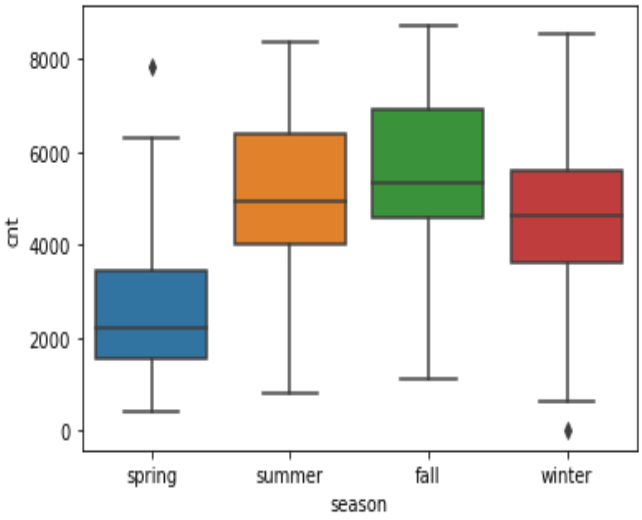
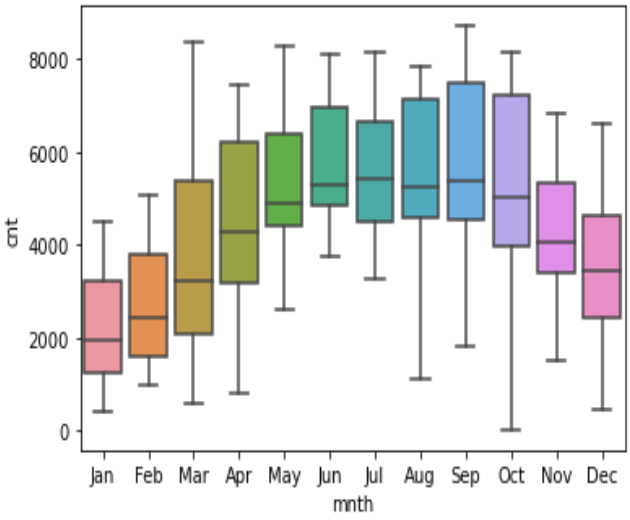
Assignment-based Subjective Questions

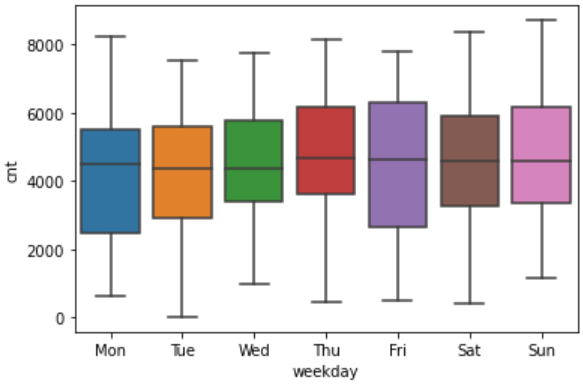
1. **From your analysis of the categorical variables from the dataset, what could you infer about their effect on the dependent variable?**

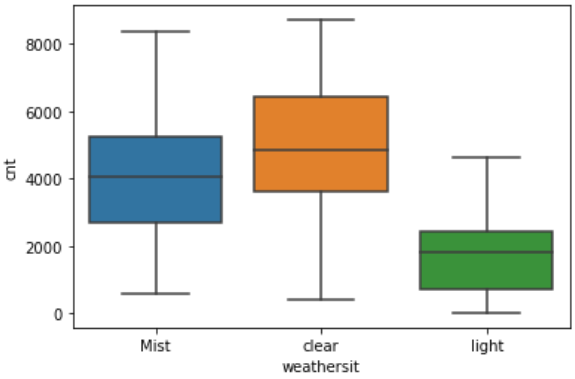
Observations:

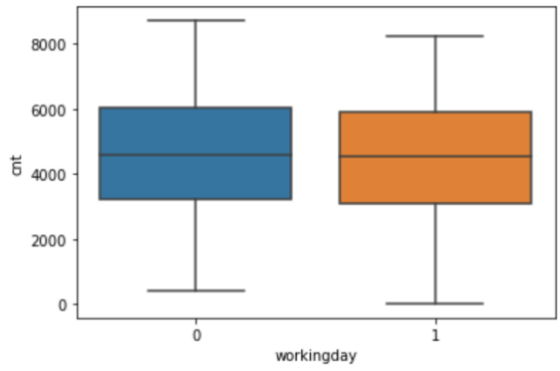
- For the fall season, we see the max no of bookings of vehicles.
- The number of bike bookings is higher in 2019 compared to 2018.
- For the month of Sep, we see max bookings.
- The no of bookings is highest on non-holiday days.
- Max bookings are found on weekends/Sundays.
- Max bookings are found on working days.
- Max bookings are found on clear weather days.

