Python for Data Science

Linear Regression

What's in this section:

Introduction to Multiple Linear Regression

Linear regression models are used to analyze the relationship between an independent variable (IV) or variables and a dependent variable (DV), a.k.a the predicted variable. If only one predictor variable (IV) is used in the model, then that is called a single linear regression model. However, the model will not be robust in design and will have little to no explanation power because in the real world there is no 1 variable that can fully explain, or predict, an outcome. Most commonly, the model will have multiple IVs which will make it a multiple linear regression model.

Assumptions for multiple linear regression:

- The DV has to be measured on a continuous level
- Linear relationship between the DV and each of the IV
  - There needs to be a theoretical reason as to why it would be expected that as one increases the other increase or decreases

- Data must not have multicollinearity
  - This is a large concern when conducting linear regressions. One way to check for multicollinearity is to run a correlation matrix on the data or to check the variance inflation factors (VIFs). If there are strong correlations between the IVs, or a high VIF, then one could drop one of the variables or conduct a Shapley's regression or Ridge regression which takes into account the highly correlated variables

- The residual errors should be approximately normally distributed
  - One can test this by using the Kolmogorov-Smirnov test, Jarque–Bera test, or the Shapiro-Wilk test for normality on the models residuals and/or by looking at a Q-Q plot of the model's residuals

    - One thing to keep in mind is that as the sample size increases, the likelihood of violating this assumption increases

- Homoscedasticity should be present
  - We can check this by plotting the data as a scatter plot and checking if there is a cone shape, or by using the Levene's test for homogeneity of variance, Breusch-Pagan test, and/or the NCV test

- Independence of errors
  - This can be tested using the Durbin-Watson test

Often times when using real world data, there will be violations. If the residuals violate the assumptions of homoscedasticity and/or normality, then you can try transforming the data or using a robust regression model (discussed below). If no transformations or corrections are made to the data/model, then one will have difficulties generalizing the model to the population, i.e. the findings should be limited to the sample used.

Data used in this section

Data in this section is from Kaggle.com from the used Miri Choi. Link to the original data set can be found here (https://www.kaggle.com/mirichoi0218/insurance). This data set contains information on insurance premium charges, age, sex, BMI, number of children, smoker status, and region. The question being asked is, what are the most influential predictors of insurance premium charges? This makes the DV be the insurance premium charge and the IVs are the other variables in the data.

Let's import Pandas as pd, load the data set, and take a look at the variables!

```
import pandas as pd

df= pd.read_csv("insurance.csv")

df.describe()
```

	age	**bmi**	**children**	**charges**
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

As you can see, by default Python ignored the categorical variables. In order to see these variables, one will have to call the method on these variables separately. Python did this because the data set contained a mix of continuous and and categorical variables and the information provided by the *.describe()* method is different for continuous and categorical data.

```
df[['sex', 'smoker']].describe()
```

	sex	smoker
count	1338	1338
unique	2	2
top	male	no
freq	676	1064

The "freq" row is the count of the most commonly occurred variable, i.e. "male" occurred 676 times and "no" occurred 1,064 times in the data set.

Multiple Linear Regression Example

Before a multiple linear regression analysis can be conducted, one needs to import the required libraries, recode the categorical data to binary form, and check the assumptions!

```
import statsmodels.formula.api as smf
import statsmodels.stats.api as sms
from scipy import stats
from statsmodels.compat import lzip
import statsmodels
import matplotlib.pyplot as plt
```

Now to recode the categorical data to binary form that way it can be included in the model. This means to code one category as 1, and the other as 0. If there are multiple categories in the variable, then one needs to create a dummy variable for each category. Always code the category outcome of interest as 1.

This is easy work using the "pd.get_dummies()" method. It automatically creates dummy variables and recodes them to binary form of 1 and 0. A 1 indicates the membership in that category and a 0 indicates non-membership.

```
df= pd.get_dummies(df)

# Dropping the dummy variable that are not needed
df.drop(['sex_male' , 'smoker_no'], axis=1 ,inplace= True)
df.describe()
```

	age	bmi	children	charges	sex_female	smoker_yes
count	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265	0.494768	0.204783

std	14.049960	6.098187	1.205493	12110.011237	0.500160	0.403694
min	18.000000	15.960000	0.000000	1121.873900	0.000000	0.000000
25%	27.000000	26.296250	0.000000	4740.287150	0.000000	0.000000
50%	39.000000	30.400000	1.000000	9382.033000	0.000000	0.000000
75%	51.000000	34.693750	2.000000	16639.912515	1.000000	0.000000
max	64.000000	53.130000	5.000000	63770.428010	1.000000	1.000000

