22. Multiple Linear Regression

- Used when input are more than one
- Multiple linear regression is an extension of simple linear regression as it takes more than one predictor variable to predict the response variable
- **y = m1x1 + m2x2 + m3x3 + ... mnxn + c**

In [102…
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [103…
```
dataset = pd.read_csv(r'Data/salary_data.csv')
dataset.head(3)
```

Out[103…

	Age	Experience	Salary
0	53	21	274930.685866
1	39	19	217753.696272
2	32	19	166660.977435

In [104…
```
dataset.shape
```

Out[104…
```
(1000, 3)
```

In [105…
```
dataset.isnull().sum()
```

Out[105…
```
Age           0
Experience    0
Salary        0
dtype: int64
```

In [106…
```
dataset.shape
```

Out[106…
```
(1000, 3)
```

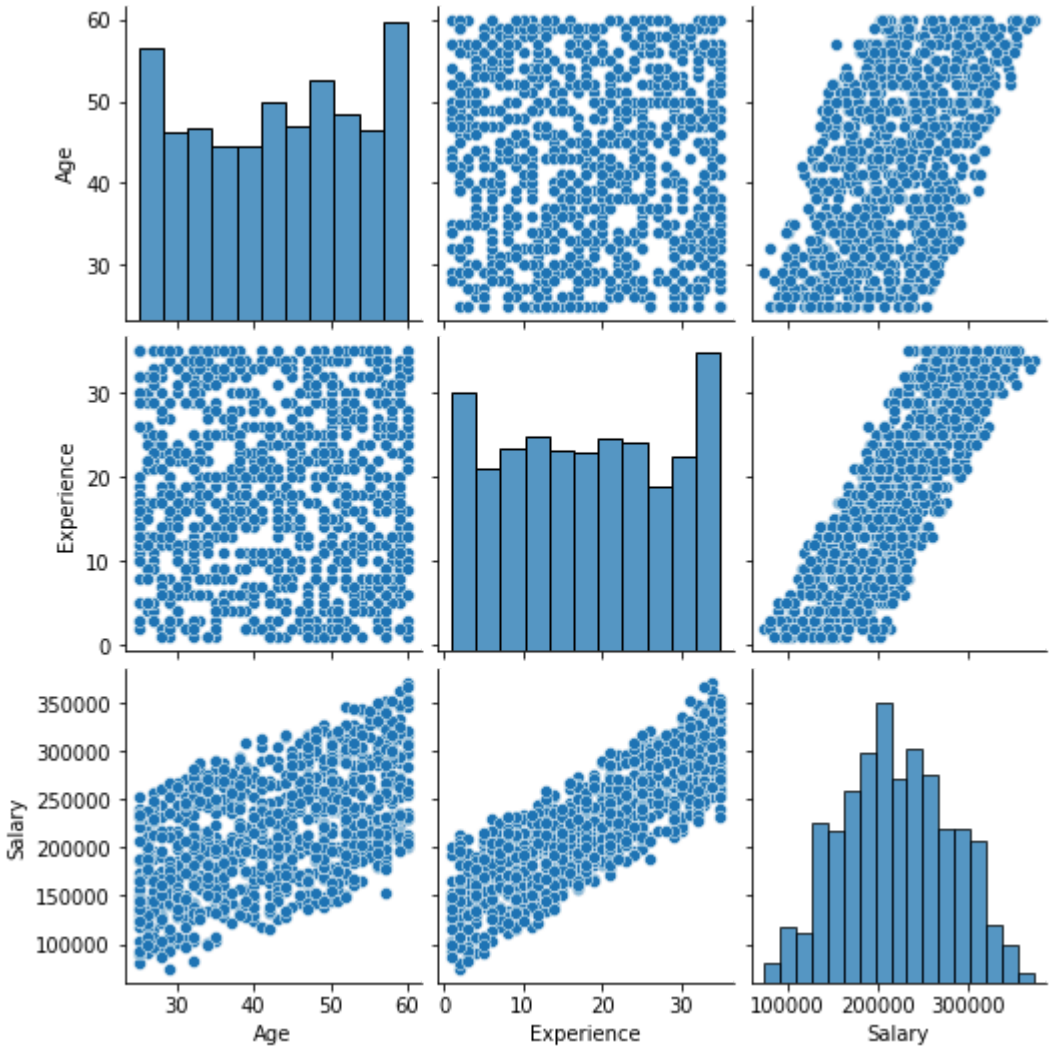
Also an important step before applying model is to check if your data needs scaling (if huge difference in data values)

- but for this exercise, we are not going to check it as we can see no much difference in values of age and experience

To Check if the data is linear before applying linear regression model

In [107…
```
sns.pairplot(data=dataset)
plt.show()
```

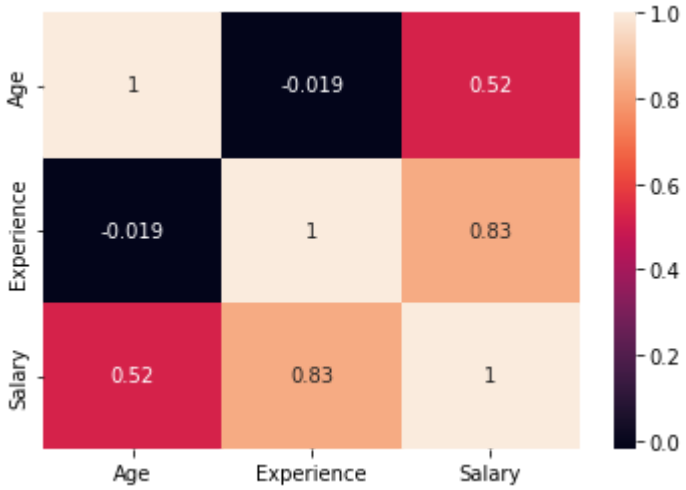
```
C:\Users\rashi\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\axisgrid.py:123: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



Use correlation frunction to check if the data is linear

In [108…