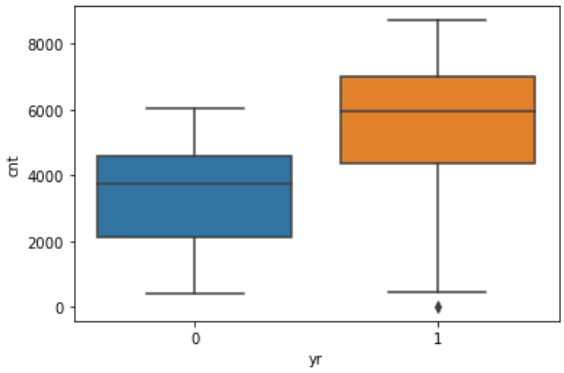


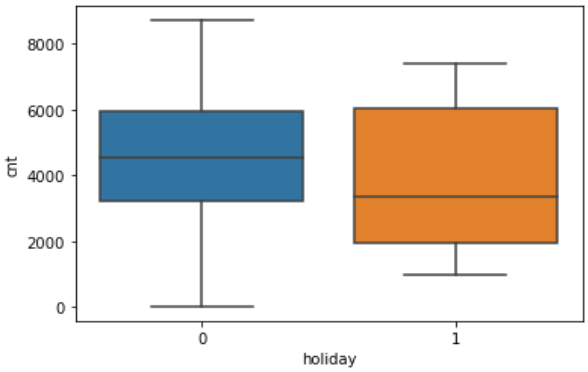







8000
6000
4000
2000
0
cnt
0
1
workingday


8000
6000
4000
2000
0
cnt
0
1
yr


8000
6000
4000
2000
0
cnt
0
1
holiday

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

The drop_first – reduces the no of columns that the dummy variable creates. Eg., let's say a column has 3 distinct values – say 1,2,3 for which we need to create a dummy variable. When we create dummy variables the values for these will be identified by 0 and 1

Dummy_1	Dummy_2	Dummy_3
0	1	0
1	0	0
0	0	1

So here we see that the values are represented by 0 and 1 against the value referring to it. So we can identify dummy 3 by just having the values for dummy_1 and dummy_2. So, this avoids redundancy as well as having multiple columns which might lead to multiple collinearities. So we use drop_first to remove the redundant column and have the dummy variables created to n-1 values to ensure the linearly independent.

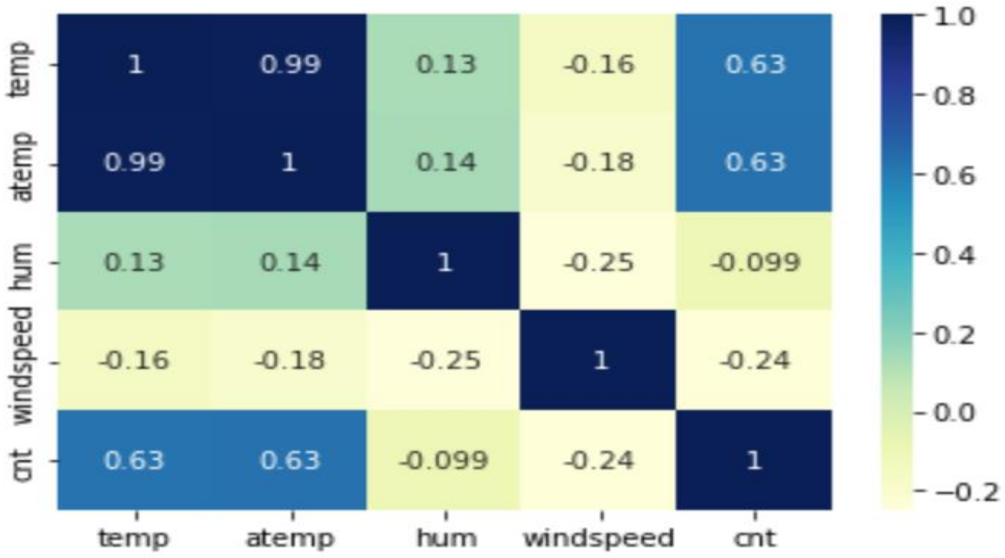
3. **Looking at the pair-plot among the numerical variables, which one has the highest correlation with the target variable?**

observations:

- It is found that temp and atemp have the highest correlation with cnt - bookings of the bike of 0.63.
- The pair plot looks like the temp and atemp follow the positive correlations against the cnt.

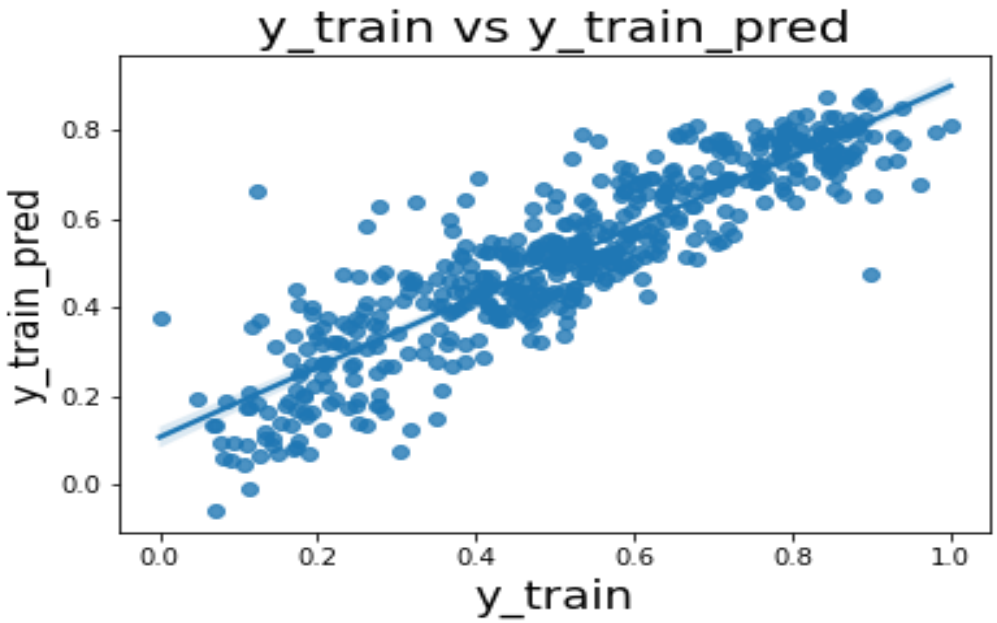
Code:

```
sns.heatmap(bikes[['temp','atemp','hum','windspeed','cnt']].corr(),annot=True,cmap='YlGnBu')
```



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

1. **Linearity:** There is a linear relationship between independent and dependent variables. When we built the model we see that model has a positive linear relationship and we can fit a straight line in the data points defined post-performing linear regression



If we check the above pair plot before linear regression we see that only temp and atemp parameters were showing linear shape where as others don't. So post model building our model shows linear behaviour post removing redundant and multicollinear features.

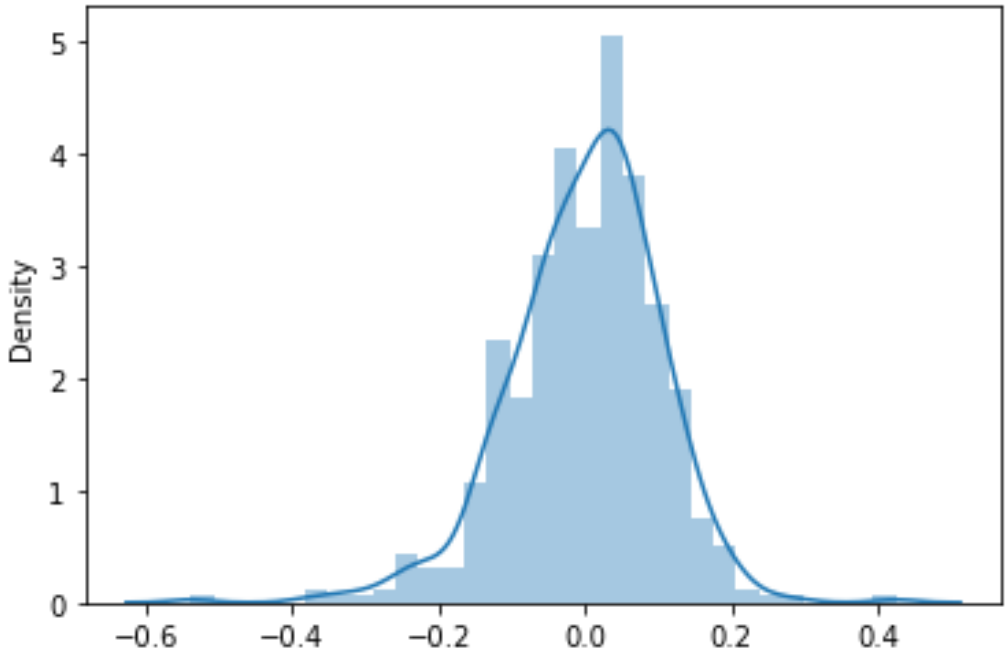
2. **Mean of residuals :** Error terms are normally distributed with mean close to 0. When we did the residual analysis we were able to get a normal distribution curve.

