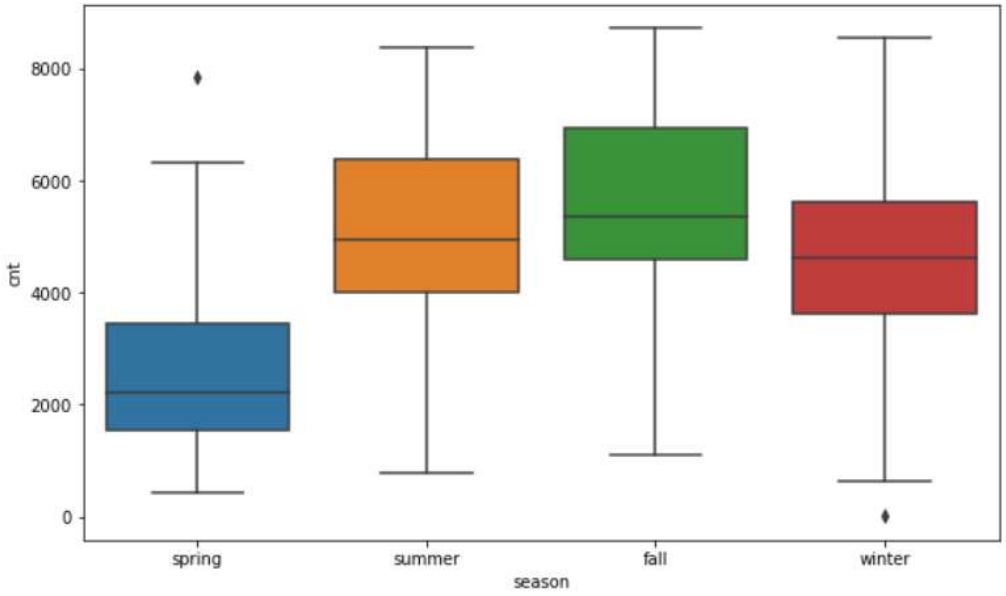
1. From your analysis of the categorical variables from the dataset, what could you infer about their effect on the dependent variable?

We have total 7 categorical variables in the dataset we have performed some Exploratory Data Analysis (Univariate and Segmented Univariate analysis) on the categorical variables to understand the their effects on the bike demands.

Please find our analysis as follows

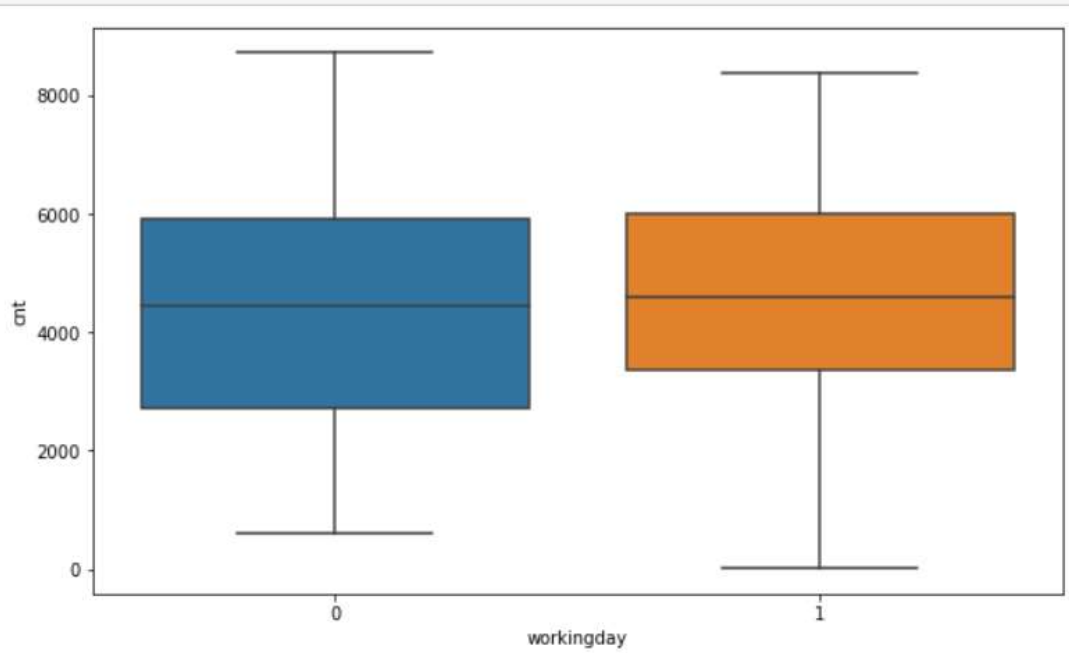
Season:

In spring season count has significant low value, while fall season is most favorable with summer being the next then winter.



