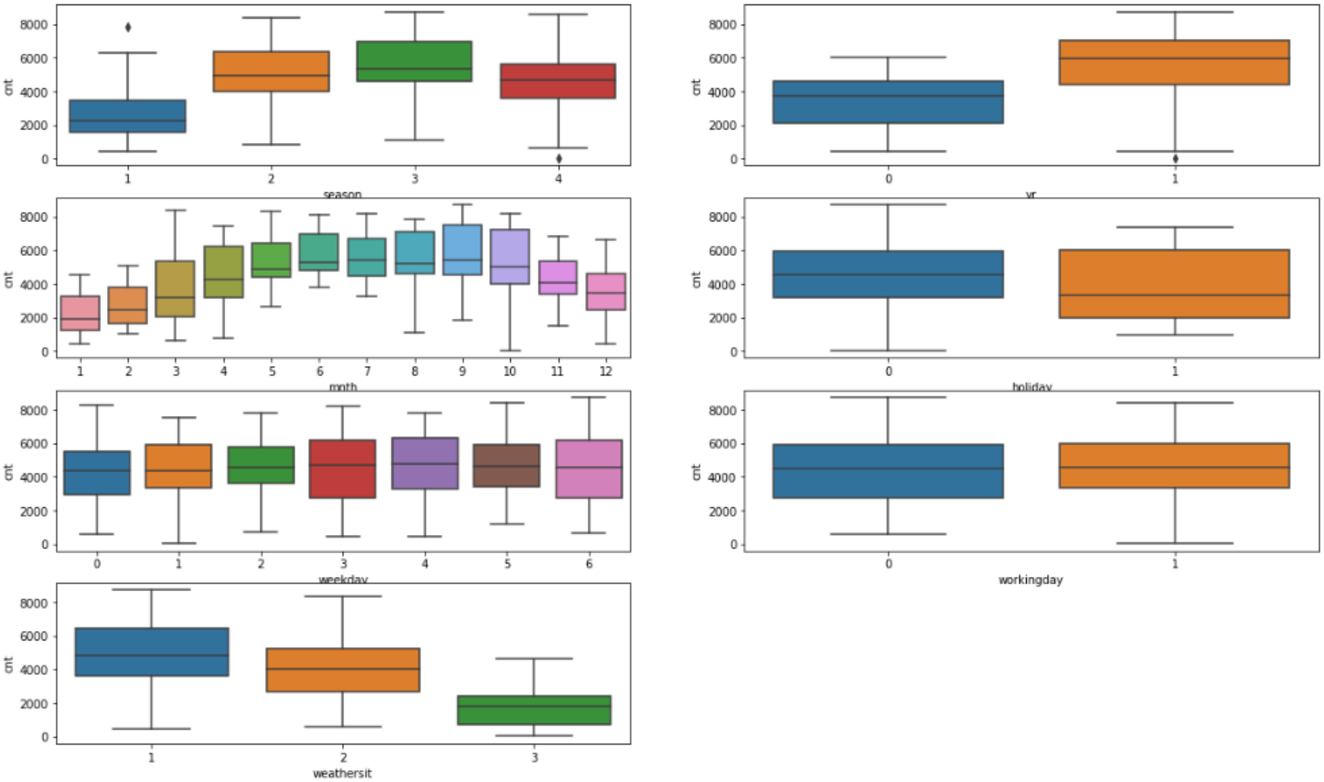
Assignment - based Subjective Questions

1. From your analysis of the categorical variables from the dataset, what could you infer about their effect on the dependent variable?



1. **Season:** Bike rentals are low on Winter, gradually increasing on Spring and Winter and slowed down during fall.

2. **Yr:** Bike rental count is highly affected by “yr” as there is a strong growth from 2018 to 2019.

3. **Month:** Average bike rentals are low on start of the year and it grows gradually until may and from Jun to September seems to be the peak rentals happening and from October till the end of the year the rentals slows down gradually.

4. **Holiday:** Average bike rentals seems to be lesser in holidays when compared to non-holidays.

5. **Weekday:** There is no significant increase or decreasing pattern on weekdays except that Tuesdays have less rentals compared to other days.

6. **Workingday:** Average rentals seems to be the similar on working day vs non-working day

7. **Weathersit:** Bike rentals are high during Clear, Few clouds, Partly cloudy, Partly cloudy and there is a clear pattern of rental decrease for other weather situation

2. Why is it important to use drop_first=True during dummy variable creation?

If a Categorical variable has data of “n” levels, then that can be captured in “n-1” number of columns. We will take the example of “Season” variable from the assignment, it has 4 levels, spring, summer, fall and winter. If these values need to be represented as columns, that needs to hold boolean values in it i.e. spring value in a specific row will be represented in columns as below

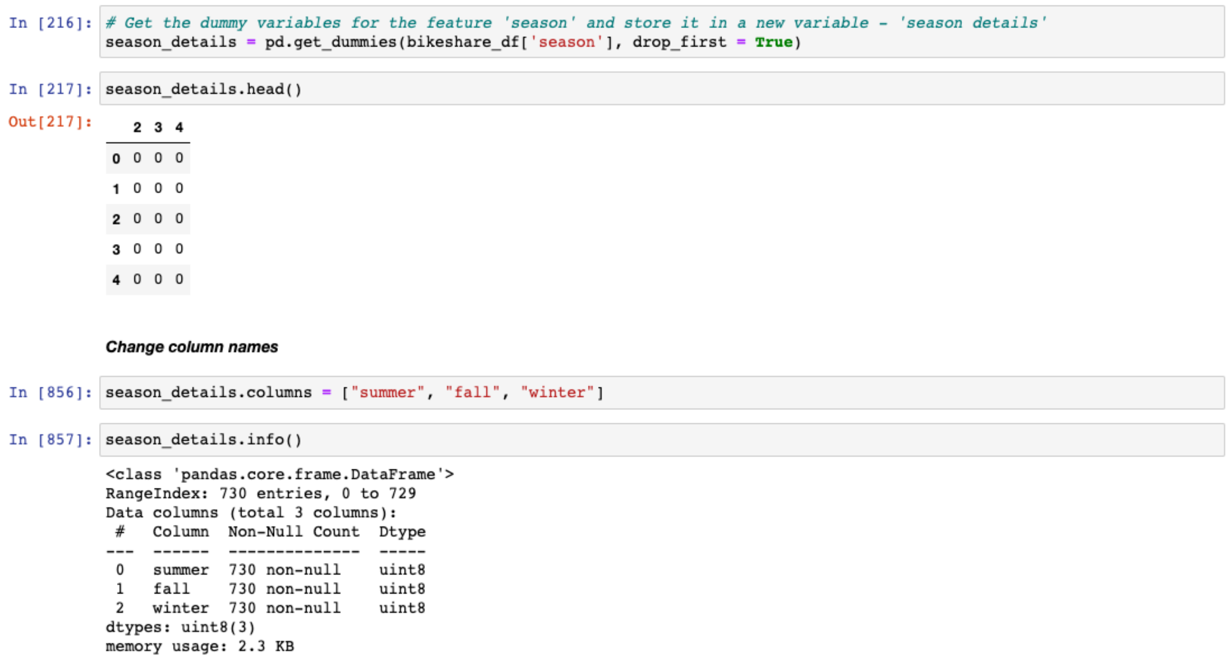
Spring	Summer	Fall	Winter
1	0	0	0

This can also represented without the “spring” column like

Summer	Fall	Winter
0	0	0

This means that if it’s not summer and not fall and not winter then it’s spring. By doing this we are reducing a feature in the model, and it’s very important to keep minimal number of features to help the model perform better.

drop_first=True will drop the first column while creating dummy variables from categorical levels.

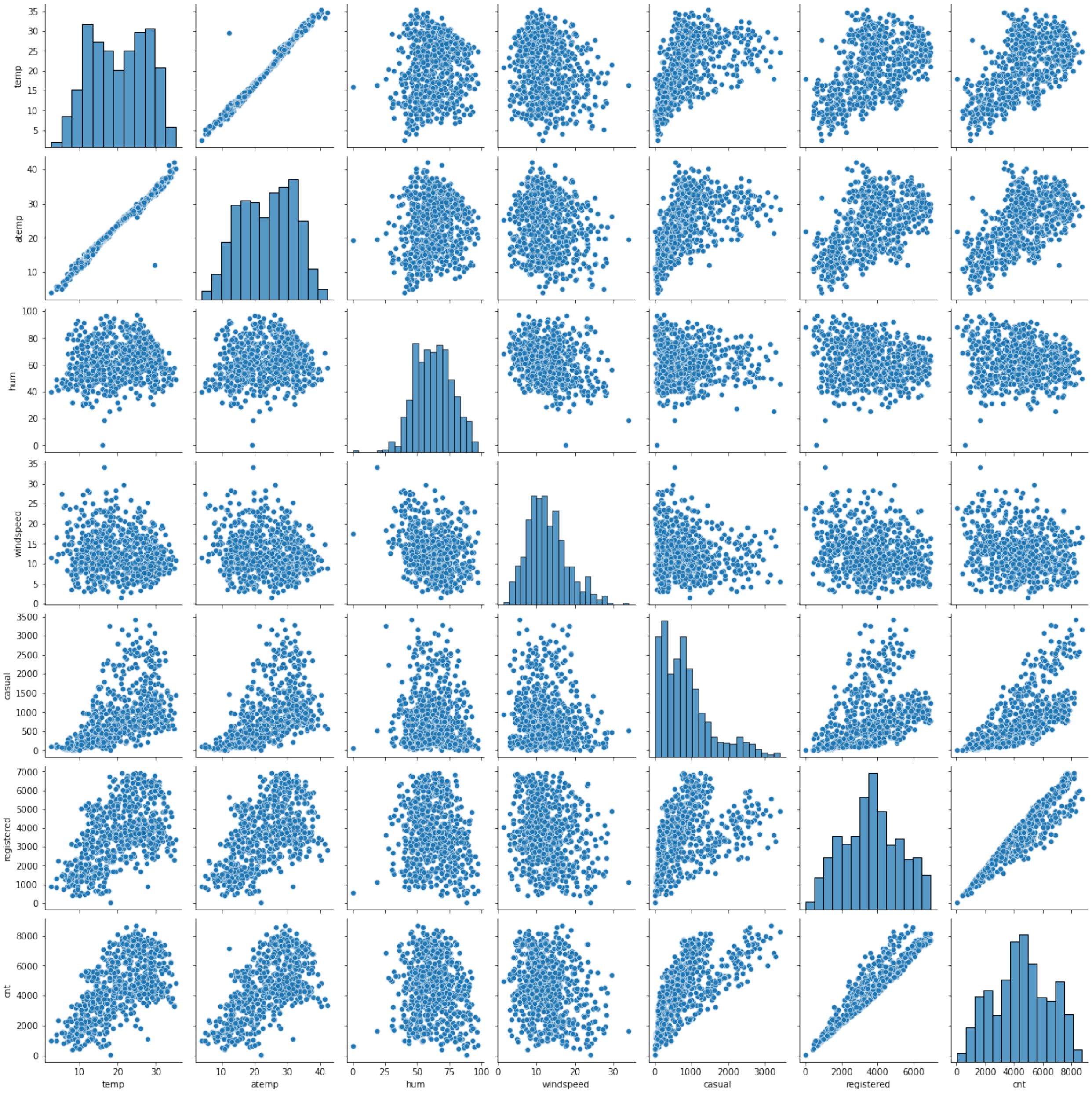


```
In [216]: # Get the dummy variables for the feature 'season' and store it in a new variable - 'season details'
          season_details = pd.get_dummies(bikeshare_df['season'], drop_first = True)

In [217]: season_details.head()

Out[217]:
             2  3  4
          0  0  0  0
          1  0  0  0
          2  0  0  0
          3  0  0  0
          4  0  0  0
```

Change column names

```
In [856]: season_details.columns = ["summer", "fall", "winter"]

In [857]: season_details.info()

          <class 'pandas.core.frame.DataFrame'>
          RangeIndex: 730 entries, 0 to 729
          Data columns (total 3 columns):
           #   Column  Non-Null Count  Dtype
          ---  ------  --------------  -----
           0   summer  730 non-null    uint8
           1   fall    730 non-null    uint8
           2   winter  730 non-null    uint8
          dtypes: uint8(3)
          memory usage: 2.3 KB
```

3. Looking at the pair-plot among the numerical variables, which one has the highest correlation with the target variable?