   Code:

   ```
   mean_res = np.mean(res)
   print("Mean of residuals is {}".format(mean_res))
   ```

   o/p

The mean of residuals is 1.3339003104687542e-16 → which is close to 0.



3. **No multicollinearity**: The VIF value is less than or equal to 5 and the p-value is less than 0.05. So there is no multicollinearity.

   VIF for the training set

```
In [343]: # calculate the vif for all the features
          vif = pd.DataFrame()
          vif['Features'] = X_train_rfe.columns
          vif['VIF'] = [variance_inflation_factor(X_train_rfe.values,i) for i in range(X_train_rfe.shape[1])]
          vif['VIF'] = round(vif['VIF'],2)
          vif = vif.sort_values(by = "VIF",ascending=False)
          vif
```

Out[343]:

	Features	VIF
2	windspeed	4.07
3	season_spring	2.96
5	season_winter	2.83
11	weathersit_clear	2.45
4	season_summer	2.03
0	yr	1.87
9	mnth_Nov	1.80
7	mnth_Jan	1.63
6	mnth_Dec	1.46
8	mnth_Jul	1.30
10	mnth_Sep	1.17
12	weathersit_light	1.12
1	holiday	1.06

4. **The observations are independent:** The value for Durbin-Watson is between 1.5 and 2.5 i.e it is 1.9 ~=2 so there is no autocorrelation and the observations are independent of each other.

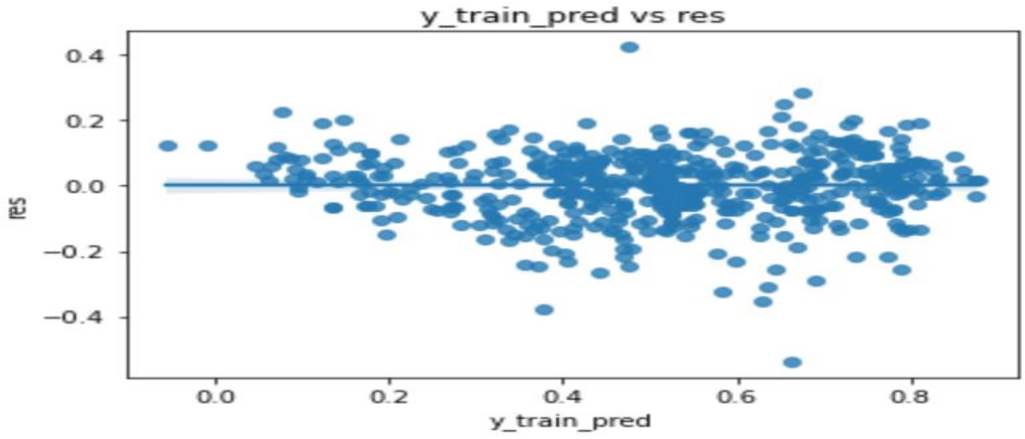
```
print(lr_model3.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    cnt   R-squared:                       0.792
Model:                            OLS   Adj. R-squared:                  0.786
Method:                 Least Squares   F-statistic:                     144.9
Date:                Wed, 10 Aug 2022   Prob (F-statistic):          1.82e-159
Time:                        00:12:12   Log-Likelihood:                 438.84
No. Observations:                 510   AIC:                            -849.7
Df Residuals:                     496   BIC:                            -790.4
Df Model:                          13
Covariance Type:            nonrobust
===========================================================================================
                              coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------
const                       0.5085      0.017     30.136      0.000       0.475       0.542
yr                          0.2457      0.009     26.616      0.000       0.228       0.264
holiday                    -0.0855      0.030     -2.887      0.004      -0.144      -0.027
windspeed                  -0.1902      0.029     -6.662      0.000      -0.246      -0.134
season_spring              -0.2503      0.018    -14.125      0.000      -0.285      -0.216
season_summer              -0.0497      0.016     -3.160      0.002      -0.081      -0.019
season_winter              -0.0231      0.018     -1.307      0.192      -0.058       0.012
mnth_Dec                   -0.1086      0.019     -5.660      0.000      -0.146      -0.071
mnth_Jan                   -0.1202      0.020     -6.056      0.000      -0.159      -0.081
mnth_Jul                   -0.0182      0.021     -0.875      0.382      -0.059       0.023
mnth_Nov                   -0.0995      0.021     -4.731      0.000      -0.141      -0.058
mnth_Sep                    0.0534      0.019      2.750      0.006       0.015       0.092
weathersit_clear            0.0869      0.010      8.800      0.000       0.068       0.106
weathersit_light           -0.2248      0.028     -7.911      0.000      -0.281      -0.169
==============================================================================
Omnibus:                       56.274   Durbin-Watson:                   1.943
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              132.272
Skew:                          -0.588   Prob(JB):                     1.89e-29
Kurtosis:                       5.200   Cond. No.                         10.0
==============================================================================
```