Assumption of Multicollinearity

Now let's check for multicollinearity. When checking, it doesn't really matter if the correlations are significant or not since that is not the purpose of running a correlation analysis in this context.

```
df.corr()
```

	age	bmi	children	charges	sex_female	smoker_yes
age	1.000000	0.109272	0.042469	0.299008	0.020856	-0.025019
bmi	0.109272	1.000000	0.012759	0.198341	-0.046371	0.003750
children	0.042469	0.012759	1.000000	0.067998	-0.017163	0.007673
charges	0.299008	0.198341	0.067998	1.000000	-0.057292	0.787251
sex_female	0.020856	-0.046371	-0.017163	-0.057292	1.000000	-0.076185
smoker_yes	-0.025019	0.003750	0.007673	0.787251	-0.076185	1.000000
region_northeast	0.002475	-0.138156	-0.022808	0.006349	0.002425	0.002811
region_northwest	-0.000407	-0.135996	0.024806	-0.039905	0.011156	-0.036945
region_southeast	-0.011642	0.270025	-0.023066	0.073982	-0.017117	0.068498
region_southwest	0.010016	-0.006205	0.021914	-0.043210	0.004184	-0.036945

There are no strong correlations between the IVs meaning there is no need to worry about multicollinearity.

Assumption of Independent Errors

To test the assumption that the errors are independent, one can use the Durbin-Watson test; this is the method *statsmodels.stats.stattools.durbin_watson()*. For this test, a value of 2, or close to it, is ideal. The statistical value ranges between 0-4 where a value closer to 0 is more evidence for positive serial correlation and a value closer to 4 is more evidence for negative serial correlation.

In order to run this test, you have to create the model within Python before hand. Skipping that portion here, it is in the respective section on the page.

```
statsmodels.stats.stattools.durbin_watson(model.resid)
2.0884229986673088
```

Given the statistical value of 2.09, the test provides evidence that there is no serial correlation present meaning the residual error terms are uncorrelated and are independent.

Assumption of Normality of the Residuals

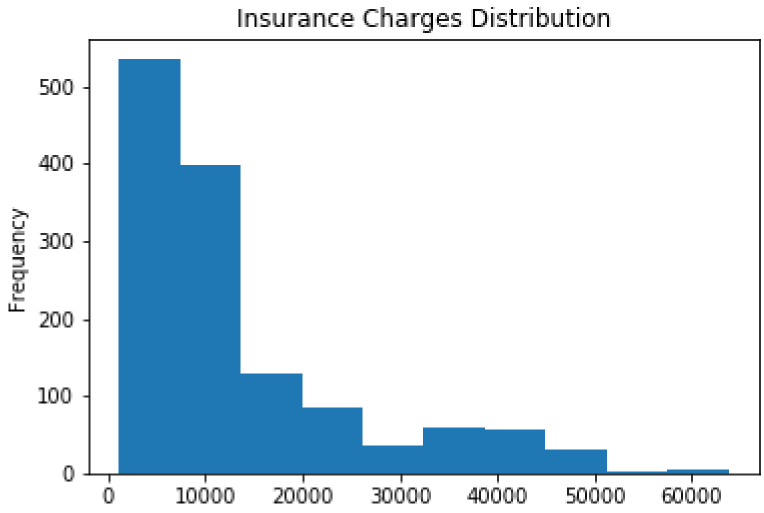
It is often miss-understood about what normality is being tested. For linear regression, it is the residuals- not the variables themselves. This can be tested with the Kolmogorov-Smirnov test, Jarque–Bera test, or the Shapiro-Wilk test for normality on the model's residuals. For this example, the Jarque-Bera test will be used which is the *sms.jarque_bera()* method.

Again, this test is to be conducted on the model's residuals. The model is developed in it's appropriate section (below) and is referenced here.

```
name = ['Jarque-Bera', 'Chi^2 two-tail prob.', 'Skew', 'Kurtosis']
test = sms.jarque_bera(model.resid)
lzip(name, test)
[('Jarque-Bera', 718.8872635707909),
('Chi^2 two-tail prob.', 7.8634686618215363e-157),
('Skew', 1.211211005167187),
('Kurtosis', 5.650794298374027)]
```

The test is significant; meaning the data violates the assumption of normality of the residuals. Given that we have a large sample in this data (N= 1,338), that is likely to occur. As the sample size increases, the more likely the data will violate this assumption. There are a couple more steps to take in order to decide what action is required. They are, explore the distribution shape of the DV and the IVs, and their errors, to see if the distribution fits a better model and/or to transform the data in order to make it normal and the residuals normal.

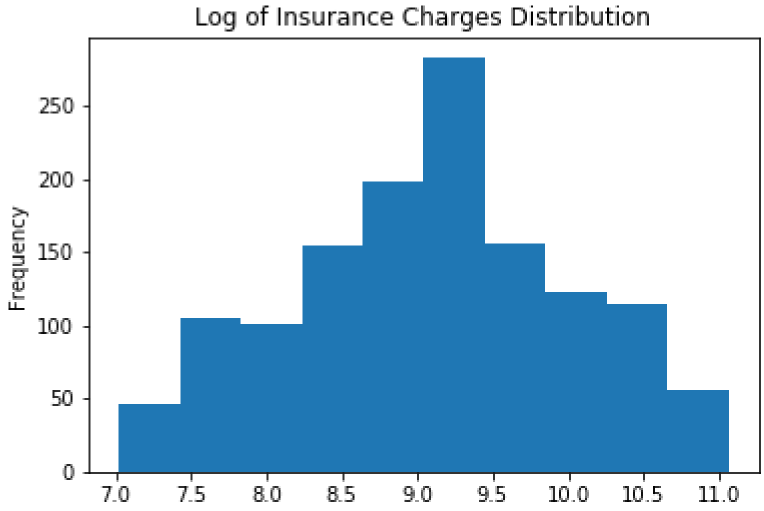
Let's start with looking at the distribution of the DV (charges). Before looking at the graph, think about the DV in the model. How is it expected to be distributed? In case of insurance charges billed to the individual, it would be reasonable to expect that most of the billing cost is low. Which would drive the distribution to the left (positive skew). Which is exactly what occurs in the data. Let's take a look!



The data follows more of a skewed normal distribution. A common way to correct this would be to take the log transformation of the DV and use it in the model. The caveat to this is that interpretation of the results can get tricky. Another regression model, such as quantile regression or a nonparametric linear regression, may also be useful given the distribution.

Let's take a look at how taking the log transformation of the DV changes the distribution of the variable. Keep in mind, we truly need to test the distribution of the residuals. This is to provide a visual of what occurs.

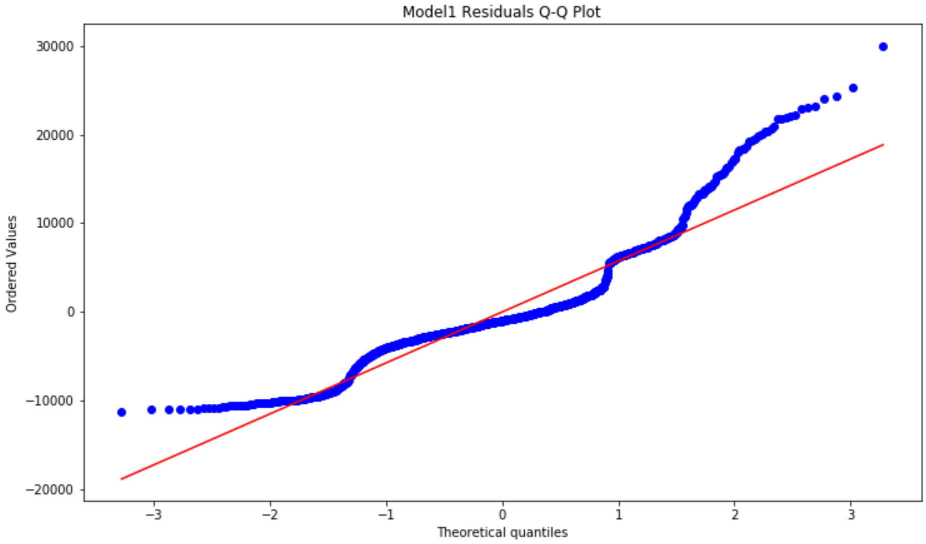
```
df['charges_log'] = np.log(df['charges'])
df['charges_log'].plot(kind='hist',
                       title= 'Log of Insurance Charges Distribution')
```