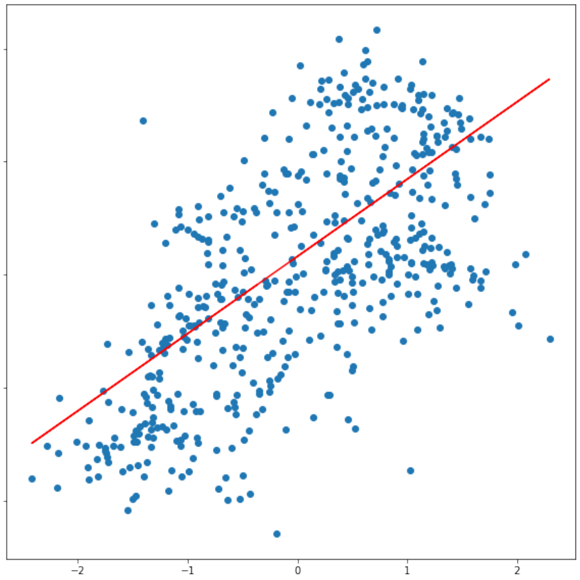
Looking at the pair plot “registered” variable is highly correlated with the target variable “cnt”.

With the features that are selected for the model, “temp” variable is highly correlated with the target variable “cnt”

4. How did you validate the assumptions of Linear Regression after building the model on the training set?

1. **Assumption 1: Variables should be in Linear Relation Ship**

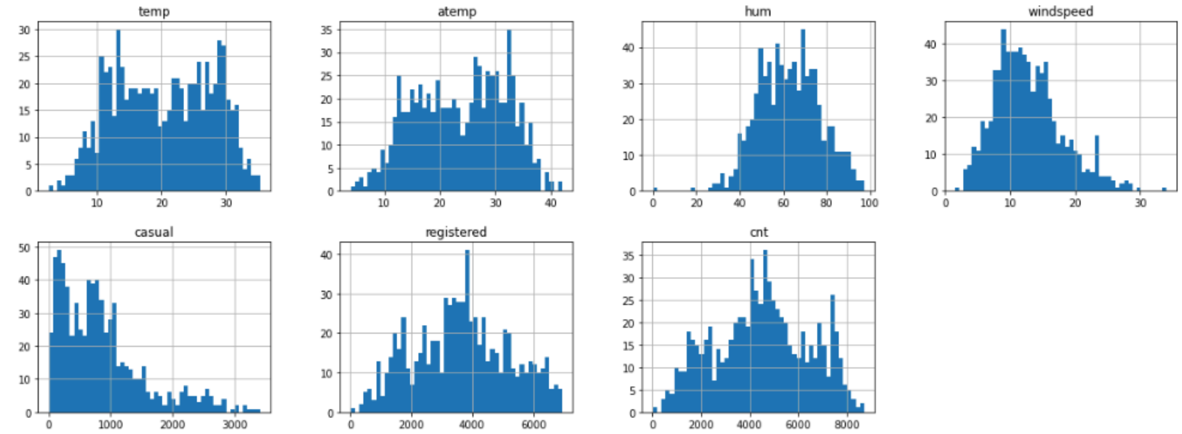
This is validated by plotting the independent variable and dependent variable once the coefficients are found.



From the above image between “temp” and “cnt” it is clear that there is a linear relationship exists.

2. **Assumption 2: All the Variables Should be Multivariate Normal**

Based on the pair plot from the question 3, it is clear that the data is normally distributed. Refer few Histograms below



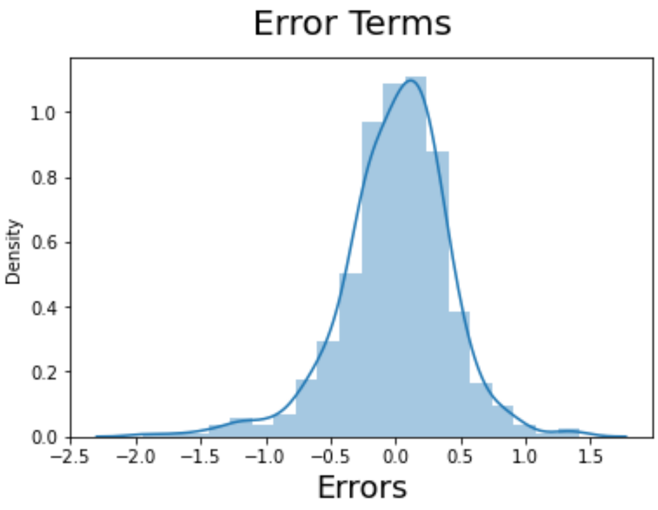
3. **Assumption 3: There Should be No OR Minimum Multicollinearity in the Data**

We have removed the features that has HIGH Multicollinearity by making sure the Variance Inflation Factor (VIF) is less than 5. Here is the table of VIF values for the selected features from our model.