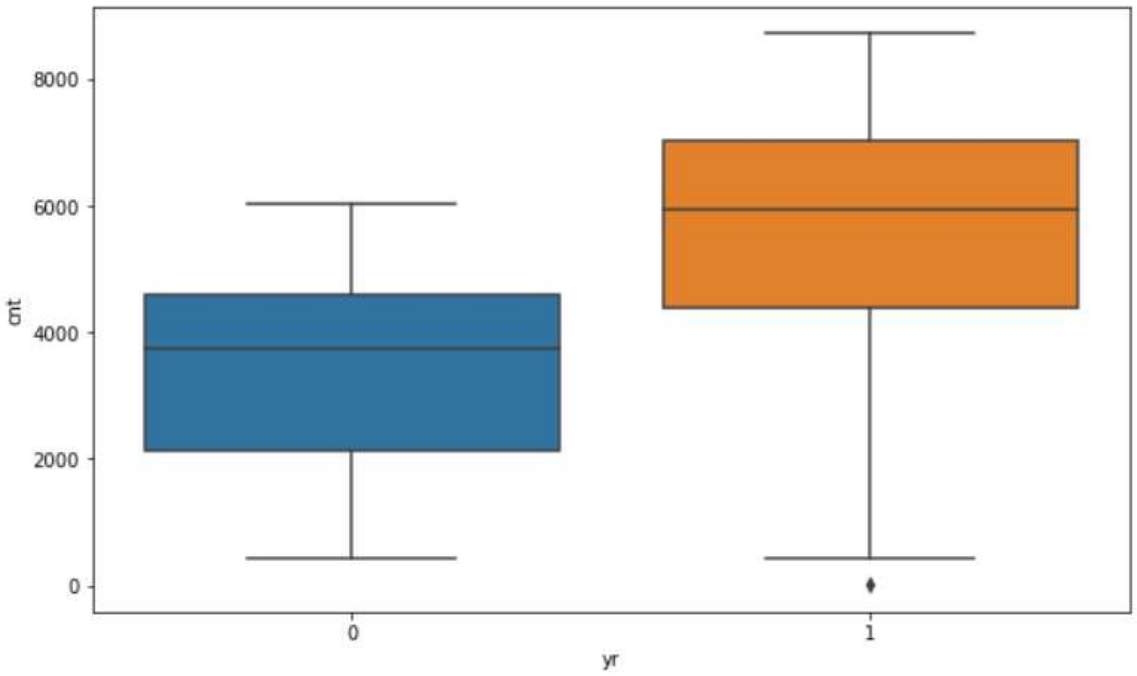
Workingday:

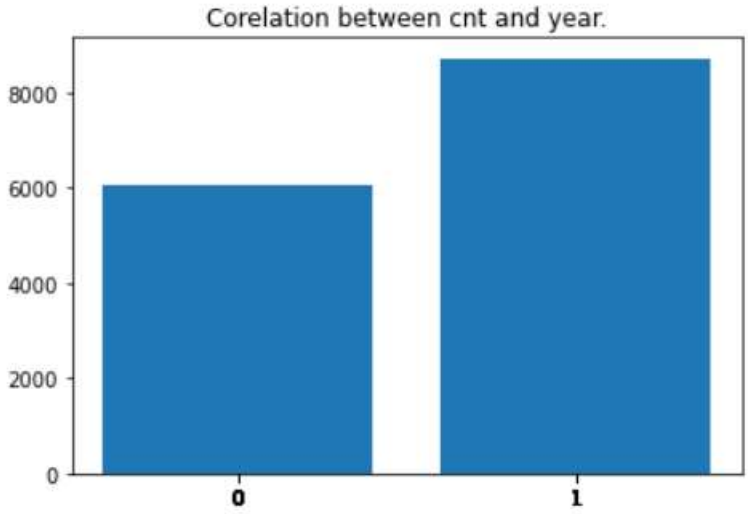
The workingday and non-workingday have similar mean in terms of demand. However there is a low min value for the working day



Yr (0: 2018, 1:2019):