```
# annote = True -> to check correlation number
sns.heatmap(data=dataset.corr(), annot=True)
plt.show()
```



Both of above graph shows correlation between output (salary) and inputs (age and expereince)

In [109…
```
# Separate features and target
x = dataset[['Age', 'Experience']]
y = dataset['Salary']
```

In [110…
```
# Check the shape of X and y
print(X.shape)  # Should be (1000, 2)
print(y.shape)  # Should be (1000,)
```

```
(1000, 2)
(1000,)
```

Train the model

In [111…
```
from sklearn.model_selection import train_test_split
```

In [112…
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.20,random_state
```

Build Model

In [113…
```
from sklearn.linear_model import LinearRegression
```

In [114…
```
lr = LinearRegression()
```

In [115…
```
lr.fit(x_train, y_train)
```

Out[115…
```
▾ LinearRegression
LinearRegression()
```

In [116…
```
dataset.shape
```

Out[116…
```
(1000, 3)
```

Test Model

In [118…
```
lr.score(x_test, y_test)
```

Out[118…
```
0.9738985132159785
```

Make Prediction

In [120…
```
lr.predict(x_test)
```

Out[120…

```
array([127673.47833523, 263638.47930118, 350142.08171943, 145791.96000071,
       229782.58458827, 217703.59681128, 207200.43711274, 250171.09339619,
       167062.09782308, 260806.16230738, 209338.55504254, 244319.02945815,
       207387.86706319, 200973.5132738 , 249421.37359439, 290122.0027354 ,
       289052.9437705 , 186999.35825527, 144722.90103581, 156239.59896145,
       211101.81307144, 145604.53005026, 309309.54336578, 279618.84303686,
       303776.81859083, 189269.38539771, 169894.41481688, 184673.81037502,
       145042.24019891, 169949.93555468, 238279.53556966, 181091.77357942,
       286727.39589025, 166930.18861044, 260674.25309473, 202792.2920405 ,
       134594.60123817, 240549.5627121 , 295974.06667344, 199210.2552449 ,
       322082.73020676, 262624.94107408, 188575.18633371, 140071.80527531,
       207387.86706319, 157870.9477777 , 239348.59453456, 260618.73235693,
       202604.86209005, 216821.96779683, 258161.27526403, 305033.30750618,
       251427.58231154, 170081.84476733, 206693.66799919, 171525.76363313,
       239723.45443546, 154851.20083345, 222861.46168533, 268477.00501213,
       270747.03215457, 315349.03725427, 306102.36647108, 222861.46168533,
       235579.12778851, 184673.81037502, 265964.02718143, 191088.16416441,
       321895.30025631, 186117.72924082, 197259.56726555, 173851.31151338,
       185555.43938947, 308934.68346488, 245388.08842305, 291378.49165075,
       217516.16686083, 207894.63617674, 253003.41038999, 291191.0617003 ,
       198328.62623045, 292072.69071475, 235579.12778851, 264388.19910298,
       316230.66626872, 137801.77813287, 211851.53287324, 211983.44208588,
       187318.69741836, 186249.63845346, 197766.3363791 , 265457.25806788,
       195871.16913756, 270559.60220412, 115594.49055824, 292260.1206652 ,
       227269.60675757, 232934.24074516, 136732.71916797, 287796.45485515,
       236141.41763986, 205249.74913339, 241486.71246435, 221736.88198262,
       228151.23577202, 266151.45713188, 120377.49553139, 200973.5132738 ,
       289052.9437705 , 299368.67351859, 133525.54227327, 207200.43711274,
       338063.09394245, 247338.7764024 , 184111.52052366, 176308.76860627,
       225506.34872867, 106535.2497255 , 194614.68022221, 182723.12239567,
       179890.80540187, 345171.64679584, 149131.04610805, 243062.5405428 ,
       157683.51782725, 202042.5722387 , 231677.75182981, 283895.07889646,
       315349.03725427, 332211.0300044 , 281250.19185311, 196884.70736465,
       242743.2013797 , 252628.55048909, 107604.3086904 , 283707.648946  ,
       203861.3510054 , 291378.49165075, 232052.61173071, 218904.56498883,
       350142.08171943, 161210.03388504, 204368.12011894, 228713.52562337,
       238973.73463365, 188012.89648236, 102446.44381636, 268102.14511122,
       251240.15236109, 144160.61118446, 249796.23349529, 249421.37359439,
       146861.01896561, 206506.23804874, 302200.99051239, 169387.64570333,
       138121.11729596, 146111.29916381, 271121.89205547, 282638.58998111,
       124278.87149008, 310003.74242978, 151269.16403785, 258480.61442713,
       340013.78192179, 256529.92644778, 320319.47217787, 301131.93154749,
       248220.40541685, 278417.87485931, 155920.25979835, 102446.44381636,
       235953.98768941, 195308.8792862 , 128367.67739923, 304845.87755573,
       171900.62353403, 207950.15691454, 287983.8848056 , 263131.71018763,
       241806.05162745, 225506.34872867, 162091.66289949, 151831.4538892 ,
       114338.00164289, 255460.86748288, 236516.27754076, 296348.92657434,
       251240.15236109, 276786.52604306, 297230.55558879, 283707.648946  ,
       168880.87658979, 277536.24584487, 222299.17183397, 275210.69796462,
       325984.10616546, 213934.13006523, 299875.44263214, 297737.32470234])
```

In [ ]: