PROJECT REPORT

Health Care Cost

Linear Regression

Students:

Sakshi Singla
Ivette Sulca

Project report

1. Description of your dataset:

The linear regression analysis presented on this report will work with a dataset related to Medical health care cost provided in the book of *Machine Learning with R by Lantz.* The dataset can be found at https://www.kaggle.com/ruslankl/health-care-cost-prediction-w-linear-regression/comments.

The dataset presents 1338 observations of health insured person with predictor variables explaining the profile of each person and the total amount they paid for the health insurance coverage. In the next table all variables are described.

Variable	Type	Description	Type
Age	Numerical	The age of the insured person	Explanatory
Sex	Categorical	Sex of the insured person. Possible values: Female or Male	Explanatory
BMI	Numerical	Body mass index in kg/m2	Explanatory
Children	Numerical	Number of children of the insured person	Explanatory
Smoker	Categorical	Indicates if the person is a smoker or not. Possible values: Yes or No	Explanatory
Region	Categorical	Region place of the insured person. Possible values: Southwest, Northwest, Southeast or Northeast.	Explanatory
Charges	Numerical	Total insurance payment	Response

2. Statement of the research problems:

- **Research problem:**
  Health care cost in The United States is one of the most expensive around the world and is affecting negatively the quality of life of American population. In this sense, the project looks forward to understand which are the main variables that influence the most on the insurance annual charge and how (based on this variables) we can predict medical insurance cost for a specific

individual. The initial hypothesis is that age, sex, body mass index, number of children, smoker status and region have a relationship with how much a person pay for insurance.

- **Methods used:**

  We chose to use Linear Regression after analyzing the dataset using residuals and correlation matrixes. It won't be used any transformation do the predictor variables or response variables since it is no needed and to choose the best model Best Subsets technique will be applied based on Mallow Cp values and R^2 adjusted.

3. The explanatory analysis:

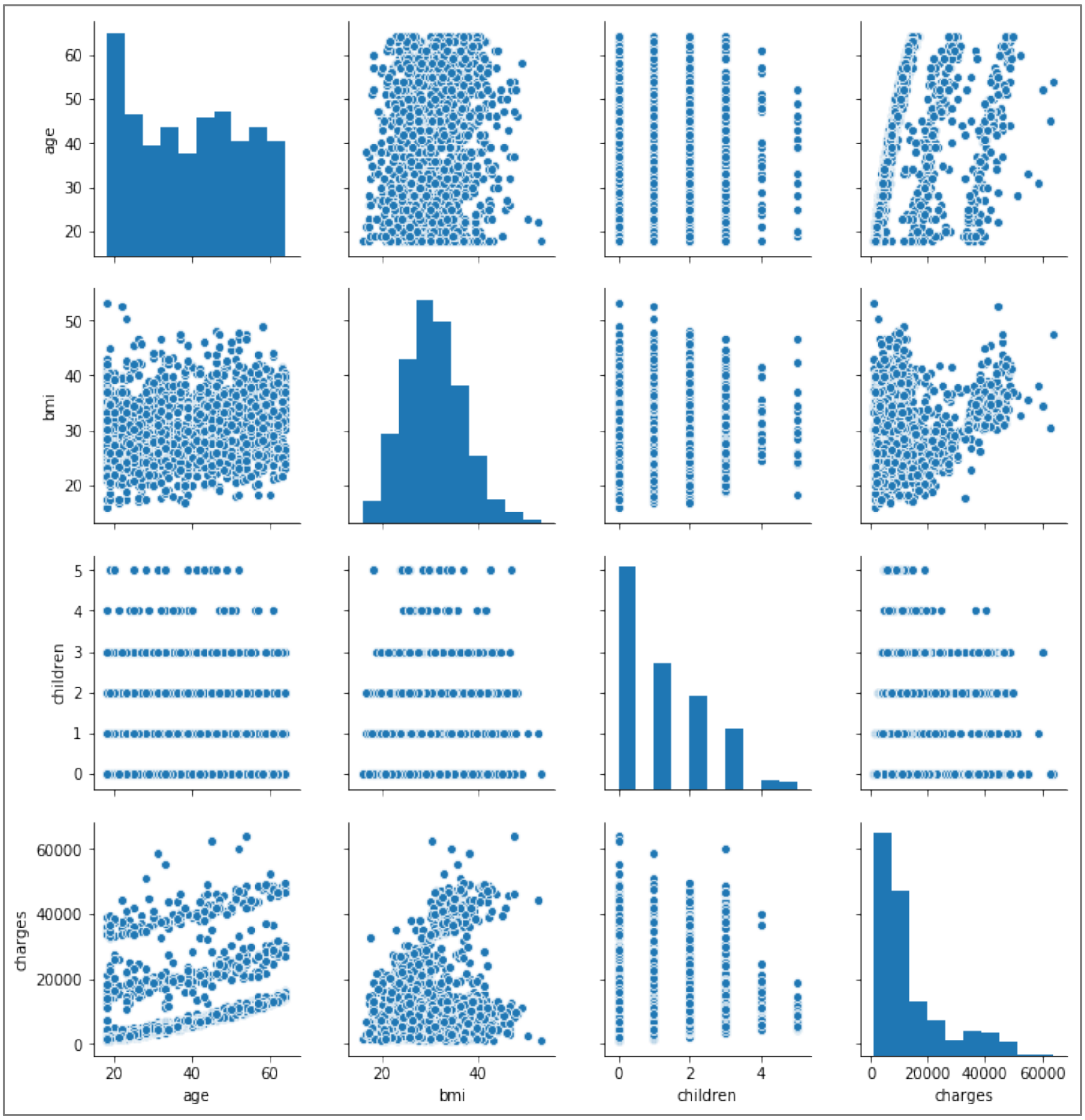
We analyze the numerical variables of the dataset with respect of the response variable "Insurance charge" in the following table. First of all, there is no missing data points and the range of values are reasonable for each type. We conclude also that half of the people paying for health insurance are 40 or more years old and the mean body mass index mean is 30 which are not good news since in general a BMI higher than 30 falls onto the obese category. In the other side, these people have in average one child and only 25% of them have 2 or more children.

Table : Statistics summary of the numerical explanatory variables

Statistc	Age	BMI	Number of children	Insurance Charges (USD)
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

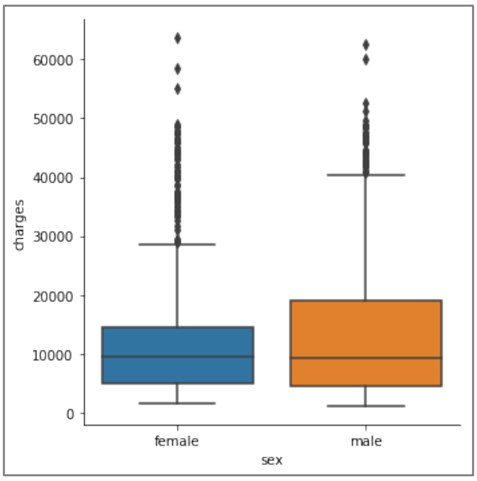
The next scatter plot matrix shows the relationship between age, bmi and children against insurance charges. Age has clearly a strong positive relationship with insurance charges while bmi just a moderate relationship which we believe after the individual tests could be excluded for the model. On the other side, the number of children seems to have a slightly negative relationship with insurance charges, however, that is not the expected since it is known that both should have a positive relationship.

Table: Scatter plot matrixes



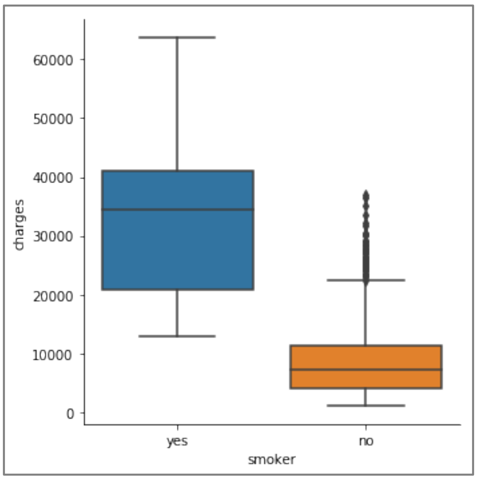
The other important categorical variable to analyze is sex. However, the next box plot shows that apparently there is no difference between sex and charges.

Table: Sex vs. Insurance charges box plot



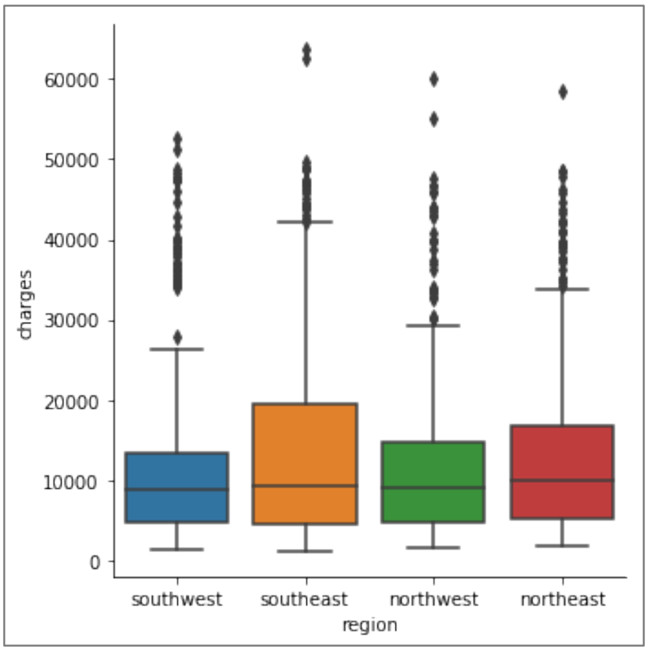
Nevertheless, smoker status seems to have a relationship with the response variable. The next plot indicates that the smokers pay more medical insurance than those who don't smoke.

Table: Smoker vs. Insurance charges box plot



In the case of the origin region of the insured, we can't find a different relationship with the predictor variable with the exception of Southeast which tends to pay a bit higher, but it is not extremely different than the other three regions. For this specific variable, we will be paying attention during the regression modelling.

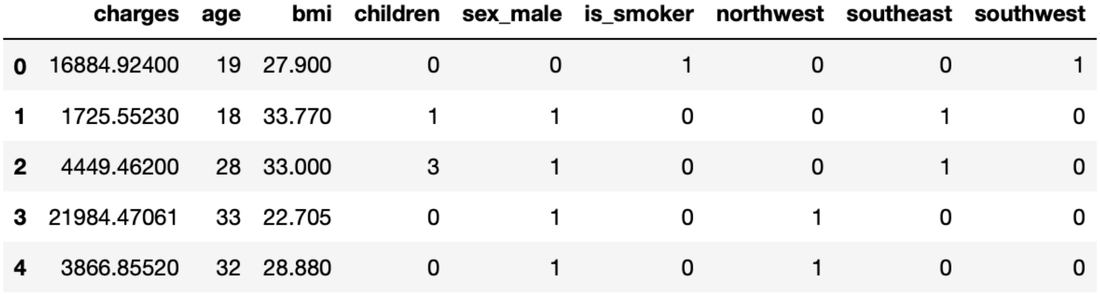
Table: Region vs. Insurance charges box plot



4. Regression analysis:

Before start fitting the model we create dummy variables for the categories sex, smoker status and region. Then, the table for fitting the variable remain as follows:

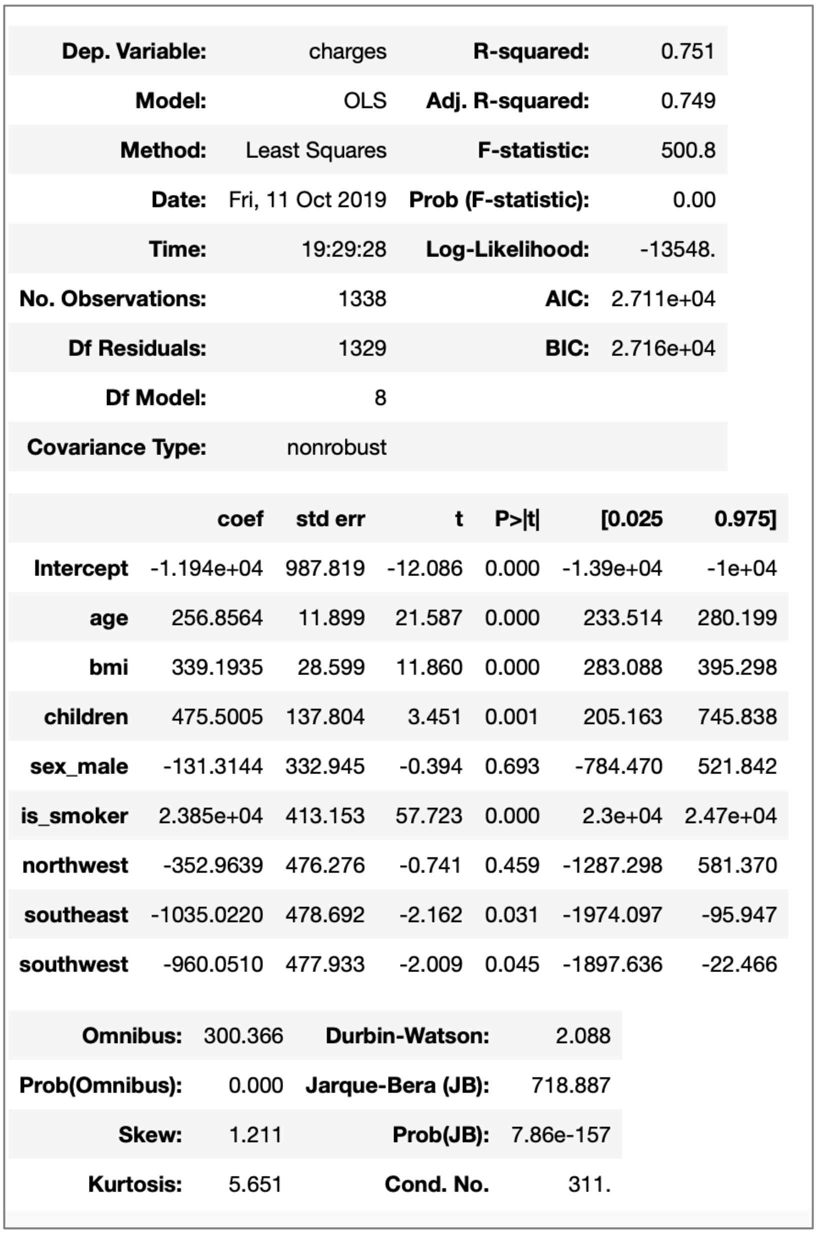
Table: Dataset with dummy variables



	charges	age	bmi	children	sex_male	is_smoker	northwest	southeast	southwest
0	16884.92400	19	27.900	0	0	1	0	0	1
1	1725.55230	18	33.770	1	1	0	0	1	0
2	4449.46200	28	33.000	3	1	0	0	1	0
3	21984.47061	33	22.705	0	1	0	1	0	0
4	3866.85520	32	28.880	0	1	0	1	0	0

We initiate the fitting model considering all the explanatory variables and doing individual tests. Considering a significance level alpha of 5%, the next table shows that both sex and region don’t have a relationship with the predictor variable insurance charges, which makes sense with the previous exploratory analysis.

Table: Fitted model considering all variables



Dep. Variable:	charges	**R-squared:**	0.751
Model:	OLS	**Adj. R-squared:**	0.749
Method:	Least Squares	**F-statistic:**	500.8
Date:	Fri, 11 Oct 2019	**Prob (F-statistic):**	0.00
Time:	19:29:28	**Log-Likelihood:**	-13548.
No. Observations:	1338	**AIC:**	2.711e+04
Df Residuals:	1329	**BIC:**	2.716e+04
Df Model:	8		
Covariance Type:	nonrobust		

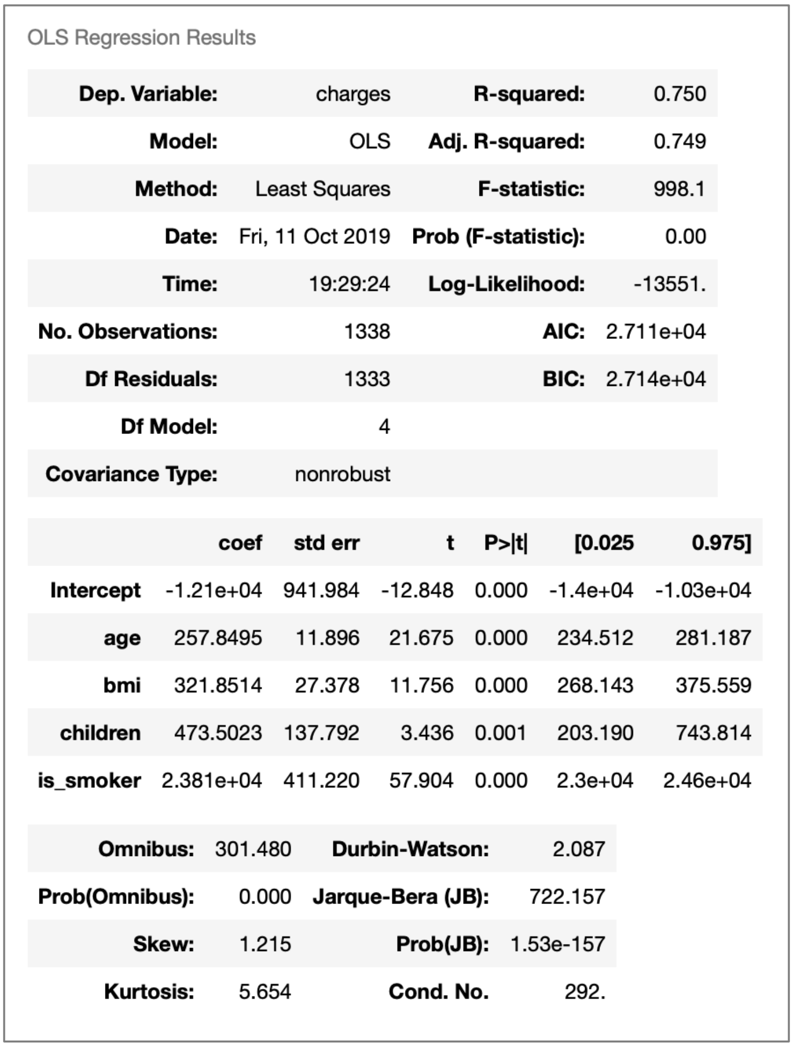
	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	-1.194e+04	987.819	-12.086	0.000	-1.39e+04	-1e+04
age	256.8564	11.899	21.587	0.000	233.514	280.199
bmi	339.1935	28.599	11.860	0.000	283.088	395.298
children	475.5005	137.804	3.451	0.001	205.163	745.838
sex_male	-131.3144	332.945	-0.394	0.693	-784.470	521.842
is_smoker	2.385e+04	413.153	57.723	0.000	2.3e+04	2.47e+04
northwest	-352.9639	476.276	-0.741	0.459	-1287.298	581.370
southeast	-1035.0220	478.692	-2.162	0.031	-1974.097	-95.947
southwest	-960.0510	477.933	-2.009	0.045	-1897.636	-22.466

Omnibus:	300.366	**Durbin-Watson:**	2.088
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	718.887
Skew:	1.211	**Prob(JB):**	7.86e-157
Kurtosis:	5.651	**Cond. No.**	311.

We exclude sex and region and fit the model again in the next table. And decide to continue with this model as the full model:

charges~age+bmi+children+is_smoker

Table: Fitted model considering only age, bmi, children and smoker status



```
OLS Regression Results

    Dep. Variable:          charges           R-squared:      0.750
           Model:               OLS       Adj. R-squared:      0.749
          Method:     Least Squares         F-statistic:      998.1
            Date:  Fri, 11 Oct 2019   Prob (F-statistic):       0.00
            Time:          19:29:24       Log-Likelihood:    -13551.
No. Observations:              1338                  AIC:  2.711e+04
    Df Residuals:              1333                  BIC:  2.714e+04
        Df Model:                 4
 Covariance Type:         nonrobust

                coef   std err        t  P>|t|    [0.025     0.975]
 Intercept  -1.21e+04  941.984  -12.848  0.000   -1.4e+04  -1.03e+04
       age  257.8495    11.896   21.675  0.000    234.512    281.187
       bmi  321.8514    27.378   11.756  0.000    268.143    375.559
  children  473.5023   137.792    3.436  0.001    203.190    743.814
 is_smoker  2.381e+04  411.220   57.904  0.000     2.3e+04   2.46e+04

       Omnibus:  301.480     Durbin-Watson:      2.087
Prob(Omnibus):    0.000  Jarque-Bera (JB):    722.157
          Skew:    1.215          Prob(JB):  1.53e-157
      Kurtosis:    5.654          Cond. No.       292.
```

Considering the second full model, the residuals remains as follows. Mostly of the data are bounding around zero and the qqplot shows nearly a line showing the normality assumption of the linear regression model.

Table: Residuals plot

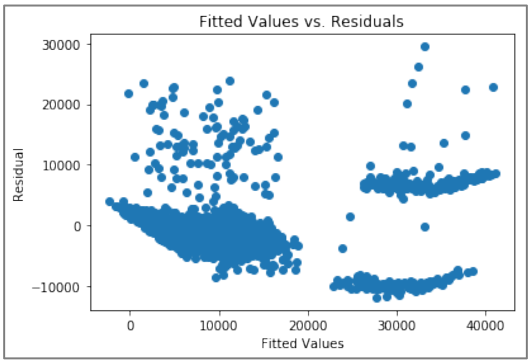
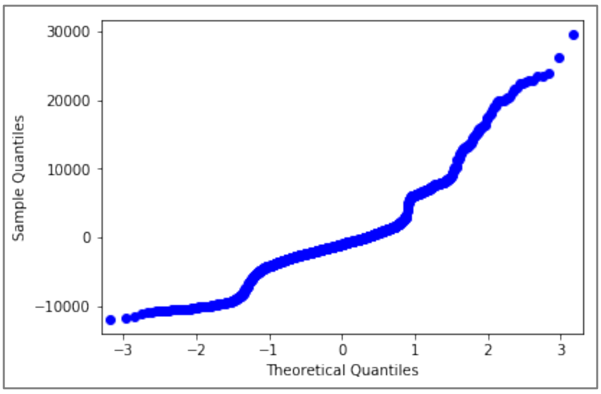


Table: QQ plot



5. Model selection

For model selection we will use Best Subsets method to find a group of model candidates with Mallow's CP, R^2 , AIC and BIC as the criteria set for the analysis. The next table presents 2^4 fitted models with all criteria computed.

Model Fitted	CP	R2_adj	R2	AIC	BIC
charges~intercept	3989.492894	0.00000	0.00000	27619.263199	27624.462130
charges~age	3515.362411	0.08872	0.08941	27495.948748	27506.346610
charges~bmi	3781.992584	0.03862	0.03934	27567.564127	27577.961989
charges~children	3966.869099	0.00388	0.00462	27615.062262	27625.460125
charges~is_smoker	690.939777	0.61948	0.61976	26327.463615	26337.861477
charges~age+bmi	3369.433604	0.11586	0.11718	27456.497842	27472.094636
charges~age+children	3501.047355	0.09111	0.09247	27493.439630	27509.036423

charges~age+is_smoker	151.677921	0.72098	0.72140	25913.324376	25928.921170
charges~bmi+children	3761.163760	0.04219	0.04363	27563.580279	27579.177073
charges~bmi+is_smoker	489.627477	0.65743	0.65794	26187.890316	26203.487110
charges~children+is_smoker	672.495400	0.62304	0.62360	26315.886123	26331.482917
charges~age+bmi+children	3355.910806	0.11812	0.12010	27454.072837	27474.868562
charges~age+bmi+is_smoker	14.808596	0.74691	0.74748	25783.835916	25804.631641
charges~age+children+is_smoker	141.203501	0.72312	0.72374	25904.027223	25924.822948
charges~bmi+children+is_smoker	472.788621	0.66072	0.66148	26175.980976	26196.776701
charges~age+bmi+children+is_smoker	5.000000	0.74894	0.74969	25774.035219	25800.029875

Finally, we observe that the next two models have lower Cp value, higher R^2 and also the best (lower) values of AIC and BIC. Those model candidates are:

Candidate 1: charges~age+bmi+children+is_smoke
Candidate 2: charges~age+bmi+is_smoker

• F-test between Full model and Reduced model:

Ho: charges~age+bmi+is_smoker (reduced model)
H1: charges~age+bmi+children+is_smoke (full model)

Full model:
SSE_full=4.89*((10)^10)
n=1338
p=9
df_full=n-p=1329

Reduced model:
SSE_red=4.90*((10)^10)
n=1338
p=5
df_red=n-p=1333

F statistic:
F_stat=((SSE_red-SSE_full)/(df_red-df_full))/(SSE_full/df_full)
F_stat=((4.90*((10)^10) - 4.89*((10)^10))/(1333-1329))/ (4.89*((10)^10) / 1329)
F_stat=0.679

P_value=0.6062

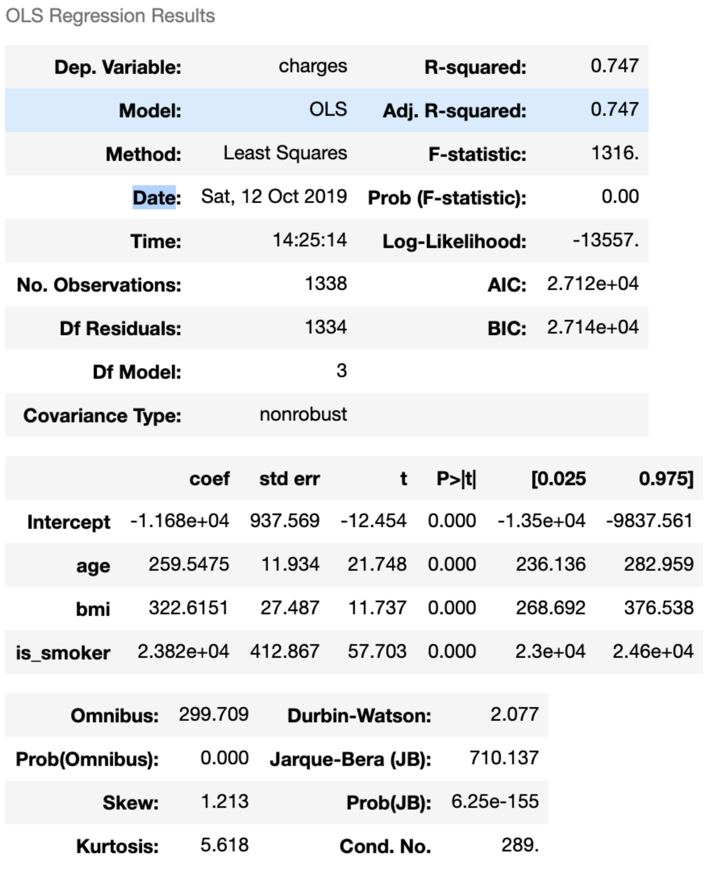
Since p-value>0.05, we don’t have enough evidence to reject the null hypothesis and conclude that the reduce model charges~age+bmi+is_smoker fits better than the full model.

6. Model diagnosis

Doing model diagnosis for the model:

charges~age+bmi+is_smoker

Table: Regression summary table



OLS Regression Results

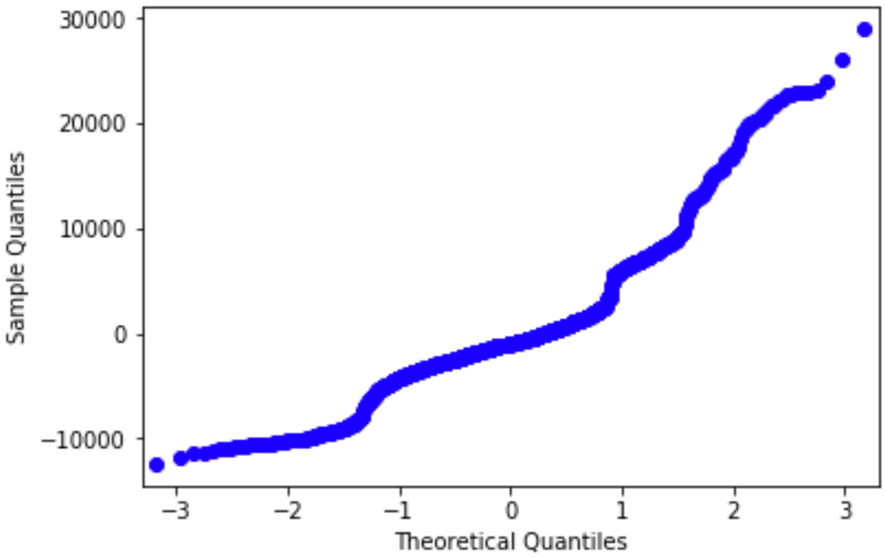
Dep. Variable:	charges	**R-squared:**	0.747
Model:	OLS	**Adj. R-squared:**	0.747
Method:	Least Squares	**F-statistic:**	1316.
Date:	Sat, 12 Oct 2019	**Prob (F-statistic):**	0.00
Time:	14:25:14	**Log-Likelihood:**	-13557.
No. Observations:	1338	**AIC:**	2.712e+04
Df Residuals:	1334	**BIC:**	2.714e+04
Df Model:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	-1.168e+04	937.569	-12.454	0.000	-1.35e+04	-9837.561
age	259.5475	11.934	21.748	0.000	236.136	282.959
bmi	322.6151	27.487	11.737	0.000	268.692	376.538
is_smoker	2.382e+04	412.867	57.703	0.000	2.3e+04	2.46e+04

Omnibus:	299.709	**Durbin-Watson:**	2.077
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	710.137
Skew:	1.213	**Prob(JB):**	6.25e-155
Kurtosis:	5.618	**Cond. No.**	289.

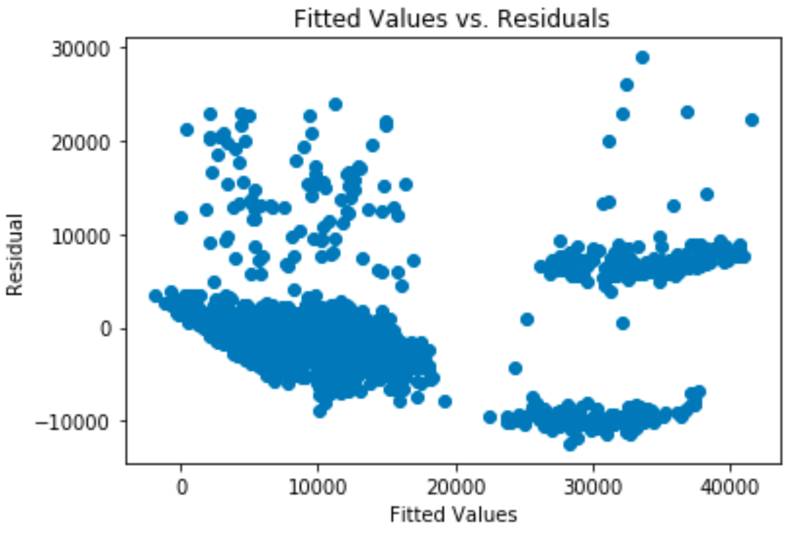
a.>Checking for the normality of error terms:

QQ-Plot:



The plot is not-linear. Therefore we can say that error terms are not following normal curve.

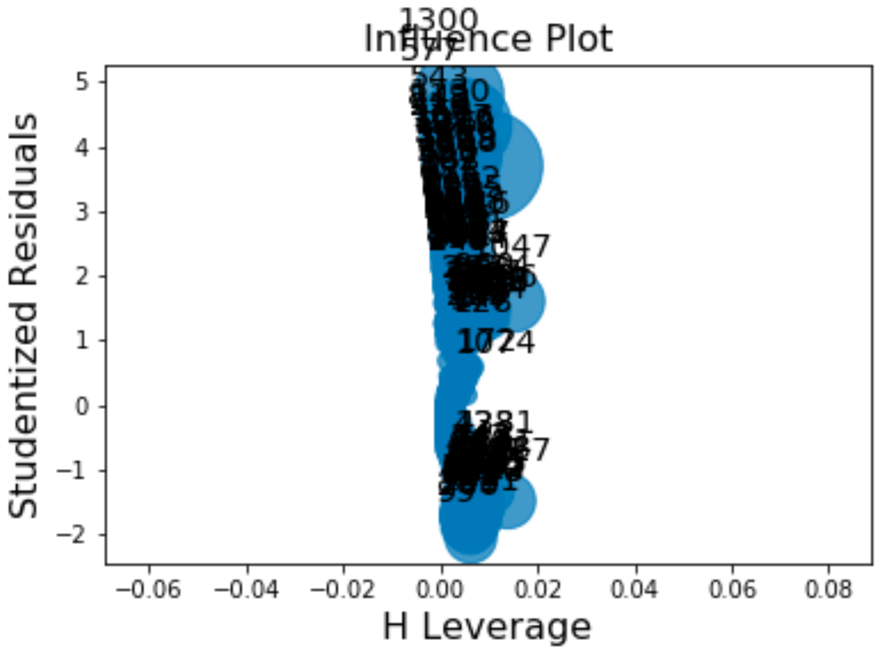
b.>Fitted values v/s residuals plot:



Not randomly distributed across zero. Therefore, Heteroskedasticity exists.

c.> Checking for the outliers and leverage points.

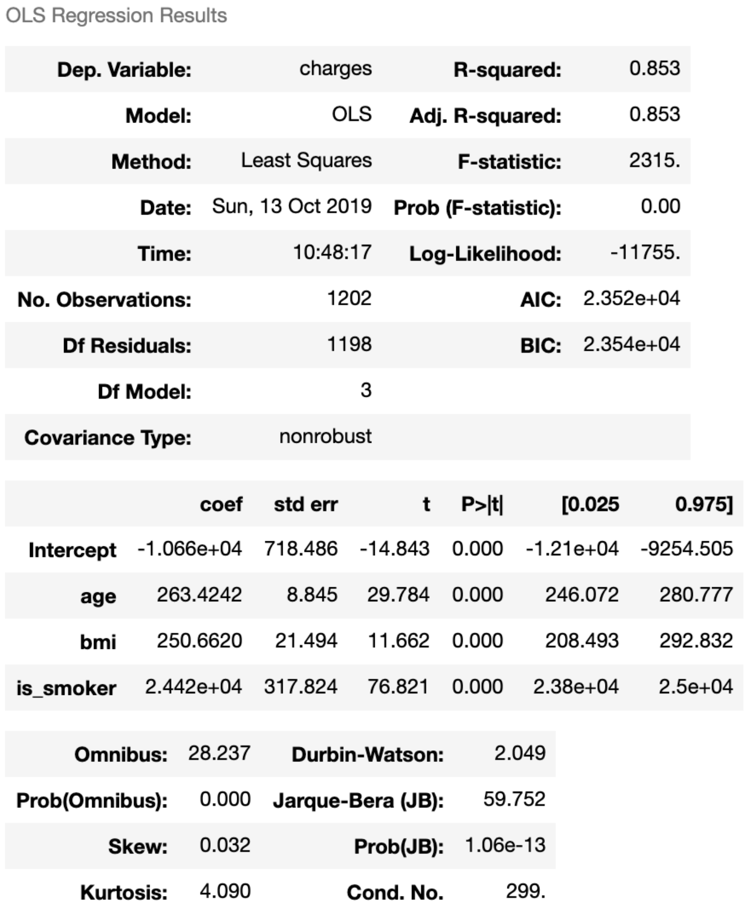
Influence Plot



Therefore, there are outliers and leverage points

Getting all the points that are considered influential using DFFITS and Cook’s distance criteria and removing them from the table.

Again performing OLS:



OLS Regression Results

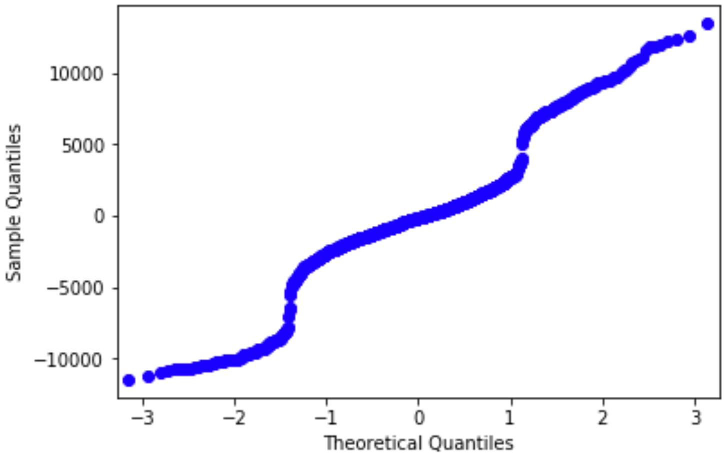
Dep. Variable:	charges	**R-squared:**	0.853
Model:	OLS	**Adj. R-squared:**	0.853
Method:	Least Squares	**F-statistic:**	2315.
Date:	Sun, 13 Oct 2019	**Prob (F-statistic):**	0.00
Time:	10:48:17	**Log-Likelihood:**	-11755.
No. Observations:	1202	**AIC:**	2.352e+04
Df Residuals:	1198	**BIC:**	2.354e+04
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1.066e+04	718.486	-14.843	0.000	-1.21e+04	-9254.505
age	263.4242	8.845	29.784	0.000	246.072	280.777
bmi	250.6620	21.494	11.662	0.000	208.493	292.832
is_smoker	2.442e+04	317.824	76.821	0.000	2.38e+04	2.5e+04

Omnibus:	28.237	**Durbin-Watson:**	2.049
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	59.752
Skew:	0.032	**Prob(JB):**	1.06e-13
Kurtosis:	4.090	**Cond. No.**	299.

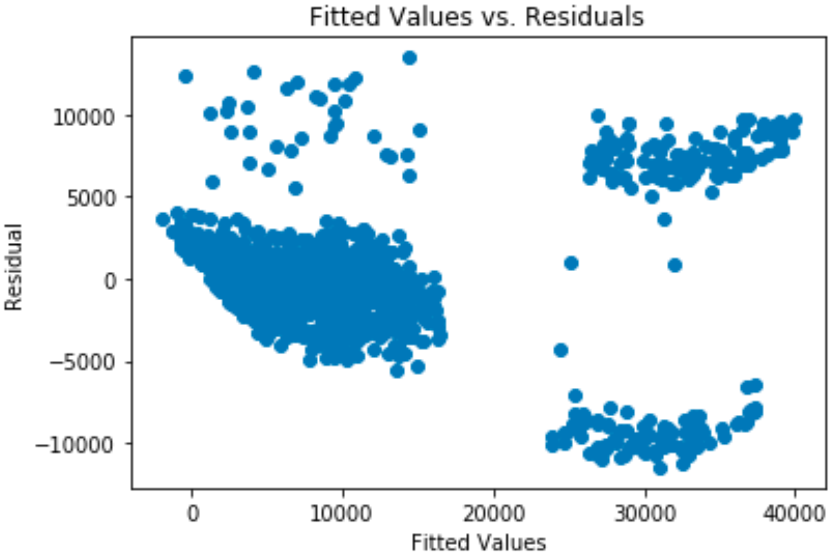
P-values are low and AIC, BIC slightly reduced and Adj. R-squared improved. Therefore, a better fit.

Again testing for normality:



We can see that **normality is considerably improved** for the model. Line is much closer to be linear than previously.

Testing for the variances:



Therefore, the range of variances is also improved now.

d.> Breusch-Pagan test for Heteroskedasticity

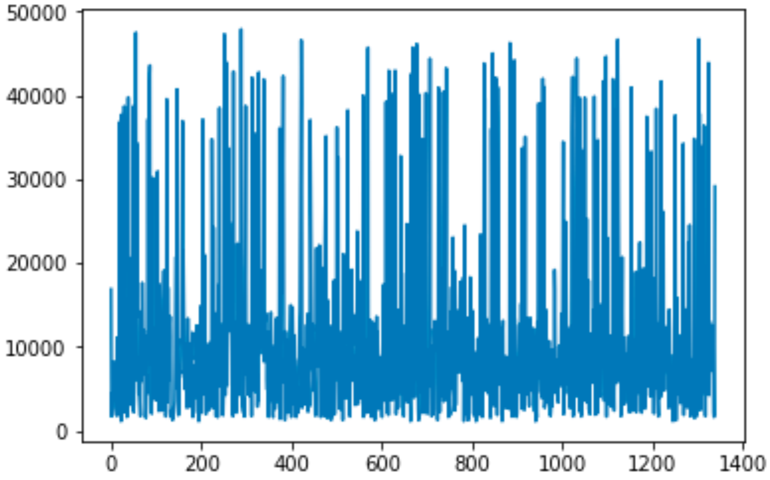
Results:

```
{'LM Statistic': 737.0934816226819,
 'LM-Test p-value': 1.8987749487608997e-159,
'F-Statistic': 633.1294257220253,
 'F-Test p-value': 1.677709707214861e-246}
```

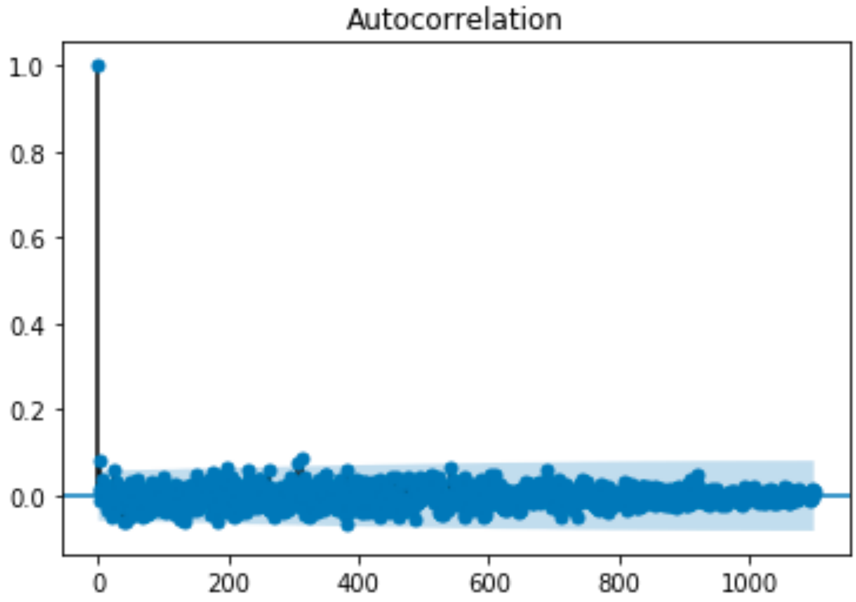
Since LM test p-value is very small. Therefore, Heteroskedasticity exists.
We will use **weighted least squares** to provide correction for Heteroskedasticity.

e.>Testing for autocorrelation:

Plotting charge variable: Quite random(Therefore good.)



Testing for autocorrelation graphically:



Plot is Showing that there is no autocorrelation.

Using **Breusch_Godfrey test** for testing autocorrelation:

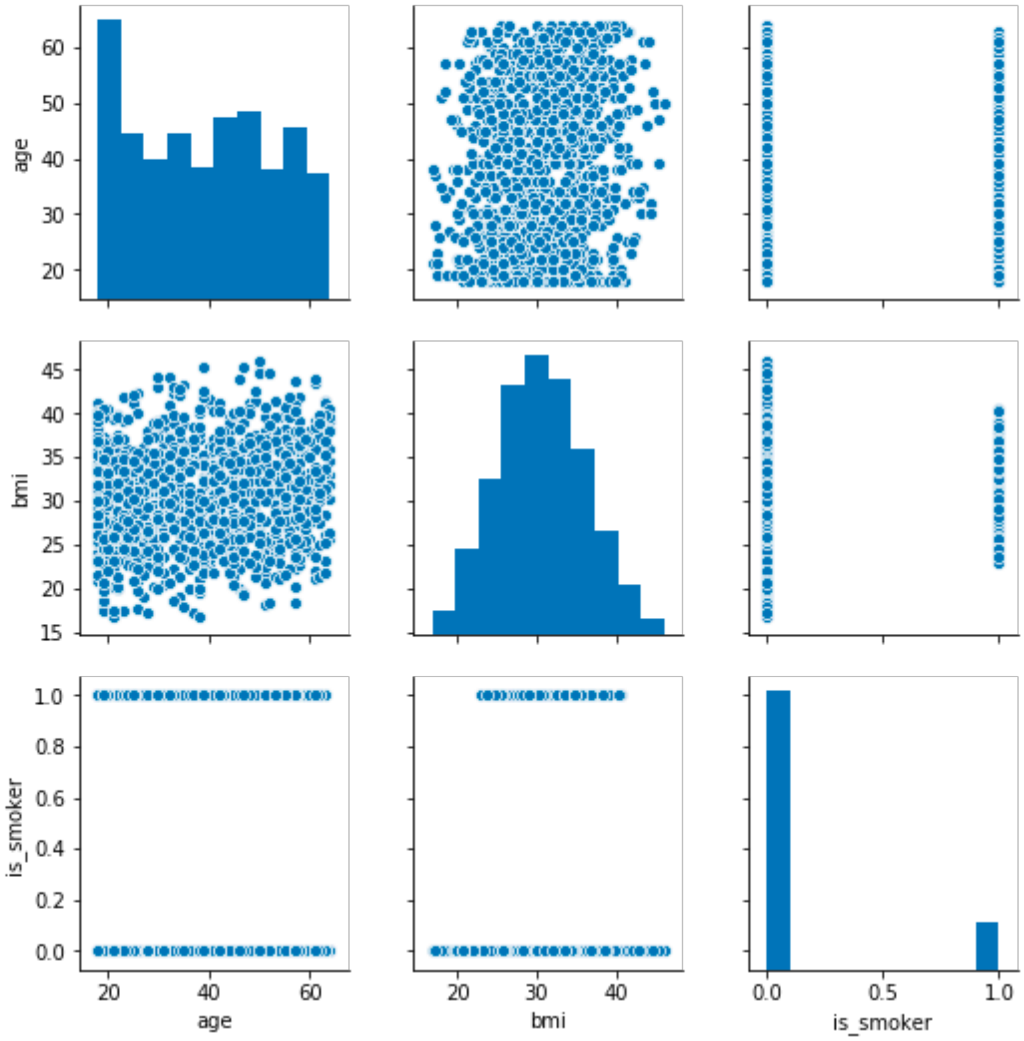
Test results:

```
{'LM Statistic': 16.67536754131242,
'LM-Test p-value': 0.7810148532361781,
'F-Statistic': 0.752008494381279,
'F-Test p-value': 0.7867214801580659}
```

As we can see, p-value for LM test is higher than 0.05. Therefore we cannot reject null hypothesis. Thus, we can say from test that there is no serial correlation.Also evident from plot.

f.> Testing for multicollinearity:

Side-by-side plots:



Not much evident multicollinearity problem evident from the plots.

Using VIF method to test multicollinearity :

```
  VIF Factor   features
0   7.919438       age
1   8.187069       bmi
2   1.177900  is_smoker
```

VIF for age and bmi is higher than 4 but less than 10.

Also, seeing correlation matrix between the variables:

```
             age   bmi  is_smoker
age         1.00  0.12      -0.05
bmi         0.12  1.00       0.05
is_smoker  -0.05  0.05       1.00
```

Correlation factor between age and bmi is very low.
Also, evident from the plot between age and bmi.

Therefore, concluding that there is no serious multicollinearity issue for the model.

g.> Performing weighted least squares to provide correction for Heteroskedasticity

MSE=56476078.25983766

βˆWLS =(X’WX)^-1* X’Wy

```
array([[-8150.40862706],
       [  260.8424167 ],
       [  173.22681875],
       [25693.62330826]])
```

Therefore final model is:

y= -8150.40+ 260.84*age + 173.23*bmi + 25693.62*is_smoker

h.> **For full model:**
Also, we performed diagnostics for model(See appendix):
charges~age+bmi+children+is_smoke

Final model:
y=-10984.74920721+260.74790992* *age* + *249.88974821**bmi + 449.63510152**children* + *24408.59159125**is_smoker

Diagnostic tests results were almost same for both models. Therefore, selecting reduced model than full model.Also, evident from F-test.

7. Final Model

Therefore final model is:

y= -8150.40+ 260.84age* + *173.23**bmi + 25693.62*is_smoker**

The final model indicates that age, body mass index and smoking status has a relationship with the final medical insurance cost for an individual. Between these three explanatory variables, the smoker status shows the highest influence on the insurance cost which is reasonable since it can lead to develop quickly new illnesses.

8. Summary

- We started with a dataset containing 1338 observations of health insured person with predictor variables explaining the profile of each person and the total amount they paid for the health insurance coverage.
- Amount paid was predicted by age, sex, body mass index, number of children, smoker status and region as predictor variables.
- Upon explanatory analysis, we concluded that sex and region is not having much impact on amount paid. Therefore, removed those variables from our analysis.
- Then, we did model selection using AIC, BIC, Mallow Cp and Adj R2 criteria and concluded with 2 models:
Candidate 1: charges~age+bmi+children+is_smoke ->Full model
Candidate 2: charges~age+bmi+is_smoker ->Reduced model
- Then we performed F-test to conclude that reduced model is better.
- Now, we started with model diagnostics for both full and reduced models and found that results were quite similar for both models.

Here were outliers and leverage points, and removing those improved normality and heteroskedasticity of the model. Therefore removed those points, and used weighted least squares for correction for heteroskedasticity. Also, there was not major auto-correction and multicollinearity in both models.

- Finally selected the reduced Model:

y= -8150.40+ 260.84*age + 173.23*bmi + 25693.62*is_smoker

Therefore cost of health insurance is depending on age, bmi and whether a person is a smoker or not.

Final Project - Linear Regression

Students:

- Sakshi Singla
- Ivette Sulca

URL_data: https://www.kaggle.com/ruslankl/health-care-cost-prediction-w-linear-regression/data (https://www.kaggle.com/ruslankl/health-care-cost-prediction-w-linear-regression/data)

1. Description of the Dataset

2. Statement of the research

3. Exploratory Analisis

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
import scipy.stats as stats
import matplotlib.cm as cm
from IPython.display import display
from mpl_toolkits.mplot3d import Axes3D
from sklearn.feature_selection import f_regression
from statsmodels.stats.anova import anova_lm
```

In [2]:

```
#Number of observations: 1338
#health_data=pd.read_csv('/Users/sakshisingla/Downloads/insurance.csv')
health_data=pd.read_csv('/Users/ivettesulca/Desktop/Linear_Regression/proje
health_data.head()
```

Out[2]:

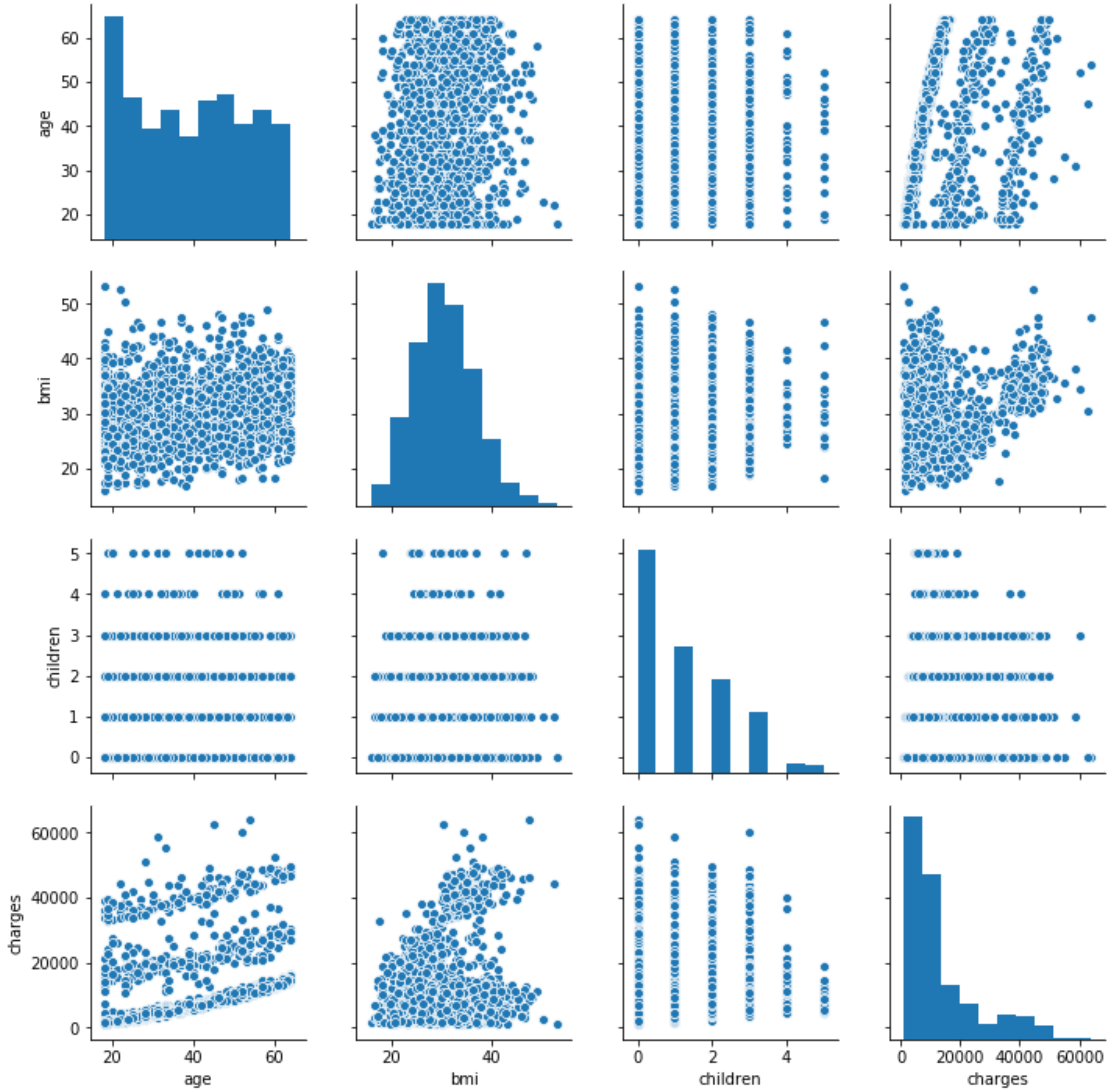
	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

In [3]:

```
#Correlation Matrix : Quantitative variables
import seaborn as sns
sns.pairplot(health_data)
```

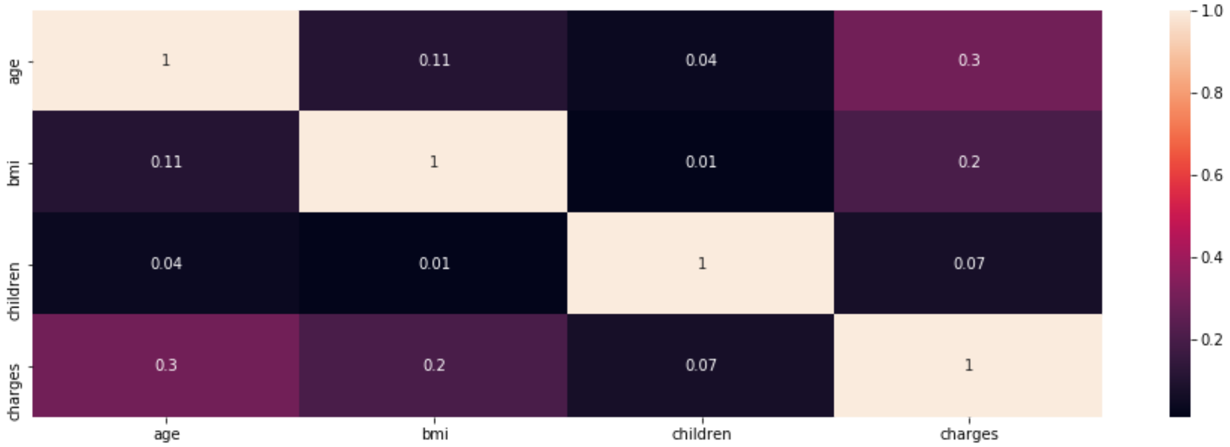
Out[3]:

```
<seaborn.axisgrid.PairGrid at 0x109934f60>
```

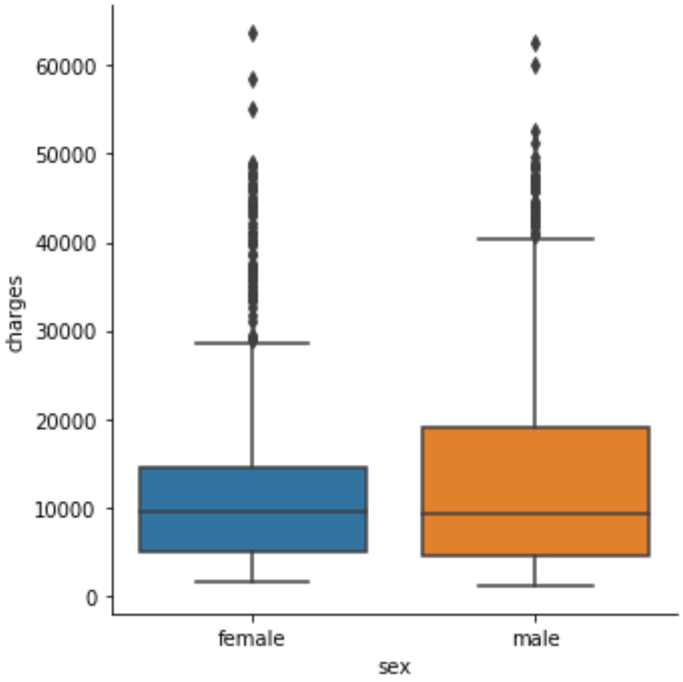


In [4]:
```
#Correlation Matrix : Quantitative variables
correlation_matrix = health_data.corr().round(2)
plt.figure(figsize = (16,5))
sns.heatmap(data=correlation_matrix, annot=True)
```

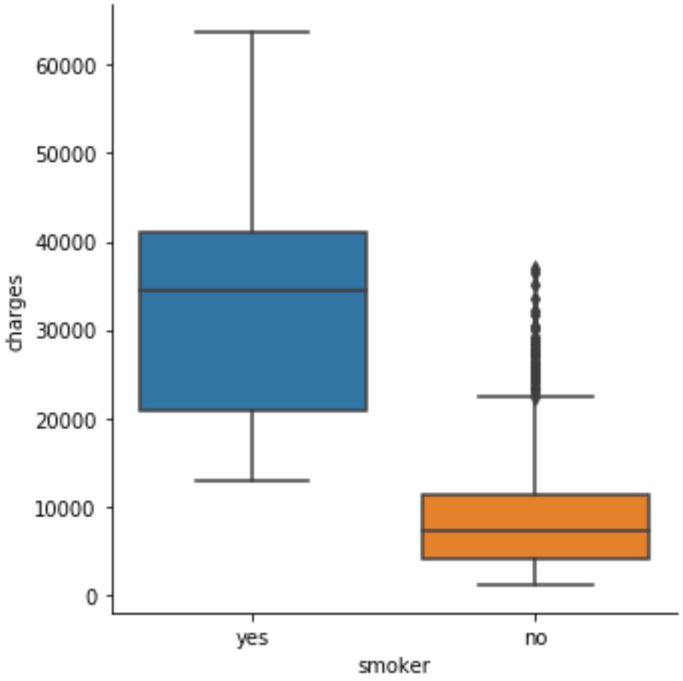
Out[4]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1c1fc20ef0>
```



