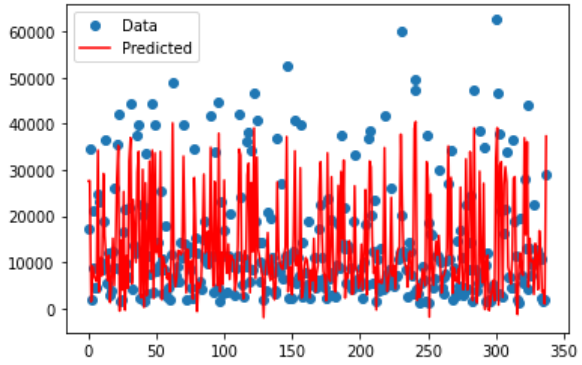
Smf.ols Linear Regression

Data: **insurance**

1. Keep only selected colums
2. Split into train and test
3. Model
4. P-values (slope)
5. Many models
6. Two parts of test data
7. Apply model on test data
8. Residuals
9. RMSE
10. Plot



Needed libraries

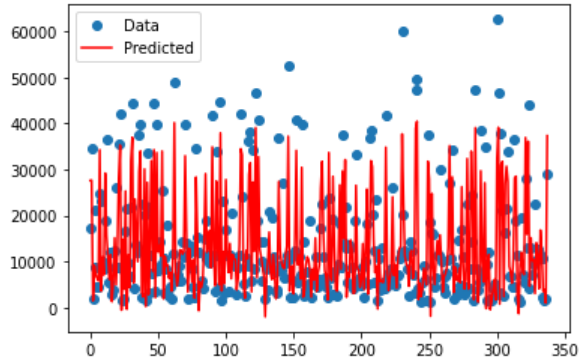
```
#Jesus is my Saviour!
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
import statsmodels.api as sm
from statsmodels.formula.api import ols
from statsmodels.stats.multicomp import pairwise_tukeyhsd
import statsmodels.formula.api as smf
pd.set_option('display.max_column',None)
from sklearn.linear_model import LinearRegression
#from sklearn.preprocessing import LabelEncoder
#le = LabelEncoder()
#from sklearn.preprocessing import StandardScaler
#sc = StandardScaler()
#from sklearn.linear_model import SGDRegressor
#sgdr = SGDRegressor()

data = pd.read_csv("C:/Users/Dr Vinod/Desktop/DataSets1/insurance.csv")
data.info()
```

Keep only potentially useful predictors with Target Variable

```
data.info()
'''
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 8 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   id        1338 non-null   int64
 1   age       1338 non-null   int64
 2   sex       1338 non-null   object
 3   bmi       1338 non-null   float64
 4   children  1338 non-null   int64
 5   smoker    1338 non-null   object
 6   region    1338 non-null   object
 7   charges   1338 non-null   float64
dtypes: float64(2), int64(3), object(3)
memory usage: 83.8+ KB
'''
#__1     remove 'id' as it is not going to be in our model
# YOU NEED TO KEEP ONLY SELECTED VARIABLES INCLUDING TARGET VARIABLE

del data['id']
data.info() # now, id will not be there!
```

Train and Test

```
#__2 now, split data into train and test in approx 70:30 ratio

trn = data.iloc[0:1000, ]
tst = data.iloc[1000:1338,]
```



```
#___3 now, build model on training set (trn) only

model_1=smf.ols(formula='charges ~ age + sex + bmi + children + smoker + region',data= trn).fit()
print(model_1.summary())
'''
print(model_1.summary())
                            OLS Regression Results
==============================================================================
Dep. Variable:                charges   R-squared:                       0.757
Model:                            OLS   Adj. R-squared:                  0.755
Method:                 Least Squares   F-statistic:                     385.7
Date:                Sun, 26 Dec 2021   Prob (F-statistic):          4.13e-298
Time:                        08:12:47   Log-Likelihood:                -10103.
No. Observations:                1000   AIC:                         2.022e+04
Df Residuals:                     991   BIC:                         2.027e+04
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      -1.208e+04   1135.373    -10.643      0.000   -1.43e+04   -9855.249
sex[T.male]     -288.5339    377.406     -0.765      0.445    -1029.140     452.073
smoker[T.yes]   2.383e+04    475.703     50.099      0.000    2.29e+04    2.48e+04
region[T.northwest]  -439.8965    543.632     -0.809      0.419   -1506.699     626.906
region[T.southeast] -1291.2899    534.513     -2.416      0.016   -2340.198    -242.382
region[T.southwest] -1263.1479    537.296     -2.351      0.019   -2317.516    -208.779
age              264.2584     13.400     19.721      0.000     237.963     290.554
bmi              339.9087     32.570     10.436      0.000     275.994     403.823
children         410.2363    156.889      2.615      0.009     102.364     718.109
==============================================================================
Omnibus:                      231.072   Durbin-Watson:                   2.063
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              550.795
Skew:                           1.233   Prob(JB):                    2.49e-120
Kurtosis:                       5.672   Cond. No.                         319.
==============================================================================
```

Look at the data choosen for building models

Look at the p_values!

```
#___3 now, build model on training set (trn) only

model_1=smf.ols(formula='charges ~ age + sex + bmi + children + smoker + region',data= trn).fit()
print(model_1.summary())
'''
print(model_1.summary())
                            OLS Regression Results
==============================================================================
Dep. Variable:                charges   R-squared:
Model:                            OLS   Adj. R-squared:                  0.7
Method:                 Least Squares   F-statistic:                     385.7
Date:                Sun, 26 Dec 2021   Prob (F-statistic):          4.13e-298
Time:                        08:12:47   Log-Likelihood:                -10103.
No. Observations:                1000   AIC:                         2.022e+04
Df Residuals:                     991   BIC:                         2.027e+04
Df Model:                           8
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept     -1.208e+04   1135.373    -10.643      0.000   -1.43e+04   -9855.249
sex[T.male]    -288.5339    377.406     -0.765      0.445   -1029.140     452.073
smoker[T.yes]  2.383e+04    475.703     50.099      0.000    2.29e+04    2.48e+04
region[T.northwest]  -439.8965    543.632     -0.809      0.419   -1506.699     626.906
region[T.southeast] -1291.2899    534.513     -2.416      0.016   -2340.198    -242.382
region[T.southwest] -1263.1479    537.296     -2.351      0.019   -2317.516    -208.779
age             264.2584     13.400     19.721      0.000     237.963     290.554
bmi             339.9087     32.570     10.436      0.000     275.994     403.823
children        410.2363    156.889      2.615      0.009     102.364     718.109
==============================================================================
Omnibus:                      231.072   Durbin-Watson:                   2.063
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              550.795
Skew:                           1.233   Prob(JB):                    2.49e-120
Kurtosis:                       5.672   Cond. No.                        319.
==============================================================================
```

In case the categorical variable is captured in integers like female = 0 and male = 1; then in the code write capital **C(sex)**; algo will understand that sex is a categorical variable

```
#___3 now, build model on training set (trn) only

model_1=smf.ols(formula='charges ~ age + sex + bmi + children + smoker + region',data= trn).fit()
print(model_1.summary())
'''
print(model_1.summary())
                            OLS Regression Results
==============================================================================
Dep. Variable:                charges   R-squared:                       0.757
Model:                            OLS   Adj. R-squared:                  0.755
Method:                 Least Squares   F-statistic:                     385.7
Date:                Sun, 26 Dec 2021   Prob (F-statistic):          4.13e-298
Time:                        08:12:47   Log-Likelihood:                -10103.
No. Observations:                1000   AIC:                         2.022e+04
Df Residuals:                     991   BIC:                         2.027e+04
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      -1.208e+04   1135.373    -10.643      0.000   -1.43e+04   -9855.249
sex[T.male]     -288.5339    377.406     -0.765      0.445   -1029.140     452.073
smoker[T.yes]   2.383e+04    475.703     50.099      0.000    2.29e+04    2.48e+04
region[T.northwest]  -439.8965    543.632     -0.809      0.419   -1506.699     626.906
region[T.southeast] -1291.2899    534.513     -2.416      0.016   -2340.198    -242.382
region[T.southwest] -1263.1479    537.296     -2.351      0.019   -2317.516    -208.779
age              264.2584     13.400     19.721      0.000     237.963     290.554
bmi              339.9087     32.570     10.436      0.000     275.994     403.823
children         410.2363    156.889      2.615      0.009     102.364     718.109
==============================================================================
Omnibus:                      231.072   Durbin-Watson:                   2.063
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              550.795
Skew:                           1.233   Prob(JB):                    2.49e-120
Kurtosis:                       5.672   Cond. No.                        319.
==============================================================================
```

You need to build more than 1 model with different sets of predictors!

And , select that model which has highest R^2 and all assumptions met!

Test data in two parts

```
#___6 make two parts of test (tst) data
# one, only having predictors and another one having only Target Variable

tst.info() # 338 observations
'''
tst.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 338 entries, 1000 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       338 non-null    int64
 1   sex       338 non-null    object
 2   bmi       338 non-null    float64
 3   children  338 non-null    int64
 4   smoker    338 non-null    object
 5   region    338 non-null    object
 6   charges   338 non-null    float64
dtypes: float64(2), int64(2), object(3)
memory usage: 18.6+ KB'''

x_tst = tst.loc[:, tst.columns != 'charges'] #only predictors
y_tst = tst.loc[:, tst.columns == 'charges'] #only TV
```

Once you finalize your model, now test it!

x_tst - DataFrame

Index	age	sex	bmi	children	smoker	region
1000	30	male	22.99	2	yes	northwest
1001	24	male	32.70	0	yes	southwest
1002	24	male	25.80	0	no	southwest
1003	48	male	29.60	0	no	southwest
1004	47	male	19.19	1	no	northeast

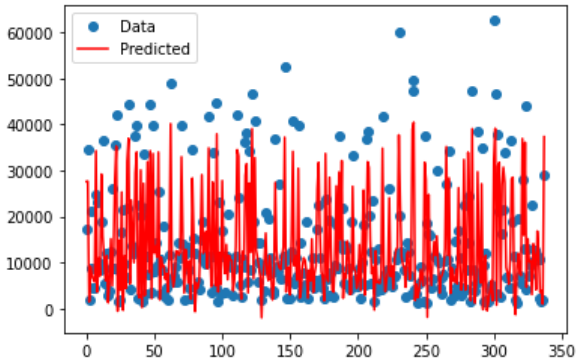
y_tst - DataFrame

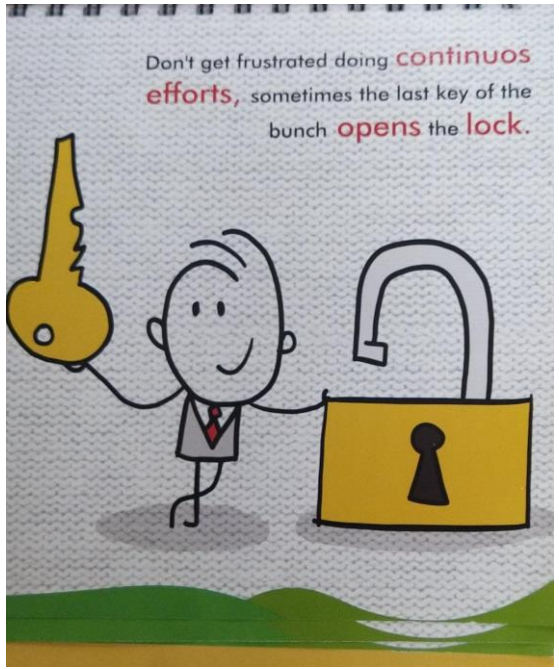
Index	charges
1000	17361.…
1001	34472.…
1002	1972.95
1003	21232.…
1004	8627.54

Predictions on test data

Y = a + b**x**
X = 100
Y = -857.07 + 19.07*100

```
#___7 apply your model on x_tst and find predictions

pred_1 = model_1.predict(x_tst)
```





pred_1 - Series

Index	0
1000	27583.41
1001	27654.65
1002	1476.90
1003	9110.76
1004	6981.43

Residuals

```
#___8 find residuals

# one problem, y_tst is a Data Frame,
# we need to convert into series

## 8(a) convert charges/TV to series
y_tst_series = y_tst.iloc[:, 0] # the first column in data frame

## 8(b) now you can find residuals! SMILE PLEASE!! Haha!!

resd_1 = y_tst_series - pred_1 # aha!!
```

This step is important!

resd_1 - Series

Index	0
1000	-10221.…
1001	6818.19
1002	496.05
1003	12121.42
1004	1646.11

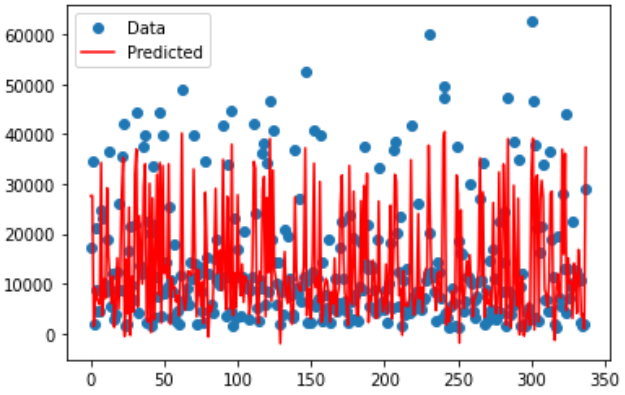
y_tst - DataFrame

Index	charges
1000	17361.…
1001	34472.…
1002	1972.95
1003	21232.…
1004	8627.54

pred_1 - Series

Index	0
1000	27583.41
1001	27654.65
1002	1476.90
1003	9110.76
1004	6981.43

RMSE: the final verdict!



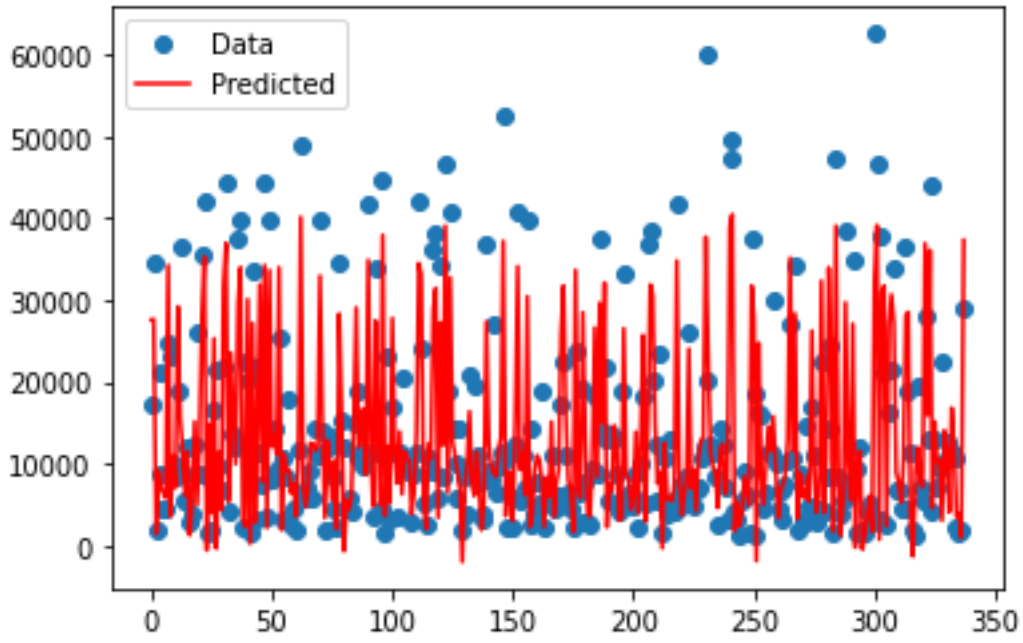
This value should be least!

```
#___9 RMSE rrot mean squared error

# 9(1) First, square the errors/residuals
se_1 = resd_1.pow(2)

# 9(2) Second, sum the squared errors/residuals

sse_1 = se_1.sum() #14030528909.308918

# 9(3) Third, divide by number of data points

msse_1 = sse_1/len(se_1) #41510440.56008556

# 9(4) Fourth, the last one: take square root! That's it

import math # need to import this library

math.sqrt(msse_1)
# Returns: 6442.85
```