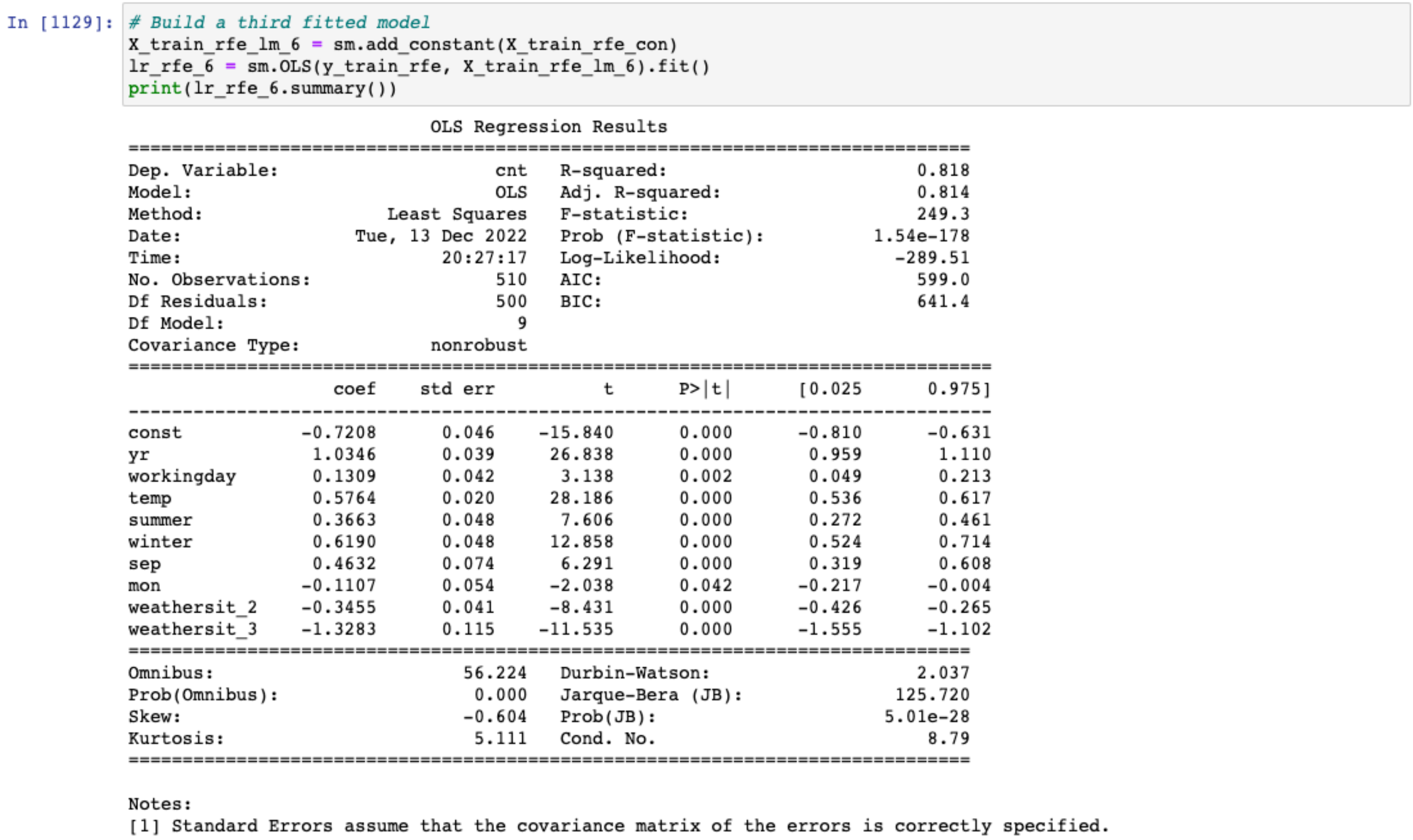
Features	VIF
winter	1.19
summer	1.18
temp	1.15
sep	1.10
workingday	1.04
mon	1.04
weathersit_2	1.04
weathersit_3	1.04
yr	1.02

4. **Assumption 4: Residuals are independent of each other**

Errors are normally distributed and are not autocorrelated to each other, it is evident from the below distplot.



5. Based on the final model, which are the top 3 features contributing significantly towards explaining the demand of the shared bikes?



```
In [1129]: # Build a third fitted model
           X_train_rfe_lm_6 = sm.add_constant(X_train_rfe_con)
           lr_rfe_6 = sm.OLS(y_train_rfe, X_train_rfe_lm_6).fit()
           print(lr_rfe_6.summary())
```

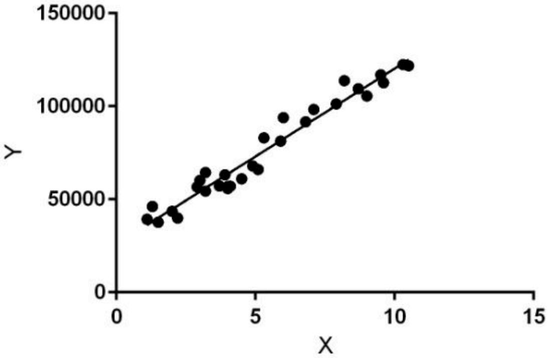
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    cnt   R-squared:                       0.818
Model:                            OLS   Adj. R-squared:                  0.814
Method:                 Least Squares   F-statistic:                     249.3
Date:                Tue, 13 Dec 2022   Prob (F-statistic):          1.54e-178
Time:                        20:27:17   Log-Likelihood:                -289.51
No. Observations:                 510   AIC:                             599.0
Df Residuals:                     500   BIC:                             641.4
Df Model:                           9
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const           -0.7208      0.046    -15.840      0.000      -0.810      -0.631
yr               1.0346      0.039     26.838      0.000       0.959       1.110
workingday       0.1309      0.042      3.138      0.002       0.049       0.213
temp             0.5764      0.020     28.186      0.000       0.536       0.617
summer           0.3663      0.048      7.606      0.000       0.272       0.461
winter           0.6190      0.048     12.858      0.000       0.524       0.714
sep              0.4632      0.074      6.291      0.000       0.319       0.608
mon             -0.1107      0.054     -2.038      0.042      -0.217      -0.004
weathersit_2    -0.3455      0.041     -8.431      0.000      -0.426      -0.265
weathersit_3    -1.3283      0.115    -11.535      0.000      -1.555      -1.102
==============================================================================
Omnibus:                       56.224   Durbin-Watson:                   2.037
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              125.720
Skew:                          -0.604   Prob(JB):                     5.01e-28
Kurtosis:                       5.111   Cond. No.                         8.79
==============================================================================

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

Based on the model, Weather Situation , Season, Year and are the top 3 features contributing significantly

General Subjective Questions

1. Explain the linear regression algorithm in detail.

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used. There are many names for a regression's dependent variable. It may be called an outcome variable, criterion variable, endogenous variable, target variable, or regressed. The independent variables can be called exogenous variables, predictor variables, or regressors.



Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression. In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model. **Hypothesis function for Linear Regression :**

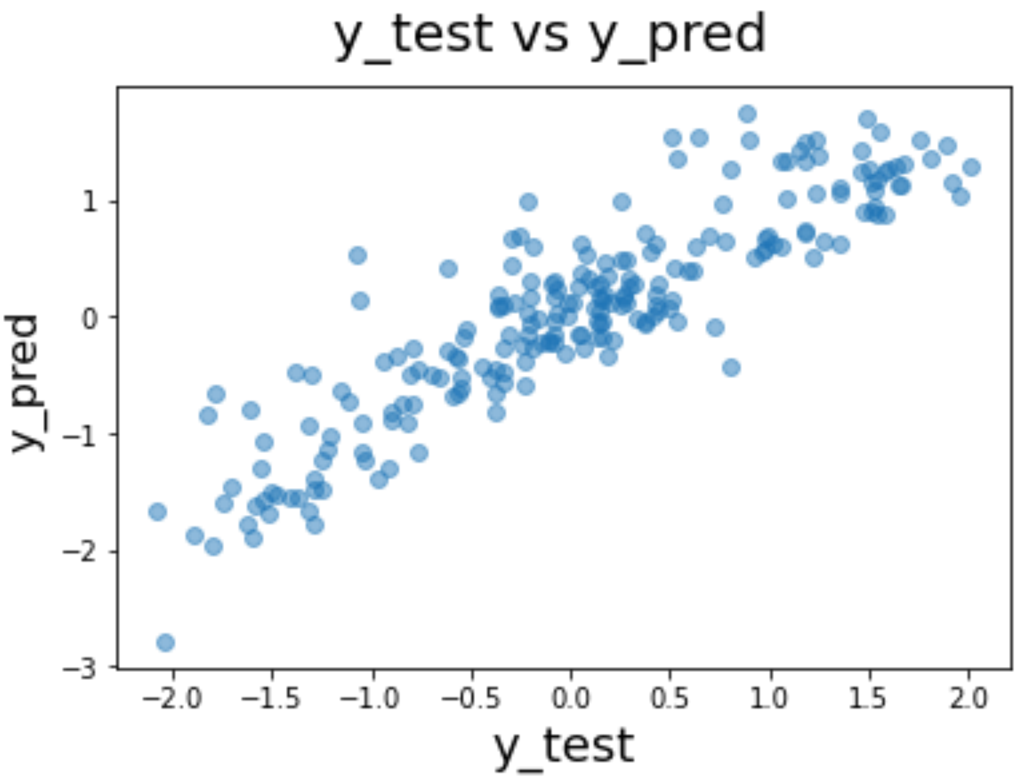
$$y = \theta_1 + \theta_2 . x$$

While training the model we are given : **x:** input training data (univariate – one input variable(parameter)) **y:** labels to data (Supervised learning) When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best θ1 and θ2 values. **θ1:** intercept **θ2:** coefficient of x Once we find the best θ1 and θ2 values, we get the best fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x. **How to update θ1 and θ2 values to get the best fit line ?**

Cost Function (J): By achieving the best-fit regression line, the model aims to predict y value such that the error difference between predicted value and true value is minimum. So, it is very important to update the θ1 and θ2 values, to reach the best value that minimize the error between predicted y value (pred) and true y value (y).

$$J = \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2$$

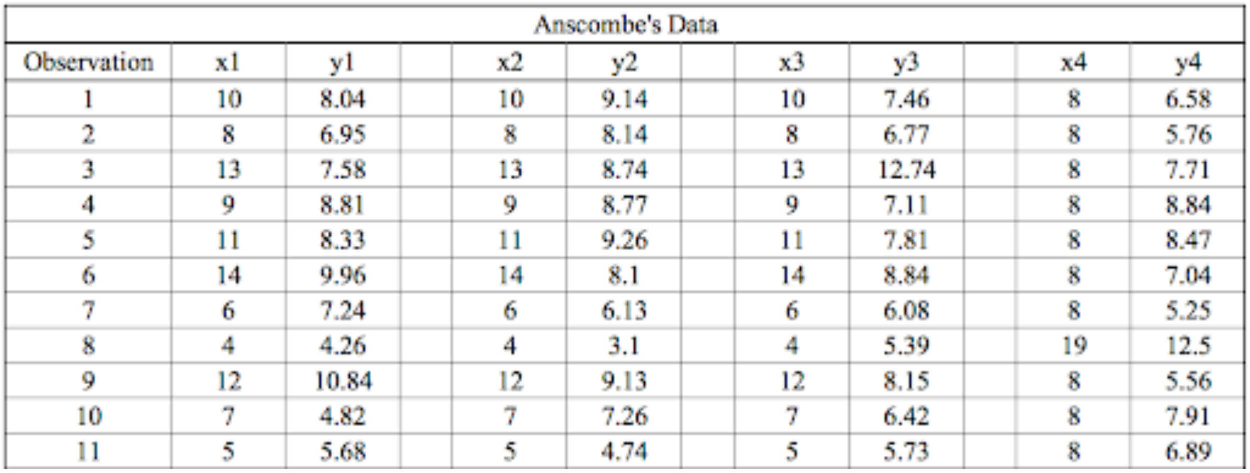
Cost function(J) of Linear Regression is the **Root Mean Squared Error (RMSE)** between predicted y value (pred) and true y value (y). **Gradient Descent:** To update θ1 and θ2 values in order to reduce Cost function (minimizing RMSE value) and achieving the best fit line the model uses Gradient Descent. The idea is to start with random θ1 and θ2 values and then iteratively updating the values, reaching minimum cost.



2. Explain the Anscombe’s quartet in detail.

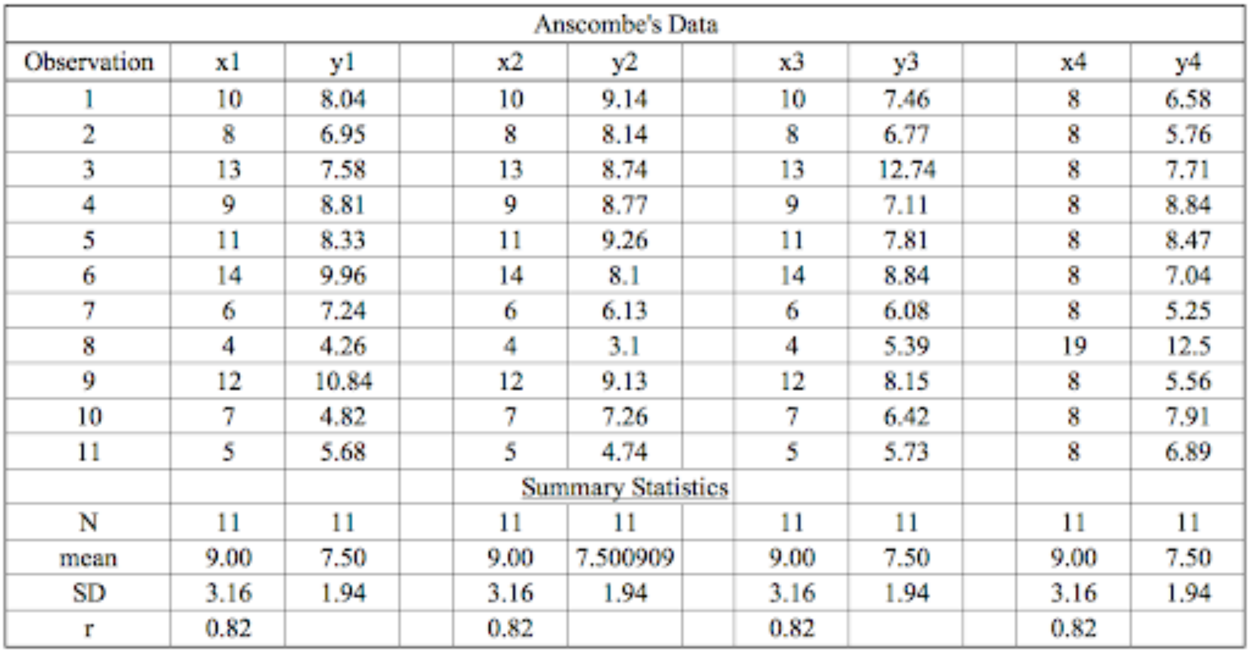
Anscombe’s quartet tells us about the importance of visualizing data before applying various algorithms to build models. This suggests the data features must be plotted to see the distribution of the samples that can help you identify the various anomalies present in the data (outliers, diversity of the data, linear separability of the data, etc.). Moreover, the linear regression can only be considered a fit for the data with linear relationships and is incapable of handling any other kind of data set.

We can define these four plots as follows:



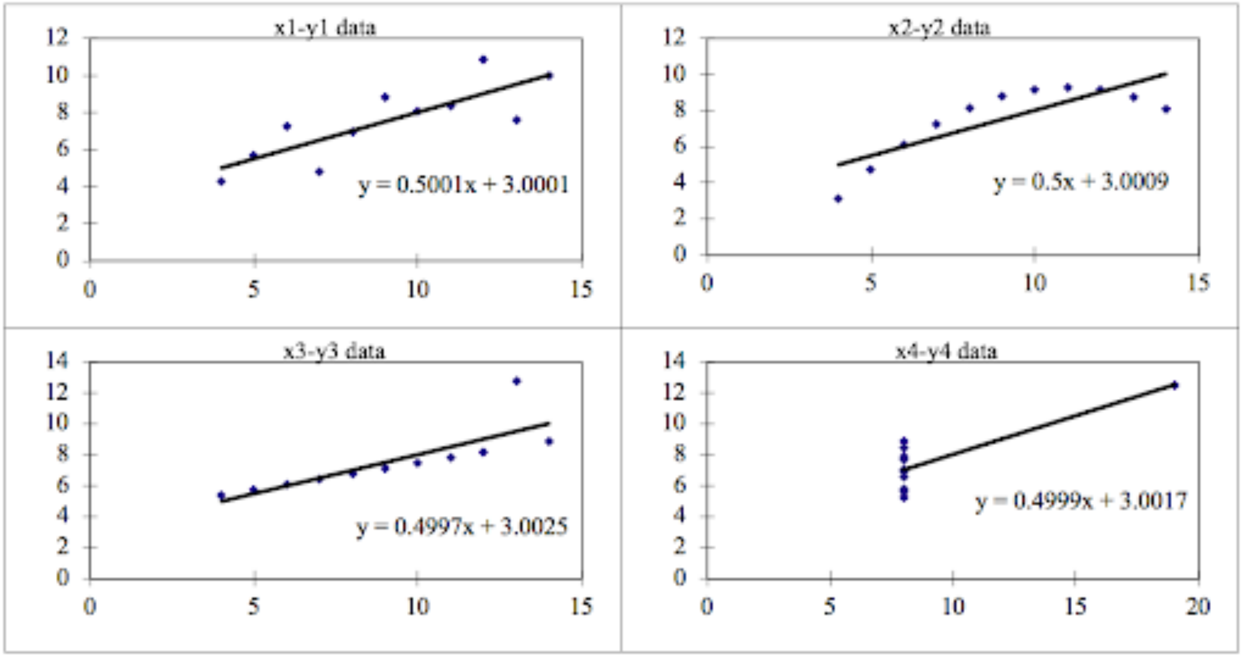
			Anscombe's Data								
Observation	x1	y1		x2	y2		x3	y3		x4	y4
1	10	8.04		10	9.14		10	7.46		8	6.58
2	8	6.95		8	8.14		8	6.77		8	5.76
3	13	7.58		13	8.74		13	12.74		8	7.71
4	9	8.81		9	8.77		9	7.11		8	8.84
5	11	8.33		11	9.26		11	7.81		8	8.47
6	14	9.96		14	8.1		14	8.84		8	7.04
7	6	7.24		6	6.13		6	6.08		8	5.25
8	4	4.26		4	3.1		4	5.39		19	12.5
9	12	10.84		12	9.13		12	8.15		8	5.56
10	7	4.82		7	7.26		7	6.42		8	7.91
11	5	5.68		5	4.74		5	5.73		8	6.89

The statistical information for these four data sets are approximately similar. We can compute them as follows:



			Anscombe's Data								
Observation	x1	y1		x2	y2		x3	y3		x4	y4
1	10	8.04		10	9.14		10	7.46		8	6.58
2	8	6.95		8	8.14		8	6.77		8	5.76
3	13	7.58		13	8.74		13	12.74		8	7.71
4	9	8.81		9	8.77		9	7.11		8	8.84
5	11	8.33		11	9.26		11	7.81		8	8.47
6	14	9.96		14	8.1		14	8.84		8	7.04
7	6	7.24		6	6.13		6	6.08		8	5.25
8	4	4.26		4	3.1		4	5.39		19	12.5
9	12	10.84		12	9.13		12	8.15		8	5.56
10	7	4.82		7	7.26		7	6.42		8	7.91
11	5	5.68		5	4.74		5	5.73		8	6.89
			Summary Statistics								
N	11	11		11	11		11	11		11	11
mean	9.00	7.50		9.00	7.500909		9.00	7.50		9.00	7.50
SD	3.16	1.94		3.16	1.94		3.16	1.94		3.16	1.94
r	0.82			0.82			0.82			0.82	

However, when these models are plotted on a scatter plot, each data set generates a different kind of plot that isn't interpretable by any regression algorithm, as you can see below:



We can describe the four data sets as:

ANSCOMBE'S QUARTET FOUR DATASETS

- **Data Set 1:** fits the linear regression model pretty well.
- **Data Set 2:** cannot fit the linear regression model because the data is non-linear.
- **Data Set 3:** shows the outliers involved in the data set, which cannot be handled by the linear regression model.
- **Data Set 4:** shows the outliers involved in the data set, which also cannot be handled by the linear regression model.