5. **Homoscedasticity**: Residuals have constant variance.

Code:

```
sns.regplot(y_train_pred,res)
plt.xlabel('y_train_pred')
plt.ylabel('res')
plt.title('y_train_pred vs res')
```



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

The top 3 significant features are as below

1. Year

2. spring season

3. When the weather condition is Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds

Below are the coefficients for them

yr 0.245724 → +ve relation

season_spring -0.250334 → -ve relation

weathersit_light -0.224786 → -ve relation.

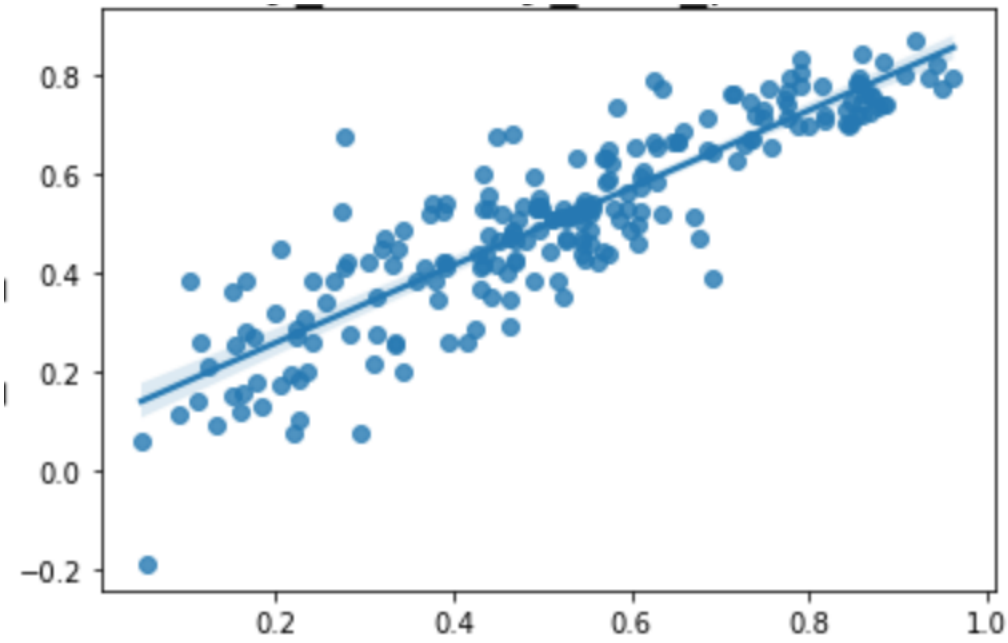
General Subjective Questions

1. Explain the linear regression algorithm in detail

Linear regression is a type of supervised machine learning algorithm that is used for analysing continuous variables.