	N	O	P	Q	R
	RESIDUAL OUTPUT				
	Observation	*Predicted Sales*	*Residuals*		*resid^2*
	1	986.1747779	-56.1748		3155.606
	2	1007.61336	-107.613		11580.64
	3	1039.771233	-19.7712		390.9017
	4	1050.490524	-60.4905		3659.104
	5	1071.929106	28.07089		787.9751
	6	1093.367688	-43.3677		1880.756
	7	1114.806271	35.19373		1238.599
	8	1125.525562	-5.52556		30.53183
	9	1125.525562	4.474438		20.0206
	10	1146.964144	53.03586		2812.802
	11	1146.964144	103.0359		10616.39
	12	1179.122017	40.87798		1671.009
0%		SUM SQUARED ERRORS=			37844.33
	MEAN SUM SQUARED ERRORS =				3153.694
5	ROOT MEAN SUM SQUARED ERRORS=				56.158

Plot

```
'''
you may like to see the plot of
actual data and predictions! Good!
Lets see how this can be done!!
'''

#_____________________________________
import matplotlib.pyplot as plt

obsno = np.arange(0,338,1)

fig, ax = plt.subplots()
ax.plot(obsno, y_tst_series , "o", label="Data")
ax.plot(obsno, pred_1, "r-", label="Predicted")
ax.legend(loc="best")
```



RMSE: refer excel workbook = RMSE

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Advt	Sales		SUMMARY OUTPUT										RESIDUAL OUTPUT				
2	92	930																
3	94	900		*Regression Statistics*										*Observation*	*Predicted Sales*	*Residuals*		*resid^2*
4	97	1020		Multiple R	0.998681									1	986.1747779	-56.1748		3155.606
5	98	990		R Square	0.997363									2	1007.61336	-107.613		11580.64
6	100	1100		Adjusted R	0.906454									3	1039.771233	-19.7712		390.9017
7	102	1050		Standard E	58.65487									4	1050.490524	-60.4905		3659.104
8	104	1150		Observatic	12									5	1071.929106	28.07089		787.9751
9	105	1120												6	1093.367688	-43.3677		1880.756
10	105	1130		ANOVA										7	1114.806271	35.19373		1238.599
11	107	1200			*df*	*SS*	*MS*	*F*	*ignificance F*					8	1125.525562	-5.52556		30.53183
12	107	1250		Regressior	1	14314756	14314756	4160.79	1.95E-14					9	1125.525562	4.474438		20.0206
13	110	1220		Residual	11	37844.33	3440.393							10	1146.964144	53.03586		2812.802
14				Total	12	14352600								11	1146.964144	103.0359		10616.39
15														12	1179.122017	40.87798		1671.009
16					*Coefficients*	*tandard Err*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*				**SUM SQUARED ERRORS=**		37844.33
17				Intercept	0	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A		**MEAN SUM SQUARED ERRORS =**				3153.694
18				Advt	10.71929	0.16618	64.50419	1.54E-15	10.35353	11.08505	10.35353	11.08505		**ROOT MEAN SUM SQUARED ERRORS=**				**56.158**

RMSE: refer excel workbook = RMSE

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Advt	Sales		SUMMARY OUTPUT										RESIDUAL OUTPUT				
2	92	930																
3	94	900		*Regression Statistics*										*Observation*	*Predicted Sales*	*Residuals*		*resid^2*
4	97	1020		Multiple R	0.998681									1	986.1747779	-56.1748		3155.606
5	98	990		R Square	0.997363									2	1007.61336	-107.613		11580.64
6	100	1100		Adjusted R	0.906454									3	1039.771233	-19.7712		390.9017
7	102	1050		Standard E	58.65487									4	1050.490524	-60.4905		3659.104
8	104	1150		Observatic	12									5	1071.929106	28.07089		787.9751
9	105	1120												6	1093.367688	-43.3677		1880.756
10	105	1130		ANOVA										7	1114.806271	35.19373		1238.599
11	107	1200			*df*	*SS*	*MS*	*F*	*ignificance F*					8	1125.525562	-5.52556		30.53183
12	107	1250		Regressior	1	14314756	14314756	4160.79	1.95E-14					9	1125.525562	4.474438		20.0206
13	110	1220		Residual	11	37844.33	3440.393							10	1146.964144	53.03586		2812.802
14				Total	12	14352600								11	1146.964144	103.0359		10616.39
15														12	1179.122017	40.87798		1671.009
16					*Coefficients*	*tandard Err*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*				**SUM SQUARED ERRORS=**		37844.33
17				Intercept	0	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A		**MEAN SUM SQUARED ERRORS =**				3153.694
18				Advt	10.71929	0.16618	64.50419	1.54E-15	10.35353	11.08505	10.35353	11.08505		**ROOT MEAN SUM SQUARED ERRORS=**				**56.158**