As you can see, Anscombe's quartet helps us to understand the importance of data visualization and how easy it is to fool a regression algorithm. So, before attempting to interpret and model the data or implement any machine learning algorithm, we first need to visualize the data set in order to help build a well-fit model.

3. What is Pearson’s R?

The Pearson correlation coefficient (r) is the most common way of measuring a linear correlation. It is a number between –1 and 1 that measures the strength and direction of the relationship between two variables.

Pearson correlation coefficient (*r*)	**Correlation type**	**Interpretation**
Between 0 and 1	Positive correlation	When one variable changes, the other variable changes in the **same direction**.
0	No correlation	There is **no relationship** between the variables.
Between 0 and –1	Negative correlation	When one variable changes, the other variable changes in the **opposite direction**.

The Pearson correlation coefficient (r) is the most widely used correlation coefficient and is known by many names:

- Pearson’s r
- Bivariate correlation
- Pearson product-moment correlation coefficient (PPMCC)
- The correlation coefficient

The Pearson correlation coefficient is a descriptive statistic, meaning that it summarizes the characteristics of a dataset. Specifically, it describes the strength and direction of the linear relationship between two quantitative variables.

Although interpretations of the relationship strength (also known as effect size) vary between disciplines, the table below gives general rules of thumb:

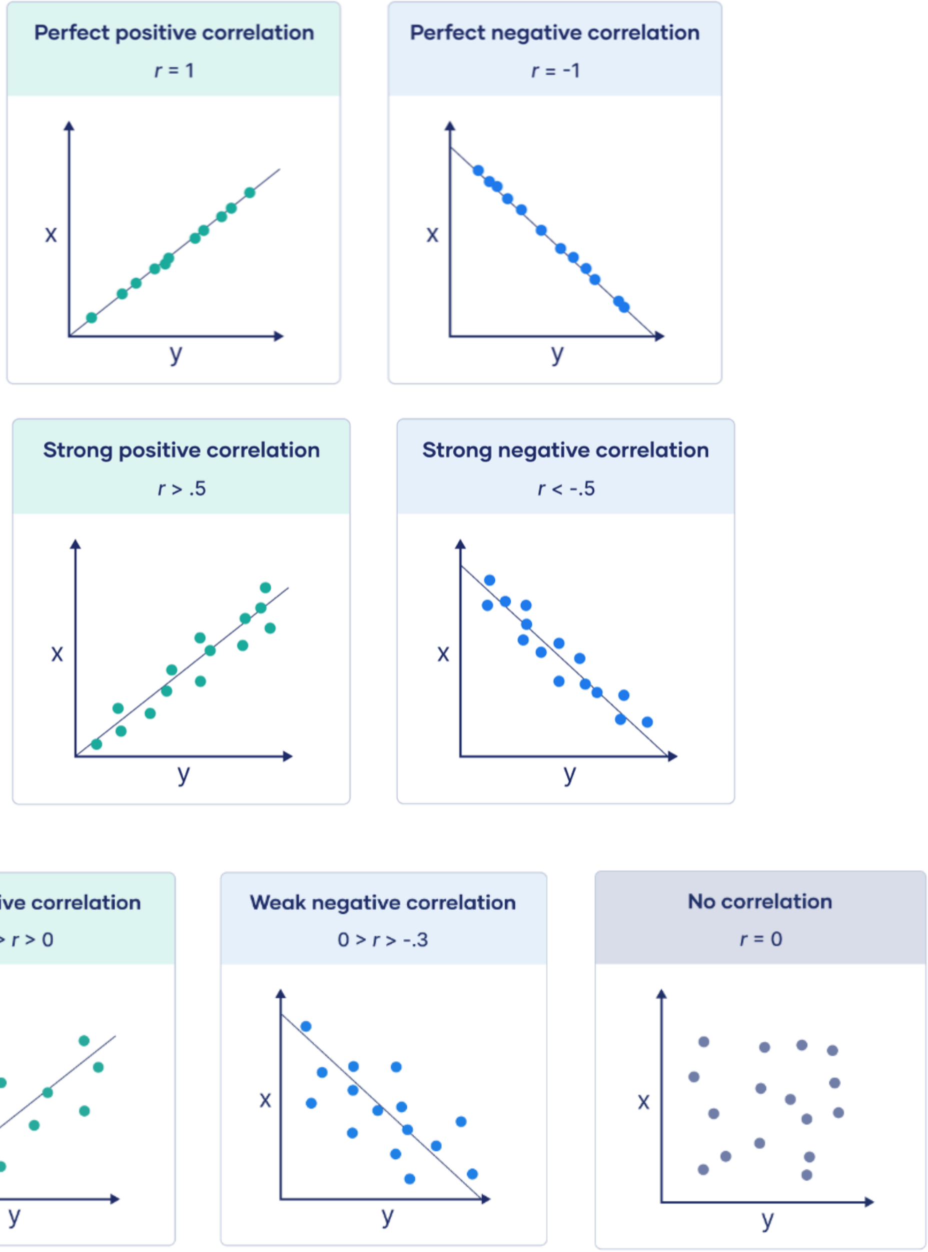
Pearson correlation coefficient (r) value	**Strength**	**Direction**
Greater than .5	Strong	Positive
Between .3 and .5	Moderate	Positive
Between 0 and .3	Weak	Positive
0	None	None
Between 0 and –.3	Weak	Negative
Between –.3 and –.5	Moderate	Negative
Less than –.5	Strong	Negative

Visualizing the Pearson correlation coefficient

Another way to think of the Pearson correlation coefficient (*r*) is as a measure of how close the observations are to a line of best fit.