You can see how it transforms the data into more of a normal distribution which would likely change the residuals as well. One should test this for each variable's residuals to find the culprit and decide what to do. The best way to graphically check for normality is to use a Q-Q plot. A Q-Q plot of the model's residuals is below. The data points (blue) should be on the red line. If not, then it indicates non-normality.

```
#Running plot & giving it a title
stats.probplot(model.resid, dist="norm", plot= plt)
plt.title("Model1 Residuals Q-Q Plot")

#Saving plot as a png
plt.savefig("Model1_Resid_qqplot.png")
```



Given the sample size of the data and the Central Limit Theorem, we are protected from violating this assumption and our findings will still be valid. For a brief touch on this if interested, see [here](http://www2.psychology.uiowa.edu/faculty/mordkoff/GradStats/part%201/I.07%20normal.pdf) (http://www2.psychology.uiowa.edu/faculty/mordkoff/GradStats/part%201/I.07%20normal.pdf). Although the results are protected, depending on the field of the study, you may have to make adjustments to keep with your field's standard of practice.

Assumption of Homoscedasticity

The assumption of homoscedasticity is a vital assumption for linear regression. If this assumption is violated, then the standard errors will be biased. The standard errors are used to conduct significance tests, and calculate the confidence intervals.

This can be tested using a residual vs. fitted values plot, looking at a scatter plot (if a cone shape is present then heteroscedasticity is present), or by using a statistical test such as Bruesch-Pagan, Cook-Weisberg test, or White general test. In this example, the Bruesch-Pagan test will be used.

```
name = ['Lagrange multiplier statistic', 'p-value',
        'f-value', 'f p-value']
test = sms.het_breuschpagan(model.resid, model.model.exog)
lzip(name, test)
```

```
[('Lagrange multiplier statistic', 121.74360137568986),
('p-value', 5.8721533542513195e-22),
('f-value', 16.628612027375389),
('f p-value', 1.1456058246340301e-23)]
```

The test is significant meaning the data violates the assumption of homoscedasticity, i.e. heteroscedasticity is present in the data. What to do? Either one can transform the variables to improve the model, or use a robust regression method that accounts for the heteroscedasticity.

In order to account for the heteroscedasticity in the data, one has to select a heteroscedasticity consistent covariance matrix (HCCM) and pass it in the “cov_type=” argument apart of the *.fit()* method. What is HCCM? Here (http://www.indiana.edu/~jslsoc/files_research/testing_tests/hccm/00TAS.pdf) is a nice read if interested more on this. There are a few HCCMs to choose from:

- HC0, not good on sample size ≤ 250
- HC1, not good on sample size ≤ 250
- HC2, good on sample size ≤ 250
- HC3, which out performs HC0, HC1, and HC2 when sample size ≤ 250
- Little difference in performance when sample is ≥ 500

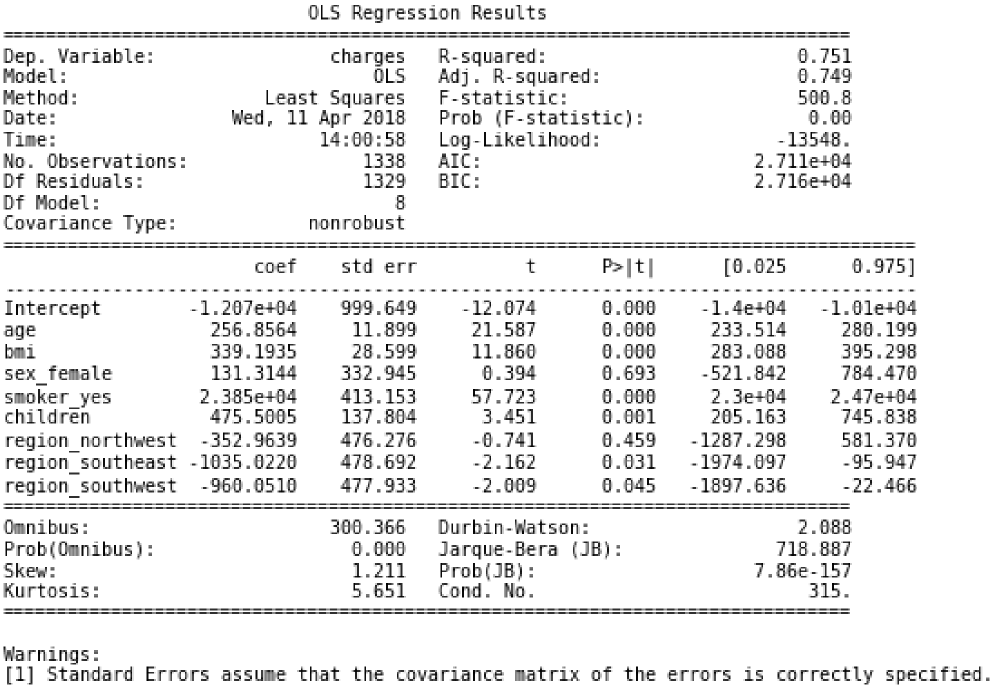
For the current model, using a robust regression technique will work. This will be demonstrated in the running of the model.