Linear regression is used to check the relationship between the dependent and independent variables. It performs regression to predict the target variable (dependent variable) based on the independent variable. The linear regression will have a straight line for the data points that represent the relationship between dependent and independent variables.

Below is the sample for linear regression



Linear regression can be

a. Single.
   Single linear regression has only one independent variable that predicts the target variable / the relationship between one independent variable and the target variable. Below is the standard equation for simple linear regression.
   $Y=\beta 0+\beta 1X1+\epsilon$

b. Multiple

Multiple linear regression has many independent variables that predict the target variable/ there are multiple independent variables which have a relationship with the target variable.
Below is the standard equation for multiple linear regression.

$$Y=\beta 0+\beta 1X1+\beta 2X2+...+\beta pXp+\epsilon$$

$\beta 0$ – intercept (always a constant)
$\beta 1$ – regression coefficient (changes in the dependent variable corresponds to a unit change in the independent variable)
Y – target variable
ϵ - error or residuals (actual – predicted value)
Xn - independent variable.

A regression line can be positive linear relation or it can be negative linear relation. Depending on the $\beta 0$ if it is a positive or negative value.

Below are the steps to be followed for performing linear regression:

1. Understand the data
    a. Import the libraries (pandas,numpy,matplotlib,seaborn)
    b. Clean and manipulate the data of nulls, duplicate values, redundant columns, outliers, standardize the data etc.
2. Visualize the data (EDA) – univariate, bivariate and multivariate analysis.
    a. Numerical data – scatter plot, histograms etc
    b. Categorical data – box plot, count plot etc.
3. Data preparation:
    a. Create dummy variables for categorical data.
    b. Remove the redundant columns
4. Split the data to train and test data sets probably in a 70:30 ratio respectively.
5. Build a linear model on the training dataset.
    a. using modules
        i. Using statsmodel
        ii. Using sklearn
    b. You can perform feature selection either by

i. Adding one feature at a time to model and see how it behaves manually – forward selection
ii. Have all the features and remove them based on their p-value and vif manually- Backward selection
iii. You can use an automated method – RFE using linear regression.

You need to perform the step of feature addition/elimination until you have p values <0.5 and vif <5.

6. Residual analysis of the train data set – RSS should be equal to mean of 0 indicating normal distribution.
7. Make prediction and evaluation of the test data set.
    a. Using the test dataset we use the trained model to see if the test dataset shows the same behaviour as the training data set.
    b. We compare the r2_score between the test and train data set and if it lies within the 2 – 3% difference max we say that the model is acceptable.

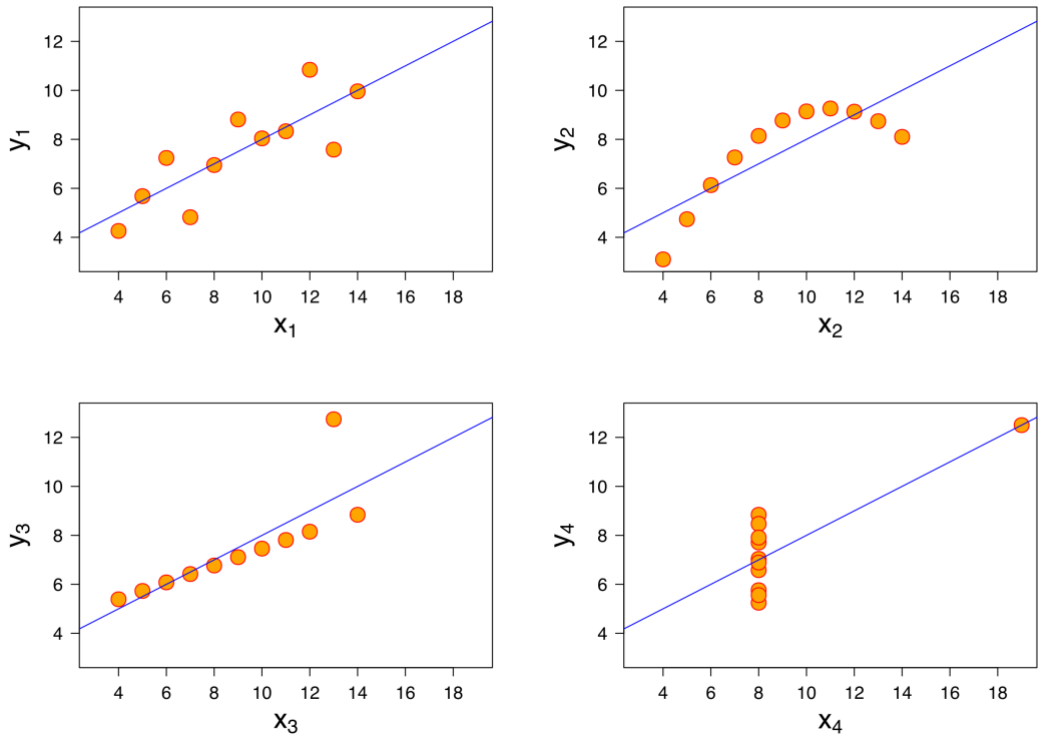
2. Explain Anscombe's quartet in detail.

