.803
Kurtosis:                       3.211   Cond. No.                         815.
==============================================================================
```

There is no significant increase in R-sqr.

Running the ANOVA test between this model and its preceding model gives us:

	df_resid	ssr	df_diff	ss_diff	F	Pr(>F)
0	196.0	0.000105	0.0	NaN	NaN	NaN
1	195.0	0.000104	1.0	0.000001	2.592047	0.109019

The p value here is **0.109** , which is still lesser than the α to enter. We may proceed to adding more variables.

Step 4:

The next most correlated variable is *month.* The F-test for this 4-variable model is given below. As the p value here is still lesser than the α to enter, we proceed to adding more variables.

```
#4 Variable Model
anova_tab = sm.stats.anova_lm(SModel3,SModel4)
anova_tab
```

	df_resid	ssr	df_diff	ss_diff	F	Pr(>F)
0	195.0	0.000104	0.0	NaN	NaN	NaN
1	194.0	0.000102	1.0	0.000001	2.233422	0.136679

Step 5:

The next most correlated/significant variable is *sex*. The F-test for this 5-variable model is given below:

```
#5 Variable Model
anova_tab = sm.stats.anova_lm(SModel4,SModel5)
anova_tab
```

	df_resid	ssr	df_diff	ss_diff	F	Pr(>F)
0	194.0	0.000102	0.0	NaN	NaN	NaN
1	193.0	0.000102	1.0	4.445120e-07	0.841529	0.360105

However, here the p value is **0.3601** > **0.15** , the α to enter. We stop here and use the current 4 variable model we have.

Summary of the Forward Selection Algorithm:

Step	Variables included	p-Value
1	dbp	0
2	dbp + age	1.63E-07
3	dbp + age + chdfate	0.109019
4	dbp + age + chdfate + month	0.136679
5	dbp + age + chdfate + month + sex	0.3601 - Exit and use 4 variable model

Interpretation of the 4-variable model:

The final model developed as the regression output as below:

```
Final_Model = sm.OLS(endog = y, exog = X[:,[0,1,4,8,6]]).fit()
print(Final_Model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.647
Model:                            OLS   Adj. R-squared:                  0.640
Method:                 Least Squares   F-statistic:                     88.83
Date:                Sat, 02 Nov 2019   Prob (F-statistic):           9.06e-43
Time:                        15:08:51   Log-Likelihood:                 1158.4
No. Observations:                 199   AIC:                            -2307.
Df Residuals:                     194   BIC:                            -2290.
Df Model:                           4
Covariance Type:            nonrobust
```

The R-squared of this model is **0.647**, while the Adjusted R-squared is **0.640**. There is negligible noise in this model and no multicollinearity.
The next table gives us the estimates of the coefficients as well as their p-values:

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.0145	0.000	32.208	0.000	0.014	0.015
x1	-6.572e-05	4.27e-06	-15.375	0.000	-7.42e-05	-5.73e-05
x2	-3.429e-05	6.47e-06	-5.303	0.000	-4.7e-05	-2.15e-05
x3	0.0002	0.000	1.663	0.098	-3.54e-05	0.000
x4	2.126e-05	1.42e-05	1.494	0.137	-6.8e-06	4.93e-05

Omnibus:	1.141	Durbin-Watson:	2.058
Prob(Omnibus):	0.565	Jarque-Bera (JB):	0.809
Skew:	0.025	Prob(JB):	0.667
Kurtosis:	3.308	Cond. No.	834.

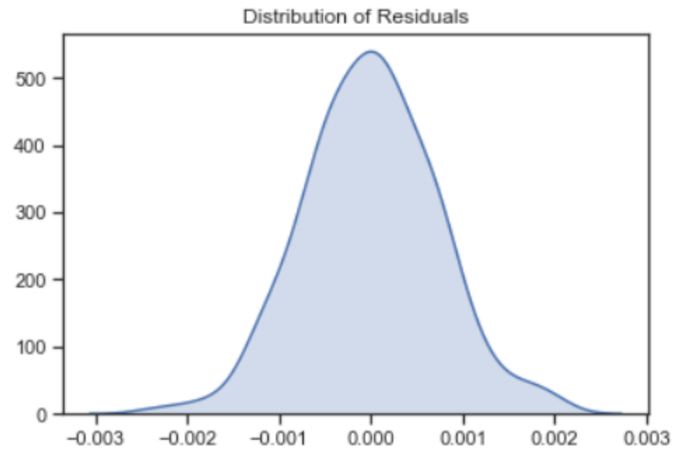
The p-values of all variable coefficients are significant (as expected) and values of the coefficients are given in the first column of the above table.

Model Diagnostics:

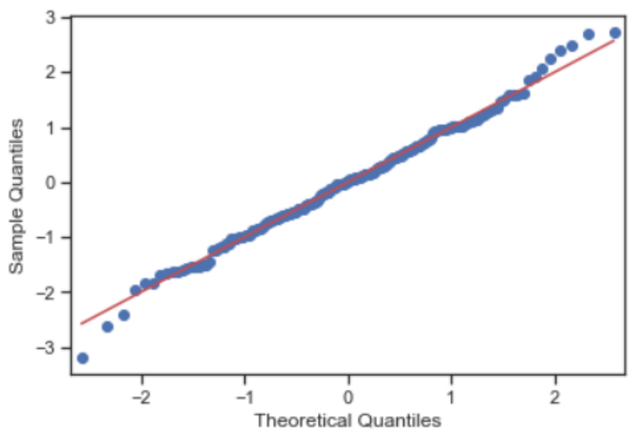
We perform model diagnosis as done in the previous sections.

1. ### Normality Assumption:

   We check the distribution of the residuals and the Normal QQ-Plot of the residuals to validate the normality assumption for our data. We require the distribution to be Gaussian and the Normal QQ-Plot to be a straight line to indicate a good fit for our model.



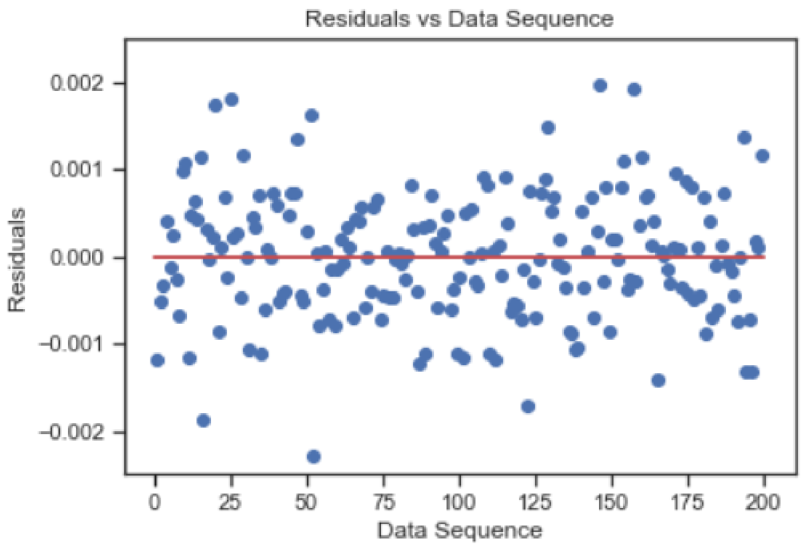
KDE plot of the residuals



Normal QQ-Plot

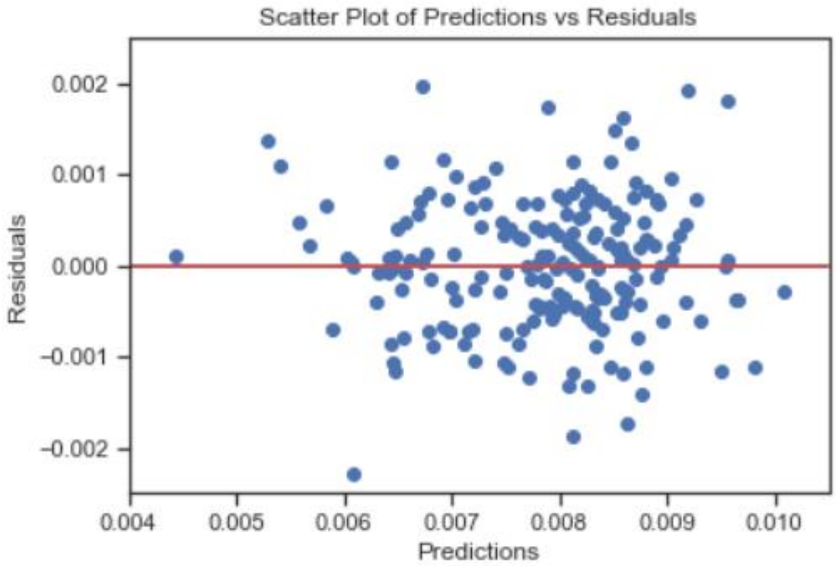
It can be observed that the distribution of the residuals is almost perfectly normal. The Normal QQ-Plot is also a linear trend, which tapers off the straight towards the end. The very slight skew indicates the presence of potential outliers.

2. Independence assumption :
   An important assumption in building a linear regression model is that the errors are independent of each other and the order in which the data was collected. We test this by plotting the residuals as a function of the data sequence and watching out for possible trends.
   With the plot below, we can verify the assumption that the residuals are independent from one another. Patterns in the plot may indicate that residuals near each other may be correlated, and thus, not independent. There are no trends or patterns when displayed in time order.



There are no trends or patterns when displayed in time order.

3. Homoscedasticity assumption:
   Another important assumption of the errors is that they all have equal variance. To test this, we construct the Residuals vs Predicted Values plot as done below:



   The plot shows no pattern which indicates that the constant variance assumption is valid.

2. All Subsets Selection:

We now search for a model created by choosing a subset of the available variables, which has the best performance when compared to all other possible combinations of subsets of the variables. We consider performance based on their scores on the following:

- Largest Adjusted R-squared value.
- Mallow's Cp-Statistic less than p that is closest to p.
- Lowest Bayesian Information Criterion

We do this by using the *regsubsets()* method available in R.