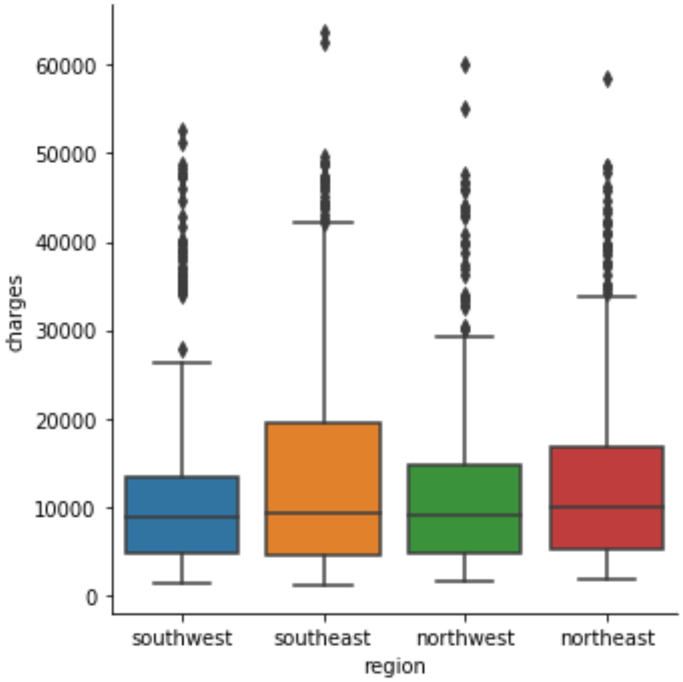
In [7]:
```
#Analysis of Categorical predictors: SEX
sns.catplot(x="sex", y="charges",kind="box", data=health_data);
# There seems to be no relationship between sex and charges
```



In [8]:
```
#Analysis of Categorical predictors: SMOKER
sns.catplot(x="smoker", y="charges",kind="box", data=health_data);
# There is a relationship between SMOKER and HEALTH CHARGES
```



In [9]:
```
#Analysis of Categorical predictors: REGION
sns.catplot(x="region", y="charges",kind="box", data=health_data);
# We dont see a relationship between REGION and HEALTH CHARGES.
```



In [10]:
```
health_data.head()
```

Out[10]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

4. Regression Model

In [11]:
```
#Creating dummy variables:
sex_cols=pd.get_dummies(health_data['sex'],drop_first=True)
sex_cols.columns=['sex_male']

smoker_cols=pd.get_dummies(health_data['smoker'],drop_first=True)
smoker_cols.columns=['is_smoker']

region_cols=pd.get_dummies(health_data['region'],drop_first=True)
region_cols

health_data_dumm=pd.concat([health_data[['charges','age','bmi','children']]
health_data_dumm.head()
```

Out[11]:

	charges	age	bmi	children	sex_male	is_smoker	northwest	southeast	southwest
0	16884.92400	19	27.900	0	0	1	0	0	1
1	1725.55230	18	33.770	1	1	0	0	1	0
2	4449.46200	28	33.000	3	1	0	0	1	0
3	21984.47061	33	22.705	0	1	0	1	0	0
4	3866.85520	32	28.880	0	1	0	1	0	0

In [12]:

```
##MODEL 1: WITH ALL VARIABLES
reg1 = smf.ols('charges~age+bmi+children+sex_male+is_smoker+northwest+south
#reg1 = smf.ols('charges~age+bmi+children+is_smoker',data=health_data_dumm)
reg1.summary()
#The individual tests shows that sex and region are not significant.
```

Out[12]:

```
OLS Regression Results

     Dep. Variable:          charges         R-squared:      0.751
             Model:              OLS    Adj. R-squared:      0.749
            Method:    Least Squares       F-statistic:      500.8
              Date: Sun, 13 Oct 2019 Prob (F-statistic):      0.00
              Time:         17:46:40    Log-Likelihood:    -13548.
 No. Observations:             1338               AIC:  2.711e+04
      Df Residuals:             1329               BIC:  2.716e+04
          Df Model:                8
  Covariance Type:        nonrobust

                    coef  std err        t  P>|t|      [0.025     0.975]
  Intercept  -1.194e+04  987.819  -12.086  0.000   -1.39e+04     -1e+04
        age   256.8564   11.899   21.587  0.000     233.514    280.199
        bmi   339.1935   28.599   11.860  0.000     283.088    395.298
   children   475.5005  137.804    3.451  0.001     205.163    745.838
   sex_male  -131.3144  332.945   -0.394  0.693    -784.470    521.842
  is_smoker  2.385e+04  413.153   57.723  0.000     2.3e+04   2.47e+04
  northwest  -352.9639  476.276   -0.741  0.459   -1287.298    581.370
  southeast -1035.0220  478.692   -2.162  0.031   -1974.097    -95.947
  southwest  -960.0510  477.933   -2.009  0.045   -1897.636    -22.466

        Omnibus: 300.366      Durbin-Watson:      2.088
 Prob(Omnibus):    0.000  Jarque-Bera (JB):    718.887
           Skew:    1.211          Prob(JB):  7.86e-157
       Kurtosis:    5.651          Cond. No.       311.


Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

In [13]:

```
##MODEL 2: DELETING SEX AND REGION
reg2 = smf.ols('charges~age+bmi+children+is_smoker',data=health_data_dumm).
reg2.summary()
```

Out[13]:

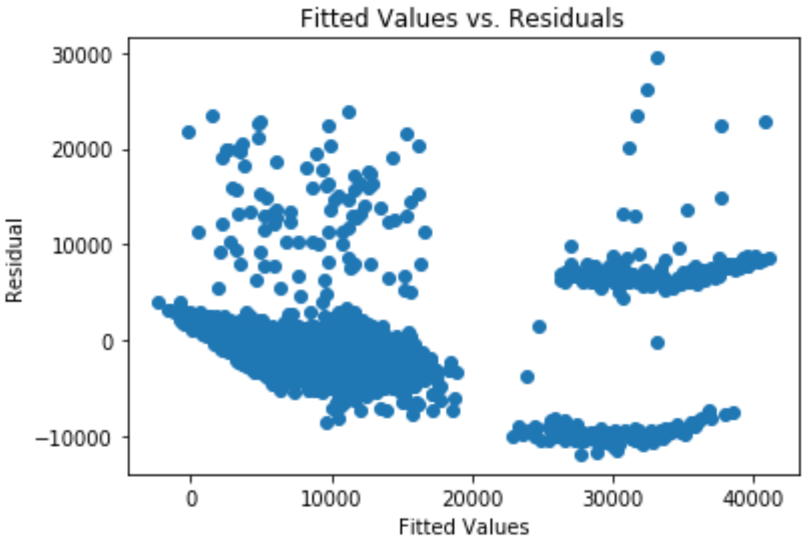
```
OLS Regression Results

     Dep. Variable:          charges          R-squared:      0.750
             Model:              OLS      Adj. R-squared:      0.749
            Method:    Least Squares         F-statistic:      998.1
              Date: Sun, 13 Oct 2019  Prob (F-statistic):       0.00
              Time:         17:46:40      Log-Likelihood:    -13551.
 No. Observations:              1338                 AIC:  2.711e+04
      Df Residuals:             1333                 BIC:  2.714e+04
          Df Model:                4
 Covariance Type:          nonrobust

                 coef    std err        t   P>|t|    [0.025      0.975]
 Intercept  -1.21e+04    941.984  -12.848   0.000   -1.4e+04   -1.03e+04
       age   257.8495     11.896   21.675   0.000   234.512     281.187
       bmi   321.8514     27.378   11.756   0.000   268.143     375.559
  children   473.5023    137.792    3.436   0.001   203.190     743.814
 is_smoker  2.381e+04    411.220   57.904   0.000    2.3e+04    2.46e+04

       Omnibus:  301.480     Durbin-Watson:      2.087
 Prob(Omnibus):    0.000  Jarque-Bera (JB):    722.157
          Skew:    1.215          Prob(JB):  1.53e-157
      Kurtosis:    5.654          Cond. No.       292.


Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

In [14]:

```
#FITTED VALUES VS. RESIDUALS: MODEL 2

p = reg2.fittedvalues
res = reg2.resid
plt.scatter(p,res)
plt.xlabel("Fitted Values")
plt.ylabel("Residual")
plt.title("Fitted Values vs. Residuals")
```

Out[14]:

```
Text(0.5, 1.0, 'Fitted Values vs. Residuals')
```



5. Model selection

In [15]:

```
##Mallow Cp and R^2
from tqdm import tnrange, tqdm_notebook
import itertools
import math

#full_model = smf.ols('charges~age+bmi+children+sex_male+is_smoker+northwes
full_model = smf.ols('charges~age+bmi+children+is_smoker',data=health_data_
Y=health_data_dumm[['charges']]
X=health_data_dumm[['age','bmi','children','is_smoker']]

SSE_p = (full_model.resid**2).sum()
MSE_p=full_model.mse_resid

###permutations of reduced model
n_pred=X.shape[1]
n_obs=X.shape[0]

list_R_adj=[]
list_CP=[]
list_predictors=[]
list_SSE=[]
list_R=[]
list_BIC=[]
list_AIC=[]

for k in range(0,n_pred+1):
    for combo in itertools.combinations(X.columns,k):
        text_predictors='charges~'
        p=1 #number of parameters

        for t in combo:
            text_predictors+=t+"+"
            p+=1
        text_predictors=text_predictors[:-1]

        if k==0:
            text_predictors='charges~1'
            p=1

        reduced_model = smf.ols(text_predictors,data=health_data_dumm).fit(

        SSE_k = (reduced_model.resid**2).sum()
        #CP and R^2
        CP = SSE_k/MSE_p -(n_obs-2*p)
        R_adj=round(reduced_model.rsquared_adj,5)
        R=round(reduced_model.rsquared,5)

        #BIC and AIC
        AIC=n_obs*math.log(2*math.pi)+n_obs*math.log(SSE_k)-n_obs*math.log(
        BIC=n_obs*math.log(2*math.pi)+n_obs*math.log(SSE_k)-n_obs*math.log(

        list_predictors.append(text_predictors)
        list_CP.append(CP)
        list_R_adj.append(R_adj)
        list_SSE.append(SSE_k)
        list_R.append(R)
        list_AIC.append(AIC)
        list_BIC.append(BIC)

df_model_selection = pd.DataFrame({'Predictors': list_predictors,'CP': list
                  'R':list_R, 'AIC':list_AIC,'BIC':list_BIC,'SSE':list_SSE
df_model_selection
```

Out[15]:

	Predictors	CP	R_adj	R	AIC	BIC
0	charges~1	3989.492894	0.00000	0.00000	27619.263199	27624.462130
1	charges~age	3515.362411	0.08872	0.08941	27495.948748	27506.346610
2	charges~bmi	3781.992584	0.03862	0.03934	27567.564127	27577.961989
3	charges~children	3966.869099	0.00388	0.00462	27615.062262	27625.460125
4	charges~is_smoker	690.939777	0.61948	0.61976	26327.463615	26337.861477
5	charges~age+bmi	3369.433604	0.11586	0.11718	27456.497842	27472.094636
6	charges~age+children	3501.047355	0.09111	0.09247	27493.439630	27509.036423
7	charges~age+is_smoker	151.677921	0.72098	0.72140	25913.324376	25928.921170
8	charges~bmi+children	3761.163760	0.04219	0.04363	27563.580279	27579.177073
9	charges~bmi+is_smoker	489.627477	0.65743	0.65794	26187.890316	26203.487110
10	charges~children+is_smoker	672.495400	0.62304	0.62360	26315.886123	26331.482917
11	charges~age+bmi+children	3355.910806	0.11812	0.12010	27454.072837	27474.868562
12	charges~age+bmi+is_smoker	14.808596	0.74691	0.74748	25783.835916	25804.631641
13	charges~age+children+is_smoker	141.203501	0.72312	0.72374	25904.027223	25924.822948
14	charges~bmi+children+is_smoker	472.788621	0.66072	0.66148	26175.980976	26196.776701
15	charges~age+bmi+children+is_smoker	5.000000	0.74894	0.74969	25774.035219	25800.029875

In [16]:

```
#Model selection based on CP:
df_model_selection.sort_values('CP',ascending=True)

#Selected models:
#charges~age+bmi+children+is_smoke
#charges~age+bmi+is_smoker
```

Out[16]:

	Predictors	CP	R_adj	R	AIC	BIC
15	charges~age+bmi+children+is_smoker	5.000000	0.74894	0.74969	25774.035219	25800.029875
12	charges~age+bmi+is_smoker	14.808596	0.74691	0.74748	25783.835916	25804.631641
13	charges~age+children+is_smoker	141.203501	0.72312	0.72374	25904.027223	25924.822948
7	charges~age+is_smoker	151.677921	0.72098	0.72140	25913.324376	25928.921170
14	charges~bmi+children+is_smoker	472.788621	0.66072	0.66148	26175.980976	26196.776701
9	charges~bmi+is_smoker	489.627477	0.65743	0.65794	26187.890316	26203.487110
10	charges~children+is_smoker	672.495400	0.62304	0.62360	26315.886123	26331.482917
4	charges~is_smoker	690.939777	0.61948	0.61976	26327.463615	26337.861477
11	charges~age+bmi+children	3355.910806	0.11812	0.12010	27454.072837	27474.868562
5	charges~age+bmi	3369.433604	0.11586	0.11718	27456.497842	27472.094636
6	charges~age+children	3501.047355	0.09111	0.09247	27493.439630	27509.036423
1	charges~age	3515.362411	0.08872	0.08941	27495.948748	27506.346610
8	charges~bmi+children	3761.163760	0.04219	0.04363	27563.580279	27579.177073
2	charges~bmi	3781.992584	0.03862	0.03934	27567.564127	27577.961989
3	charges~children	3966.869099	0.00388	0.00462	27615.062262	27625.460125
0	charges~1	3989.492894	0.00000	0.00000	27619.263199	27624.462130

In [17]:

```
#Model selection based on R_adj:
df_model_selection.sort_values('R_adj',ascending=False)

#Models selected
#charges~age+bmi+children+is_smoker
#charges~age+bmi+is_smoke
```

Out[17]:

	Predictors	CP	R_adj	R	AIC	BIC
15	charges~age+bmi+children+is_smoker	5.000000	0.74894	0.74969	25774.035219	25800.029875
12	charges~age+bmi+is_smoker	14.808596	0.74691	0.74748	25783.835916	25804.631641
13	charges~age+children+is_smoker	141.203501	0.72312	0.72374	25904.027223	25924.822948
7	charges~age+is_smoker	151.677921	0.72098	0.72140	25913.324376	25928.921170
14	charges~bmi+children+is_smoker	472.788621	0.66072	0.66148	26175.980976	26196.776701
9	charges~bmi+is_smoker	489.627477	0.65743	0.65794	26187.890316	26203.487110
10	charges~children+is_smoker	672.495400	0.62304	0.62360	26315.886123	26331.482917
4	charges~is_smoker	690.939777	0.61948	0.61976	26327.463615	26337.861477
11	charges~age+bmi+children	3355.910806	0.11812	0.12010	27454.072837	27474.868562
5	charges~age+bmi	3369.433604	0.11586	0.11718	27456.497842	27472.094636
6	charges~age+children	3501.047355	0.09111	0.09247	27493.439630	27509.036423
1	charges~age	3515.362411	0.08872	0.08941	27495.948748	27506.346610
8	charges~bmi+children	3761.163760	0.04219	0.04363	27563.580279	27579.177073
2	charges~bmi	3781.992584	0.03862	0.03934	27567.564127	27577.961989
3	charges~children	3966.869099	0.00388	0.00462	27615.062262	27625.460125
0	charges~1	3989.492894	0.00000	0.00000	27619.263199	27624.462130

In [18]:

```
#Model selection based on AIC:
df_model_selection.sort_values('AIC',ascending=True)

#Models selected
#charges~age+bmi+children+is_smoker
#charges~age+bmi+is_smoke
```

Out[18]:

	Predictors	CP	R_adj	R	AIC	BIC
15	charges~age+bmi+children+is_smoker	5.000000	0.74894	0.74969	25774.035219	25800.029875
12	charges~age+bmi+is_smoker	14.808596	0.74691	0.74748	25783.835916	25804.631641
13	charges~age+children+is_smoker	141.203501	0.72312	0.72374	25904.027223	25924.822948
7	charges~age+is_smoker	151.677921	0.72098	0.72140	25913.324376	25928.921170
14	charges~bmi+children+is_smoker	472.788621	0.66072	0.66148	26175.980976	26196.776701
9	charges~bmi+is_smoker	489.627477	0.65743	0.65794	26187.890316	26203.487110
10	charges~children+is_smoker	672.495400	0.62304	0.62360	26315.886123	26331.482917
4	charges~is_smoker	690.939777	0.61948	0.61976	26327.463615	26337.861477
11	charges~age+bmi+children	3355.910806	0.11812	0.12010	27454.072837	27474.868562
5	charges~age+bmi	3369.433604	0.11586	0.11718	27456.497842	27472.094636
6	charges~age+children	3501.047355	0.09111	0.09247	27493.439630	27509.036423
1	charges~age	3515.362411	0.08872	0.08941	27495.948748	27506.346610
8	charges~bmi+children	3761.163760	0.04219	0.04363	27563.580279	27579.177073
2	charges~bmi	3781.992584	0.03862	0.03934	27567.564127	27577.961989
3	charges~children	3966.869099	0.00388	0.00462	27615.062262	27625.460125
0	charges~1	3989.492894	0.00000	0.00000	27619.263199	27624.462130

In [19]:

```
#Model selection based on BIC:
df_model_selection.sort_values('BIC',ascending=True)

#Models selected
#charges~age+bmi+children+is_smoker
#charges~age+bmi+is_smoke
```

Out[19]:

	Predictors	CP	R_adj	R	AIC	BIC
15	charges~age+bmi+children+is_smoker	5.000000	0.74894	0.74969	25774.035219	25800.029875
12	charges~age+bmi+is_smoker	14.808596	0.74691	0.74748	25783.835916	25804.631641
13	charges~age+children+is_smoker	141.203501	0.72312	0.72374	25904.027223	25924.822948
7	charges~age+is_smoker	151.677921	0.72098	0.72140	25913.324376	25928.921170
14	charges~bmi+children+is_smoker	472.788621	0.66072	0.66148	26175.980976	26196.776701
9	charges~bmi+is_smoker	489.627477	0.65743	0.65794	26187.890316	26203.487110
10	charges~children+is_smoker	672.495400	0.62304	0.62360	26315.886123	26331.482917
4	charges~is_smoker	690.939777	0.61948	0.61976	26327.463615	26337.861477
5	charges~age+bmi	3369.433604	0.11586	0.11718	27456.497842	27472.094636
11	charges~age+bmi+children	3355.910806	0.11812	0.12010	27454.072837	27474.868562
1	charges~age	3515.362411	0.08872	0.08941	27495.948748	27506.346610
6	charges~age+children	3501.047355	0.09111	0.09247	27493.439630	27509.036423
2	charges~bmi	3781.992584	0.03862	0.03934	27567.564127	27577.961989
8	charges~bmi+children	3761.163760	0.04219	0.04363	27563.580279	27579.177073
0	charges~1	3989.492894	0.00000	0.00000	27619.263199	27624.462130
3	charges~children	3966.869099	0.00388	0.00462	27615.062262	27625.460125

In [20]:

```
#CONCLUSION: We select these two models:
#Selected models:
#charges~age+bmi+children+is_smoker
#charges~age+bmi+is_smoker
```

In [21]:

```
#ANOVA Full model
sm.stats.anova_lm(reg1, typ=1)
```

Out[21]:

	df	sum_sq	mean_sq	F	PR(>F)
age	1.0	1.753019e+10	1.753019e+10	477.023920	1.311803e-90
bmi	1.0	5.446449e+09	5.446449e+09	148.206388	2.114426e-32
children	1.0	5.715190e+08	5.715190e+08	15.551926	8.445622e-05
sex_male	1.0	5.824524e+08	5.824524e+08	15.849440	7.229729e-05
is_smoker	1.0	1.228706e+11	1.228706e+11	3343.502231	0.000000e+00
northwest	1.0	2.167002e+07	2.167002e+07	0.589675	4.426812e-01
southeast	1.0	6.347488e+07	6.347488e+07	1.727251	1.889892e-01
southwest	1.0	1.482863e+08	1.482863e+08	4.035102	4.476493e-02
Residual	1329.0	4.883953e+10	3.674908e+07	NaN	NaN