RMSE: refer excel workbook = RMSE

Step 1: square the errors/residuals

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
1	Advt	Sales		SUMMARY OUTPUT										RESIDUAL OUTPUT				
2	92	930																
3	94	900		*Regression Statistics*										*Observation*	*Predicted Sales*	*Residuals*		*resid^2*
4	97	1020		Multiple R	0.998681									1	986.1747779	-56.1748		3155.606
5	98	990		R Square	0.997363									2	1007.61336	-107.613		11580.64
6	100	1100		Adjusted R	0.906454									3	1039.771233	-19.7712		390.9017
7	102	1050		Standard E	58.65487									4	1050.490524	-60.4905		3659.104
8	104	1150		Observatic	12									5	1071.929106	28.07089		787.9751
9	105	1120												6	1093.367688	-43.3677		1880.756
10	105	1130		ANOVA										7	1114.806271	35.19373		1238.599
11	107	1200			*df*	*SS*	*MS*	*F*	*ignificance F*					8	1125.525562	-5.52556		30.53183
12	107	1250		Regressior	1	14314756	14314756	4160.79	1.95E-14					9	1125.525562	4.474438		20.0206
13	110	1220		Residual	11	37844.33	3440.393							10	1146.964144	53.03586		2812.802
14				Total	12	14352600								11	1146.964144	103.0359		10616.39
15														12	1179.122017	40.87798		1671.009
16					*Coefficients*	*tandard Err*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*			**SUM SQUARED ERRORS=**			37844.33
17				Intercept	0	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A		**MEAN SUM SQUARED ERRORS =**				3153.694
18				Advt	10.71929	0.16618	64.50419	1.54E-15	10.35353	11.08505	10.35353	11.08505		**ROOT MEAN SUM SQUARED ERRORS=**				**56.158**

RMSE: refer excel workbook = RMSE

Step 1: square the errors/residuals

Step 2: ADD the squared errors/residuals

Advt	Sales
92	930
94	900
97	1020
98	990
100	1100
102	1050
104	1150
105	1120
105	1130
107	1200
107	1250
110	1220

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.998681
R Square	0.997363
Adjusted R	0.906454
Standard E	58.65487
Observatic	12

ANOVA

	df	*SS*	*MS*	*F*	*ignificance F*
Regressior	1	14314756	14314756	4160.79	1.95E-14
Residual	11	37844.33	3440.393		
Total	12	14352600			

	Coefficients	*tandard Err*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*
Intercept	0	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
Advt	10.71929	0.16618	64.50419	1.54E-15	10.35353	11.08505	10.35353	11.08505

RESIDUAL OUTPUT

Observation	*Predicted Sales*	*Residuals*	*resid^2*
1	986.1747779	-56.1748	3155.606
2	1007.61336	-107.613	11580.64
3	1039.771233	-19.7712	390.9017
4	1050.490524	-60.4905	3659.104
5	1071.929106	28.07089	787.9751
6	1093.367688	-43.3677	1880.756
7	1114.806271	35.19373	1238.599
[illegible]	1125.525562	-5.52556	30.53183
9	11[illegible]525562	4.474438	20.0206
10	1146.96[illegible]44	53.03586	2812.802
11	1146.964144	[illegible]03.0359	10616.39
12	1179.122017	40.87[illegible]98	1671.009

SUM SQUARED ERRORS=	37844.33
MEAN SUM SQUARED ERRORS =	3153.694
ROOT MEAN SUM SQUARED ERRORS=	56.158

RMSE: refer excel workbook = RMSE

Step 1: square the errors/residuals

Step 2: ADD the squared errors/residuals

Step 3: Take the AVERAGE of ADDED squared errors/residuals

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Advt	Sales		SUMMARY OUTPUT										RESIDUAL OUTPUT				
2	92	930																
3	94	900		*Regression Statistics*										*Observation*	*Predicted Sales*	*Residuals*		*resid^2*
4	97	1020		Multiple R	0.998681									1	986.1747779	-56.1748		3155.606
5	98	990		R Square	0.997363									2	1007.61336	-107.613		11580.64
6	100	1100		Adjusted R	0.906454									3	1039.771233	-19.7712		390.9017
7	102	1050		Standard E	58.65487									4	1050.490524	-60.4905		3659.104
8	104	1150		Observatic	12									5	1071.929106	28.07089		787.9751
9	105	1120												6	1093.367688	-43.3677		1880.756
10	105	1130		ANOVA											1114.806271	35.19373		1238.599
11	107	1200			*df*	*SS*	*MS*	*F*	*ignificance F*					8	25.525562	-5.52556		30.53183
12	107	1250		Regressior	1	14314756	14314756	4160.79	1.95E-14					9	1125.25562	4.474438		20.0206
13	110	1220		Residual	11	37844.33	3440.393							10	1146.9641	53.03586		2812.802
14				Total	12	14352600								11	1146.964144	103.0359		10616.39
15														12	1179.122017	40.87798		1671.009
16					*Coefficients*	*tandard Err*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*				SUM SQUARED ERRORS=		37844.33
17				Intercept	0	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A		MEAN SUM SQUARED ERRORS =				3153.694
18				Advt	10.71929	0.16618	64.50419	1.54E-15	10.35353	11.08505	10.35353	11.08505		ROOT MEAN SUM SQUARED ERRORS=				56.158