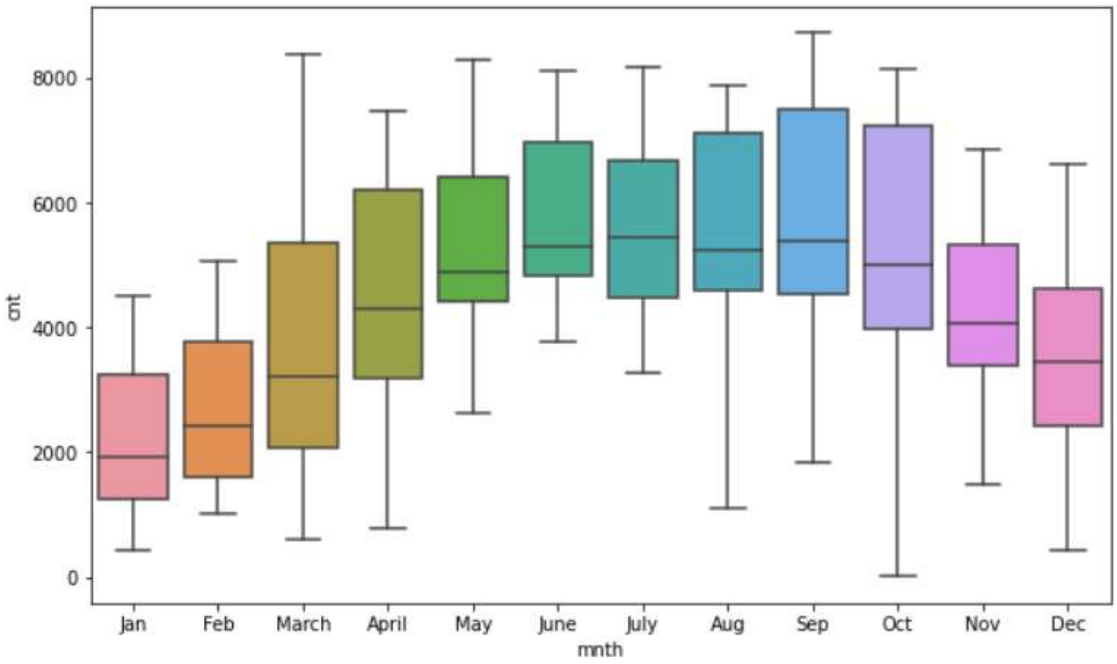
There was a clear surge in demand from 2018 to 2019. Bike sharing was getting popular during pre-covid time.¶





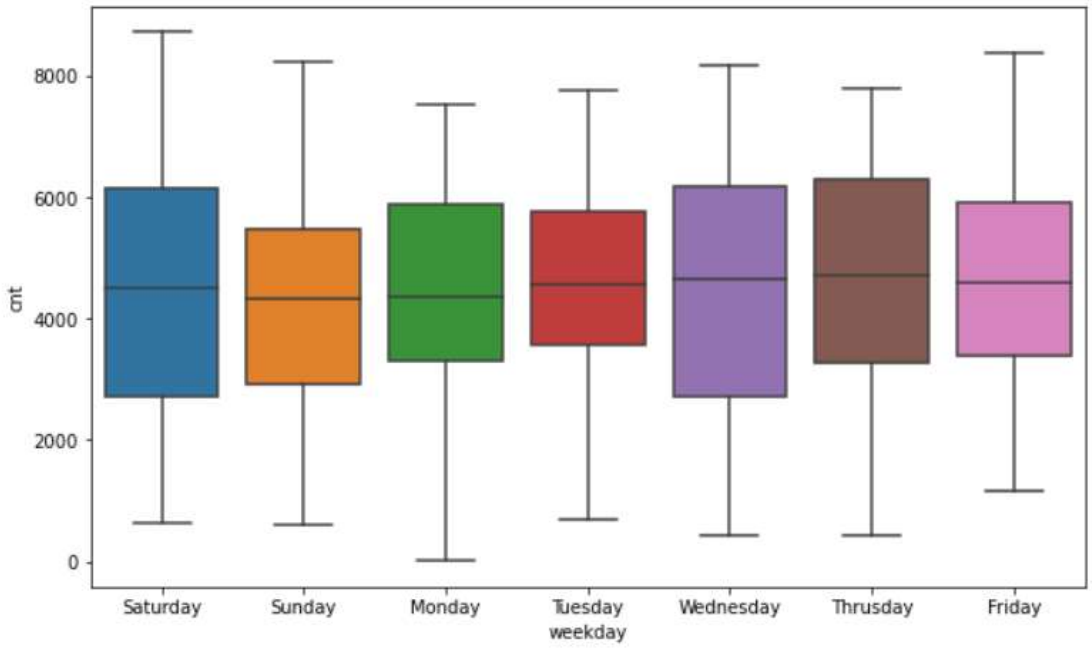
mnth:

May to Oct has high median while in Jan it is lowest then it is gradually increasing till July then again going down in winter months. It means cnt increase in higher temperature months