In [22]:

```
#ANOVA Reduced model
sm.stats.anova_lm(reg2, typ=1)
```

Out[22]:

	df	sum_sq	mean_sq	F	PR(>F)
age	1.0	1.753019e+10	1.753019e+10	476.130483	1.675025e-90
bmi	1.0	5.446449e+09	5.446449e+09	147.928806	2.371721e-32
children	1.0	5.715190e+08	5.715190e+08	15.522798	8.573928e-05
is_smoker	1.0	1.234476e+11	1.234476e+11	3352.910806	0.000000e+00
Residual	1333.0	4.907845e+10	3.681804e+07	NaN	NaN

In [23]:

```
#####F-Test from ANOVA TABLE####
#Full Model
SSE_full=4.89*((10)**10)
n=1338
p=9
df_full=n-p

#Reduced Model

SSE_red=4.90*((10)**10)
n=1338
p=5
df_red=n-p


#F-statistic
F_stat=((SSE_red-SSE_full)/(df_red-df_full))/(SSE_full/df_full)
F_stat
```

Out[23]:

```
0.6794478527607362
```

In [24]:

```
from scipy.stats import f
from scipy.stats import norm

p_value=1-f.cdf(0.67944, df_red-df_full, df_full)
p_value
```

Out[24]:

```
0.6062384669375296
```

6. Model Diagnosis

In [26]:

```
health_data_dumm.head()
```

Out[26]:

	charges	age	bmi	children	sex_male	is_smoker	northwest	southeast	southwest
0	16884.92400	19	27.900	0	0	1	0	0	1
1	1725.55230	18	33.770	1	1	0	0	1	0
2	4449.46200	28	33.000	3	1	0	0	1	0
3	21984.47061	33	22.705	0	1	0	1	0	0
4	3866.85520	32	28.880	0	1	0	1	0	0

In [28]:

```
#Regression for full model
reg1 = smf.ols('charges~age+bmi+children+is_smoker',data=health_data_dumm).
reg1.summary()
```

Out[28]:

```
OLS Regression Results

     Dep. Variable:          charges         R-squared:      0.750
            Model:               OLS    Adj. R-squared:      0.749
           Method:     Least Squares       F-statistic:      998.1
             Date:  Sat, 12 Oct 2019  Prob (F-statistic):     0.00
             Time:          14:24:48    Log-Likelihood:    -13551.
 No. Observations:              1338               AIC:  2.711e+04
     Df Residuals:              1333               BIC:  2.714e+04
         Df Model:                 4
  Covariance Type:         nonrobust

                coef    std err        t   P>|t|    [0.025     0.975]
 Intercept  -1.21e+04   941.984  -12.848   0.000   -1.4e+04  -1.03e+04
       age   257.8495    11.896   21.675   0.000    234.512    281.187
       bmi   321.8514    27.378   11.756   0.000    268.143    375.559
  children   473.5023   137.792    3.436   0.001    203.190    743.814
 is_smoker  2.381e+04   411.220   57.904   0.000     2.3e+04   2.46e+04

      Omnibus:  301.480    Durbin-Watson:       2.087
Prob(Omnibus):    0.000  Jarque-Bera (JB):    722.157
         Skew:    1.215          Prob(JB):  1.53e-157
     Kurtosis:    5.654          Cond. No.       292.


Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

In [29]:

```
#Regression for reduced model
reg2 = smf.ols('charges~age+bmi+is_smoker',data=health_data_dumm).fit()
reg2.summary()
```

Out[29]:

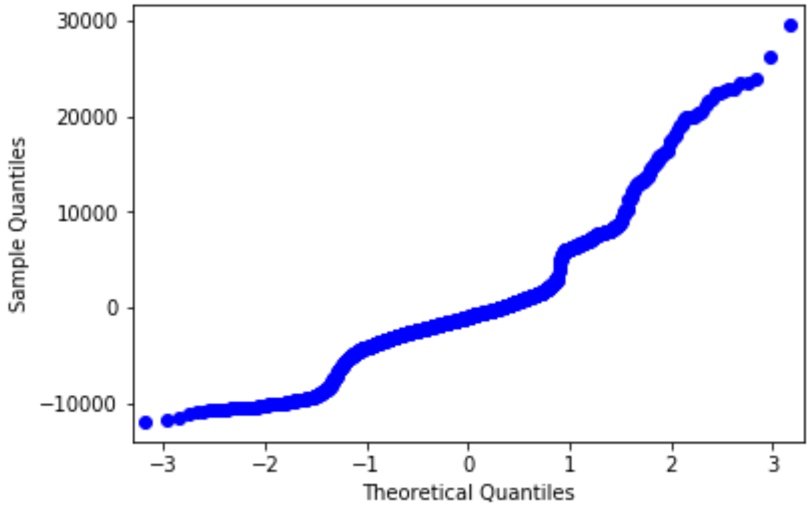
```
OLS Regression Results

       Dep. Variable:          charges        R-squared:       0.747
              Model:              OLS    Adj. R-squared:       0.747
             Method:    Least Squares       F-statistic:       1316.
               Date: Sat, 12 Oct 2019  Prob (F-statistic):      0.00
               Time:         14:25:14    Log-Likelihood:     -13557.
   No. Observations:             1338               AIC:   2.712e+04
        Df Residuals:            1334               BIC:   2.714e+04
           Df Model:                3
    Covariance Type:        nonrobust

                  coef   std err        t  P>|t|    [0.025     0.975]
 Intercept  -1.168e+04   937.569  -12.454  0.000  -1.35e+04  -9837.561
       age    259.5475    11.934   21.748  0.000    236.136    282.959
       bmi    322.6151    27.487   11.737  0.000    268.692    376.538
 is_smoker   2.382e+04   412.867   57.703  0.000     2.3e+04   2.46e+04

        Omnibus: 299.709     Durbin-Watson:      2.077
  Prob(Omnibus):   0.000  Jarque-Bera (JB):    710.137
           Skew:   1.213          Prob(JB):  6.25e-155
       Kurtosis:   5.618          Cond. No.       289.


Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

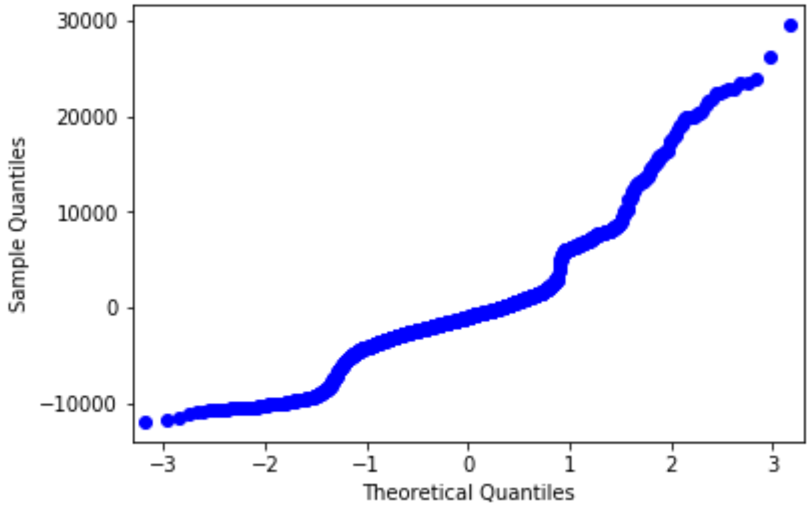
In [40]:

```
#2. check residual for full model
# Normality
JB, JBpv,skw,kurt = sm.stats.stattools.jarque_bera(reg.resid)
print(JB,JBpv,skw,kurt)
# p-value is not good
sm.qqplot(reg1.resid)
```

```
722.1565054761726 1.5335830526321106e-157 1.2152404041806357 5.6544760234
446745
```

Out[40]:



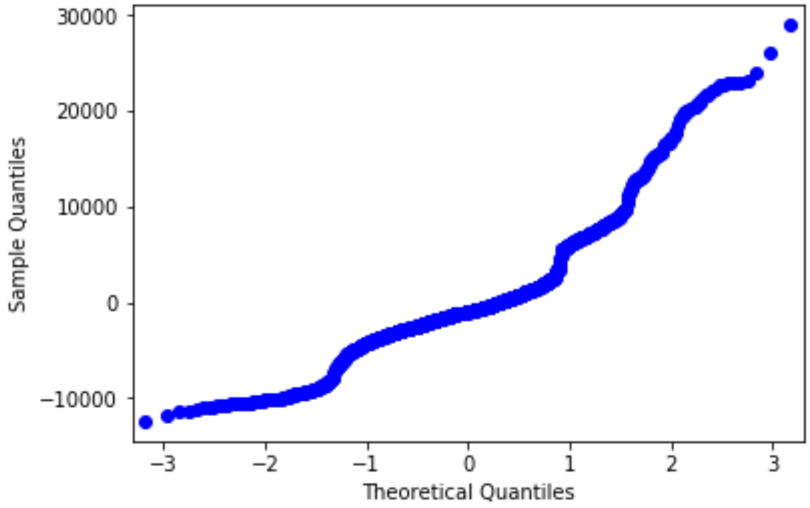


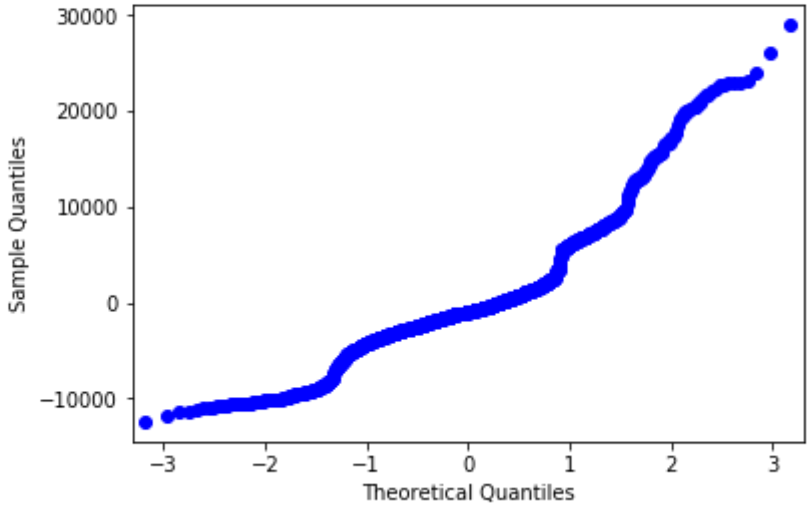
In [41]:

```
#2. check residual for reduced model
# Normality
JB, JBpv,skw,kurt = sm.stats.stattools.jarque_bera(reg2.resid)
print(JB,JBpv,skw,kurt)
# p-value is not good
sm.qqplot(reg2.resid)
```

```
710.137418335741 6.246243519708224e-155 1.2130482786943517 5.617622281017
362
```

Out[41]:





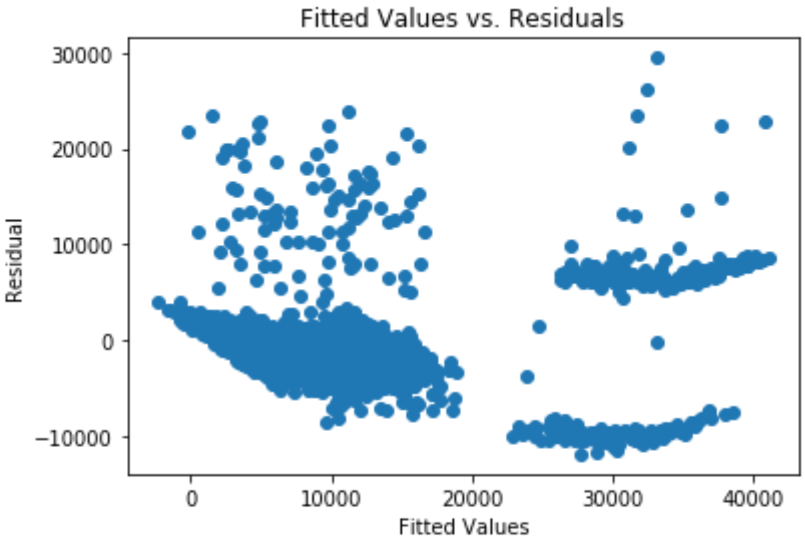
Model 2 is slightly better but both are having non normality

In [42]:

```
#2.2 Fitted Values vs. Residuals for full model
p = reg1.fittedvalues
res = reg1.resid
plt.scatter(p,res)
plt.xlabel("Fitted Values")
plt.ylabel("Residual")
plt.title("Fitted Values vs. Residuals")
#nonlinearity is there
```

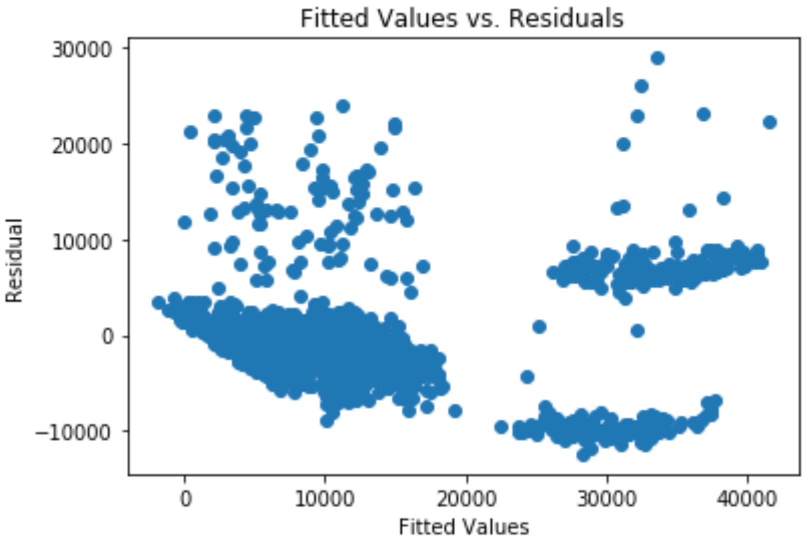
Out[42]:

```
Text(0.5, 1.0, 'Fitted Values vs. Residuals')
```



In [43]:
```
#2.2 Fitted Values vs. Residuals for reduced model
p = reg2.fittedvalues
res = reg2.resid
plt.scatter(p,res)
plt.xlabel("Fitted Values")
plt.ylabel("Residual")
plt.title("Fitted Values vs. Residuals")
#nonlinearity is there
```

Out[43]:
```
Text(0.5, 1.0, 'Fitted Values vs. Residuals')
```



Almost same results. There is some variance in residuals.

Diagnostic tests and correction for full model

```
In [45]: #influential points
         #object for the analysis of influential points
         infl1 = reg1.get_influence()
         infl2 = reg2.get_influence()
         #members
         print(dir(infl))
```

```
['__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__',
'__format__', '__ge__', '__getattribute__', '__gt__', '__hash__', '__init
__', '__init_subclass__', '__le__', '__lt__', '__module__', '__ne__', '__
new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__size
of__', '__str__', '__subclasshook__', '__weakref__', '_cache', '_get_drop
_vari', '_ols_xnoti', '_res_looo', 'aux_regression_endog', 'aux_regressio
n_exog', 'cooks_distance', 'cov_ratio', 'det_cov_params_not_obsi', 'dfbet
as', 'dffits', 'dffits_internal', 'endog', 'ess_press', 'exog', 'get_resi
d_studentized_external', 'hat_diag_factor', 'hat_matrix_diag', 'influenc
e', 'k_vars', 'model_class', 'nobs', 'params_not_obsi', 'resid_press', 'r
esid_std', 'resid_studentized_external', 'resid_studentized_internal', 'r
esid_var', 'results', 'sigma2_not_obsi', 'sigma_est', 'summary_frame', 's
ummary_table']
```

For model 1:Influential points

```
In [46]: #leverage
         print(infl1.hat_matrix_diag)
```

```
[0.00572181 0.00302181 0.0035557  ... 0.00427337 0.00307642 0.00644726]
```

```
In [47]: #internally studentized residuals
         print(infl1.resid_studentized_internal)
```

```
[-1.43833415 -0.35575698 -0.44727795 ... -0.45730802  0.06472097
 -1.2652221 ]
```

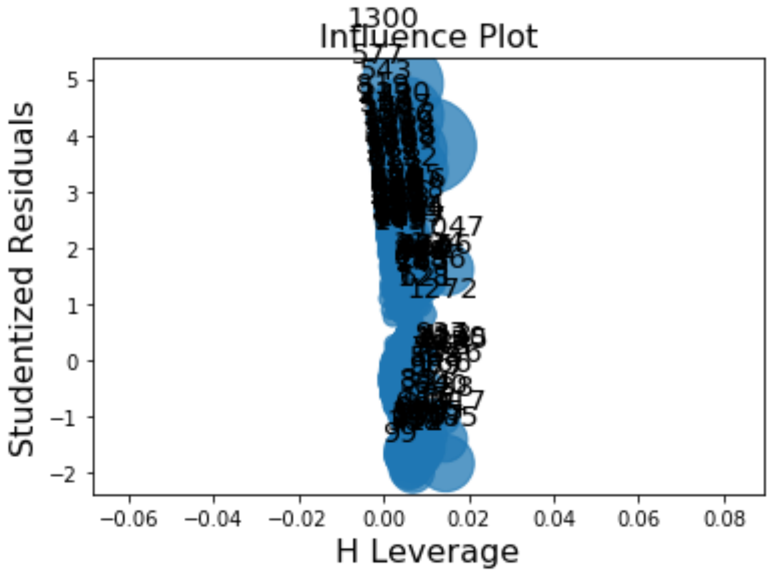
```
In [48]: #externally studentized residuals
         print(infl1.resid_studentized_external)
```

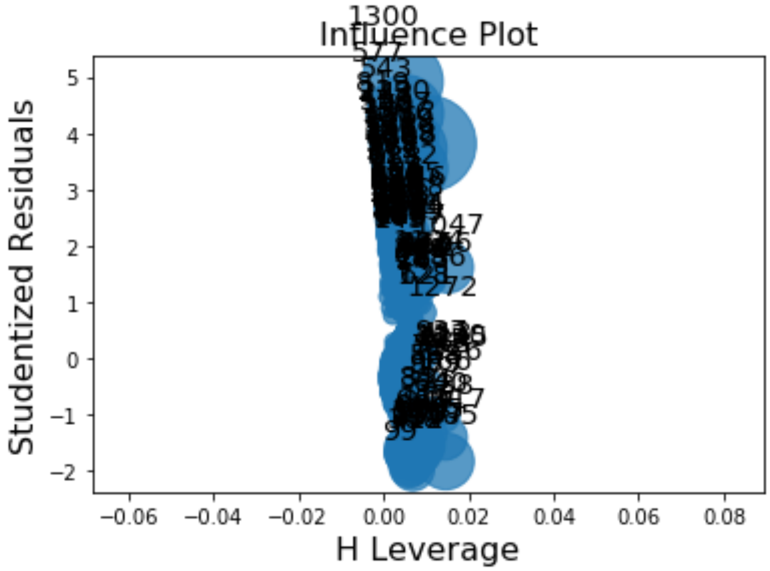
```
[-1.43891156 -0.35564039 -0.4471437  ... -0.45717232  0.06469679
 -1.26550753]
```

In [49]:

```
#graphical representation of the influences()
sm.graphics.influence_plot(reg1)
```

Out[49]:





In [50]:

```
#too messy. define your rules.
#threshold leverage
residus = reg1.resid.as_matrix() #residuals
leviers = infl1.hat_matrix_diag  #leverage
n = health_data_dumm.shape[0]
p=5
seuil_levier = 2*p/n # people choose 2.5 or 3 as well
print(seuil_levier)
#identification
atyp_levier = leviers > seuil_levier
print(atyp_levier)
```

```
0.007473841554559043
[False False False ... False False False]
```

```
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:3: FutureWar
ning: Method .as_matrix will be removed in a future version. Use .values
instead.
  This is separate from the ipykernel package so we can avoid doing impor
ts until
```

In [52]:

```
#too hard to read
print(health_data_dumm.index[atyp_levier],leviers[atyp_levier])
```

```
Int64Index([  14,   32,   39,   71,   83,   98,  116,  128,  166,  185,
250,
             265,  281,  292,  301,  344,  380,  412,  413,  425,  438,
494,
             543,  549,  568,  621,  640,  660,  664,  674,  803,  847,
860,
             877,  901,  932,  937,  969,  984,  994, 1047, 1085, 1116, 1
124,
            1130, 1156, 1186, 1245, 1265, 1272, 1307, 1317],
           dtype='int64') [0.00768125 0.01073336 0.00750771 0.0092878  0.
00754973 0.00823485
 0.00923105 0.00750688 0.01156398 0.00809413 0.00889485 0.00820321
 0.0079519  0.00964437 0.0077778  0.00768758 0.00764957 0.00810981
 0.01061097 0.00979143 0.01403164 0.00967911 0.01048509 0.00916477
 0.00903956 0.00820012 0.01194986 0.00801453 0.00846788 0.0075313
 0.00900637 0.01052002 0.009902   0.00924631 0.00787025 0.00946798
 0.00970682 0.00908174 0.01053691 0.00818723 0.01513658 0.01466071
 0.00884004 0.00788926 0.00980432 0.01032    0.00758461 0.01020894
 0.00820157 0.00942602 0.00829636 0.01447816]
```

In [54]:

```
#threshold externally studentized residuals
import scipy
seuil_stud = scipy.stats.t.ppf(0.975,df=n-p-1)

#detection - absolute value > threshold
reg_studs=infl1.resid_studentized_external
atyp_stud = np.abs(reg_studs) > seuil_stud
#which ones?
print(health_data_dumm.index[atyp_stud],reg_studs[atyp_stud])
```

```
Int64Index([   3,    9,   34,   62,   99,  102,  115,  138,  140,  143,
219,
             242,  245,  289,  291,  305,  306,  321,  340,  355,  379,
387,
             397,  429,  430,  443,  468,  488,  491,  516,  520,  526,
539,
             543,  573,  577,  583,  587,  599,  637,  658,  688,  696,
739,
             806,  819,  858,  876,  925,  936,  959,  964,  980,  987, 1
008,
            1012, 1019, 1027, 1039, 1104, 1123, 1134, 1142, 1146, 1157, 1
195,
            1206, 1211, 1230, 1258, 1300, 1328],
           dtype='int64') [ 3.024589    2.8541619   3.32940668  2.8722592
3 -1.96976668  3.16576247
  2.92844454  2.28425829  3.73945972  2.02564622  3.904287    3.9788698
  2.1469096   2.44338612  2.46021772  2.04615842  2.52300445  3.09074804
  2.64386329  2.63903414  2.0685811   3.3766269   2.19656462  2.10613006
  3.25262268  2.1402063   3.29036127  2.2478375   2.1192884   3.78728912
  2.65600572  3.3921089   2.59479685  3.80964778  2.54744952  4.34701078
  2.21264374  2.19447207  3.16970781  2.44442791  2.32744755  2.97680119
  2.69212211  2.15513084  2.72385589  3.90017714  2.15195422  2.71356556
  2.19069372  3.70341629  2.63183422  2.04561694  2.48457778  3.2218622
  3.29779974  3.37565129  3.5207505   3.60873616  3.30943844  2.21198929
  2.17105574  2.25914333  2.94832248  2.46031465  2.01479649  2.5877892
  3.58969042  2.261063    3.7185462   2.38686515  4.91644271  3.28417563]
```

In [55]:

```
#suspicious observations with one of the two criteria
pbm_infl = np.logical_or(atyp_levier,atyp_stud)
print(health_data_dumm.index[pbm_infl])
```

```
Int64Index([   3,    9,   14,   32,   34,   39,   62,   71,   83,   98,
            ...
            1211, 1230, 1245, 1258, 1265, 1272, 1300, 1307, 1317, 1328],
           dtype='int64', length=123)
```

In [59]:

```
#DFFITS for detecting influential points

inflsum=infl1.summary_frame()
reg_dffits=inflsum.dffits
seuil_dffits=2*np.sqrt((p+1)/(n-p-1))
atyp_dffits = np.abs(reg_dffits) > seuil_dffits
# print(health_data_dumm.index[atyp_dffits],reg_dffits[atyp_dffits])
influ_DFFITS=health_data_dumm.index[atyp_dffits]
```

In [79]:
```
import scipy
seuil_stud = 5/1338
#detection - absolute value > threshold
cook_studs,pvalue=infl1.cooks_distance
atyp_cook = np.abs(cook_studs) > seuil_stud
#which ones?
print(health_data_dumm.index[atyp_cook],cook_studs[atyp_cook])
```

```
Int64Index([   3,    9,   34,   62,   69,   98,   99,  102,  115,  140,
219,
             242,  250,  266,  289,  321,  380,  387,  412,  430,  468,
488,
             491,  494,  516,  526,  543,  573,  577,  599,  688,  730,
739,
             754,  780,  793,  803,  806,  819,  847,  860,  896,  936,
994,
            1008, 1011, 1012, 1019, 1027, 1039, 1047, 1085, 1100, 1142, 1
146,
            1156, 1195, 1206, 1230, 1258, 1300, 1307, 1317, 1328],
           dtype='int64') [0.00521066 0.00638295 0.01072969 0.00706031 0.
00404725 0.00469132
 0.00538363 0.00640598 0.00587598 0.00778474 0.00996854 0.00723109
 0.00495071 0.00459631 0.00457148 0.01172495 0.00455505 0.00555543
 0.0048788  0.00693583 0.00455522 0.00553413 0.00385997 0.00468374
 0.00854248 0.00694026 0.03044872 0.00445184 0.01907625 0.00544371
 0.00377593 0.0039572  0.00463895 0.00389433 0.00394341 0.0048317
 0.00389982 0.00488223 0.01487195 0.00380881 0.00405566 0.00389113
 0.0039719  0.0046322  0.00587941 0.00469559 0.01569061 0.00710086
 0.01302549 0.00684464 0.00801497 0.00992318 0.00417362 0.00525094
 0.00747265 0.00430255 0.00618261 0.00760074 0.01728522 0.0050502
 0.02139853 0.00407646 0.0058161  0.00666209]
```

In [80]:
```
#Removing influential points
pbm_infl1 = np.logical_or(atyp_levier,atyp_stud)
pbm_infl2 = np.logical_or(atyp_cook,atyp_dffits)
pbm_infl = np.logical_or(pbm_infl1,pbm_infl2)
infl_pts=(health_data_dumm.index[pbm_infl])
```

In [75]:
```
health_data_without_influential=health_data_dumm.drop(infl_pts)
```

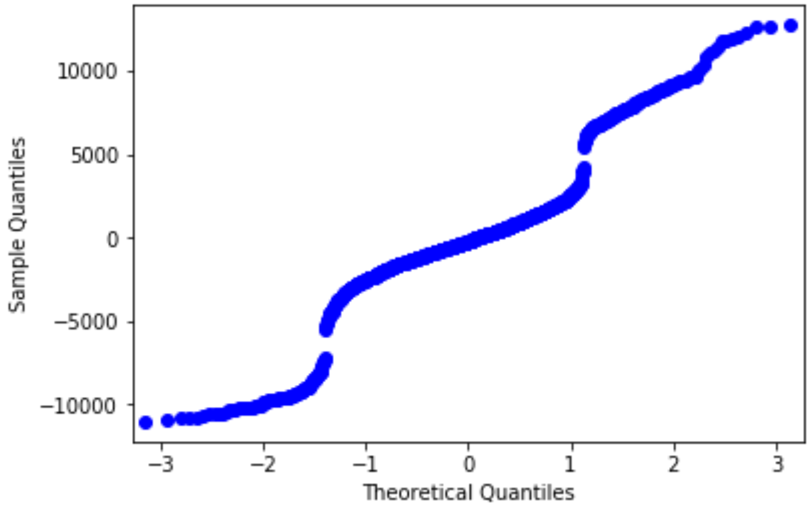
In [76]:
```
reg3 = smf.ols('charges~age+bmi+children+is_smoker',data=health_data_withou
reg3.summary()
```

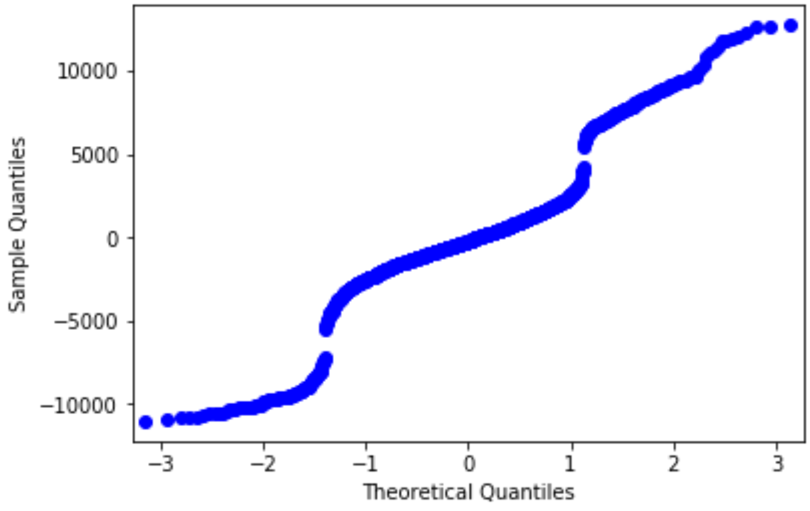
Out[76]:
```
OLS Regression Results

    Dep. Variable:         charges           R-squared:      0.855
            Model:             OLS      Adj. R-squared:      0.854
           Method:   Least Squares         F-statistic:      1762.
             Date: Sat, 12 Oct 2019  Prob (F-statistic):      0.00
             Time:        21:09:25      Log-Likelihood:    -11747.
 No. Observations:            1202                 AIC:  2.350e+04
     Df Residuals:            1197                 BIC:  2.353e+04
         Df Model:               4
  Covariance Type:       nonrobust

                  coef   std err        t  P>|t|     [0.025     0.975]
 Intercept  -1.097e+04   718.108  -15.281  0.000  -1.24e+04  -9564.594
       age    260.7491     8.814   29.583  0.000    243.456    278.042
       bmi    249.5205    21.361   11.681  0.000    207.612    291.429
  children    449.3417   111.718    4.022  0.000    230.156    668.527
 is_smoker   2.441e+04   315.835   77.281  0.000    2.38e+04    2.5e+04

        Omnibus: 29.546     Durbin-Watson:    2.068
 Prob(Omnibus):  0.000  Jarque-Bera (JB):   63.950
           Skew:  0.033          Prob(JB): 1.30e-14
       Kurtosis:  4.128          Cond. No.     301.


Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

In [77]:
```
#2. check residual
#2.1 Normality
JB, JBpv,skw,kurt = sm.stats.stattools.jarque_bera(reg3.resid)
print(JB,JBpv,skw,kurt)
# p-value is not good
sm.qqplot(reg3.resid)
```

```
63.949777371101284 1.298620589821719e-14 0.03294510562484799 4.1280629038
71845
```

Out[77]:



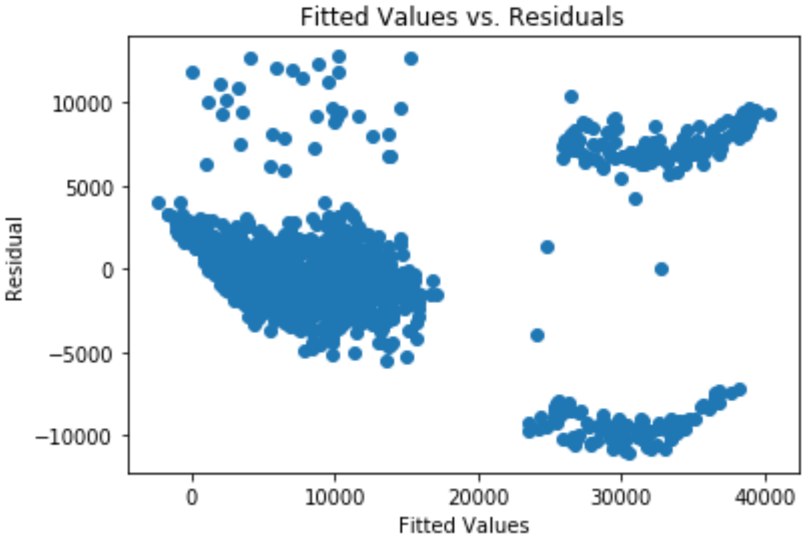


In [78]:

```
#2.2 Fitted Values vs. Residuals
p = reg3.fittedvalues
res = reg3.resid
plt.scatter(p,res)
plt.xlabel("Fitted Values")
plt.ylabel("Residual")
plt.title("Fitted Values vs. Residuals")
#nonlinearity
```

Out[78]:

```
Text(0.5, 1.0, 'Fitted Values vs. Residuals')
```



Therefore removing influential points have improved both normality and homoscedascity

In [82]:

```
#Breusch-Pagan for Heteroskedasticity
from statsmodels.stats.diagnostic import het_breuschpagan
bp_test = het_breuschpagan(reg3.resid, reg3.model.exog)
labels = ['LM Statistic', 'LM-Test p-value', 'F-Statistic', 'F-Test p-value
print(dict(zip(labels, bp_test)))
```

```
{'LM Statistic': 741.9669535355475, 'LM-Test p-value': 2.8473890576841394
e-159, 'F-Statistic': 482.6470892731169, 'F-Test p-value': 8.416450376812
329e-248}
```

In [83]:
```
#Breusch-Pagan for Heteroskedasticity
from statsmodels.stats.diagnostic import het_breuschpagan
bp_test = het_breuschpagan(reg1.resid, reg1.model.exog)
labels = ['LM Statistic', 'LM-Test p-value', 'F-Statistic', 'F-Test p-value
print(dict(zip(labels, bp_test)))
```

```
{'LM Statistic': 117.15665244370166, 'LM-Test p-value': 2.161944101621577
3e-24, 'F-Statistic': 31.979905124611555, 'F-Test p-value': 1.77558219771
74608e-25}
```

In [86]:
```
# therefore heteroskedascity exists
# use weighted least squares
```

In [141]:
```
from patsy import dmatrices
y, X = dmatrices('charges~age+bmi+children+is_smoker', health_data_without_
```

In [142]:
```
X.head()
```

Out[142]:

	Intercept	age	bmi	children	is_smoker
0	1.0	19.0	27.90	0.0	1.0
1	1.0	18.0	33.77	1.0	0.0
2	1.0	28.0	33.00	3.0	0.0
4	1.0	32.0	28.88	0.0	0.0
5	1.0	31.0	25.74	0.0	0.0

In [143]:
```
y.head()
```

Out[143]:

	charges
0	16884.9240
1	1725.5523
2	4449.4620
4	3866.8552
5	3756.6216

In [146]:
```
XTXI = np.linalg.inv((X.T).dot(X))
H=X.dot(XTXI.dot(X.T))
```

In [150]:
```
H.shape
```

Out[150]:
```
(1202, 1202)
```

In [151]:
```
I=np.identity(1202)
```

In [174]:
```
l=np.array((y.T).dot(I-H))
```

In [175]:
```
r=np.array(y)
```

In [177]:
```
SSE=l.dot(r)
```

In [180]:
```
MSE=SSE/(1202/5)
MSE=90165854.54827976
```

In [192]:
```
mat=MSE*(I-H)
mat=1/mat
```

In [194]:
```
W = np.diag(np.diag(mat))
W
```

Out[194]:
```
array([[1.11658995e-08, 0.00000000e+00, 0.00000000e+00, ...,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
        [0.00000000e+00, 1.11287143e-08, 0.00000000e+00, ...,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
        [0.00000000e+00, 0.00000000e+00, 1.11432905e-08, ...,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
        ...,
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
         1.11447609e-08, 0.00000000e+00, 0.00000000e+00],
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
         0.00000000e+00, 1.11280493e-08, 0.00000000e+00],
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
         0.00000000e+00, 0.00000000e+00, 1.11763169e-08]])
```

In [197]:
```
# β^WLS =(XTWX)−1XTWy
X=np.array(X)
y=np.array(y)
```

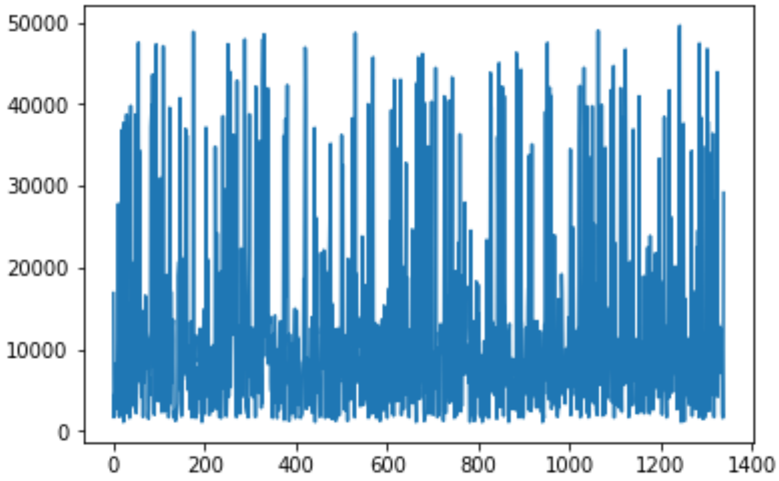
In [200]:
```
(((np.linalg.inv(((X.T).dot(W)).dot(X))).dot(X.T)).dot(W)).dot(y)
```

Out[200]:
```
array([[-10984.74920721],
       [   260.74790992],
       [   249.88974821],
       [   449.63510152],
       [ 24408.59159125]])
```

Therefore Model is: y=-10984.74920721+260.74790992*age+249.88974821*bmi+
449.63510152*children+24408.59159125*is_smoker

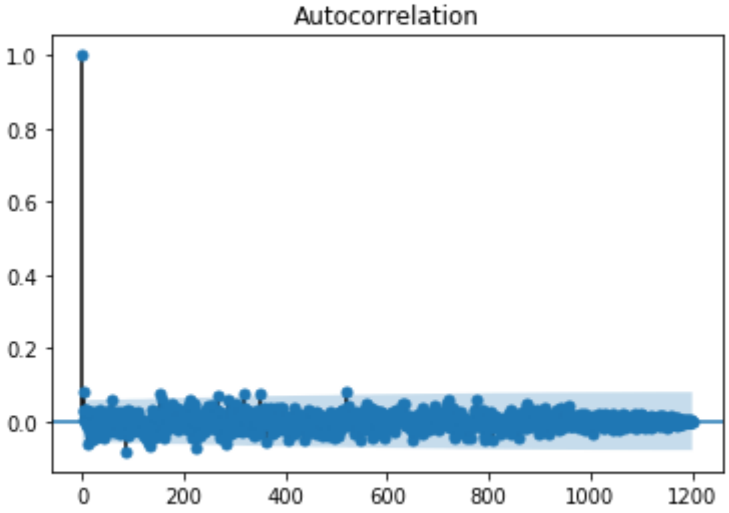
In [88]:

```
#Autocorrelation
charge=health_data_without_influential['charges']
plt.plot(charge)
plt.show()
```



In [94]:

```
#show autocorrelation graphically
sm.graphics.tsa.plot_acf(charge, lags=1200)
plt.show()
```



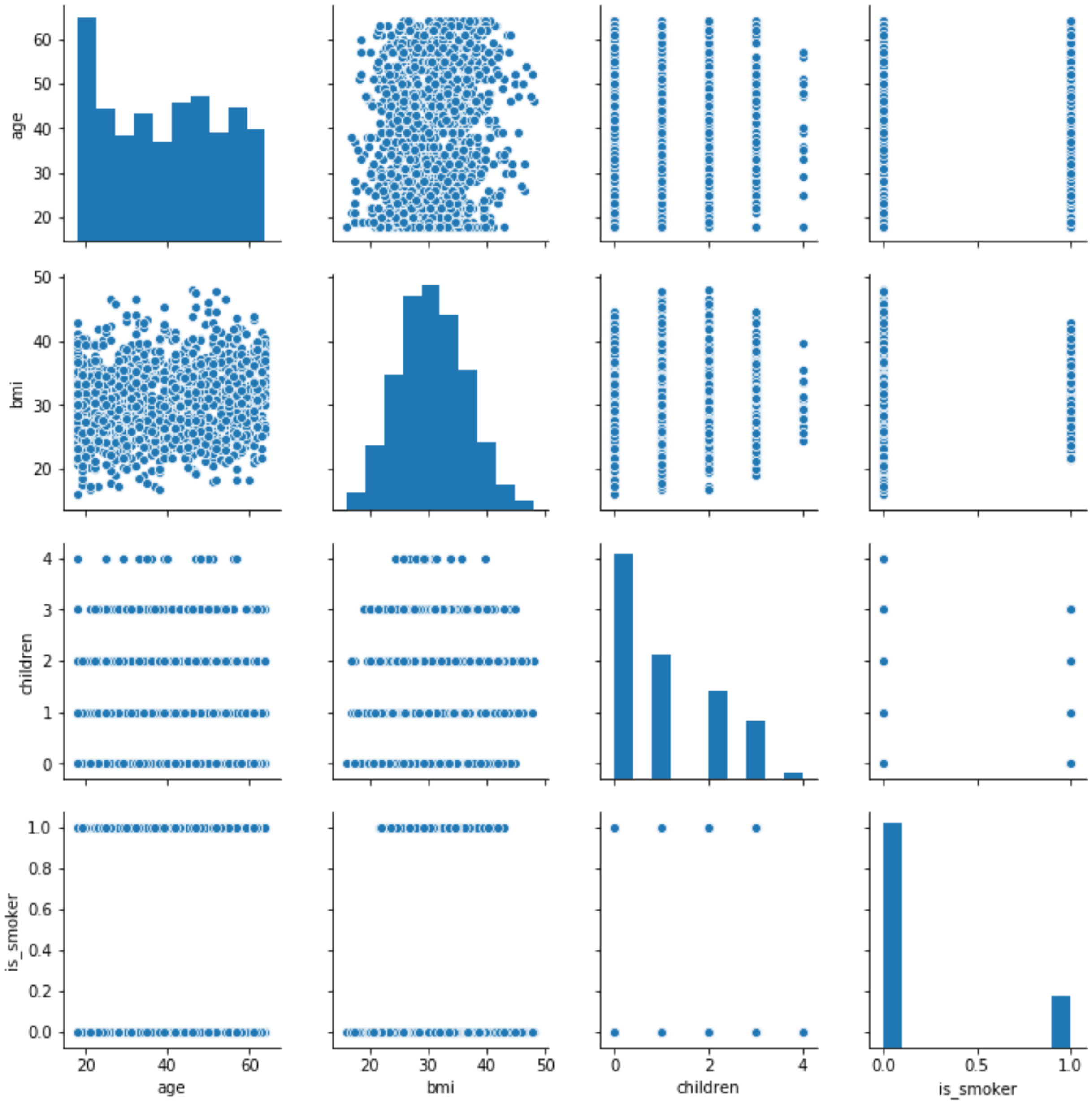
In [95]:
```
bg_test=sm.stats.diagnostic.acorr_breusch_godfrey(reg3)
labels = ['LM Statistic', 'LM-Test p-value', 'F-Statistic', 'F-Test p-value
print(dict(zip(labels, bg_test)))
```

```
{'LM Statistic': 16.051171595369393, 'LM-Test p-value': 0.813338010638982
8, 'F-Statistic': 0.7228629620451152, 'F-Test p-value': 0.819325659425108
3}
```

There p-value is higher than 0.05. Therefore we cannot reject null hypothesis. Thus, we can say from test that there is no serial correlation.Also evident from plot.

In [139]:
```
import seaborn as sns
sns.pairplot(health_data_without_influential[['age','bmi','children','is_sm
```

Out[139]:
```
<seaborn.axisgrid.PairGrid at 0x1c2775c1d0>
```



In [97]:

```
#Multicollinearity
health_data_new=health_data_without_influential[['age','bmi','children','is
cm = health_data_new.corr().round(2)
print(cm)
#doesnt seem to be correlated.
```

```
             age   bmi  children  is_smoker
age         1.00  0.12      0.08      -0.02
bmi         0.12  1.00      0.02       0.00
children    0.08  0.02      1.00       0.00
is_smoker  -0.02  0.00      0.00       1.00
```

In [ ]:

```
from patsy import dmatrices
y, X = dmatrices('charges~age+bmi+children+is_smoker', salesdata, return_ty
```

In [98]:

```
#VIF
from statsmodels.stats.outliers_influence import variance_inflation_factor
vif = pd.DataFrame()
vif["VIF Factor"] = [variance_inflation_factor(health_data_new.values, i) f
vif["features"] = health_data_new.columns
print(vif)
```

```
   VIF Factor   features
0    7.819684        age
1    8.215770        bmi
2    1.819139   children
3    1.219588  is_smoker
```

In [ ]:

```
# Therefore age and BMI may have multicollinearity problem.
# But looking at the graph, the problem is not serious. Therefore can be ig
```

Diagnostic tests and correction for reduced model

```
In [99]: reg2 = smf.ols('charges~age+bmi+is_smoker',data=health_data_dumm).fit()
         reg2.summary()
```

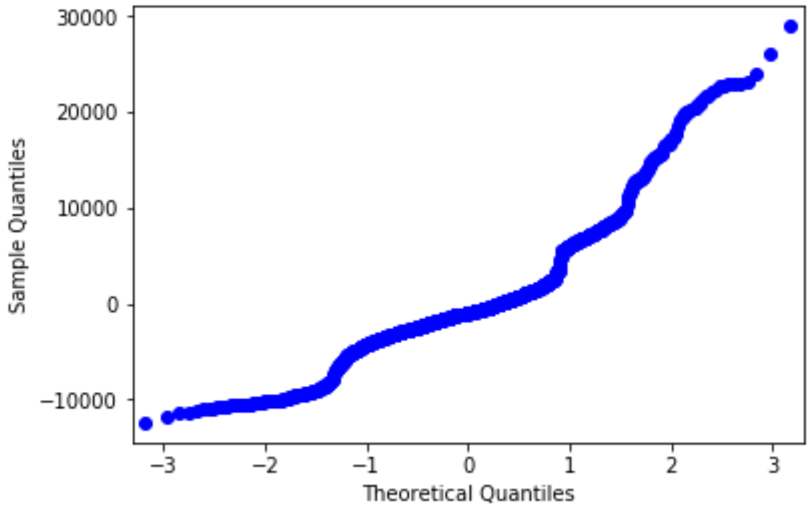
```
Out[99]:
OLS Regression Results

      Dep. Variable:          charges        R-squared:      0.747
              Model:              OLS   Adj. R-squared:      0.747
             Method:    Least Squares      F-statistic:      1316.
               Date: Sun, 13 Oct 2019 Prob (F-statistic):    0.00
               Time:         10:39:37   Log-Likelihood:    -13557.
   No. Observations:             1338              AIC:  2.712e+04
       Df Residuals:             1334              BIC:  2.714e+04
           Df Model:                3
    Covariance Type:        nonrobust

                  coef   std err        t  P>|t|    [0.025     0.975]
 Intercept  -1.168e+04   937.569  -12.454  0.000  -1.35e+04  -9837.561
       age   259.5475    11.934   21.748  0.000    236.136    282.959
       bmi   322.6151    27.487   11.737  0.000    268.692    376.538
 is_smoker  2.382e+04   412.867   57.703  0.000    2.3e+04   2.46e+04

       Omnibus: 299.709    Durbin-Watson:     2.077
Prob(Omnibus):   0.000  Jarque-Bera (JB):   710.137
         Skew:   1.213          Prob(JB): 6.25e-155
     Kurtosis:   5.618         Cond. No.       289.


Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

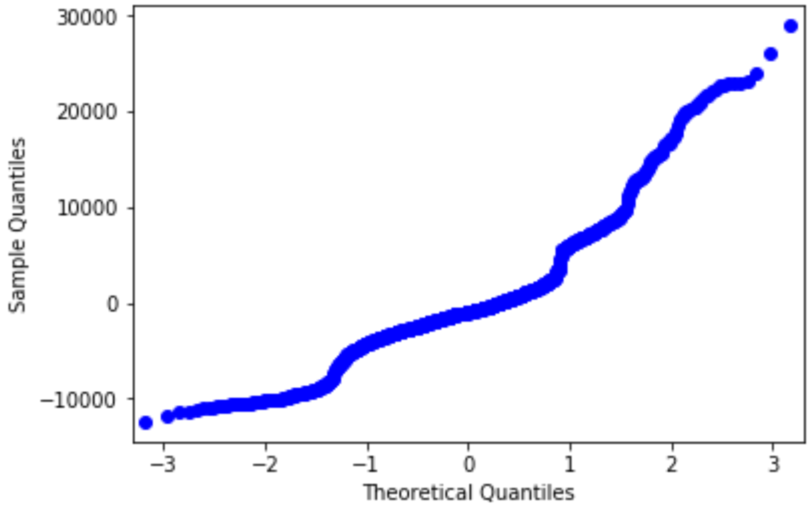
In [100]:

```
#2. check residual
#2.1 Normality
JB, JBpv,skw,kurt = sm.stats.stattools.jarque_bera(reg2.resid)
print(JB,JBpv,skw,kurt)
# p-value is not good
sm.qqplot(reg2.resid)
```

```
710.137418335741 6.246243519708224e-155 1.2130482786943517 5.617622281017
362
```

Out[100]:



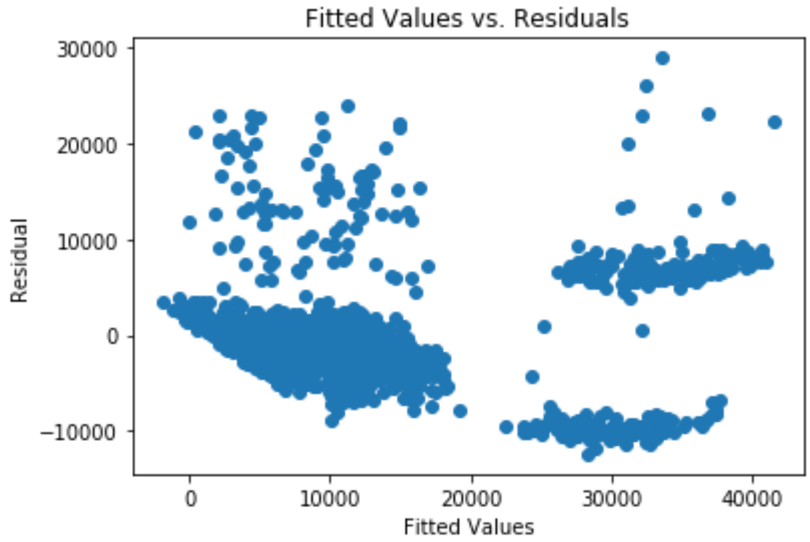


In [101]:

```
#2.2 Fitted Values vs. Residuals
p = reg2.fittedvalues
res = reg2.resid
plt.scatter(p,res)
plt.xlabel("Fitted Values")
plt.ylabel("Residual")
plt.title("Fitted Values vs. Residuals")
#nonlinearity
```

Out[101]:

```
Text(0.5, 1.0, 'Fitted Values vs. Residuals')
```



In [102]:

```
infl2 = reg2.get_influence()
```

In [103]:

```
#leverage
print(infl2.hat_matrix_diag)
```

```
[0.00516551 0.00302163 0.0016012  ... 0.00373312 0.00254361 0.00571561]
```

In [105]:

```
#internally studentized residuals
print(infl2.resid_studentized_internal)
```

```
[-1.51307456 -0.35576963 -0.29361057 ... -0.5350423  -0.01465878
 -1.35250531]
```

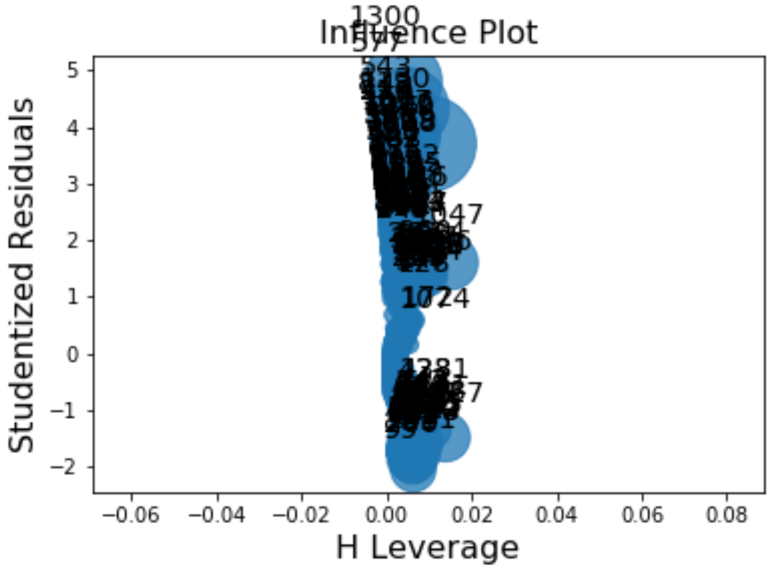
In [106]:

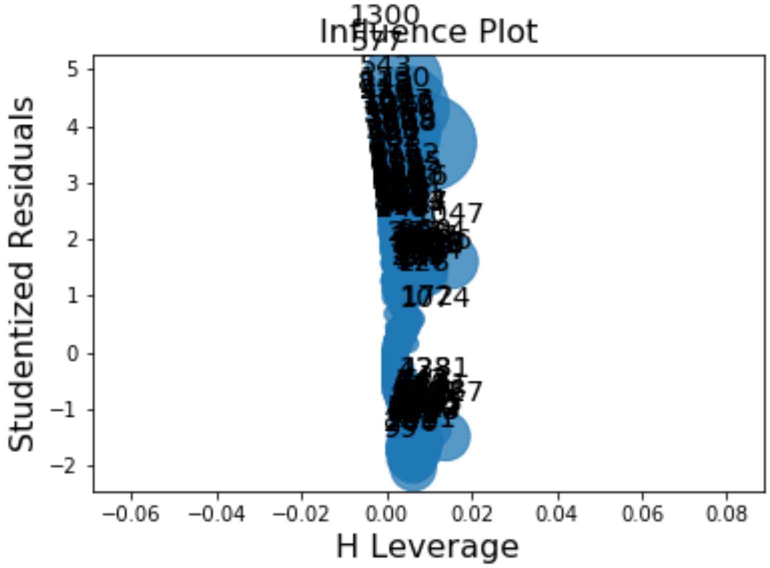
```
#externally studentized residuals
print(infl2.resid_studentized_external)
```

```
[-1.51380688 -0.35565314 -0.29350998 ... -0.53489911 -0.01465329
 -1.35292621]
```

In [107]:
```
#graphical representation of the influences()
sm.graphics.influence_plot(reg2)
```

Out[107]:





In [109]:

```
#too messy. define your rules.
#threshold leverage
residus = reg2.resid.as_matrix() #residuals
leviers = infl2.hat_matrix_diag  #leverage
n = health_data_dumm.shape[0]
p=4
seuil_levier = 2*p/n # people choose 2.5 or 3 as well
print(seuil_levier)
#identification
atyp_levier = leviers > seuil_levier
print(atyp_levier)
```

```
0.005979073243647235
[False False False ... False False False]

/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:3: FutureWar
ning: Method .as_matrix will be removed in a future version. Use .values
instead.
  This is separate from the ipykernel package so we can avoid doing impor
ts until
```

In [110]:

```
#too hard to read
print(health_data_dumm.index[atyp_levier],leviers[atyp_levier])
```

```
Int64Index([  11,   14,   39,   64,   94,   98,   99,  109,  116,  128,
161,
             172,  175,  185,  244,  250,  263,  265,  266,  281,  286,
292,
             301,  328,  330,  362,  377,  380,  401,  411,  412,  419,
420,
             438,  442,  454,  530,  543,  547,  549,  593,  607,  660,
664,
             674,  759,  793,  803,  847,  860,  890,  896,  901,  930,
951,
             989,  994, 1011, 1033, 1047, 1062, 1074, 1085, 1088, 1100, 1
124,
            1131, 1156, 1231, 1240, 1241, 1265, 1282, 1284, 1288, 1317, 1
321],
           dtype='int64') [0.0063642  0.00706826 0.00675891 0.00606671 0.
00611992 0.00754275
 0.00628875 0.00613353 0.00850487 0.0070598  0.006276   0.00648109
 0.00661421 0.00627568 0.00627013 0.00839749 0.00614096 0.00645495
 0.00611174 0.0062523  0.00706097 0.00921601 0.00601672 0.00618415
 0.00599884 0.00642295 0.00653768 0.00718541 0.00670611 0.00610916
 0.00764093 0.00640972 0.0061904  0.00643319 0.00631477 0.00650813
 0.00720811 0.00974793 0.00654859 0.00847778 0.00612414 0.00655212
 0.00616715 0.00773507 0.00716227 0.00669388 0.00668795 0.00842894
 0.01051796 0.00951611 0.00662797 0.00613405 0.00711941 0.00717058
 0.00697911 0.00628516 0.00623567 0.00609655 0.00660366 0.01513248
 0.00706183 0.00614969 0.00679646 0.00707784 0.006388   0.00788621
 0.00662671 0.00973398 0.00626155 0.00656252 0.00662942 0.00746722
 0.00657225 0.00598304 0.00681398 0.01390984 0.00627629]
```

In [111]:

```
#threshold externally studentized residuals
import scipy
seuil_stud = scipy.stats.t.ppf(0.975,df=n-p-1)

#detection - absolute value > threshold
reg_studs=infl2.resid_studentized_external
atyp_stud = np.abs(reg_studs) > seuil_stud
#which ones?
print(health_data_dumm.index[atyp_stud],reg_studs[atyp_stud])
```

```
Int64Index([   3,    9,   34,   62,   99,  102,  115,  138,  140,  143,
219,
             242,  245,  289,  291,  305,  306,  321,  340,  355,  379,
387,
             397,  429,  430,  443,  468,  488,  491,  516,  520,  526,
539,
             543,  573,  577,  583,  587,  599,  637,  658,  688,  696,
739,
             770,  806,  819,  858,  876,  925,  936,  959,  964,  980,
987,
            1008, 1012, 1019, 1027, 1039, 1104, 1123, 1134, 1142, 1146, 1
157,
            1195, 1206, 1211, 1230, 1258, 1300, 1328],
           dtype='int64') [ 2.92861178  2.75082316  3.30920227  2.8473245
9 -2.0467631   3.0724268
  2.82439838  2.4181574   3.79783728  2.09108346  3.80638518  3.95153954
  2.04829857  2.57785601  2.4462419   2.1110727   2.5870492   3.30393266
  2.55171394  2.54064006  2.04671228  3.43157414  2.10695987  2.24832186
  3.15817807  2.11824433  3.27391006  2.14835823  2.01936695  3.76962941
  2.55639917  3.45494418  2.49412033  3.69787647  2.52290213  4.32142422
  2.19959623  2.17807235  3.22338764  2.50554811  2.30788657  2.95631846
  2.74799761  2.21745085  1.99790168  2.70419838  3.79588473  2.14003656
  2.69323719  2.24937384  3.75820336  2.61091377  2.10376094  2.46399629
  3.20047291  3.3600212   3.57791131  3.5822785   3.51314767  3.37300615
  2.11796841  2.15847854  2.16691293  2.8469783   2.3558629   2.08280893
  2.73048722  3.64017089  2.3222563   3.84512904  2.51941218  4.80382808
  3.34711085]
```

In [114]:

```
#DFFITS for detecting influential points

inflsum=infl2.summary_frame()
reg_dffits=inflsum.dffits
seuil_dffits=2*np.sqrt((p+1)/(n-p-1))
atyp_dffits = np.abs(reg_dffits) > seuil_dffits
# print(health_data_dumm.index[atyp_dffits],reg_dffits[atyp_dffits])
influ_DFFITS=health_data_dumm.index[atyp_dffits]
```