Anscombe's quartet is a collection of 4 datasets which are nearly identical to each other statistically (they have the same mean, sd, variance, correlation values etc). However when plotted and when they look different.

For eg. ,

Below when you look at the table you see that set I, II, III and IV have the same values for mean, sd, variance, and correlation. But when you look at the plot you see the difference

	I		II		III		IV	
	x	*y*	*x*	*y*	*x*	*y*	*x*	*y*
	10	8.04	10	9.14	10	7.46	8	6.58
	8	6.95	8	8.14	8	6.77	8	5.76
	13	7.58	13	8.74	13	12.74	8	7.71
	9	8.81	9	8.77	9	7.11	8	8.84
	11	8.33	11	9.26	11	7.81	8	8.47
	14	9.96	14	8.1	14	8.84	8	7.04
	6	7.24	6	6.13	6	6.08	8	5.25
	4	4.26	4	3.1	4	5.39	19	12.5
	12	10.84	12	9.13	12	8.15	8	5.56
	7	4.82	7	7.26	7	6.42	8	7.91
	5	5.68	5	4.74	5	5.73	8	6.89
Mean	9.00	7.50	9.00	7.50	9.00	7.50	9.00	7.50
Standard deviation	3.16	1.94	3.16	1.94	3.16	1.94	3.16	1.94
Variance	1.78	1.39	1.78	1.39	1.78	1.39	1.78	1.39
Correlation	0.82		0.82		0.82		0.82	



X1 – plot shows the linear relationship

X2 – The plot shows a non-linear relationship

X3 – The plot show a linear relationship however when you look at the data points it falls almost on the straight line except for one point which can be an outlier

X4: plot where the line does not pass through all the data points.

3. What is Pearson's R?

Pearson's R is also known as the coefficient of determination. Which determines the relationship of the data. It can be positive, negative or no correlations at all.