Linear Regression Model

Now that all the assumptions have been checked, time to run the model. The first model to be ran will be the original model used to test the assumptions. Note that when using dummy variables, for each category of the orignal variable, one needs to drop 1 dummy variable, i.e. (k-1).

```
model = smf.ols("charges ~ age + bmi + sex_female + smoker_yes + children + re

model.summary()
```



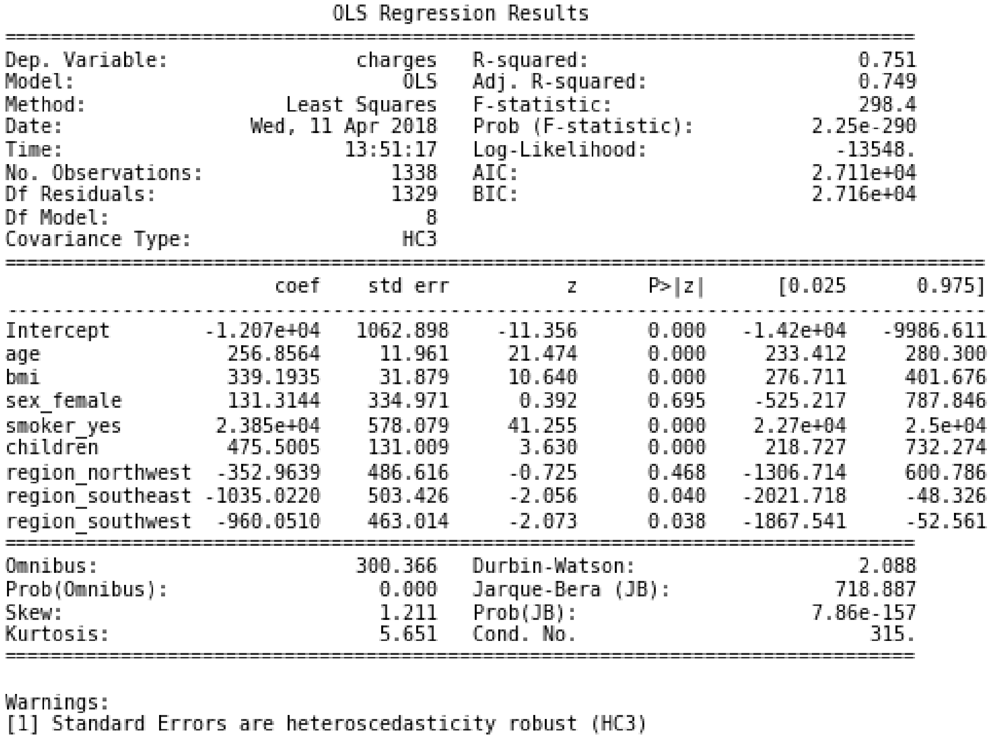
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                charges   R-squared:                       0.751
Model:                            OLS   Adj. R-squared:                  0.749
Method:                 Least Squares   F-statistic:                     500.8
Date:                Wed, 11 Apr 2018   Prob (F-statistic):               0.00
Time:                        14:00:58   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        -1.207e+04    999.649    -12.074      0.000    -1.4e+04   -1.01e+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
sex_female         131.3144    332.945      0.394      0.693    -521.842     784.470
smoker_yes        2.385e+04    413.153     57.723      0.000      2.3e+04    2.47e+04
children           475.5005    137.804      3.451      0.001     205.163     745.838
region_northwest  -352.9639    476.276     -0.741      0.459   -1287.298     581.370
region_southeast -1035.0220    478.692     -2.162      0.031   -1974.097     -95.947
region_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.                         315.
==============================================================================