```
Subset selection object
Call: regsubsets.formula(sbpinv ~ dbp + scl + followup + age + bmi +
    month + sex + chdfate + dscl + dfol + dage + dbmi + dmo,
    data = p4dataext, nbest = 2, nvmax = 10)
13 Variables  (and intercept)
         Forced in Forced out
dbp          FALSE      FALSE
scl          FALSE      FALSE
followup     FALSE      FALSE
age          FALSE      FALSE
bmi          FALSE      FALSE
month        FALSE      FALSE
sex          FALSE      FALSE
chdfate      FALSE      FALSE
dscl         FALSE      FALSE
dfol         FALSE      FALSE
dage         FALSE      FALSE
dbmi         FALSE      FALSE
dmo          FALSE      FALSE
2 subsets of each size up to 10
Selection Algorithm: exhaustive
```

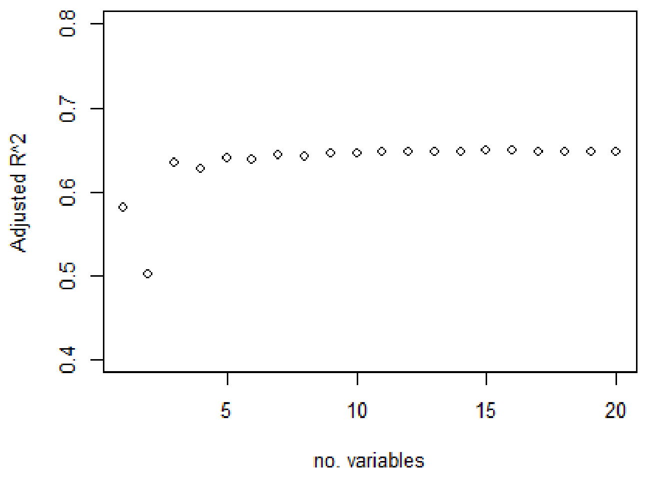
```
          dbp scl followup age bmi month sex chdfate dscl dfol dage dbmi dmo
1  ( 1 )  "*" " " " "      " " " " " "   " " " "     " "  " "  " "  " "  " "
1  ( 2 )  " " " " " "      " " " " " "   " " " "     " "  " "  "*"  " "  " "
2  ( 1 )  "*" " " " "      "*" " " " "   " " " "     " "  " "  " "  " "  " "
2  ( 2 )  "*" " " " "      " " " " " "   " " " "     " "  " "  "*"  " "  " "
3  ( 1 )  "*" " " " "      "*" " " " "   " " " "     " "  " "  "*"  " "  " "
3  ( 2 )  "*" " " " "      "*" " " " "   " " "*"     " "  " "  " "  " "  " "
4  ( 1 )  "*" " " " "      "*" " " " "   " " "*"     " "  " "  "*"  " "  " "
4  ( 2 )  "*" " " " "      "*" " " " "   " " " "     " "  " "  "*"  " "  "*"
5  ( 1 )  "*" " " " "      "*" " " " "   " " "*"     " "  " "  "*"  " "  "*"
5  ( 2 )  "*" " " " "      "*" " " "*"   " " "*"     " "  " "  "*"  " "  " "
6  ( 1 )  "*" " " " "      "*" " " " "   "*" "*"     " "  " "  "*"  " "  "*"
6  ( 2 )  "*" " " " "      "*" " " "*"   "*" "*"     " "  " "  "*"  " "  " "
7  ( 1 )  "*" " " " "      "*" "*" " "   " " "*"     " "  " "  "*"  "*"  "*"
7  ( 2 )  "*" " " " "      "*" "*" "*"   " " "*"     " "  " "  "*"  "*"  " "
8  ( 1 )  "*" " " " "      "*" "*" " "   "*" "*"     " "  " "  "*"  "*"  "*"
8  ( 2 )  "*" " " " "      "*" "*" "*"   "*" "*"     " "  " "  "*"  "*"  " "
9  ( 1 )  "*" " " " "      "*" "*" "*"   "*" "*"     " "  " "  "*"  "*"  "*"
9  ( 2 )  "*" " " "*"      "*" "*" " "   "*" "*"     " "  " "  "*"  "*"  "*"
10  ( 1 ) "*" "*" " "      "*" "*" " "   "*" "*"     "*"  " "  "*"  "*"  "*"
10  ( 2 ) "*" "*" " "      "*" "*" "*"   "*" "*"     "*"  " "  "*"  "*"  " "
```

We decide which model to pick from the 20 built above by checking the performance metrics we mentioned earlier.

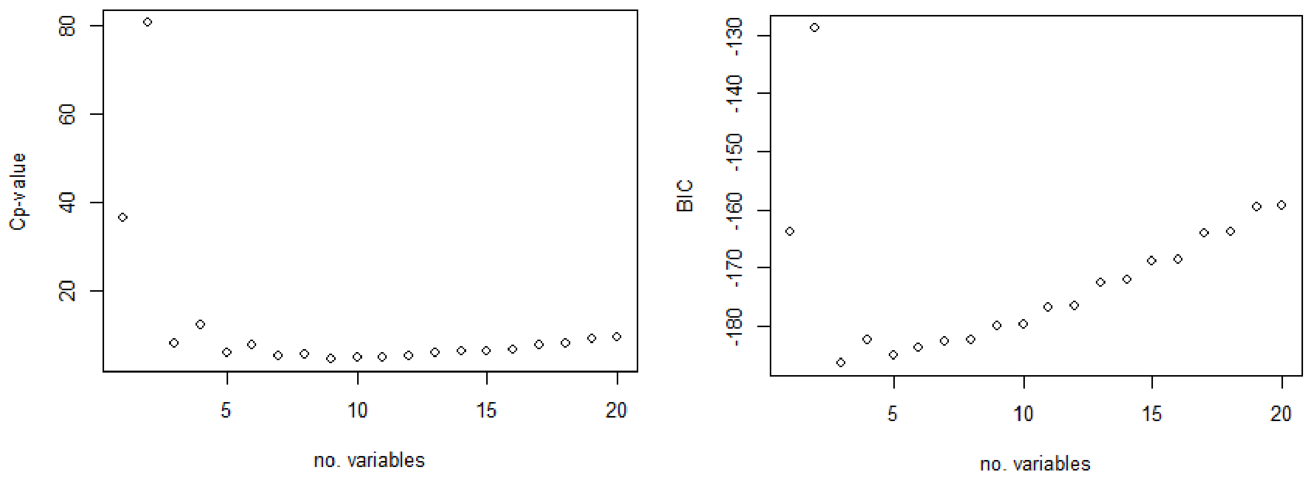
	Nvar	AdjR2	Cp	BIC
1	1	0.5813975	36.372433	-163.7169
2	1	0.5012476	80.673360	-128.8544
3	2	0.6343364	8.085786	-186.3427
4	2	0.6271859	12.017973	-182.4888
5	3	0.6399515	5.987717	-185.1468
6	3	0.6372826	7.447907	-183.6772
7	4	0.6432638	5.174609	-182.7158
8	4	0.6426208	5.524641	-182.3574
9	5	0.6462563	4.553260	-180.1273
10	5	0.6457950	4.803065	-179.8680
11	6	0.6478140	4.721661	-176.7460
12	6	0.6473451	4.974218	-176.4813
13	7	0.6479270	5.672971	-172.5557
14	7	0.6472698	6.025140	-172.1846
15	8	0.6490464	6.088402	-168.9408
16	8	0.6483866	6.440126	-168.5670
17	9	0.6481359	7.586571	-164.1820
18	9	0.6475725	7.885303	-163.8636
19	10	0.6475139	8.927404	-159.5929
20	10	0.6468740	9.264953	-159.2320

The highlighted subset above , model number **15**, has the best set of performance metrics of the subsets. This corresponds to a model with variables :
dbp, age , bmi, sex, chdfate, dage, dbmi, dmo

We can analyse the trend of the performance metrics as we add more variables :



After the addition of the 6th variable, the Adjusted R-sqr value remains constant.



After a point, an increase in the number of variables only increases BIC.

Interpretation of the resulting 8-variable model:

The final model developed by considering the variables mentioned above (*dbp, age , bmi, sex, chdfate, dage, dbmi, dmo)* has the regression output as below:

```
#Final All Subsets Regression Model
ASRModel = sm.OLS(endog = y, exog = X[:,[0,1,4,5,7,8,11,12,13]]).fit()
print(ASRModel.summary())
```

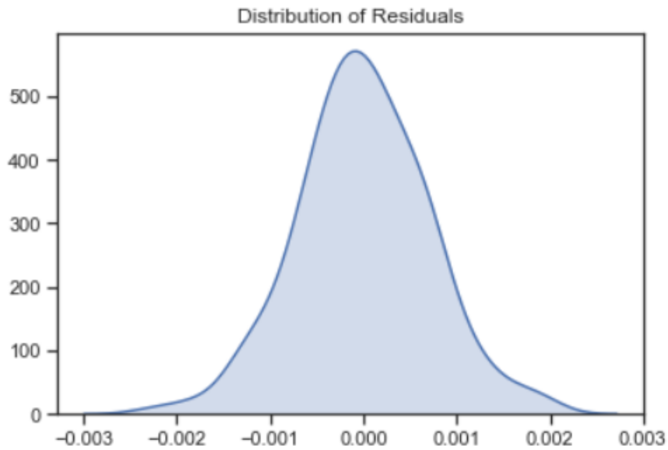
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.663
Model:                            OLS   Adj. R-squared:                  0.649
Method:                 Least Squares   F-statistic:                     46.77
Date:                Sat, 02 Nov 2019   Prob (F-statistic):           5.64e-41
Time:                        16:16:23   Log-Likelihood:                 1163.1
No. Observations:                 199   AIC:                            -2308.
Df Residuals:                     190   BIC:                            -2279.
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0226      0.003      7.784      0.000       0.017       0.028
x1            -0.0002   3.51e-05     -4.689      0.000      -0.000   -9.54e-05
x2            -0.0001   4.31e-05     -2.763      0.006      -0.000   -3.41e-05
x3            -0.0002      0.000     -1.619      0.107      -0.000    3.55e-05
x4             0.0001      0.000      1.269      0.206   -7.41e-05       0.000
x5             0.0002      0.000      2.009      0.046    4.23e-06       0.000
x6          1.035e-06   5.19e-07      1.995      0.048    1.14e-08    2.06e-06
x7          1.946e-06   1.19e-06      1.635      0.104   -4.02e-07    4.29e-06
x8          2.728e-07   1.73e-07      1.574      0.117   -6.91e-08    6.15e-07
==============================================================================
Omnibus:                        1.659   Durbin-Watson:                   2.007
Prob(Omnibus):                  0.436   Jarque-Bera (JB):                1.395
Skew:                          -0.010   Prob(JB):                        0.498
Kurtosis:                       3.410   Cond. No.                     2.52e+05
==============================================================================
```