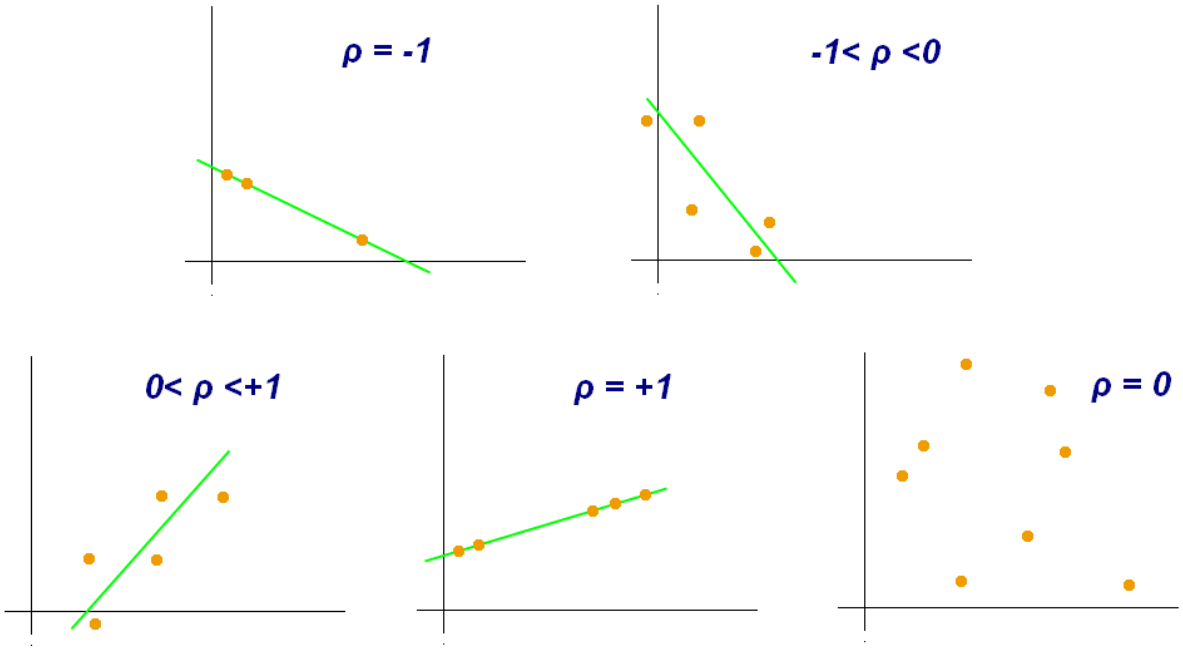
The value for R lies between -1 to +1

+1 indicates positive relation.

-1 indicates negative relation

0 indicates no relationship.

For eg.



R-square = 1 – (RSS/TSS)

Here the 1st two graphs show a negative relation

The 3rd and 4th graphs show a positive relation

The 5th graph shows no relation.

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

Scaling is the process of normalizing or standardizing the data for the features. In a given dataset you might find multiple independent numerical features having varying data distribution (df. describe()) that can have very high magnitudes compared to others. This will lead to incorrect modelling as it affects the coefficients of independent variables. To overcome this we use scaling which ensures that all the values are in the same scale or range.

A. Normalized scaling – is also called MinMaxScaler where the data is in the range of min and max values i.e equal to 0 and 1 respectively

MinMaxScaling = (x – min(x))/(max(x) – min(x))

B. Standard scaling – here the data is in the range of mean and standard deviation.
Standard Scaling = (x – mean(x))/sd(x)

For eg.

In our bike sharing app – we see that temp,atemp,hum and windspeed had high data distribution compared to other columns so applying MinMaxScaler ensures that they are in same scale.

```
In [152]: # check the data distribution
          bikes.describe()
```

Out[152]:

	yr	holiday	workingday	temp	atemp	hum	windspeed	cnt	season_spring	season_summer	...
count	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	...
mean	0.500000	0.028767	0.690411	20.319259	23.726322	62.765175	12.763620	4508.006849	0.246575	0.252055	...
std	0.500343	0.167266	0.462641	7.506729	8.150308	14.237589	5.195841	1936.011647	0.431313	0.434490	...
min	0.000000	0.000000	0.000000	2.424346	3.953480	0.000000	1.500244	22.000000	0.000000	0.000000	...
25%	0.000000	0.000000	0.000000	13.811885	16.889713	52.000000	9.041650	3169.750000	0.000000	0.000000	...
50%	0.500000	0.000000	1.000000	20.465826	24.368225	62.625000	12.125325	4548.500000	0.000000	0.000000	...
75%	1.000000	0.000000	1.000000	26.880615	30.445775	72.989575	15.625589	5966.000000	0.000000	1.000000	...
max	1.000000	1.000000	1.000000	35.328347	42.044800	97.250000	34.000021	8714.000000	1.000000	1.000000	...

8 rows × 30 columns

We see that except temp,atemp,hum,windspeed and cnt all other numerical variables have the min as 0 and max as 1 in the data distribution

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

When the R-square value is 1 the value of VIF will be equal to infinite.
VIF = 1/(1-R-square^2)
VIF = 1/(1-1)
VIF = 1/0
VIF = inifinite
We get R-square as 1 when there is a 100% correlation between independent variables.
An infinite VIF means that a given independent variable can perfectly determine the other variables in the model. This leads to a multicollinearity problem. So we need to drop one of the variables from the dataset.

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

Q – Q plots are called quantile-quantile plots. As the name suggests it is a plot of 2 quantiles. Quantiles divide the sorted data into 4 equal parts (0-25,25 - 50,50 – 75,75 to max)
Here 50% is the median.
Q – Q plots are used to find if two sets of data come from a common distribution (normal/uniform/exponential etc) graphically.

It also helps to identify if the data sets are from populations having common statistical distribution.

We can use the Q -Q plot in linear regression to identify the training and test data to determine if they are from the same population or not and have the same statistical distribution.

It can be used to get below insights about the data:

a. if they are from the same population with the same distribution.
b. If they use the same scale and location for the data.
c. If they have similar distribution shapes – like linear, uniform, exponential.
d. They follow the same data distribution pattern like skewness, left skewed or right skewed.