RMSE: refer excel workbook = RMSE

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Advt	Sales		SUMMARY OUTPUT										RESIDUAL OUTPUT				
2	92	930																
3	94	900		*Regression Statistics*										*Observation*	*Predicted Sales*	*Residuals*		*resid^2*
4	97	1020		Multiple R	0.998681									1	986.1747779	-56.1748		3155.606
5	98	990		R Square	0.997363									2	1007.61336	-107.613		11580.64
6	100	1100		Adjusted R	0.906454									3	1039.771233	-19.7712		390.9017
7	102	1050		Standard E	58.65487									4	1050.490524	-60.4905		3659.104
8	104	1150		Observatic	12									5	1071.929106	28.07089		787.9751
9	105	1120												6	1093.367688	-43.3677		1880.756
10	105	1130		ANOVA											1114.806271	35.19373		1238.599
11	107	1200			*df*	*SS*	*MS*	*F*	*ignificance F*					8	25.525562	-5.52556		30.53183
12	107	1250		Regressior	1	14314756	14314756	4160.79	1.95E-14					9	1125.25562	4.474438		20.0206
13	110	1220		Residual	11	37844.33	3440.393							10	1146.9641	53.03586		2812.802
14				Total	12	14352600								11	1146.964144	103.0359		10616.39
15														12	1179.122017	40.87798		1671.009
16					*Coefficients*	*tandard Erro*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*				*SUM SQUARED ERRORS=*		37844.33
17				Intercept	0	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A				*MEAN SUM SQUARED ERRORS =*		3153.694
18				Advt	10.71929	0.16618	64.50419	1.54E-15	10.35353	11.08505	10.35353	11.08505				*ROOT MEAN SUM SQUARED ERRORS=*		**56.158**

Step 1: square the errors/residuals

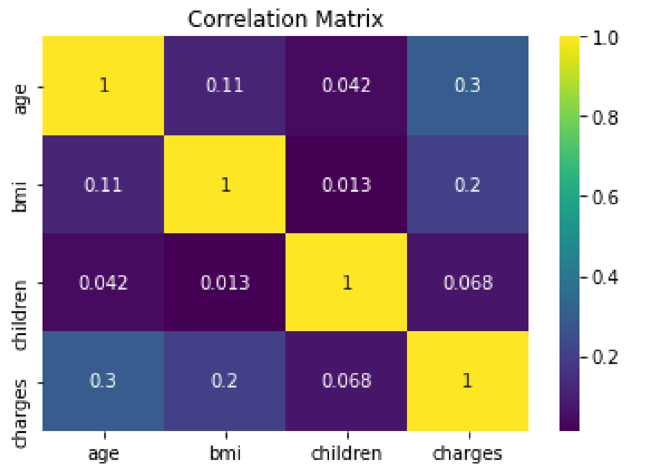
Step 2: ADD the squared errors/residuals

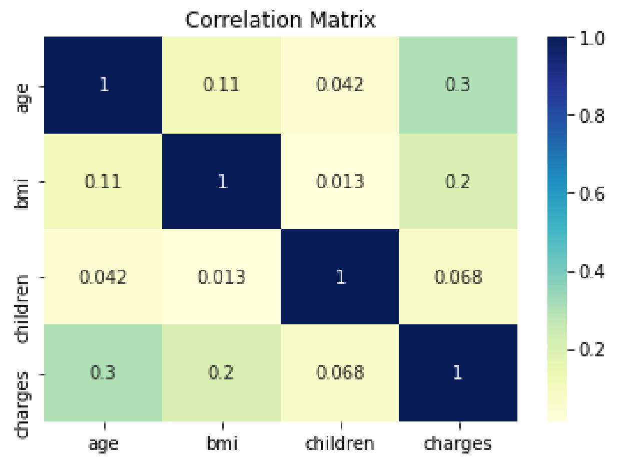
Step 3: Take the AVERAGE of ADDED squared errors/residuals

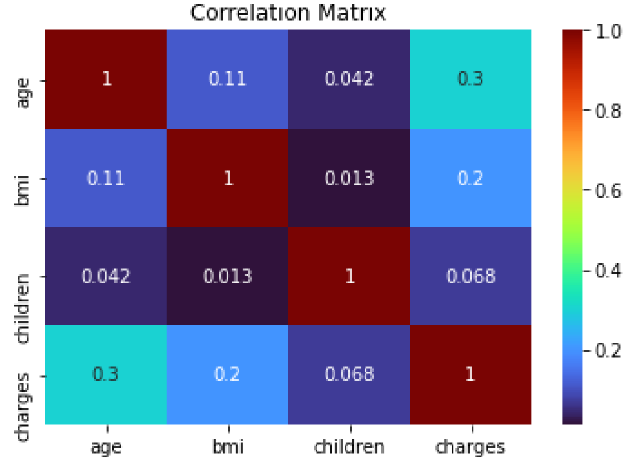
Step 4: SQUAREROOT of the AVERAGE of ADDED squared errors/residuals

Correlation Matrix/Heat Map

```
# correlation/heat map
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sb

# first collect all continuous vars in a data frame
hm = data.loc[:,['age', 'bmi', 'children', 'charges']]

print(hm.corr())
'''
print(hm.corr())
               age       bmi  children   charges
age       1.000000  0.109272  0.042469  0.299008
bmi       0.109272  1.000000  0.012759  0.198341
children  0.042469  0.012759  1.000000  0.067998
charges   0.299008  0.198341  0.067998  1.000000
'''

# plotting correlation heatmap
sb.heatmap(hm.corr(), cmap="viridis", annot=True)
plt.title('Correlation Matrix')
```

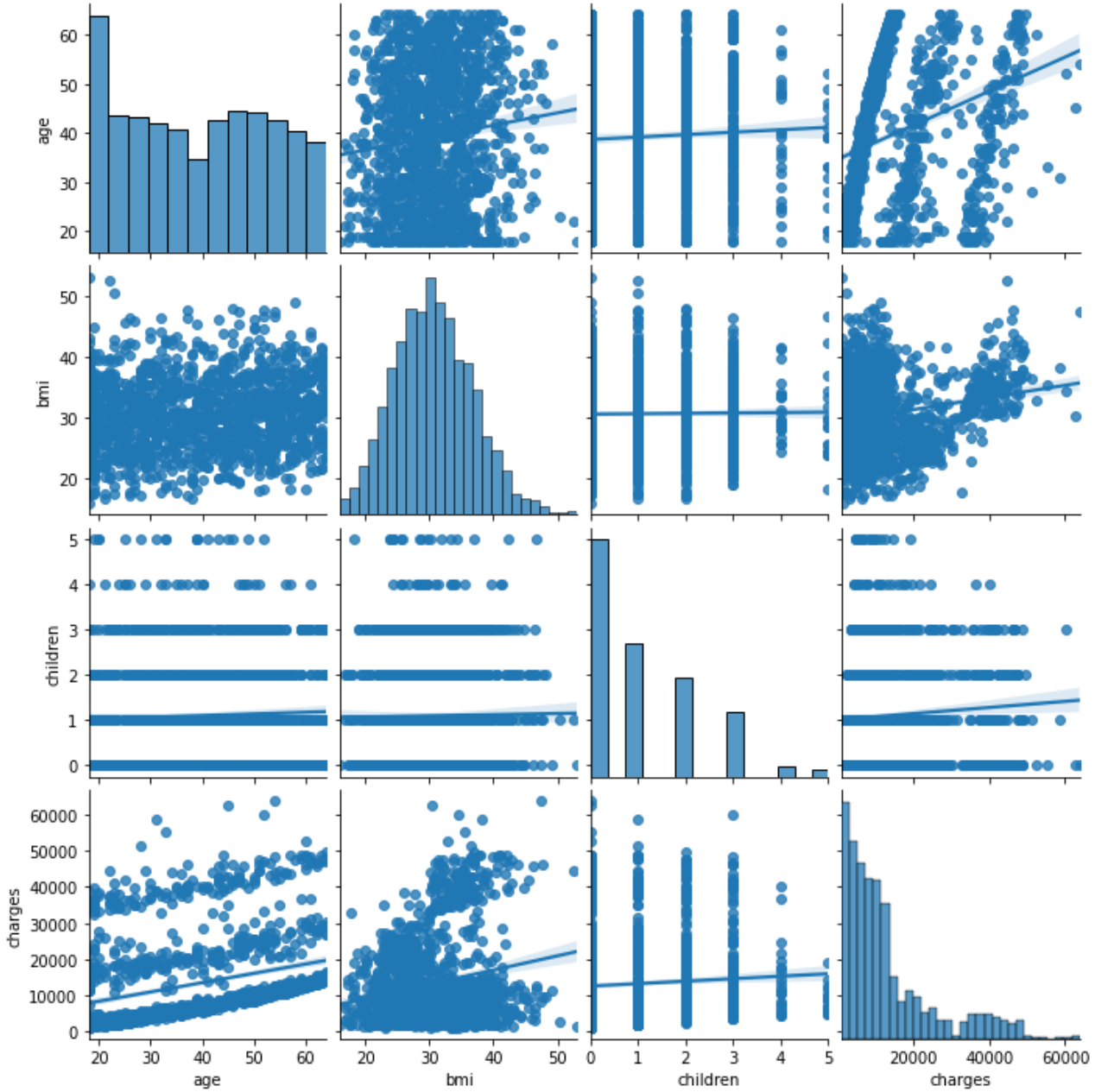






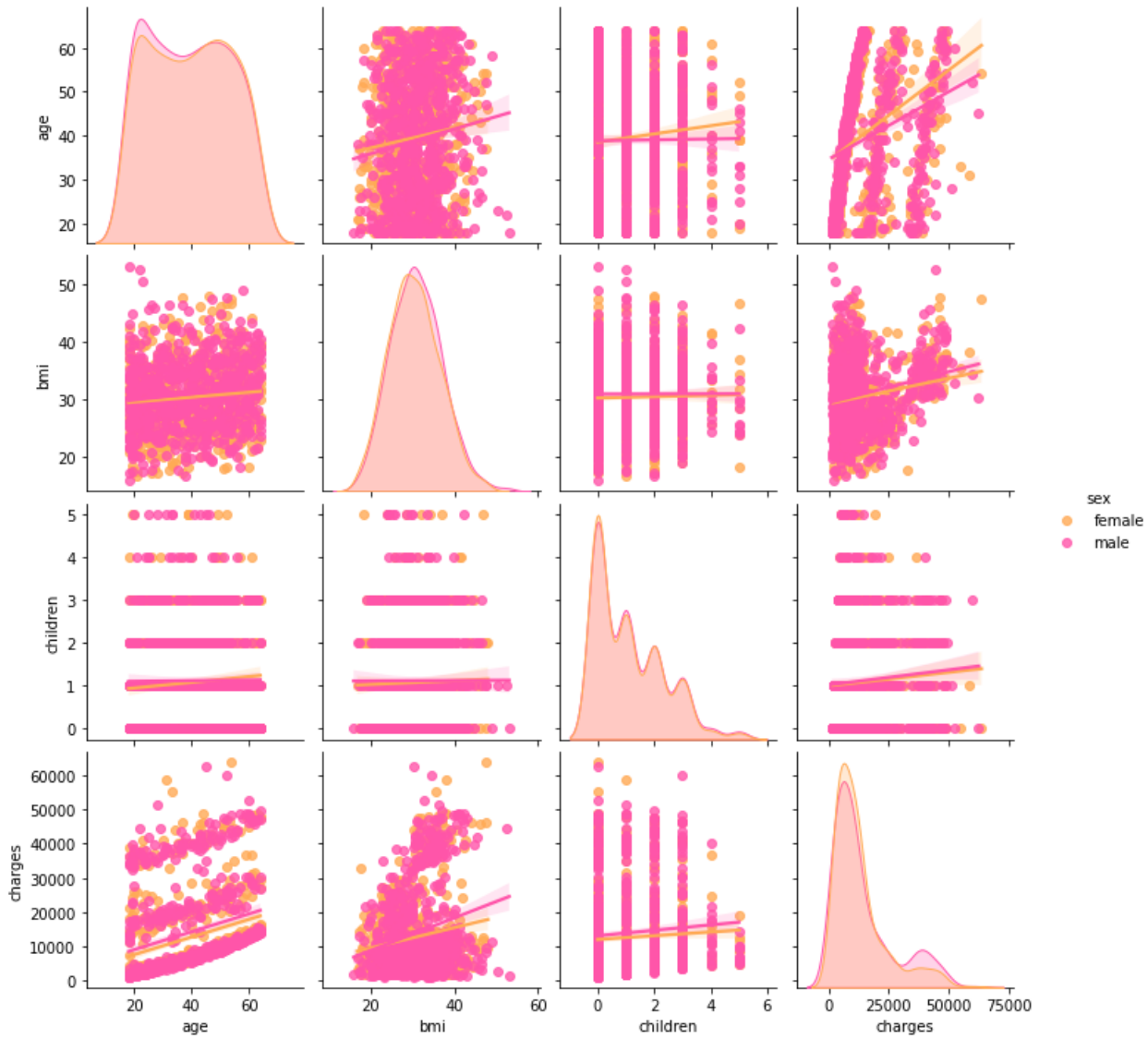
Simple pair plot

```
# pair plot SIMPLE

sb.pairplot(hm, kind = 'reg')
```

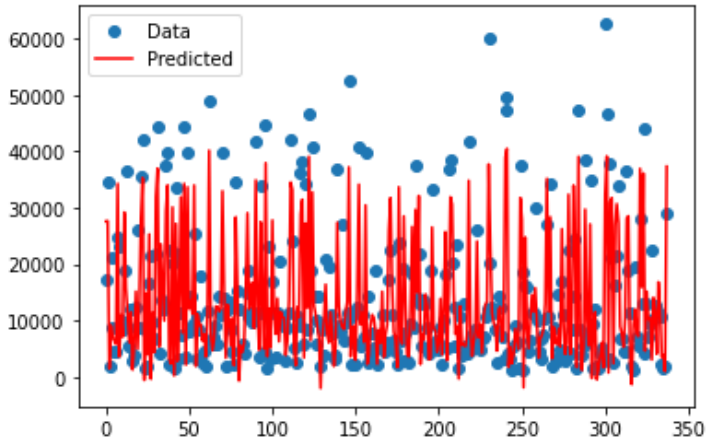


Pair Plots, hue by sex

```
# pair plot HUE by a categorical var, say sex
pp = data.loc[:,['age', 'bmi', 'children', 'charges', 'sex']]

sb.pairplot(pp, hue = 'sex', kind = 'reg', palette = 'spring_r')
```




You win by trying. And failing. Test, try,
fail, measure evolve, repeat, persist.
TRY
FAIL
WIN


Data
Predicted
60000
50000
40000
30000
20000
10000
0
0
50
100
150
200
250
300
350