The Pearson correlation coefficient also tells you whether the slope of the line of best fit is negative or positive. When the slope is negative, *r* is negative. When the slope is positive, *r* is positive.

When *r* is 1 or –1, all the points fall exactly on the line of best fit:



4. What is scaling? Why is scaling performed? What is the difference between normalized scaling and standardized scaling?

It is a step of data Pre-Processing which is applied to independent variables to normalize the data within a particular range. It also helps in speeding up the calculations in an algorithm.

Most of the times, collected data set contains features highly varying in magnitudes, units and range. If scaling is not done then algorithm only takes magnitude in account and not units hence incorrect modeling. To solve this issue, we have to do scaling to bring all the variables to the same level of magnitude.

It is important to note that scaling just affects the coefficients and none of the other parameters like t-statistic, F-statistic, p-values, R-squared, etc.

Normalization/Min-Max Scaling:

It brings all of the data in the range of 0 and 1 .
sklearn.preprocessing.MinMaxScaler helps to implement normalization in python.

$$\text{MinMax Scaling: } x = \frac{x - min(x)}{max(x) - min(x)}$$

Standardization Scaling:

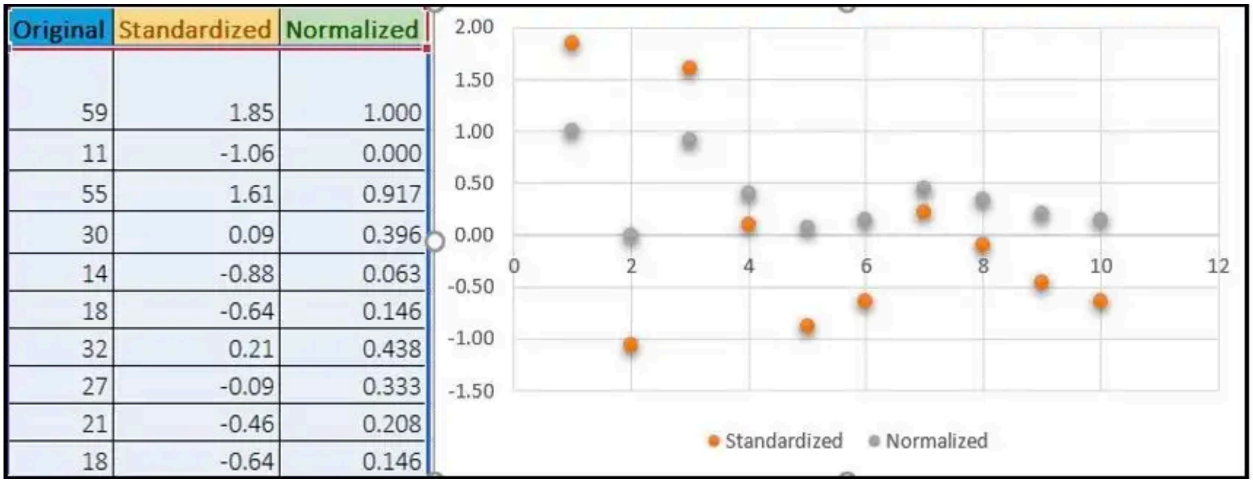
Standardization replaces the values by their Z scores. It brings all of the data into a standard normal distribution which has mean (μ) zero and standard deviation one (σ).

$$\text{Standardisation: } x = \frac{x - mean(x)}{sd(x)}$$

- sklearn.preprocessing.scale helps to implement standardization in python.
- One disadvantage of normalization over standardization is that it loses some information in the data, especially about outliers.

Example:

Below shows example of Standardized and Normalized scaling on original values.



Original	Standardized	Normalized
59	1.85	1.000
11	-1.06	0.000
55	1.61	0.917
30	0.09	0.396
14	-0.88	0.063
18	-0.64	0.146
32	0.21	0.438
27	-0.09	0.333
21	-0.46	0.208
18	-0.64	0.146

5. You might have observed that sometimes the value of VIF is infinite. Why does this happen?

If there is perfect correlation, then VIF = infinity. This shows a perfect correlation between two independent variables.

$$VIF_i = \frac{1}{1-R_i^{\,2}}$$

In the case of perfect correlation, we get R2 =1, which lead to 1/(1-R2) infinity.

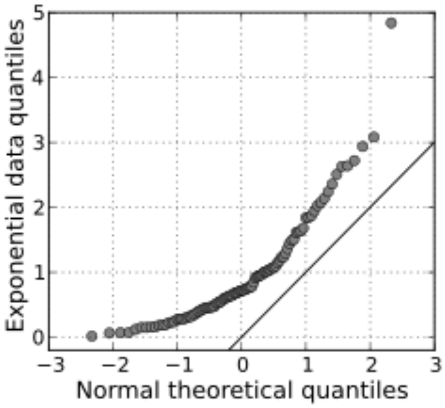
To solve this problem we need to drop one of the variables from the dataset which is causing this perfect multicollinearity.

An infinite VIF value indicates that the corresponding variable may be expressed exactly by a linear combination of other variables (which show an infinite VIF as well).

6. What is a Q-Q plot? Explain the use and importance of a Q-Q plot in linear regression.

Q-Q Plots (Quantile-Quantile plots) are plots of two quantiles against each other. A quantile is a fraction where certain values fall below that quantile. For example, the median is a quantile where 50% of the data fall below that point and 50% lie above it. The purpose of Q Q plots is to find out if two sets of data come from the same distribution. A 45 degree angle is plotted on the Q Q plot; if the two data sets come from a common distribution, the points will fall on that reference line.

A Q Q plot showing the 45 degree reference line:



If the two distributions being compared are similar, the points in the Q–Q plot will approximately lie on the line y = x. If the distributions are linearly related, the points in the Q–Q plot will approximately lie on a line, but not necessarily on the line y = x. Q–Q plots can also be used as a graphical means of estimating parameters in a location-scale family of distributions.

A Q–Q plot is used to compare the shapes of distributions, providing a graphical view of how properties such as location, scale, and skewness are similar or different in the two distributions.