The R-squared of this model is 0.663, while the Adjusted R-squared is 0.649. There is negligible noise and multicollinearity in this model.

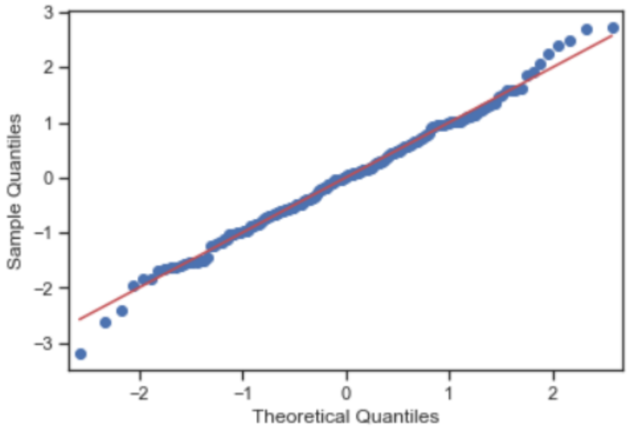
Model Diagnostics:

We perform model diagnosis as done in the previous sections.

1. Normality Assumption:
   We check the distribution of the residuals and the Normal QQ-Plot of the residuals to validate the normality assumption for our data. We require the distribution to be Gaussian and the Normal QQ-Plot to be a straight line to indicate a good fit for our model.