In [113]:

```
import scipy
seuil_stud = 4/1338
#detection - absolute value > threshold
cook_studs,pvalue=infl2.cooks_distance
atyp_cook = np.abs(cook_studs) > seuil_stud
#which ones?
print(health_data_dumm.index[atyp_cook],cook_studs[atyp_cook])
```

```
Int64Index([   3,    9,   11,   34,   58,   62,   64,   70,   85,   98,
            ...
            1258, 1265, 1274, 1282, 1288, 1300, 1306, 1317, 1321, 1328],
           dtype='int64', length=129) [0.00488022 0.00610712 0.00344809
0.0132411  0.00406817 0.00864387
 0.00402056 0.00360179 0.0035866  0.00591095 0.00661214 0.00629806
 0.00544821 0.00356631 0.00839059 0.00299935 0.003716   0.00423889
 0.00342835 0.00300864 0.00327494 0.00990315 0.00401993 0.00368387
 0.0088774  0.00526532 0.00310951 0.00574089 0.00368822 0.00335215
 0.00488004 0.0033297  0.00347813 0.00440093 0.00316687 0.00321897
 0.00316238 0.00459028 0.00301787 0.00485195 0.00599806 0.00459281
 0.00521239 0.0035014  0.00683758 0.0032479  0.00303559 0.00563559
 0.00553543 0.00367222 0.00430865 0.01057842 0.00751379 0.00326599
 0.03333532 0.00340006 0.00543154 0.02354643 0.00442619 0.00360199
 0.00606152 0.00396194 0.00308403 0.00345388 0.00309855 0.00441755
 0.00351474 0.00464275 0.00312944 0.00374669 0.00560544 0.0030143
 0.00314435 0.00310529 0.00598529 0.00350381 0.00402282 0.00600414
 0.01536903 0.0047569  0.00386977 0.0053002  0.00416287 0.00362472
 0.00366429 0.00323593 0.00316477 0.00307818 0.00413606 0.00372108
 0.00325302 0.00465854 0.00360046 0.00340309 0.00625975 0.00506091
 0.008801   0.00762718 0.01380662 0.00341493 0.00453258 0.00745494
 0.00984929 0.00327386 0.00375446 0.00388512 0.00444223 0.00436732
 0.00311226 0.00479762 0.00754005 0.00445789 0.0031135  0.00334656
 0.00479427 0.00858699 0.01668255 0.00358104 0.00339601 0.00424356
 0.00460459 0.00337342 0.00406792 0.00320201 0.0217505  0.00416152
 0.00775626 0.00327741 0.00726581]
```

In [119]:

```
pbm_infl1 = np.logical_or(atyp_levier,atyp_stud)
pbm_infl2 = np.logical_or(atyp_cook,atyp_dffits)
pbm_infl = np.logical_or(pbm_infl1,pbm_infl2)
infl_pts=(health_data_dumm.index[pbm_infl])
```

In [120]:

```
health_data_without_influential2=health_data_dumm.drop(infl_pts)
```

In [121]:

```
reg4 = smf.ols('charges~age+bmi+is_smoker',data=health_data_without_influen
reg4.summary()
```

Out[121]:

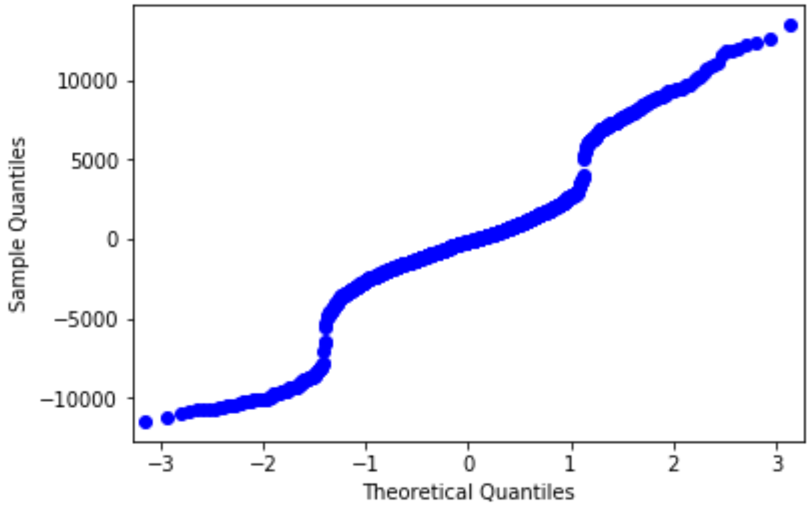
```
OLS Regression Results

       Dep. Variable:           charges           R-squared:      0.853
              Model:               OLS      Adj. R-squared:      0.853
             Method:     Least Squares         F-statistic:      2315.
               Date:  Sun, 13 Oct 2019  Prob (F-statistic):       0.00
               Time:          10:48:17      Log-Likelihood:    -11755.
   No. Observations:              1202                 AIC:  2.352e+04
       Df Residuals:              1198                 BIC:  2.354e+04
           Df Model:                 3
    Covariance Type:         nonrobust

                   coef  std err       t  P>|t|    [0.025     0.975]
  Intercept  -1.066e+04  718.486 -14.843  0.000  -1.21e+04  -9254.505
        age    263.4242    8.845  29.784  0.000    246.072    280.777
        bmi    250.6620   21.494  11.662  0.000    208.493    292.832
  is_smoker   2.442e+04  317.824  76.821  0.000    2.38e+04    2.5e+04

        Omnibus:  28.237      Durbin-Watson:     2.049
  Prob(Omnibus):   0.000   Jarque-Bera (JB):    59.752
           Skew:   0.032           Prob(JB):  1.06e-13
       Kurtosis:   4.090           Cond. No.      299.


Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

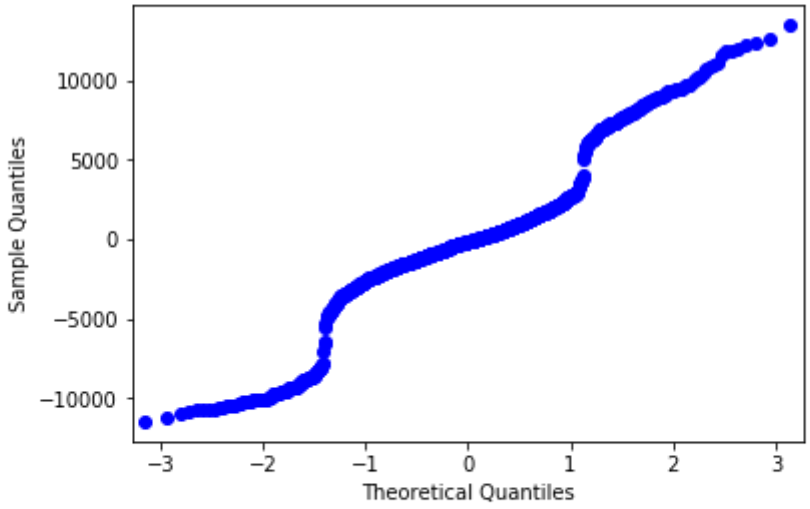
In [122]:

```
#2. check residual
#2.1 Normality
JB, JBpv,skw,kurt = sm.stats.stattools.jarque_bera(reg4.resid)
print(JB,JBpv,skw,kurt)
# p-value is not good
sm.qqplot(reg4.resid)
```

```
59.75244561532769 1.0590617781850974e-13 0.0323503356784574686 4.090355126
232836
```

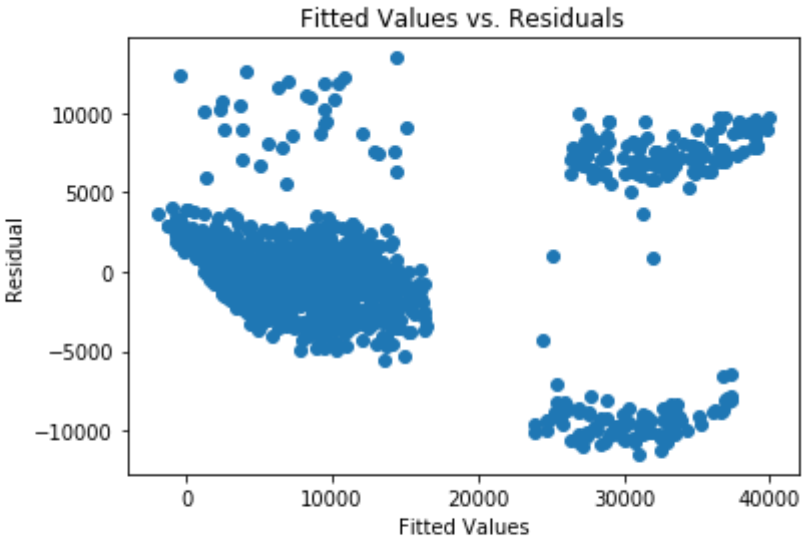
Out[122]:





```
In [123]: #2.2 Fitted Values vs. Residuals
          p = reg4.fittedvalues
          res = reg4.resid
          plt.scatter(p,res)
          plt.xlabel("Fitted Values")
          plt.ylabel("Residual")
          plt.title("Fitted Values vs. Residuals")
          #nonlinearity
```

```
Out[123]: Text(0.5, 1.0, 'Fitted Values vs. Residuals')
```



```
In [124]: #Breusch-Pagan for Heteroskedasticity
          from statsmodels.stats.diagnostic import het_breuschpagan
          bp_test = het_breuschpagan(reg4.resid, reg4.model.exog)
          labels = ['LM Statistic', 'LM-Test p-value', 'F-Statistic', 'F-Test p-value
          print(dict(zip(labels, bp_test)))
```

```
{'LM Statistic': 737.0934816226819, 'LM-Test p-value': 1.8987749487608997
e-159, 'F-Statistic': 633.1294257220253, 'F-Test p-value': 1.677709707214
861e-246}
```

```
In [125]: #Breusch-Pagan for Heteroskedasticity
          from statsmodels.stats.diagnostic import het_breuschpagan
          bp_test = het_breuschpagan(reg2.resid, reg2.model.exog)
          labels = ['LM Statistic', 'LM-Test p-value', 'F-Statistic', 'F-Test p-value
          print(dict(zip(labels, bp_test)))
```

```
{'LM Statistic': 112.73640444098342, 'LM-Test p-value': 2.827322043783107
4e-24, 'F-Statistic': 40.91374407633964, 'F-Test p-value': 2.715975903497
3015e-25}
```

```
In [126]: # therefore heteroskedascity exists
          # use weighted least squares
```

```
In [202]: from patsy import dmatrices
          y, X = dmatrices('charges~age+bmi+is_smoker', health_data_without_influenti
```

```
In [208]: X=np.array(X)
          y=np.array(y)
          y.shape

Out[208]: (1157, 1)
```

```
In [206]: XTXI = np.linalg.inv((X.T).dot(X))
          H=X.dot(XTXI.dot(X.T))
          H.shape

Out[206]: (1157, 1157)
```

```
In [212]: I=np.identity(1157)
```

```
In [215]: l=(y.T).dot(I-H)
```

```
In [217]: SSE=l.dot(y)
          SSE

Out[217]: array([[1.63357056e+10]])
```

```
In [220]: MSE=SSE/(1157/4)
          MSE=56476078.25983766
```

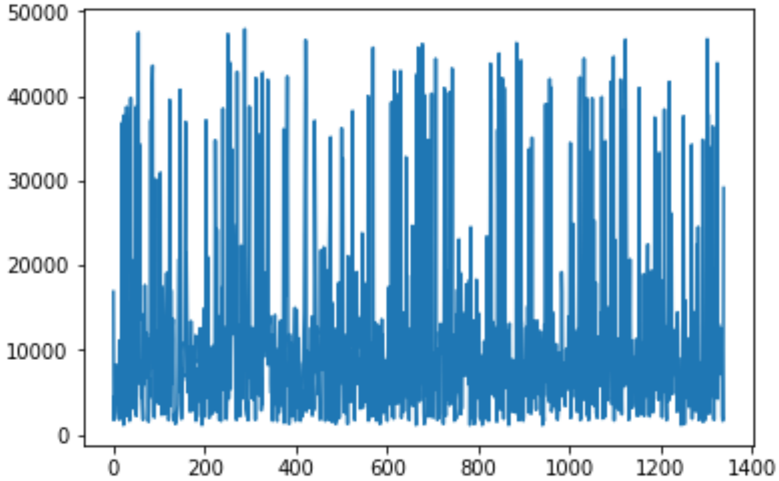
```
In [222]: mat=MSE*(I-H)
          mat=1/mat
          W = np.diag(np.diag(mat))
          W

Out[222]: array([[1.78416846e-08, 0.00000000e+00, 0.00000000e+00, ...,
                   0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
                  [0.00000000e+00, 1.77707166e-08, 0.00000000e+00, ...,
                   0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
                  [0.00000000e+00, 0.00000000e+00, 1.77396514e-08, ...,
                   0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
                  ...,
                  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
                   1.77889217e-08, 0.00000000e+00, 0.00000000e+00],
                  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
                   0.00000000e+00, 1.77589907e-08, 0.00000000e+00],
                  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
                   0.00000000e+00, 0.00000000e+00, 1.78615914e-08]])
```

```
In [223]: (((np.linalg.inv(((X.T).dot(W)).dot(X))).dot(X.T)).dot(W)).dot(y)
```

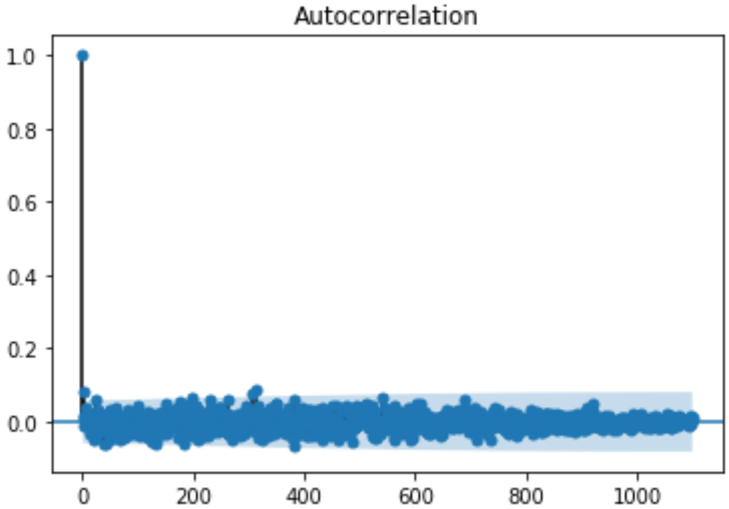
```
Out[223]: array([[-8150.40862706],
                 [  260.8424167 ],
                 [  173.22681875],
                 [25693.62330826]])
```

Therefore model is: y=-8150.40862706 + 260.8424167*age + 173.22681875*bmi + 25693.62330826*is_smoker

```
In [127]: charge=health_data_without_influential2['charges']
          plt.plot(charge)
          plt.show()
```



```
In [129]: #show autocorrelation graphically
          sm.graphics.tsa.plot_acf(charge, lags=1100)
          plt.show()
```



In [130]:

```
bg_test=sm.stats.diagnostic.acorr_breusch_godfrey(reg4)
labels = ['LM Statistic', 'LM-Test p-value', 'F-Statistic', 'F-Test p-value
print(dict(zip(labels, bg_test)))
```

```
{'LM Statistic': 16.67536754131242, 'LM-Test p-value': 0.781014853236178
1, 'F-Statistic': 0.752008494381279, 'F-Test p-value': 0.786721480158065
9}
```

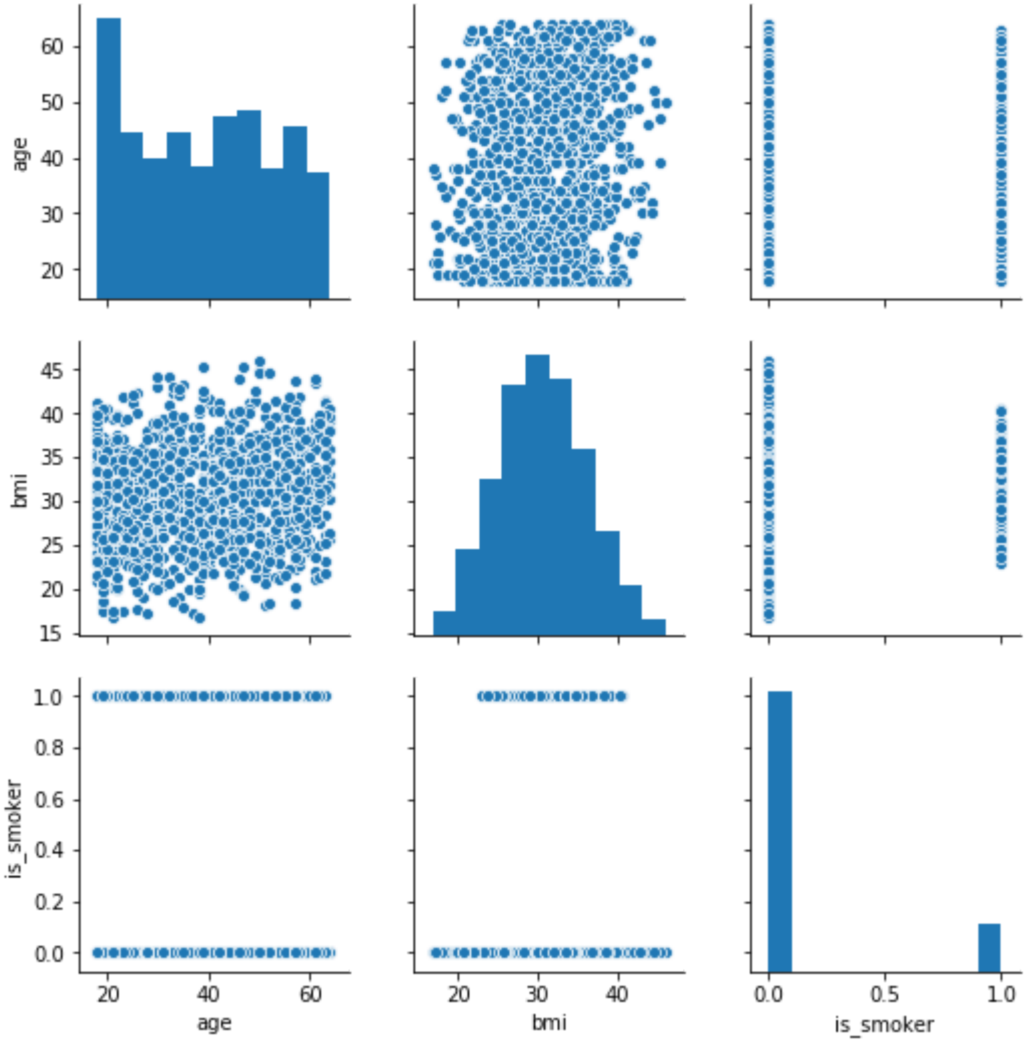
There p-value is higher than 0.05. Therefore we cannot reject null hypothesis. Thus, we can say from test that there is no serial correlation.Also evident from plot.

In [132]:

```
import seaborn as sns
sns.pairplot(health_data_without_influential2[['age','bmi','is_smoker']])
```

Out[132]:

```
<seaborn.axisgrid.PairGrid at 0x1c24e70e10>
```



In [133]:

```
#Multicollinearity
health_data_new2=health_data_without_influential2[['age','bmi','is_smoker']
cm = health_data_new2.corr().round(2)
print(cm)
#doesnt seem to be correlated.
```

```
             age   bmi  is_smoker
age         1.00  0.12      -0.05
bmi         0.12  1.00       0.05
is_smoker  -0.05  0.05       1.00
```

In [135]:

```
from patsy import dmatrices
y, X = dmatrices('charges~age+bmi+is_smoker', health_data_without_influenti
```

In [137]:

```
#VIF
from statsmodels.stats.outliers_influence import variance_inflation_factor
vif = pd.DataFrame()
vif["VIF Factor"] = [variance_inflation_factor(health_data_new2.values, i)
vif["features"] = health_data_new2.columns
print(vif)
```

```
   VIF Factor   features
0    7.919438        age
1    8.187069        bmi
2    1.177900  is_smoker
```

In [138]:

```
# Therefore age and BMI may have multicollinearity problem.
# But looking at the graph, the problem is not serious. Therefore can be ig
```

7. Therefore, the final model is:

y=-8150.40862706 + 260.8424167age + 173.22681875bmi + 25693.623308
26*is_smoker

In [ ]:

```
```