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

Remember, the model above is not taking into account the presence of heteroscedasticity. This is noted in “Covariance Type: nonrobust”. Let’s see the results of the model that accounts for this. In order to do this, pass the desried HCCM into the “cov_type=” argument within the *.fit()* method.

```
model3 = smf.ols("charges ~ age + bmi + sex_female + smoker_yes + children + r

model3.summary()
```



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                charges   R-squared:                       0.751
Model:                            OLS   Adj. R-squared:                  0.749
Method:                 Least Squares   F-statistic:                     298.4
Date:                Wed, 11 Apr 2018   Prob (F-statistic):         2.25e-290
Time:                        13:51:17   Log-Likelihood:                -13548.
No. Observations:                1338   AIC:                         2.711e+04
Df Residuals:                    1329   BIC:                         2.716e+04
Df Model:                           8
Covariance Type:                  HC3
=======================================================================================
                       coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------
Intercept        -1.207e+04   1062.898    -11.356      0.000   -1.42e+04   -9986.611
age                256.8564     11.961     21.474      0.000     233.412     280.300
bmi                339.1935     31.879     10.640      0.000     276.711     401.676
sex_female         131.3144    334.971      0.392      0.695    -525.217     787.846
smoker_yes        2.385e+04    578.079     41.255      0.000    2.27e+04      2.5e+04
children           475.5005    131.009      3.630      0.000     218.727     732.274
region_northwest  -352.9639    486.616     -0.725      0.468   -1306.714     600.786
region_southeast -1035.0220    503.426     -2.056      0.040   -2021.718     -48.326
region_southwest  -960.0510    463.014     -2.073      0.038   -1867.541     -52.561
==============================================================================
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.                         315.
==============================================================================

Warnings:
[1] Standard Errors are heteroscedasticity robust (HC3)
```

There are differences in the models and their results. Accounting for the heteroscedasiticity has altered the F-statistics, the standard errors, and etc. The second model ran, model3, is a better model.

As you can see, the statistical tests used to test the assumptions are provided at the bottom of the model’s summary. You don’t have to test for the assumptions of independent errors (Durban-Watson), normality of residuals (Jarque-Bera), or the presence of homoscedasticity (Omnibus/Breusch-Pagan) before running the model because they are provided with the model. I wanted you to see them beforehand with a bit more explanation of each assumption. One does however, have to check for the presence of multicollinearity before running the model.

Linear Regression Interpretation

First one has to look to see if the model is significant. This information is found in “Prov (F-statistic)= “, the current model is significant. Now one can look at the affects of each IV on the DV (the IV coefficients) and see if it is a significant predictor or not (P>|z|). As usual, one wants a p-value < 0.05. All the IVs are significant predictors of insurance premium charges except for sex_female. Indicating that sex doesn't matter in this context.

The coefficient (coef) can be interpreted as the affect in unit change in terms of the DV. Meaning, for every 1 unit increase in the IV, the DV will increase or decrease by the coefficient amount. In the example above, for every year increase in age, there will be a $256.86 increase in the insurance premium charge. Let's review the linear regression equation to see how this data makes as a predictive/explanatory model. It is predictive when used to predict future outcomes, and explanatory when used to explain the influence of each IV.

$$y = b_0 + m_1X_1 + m_2X_2 + m_nX_n$$

Where:

- **b_0** is the intercept of the model, and
- **m_nX_n** is the coefficient (m) to the respective variable (X)

Using the current model, one can write the formula as:

```
Charges = -12,070 + 256.86(age) + 339.19(bmi) + 131.31(sex_female) +
23,850(smoker_yes) + 474.50(number of children) - 352.96(region_northwest) -
1,035.02(region_southeast) - 960.05(region_soughtwest)
```

Now to write up the model’s performance.

Model write up

Multiple regression analysis was used to test if age, BMI, sex (female), smoking status, number of children, and region significantly predicted the cost of insurance premiums. The results of the regression indicated the nine predictors explained 75.1% of the variance (R^2= 0.75, F(8,1329)= 298.4, p< 0.01). The predicted insurance premium charge is equal to -12,070 + 256.86(age) + 339.19(bmi) + 131.31(sex_female) + 23,850(smoker_yes) + 474.50(number of children) – 352.96(region_northwest) – 1,035.02(region_southeast) – 960.05(region_soughtwest), where sex_female is coded as 1= female, 0= male; smoker_yes is coded as 1= smoker, 0= non-smoker; region_[location] is coded as 1= in region, 0= not in region. All of the independent variables used in the model were significant predictors of insurance premium charge, except for sex