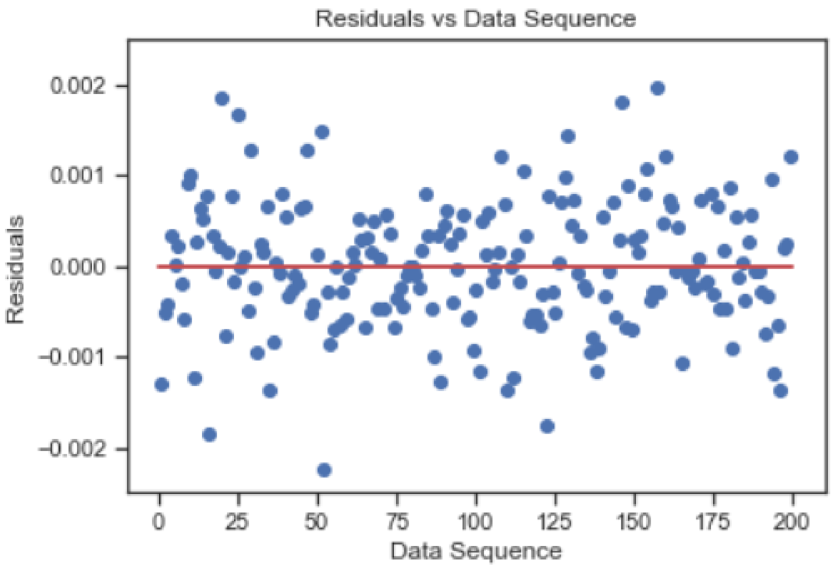
KDE plot of the residuals



Normal QQ-Plot

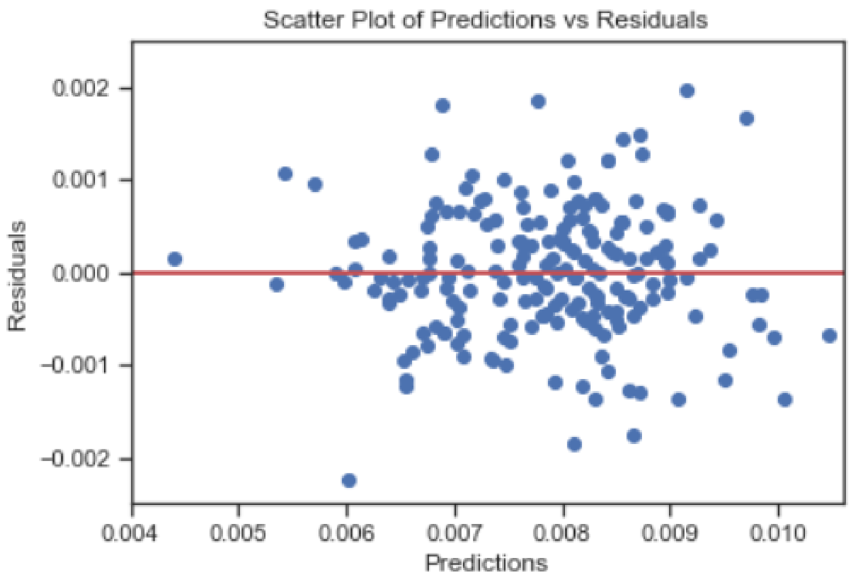
   It can be observed that the distribution of the residuals is almost perfectly normal. The Normal QQ-Plot is also a linear trend, which tapers off the straight towards the end. The very slight skew indicates the presence of potential outliers.

2. Independence assumption :
   An important assumption in building a linear regression model is that the errors are independent of each other and the order in which the data was collected. We test this by plotting the residuals as a function of the data sequence and watching out for possible trends.
   With the plot below, we can verify the assumption that the residuals are independent from one another. Patterns in the plot may indicate that residuals near each other may be correlated, and thus, not independent. There are no trends or patterns when displayed in time order.



There are no trends or patterns when displayed in time order.

3. Homoscedasticity assumption:
   Another important assumption of the errors is that they all have equal variance. To test this, we construct the Residuals vs Predicted Values plot as done below:



The plot shows no pattern which indicates that the constant variance assumption is valid.

Conclusion:

A comparison of the two models built using forward selection and all subsets selection tells us that the 8-variable all subsets selection regression model gives us a better model, although only marginally. Sometimes it might be better to weigh in the extra complexity when building a model with more variables.

	Adj-R^2	BIC
Forward Selection	0.64	-2290
All Subsets Selection	0.649	-2279