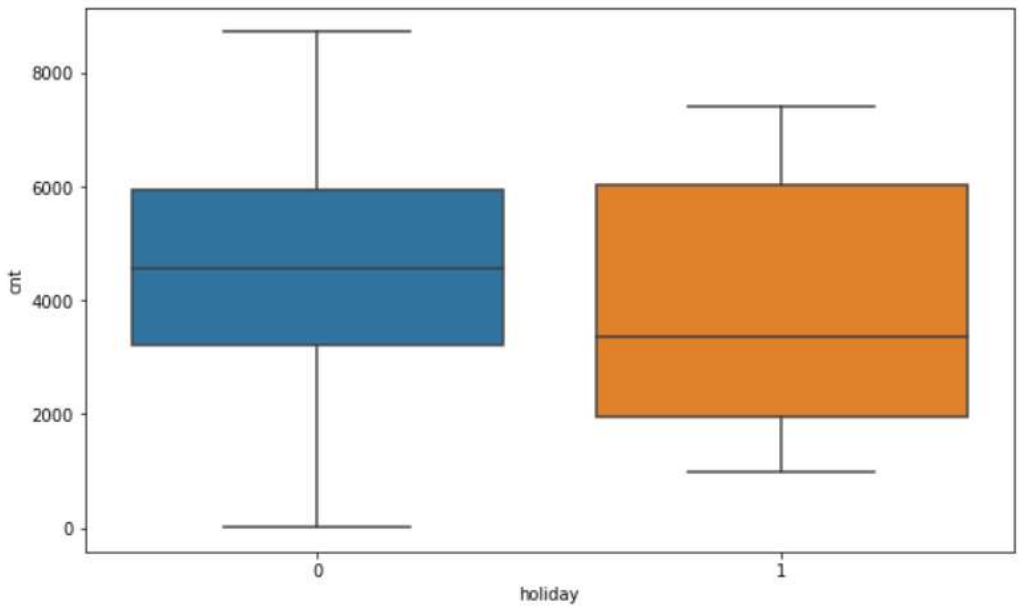
Weekday:

All Weekdays has fairly similar median, there is no significant difference between them. Monday has Low Min value and Low Max Value than others.



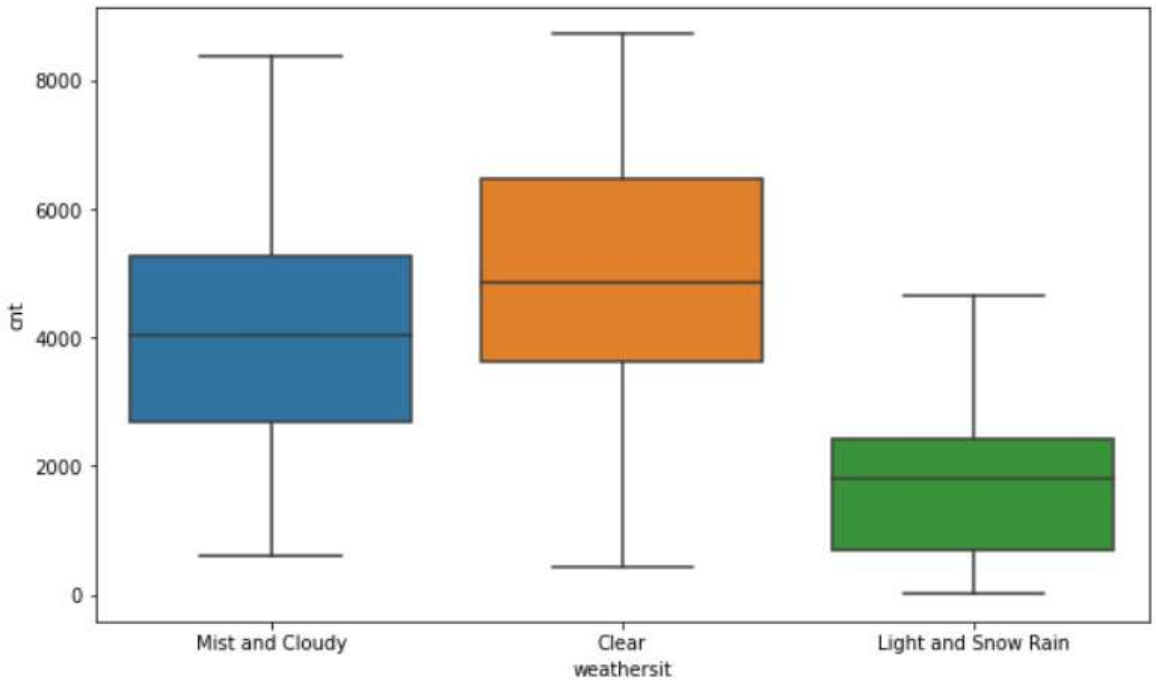
Holiday

cnt median is low when there is a holiday with large range on working days it has high median and low range



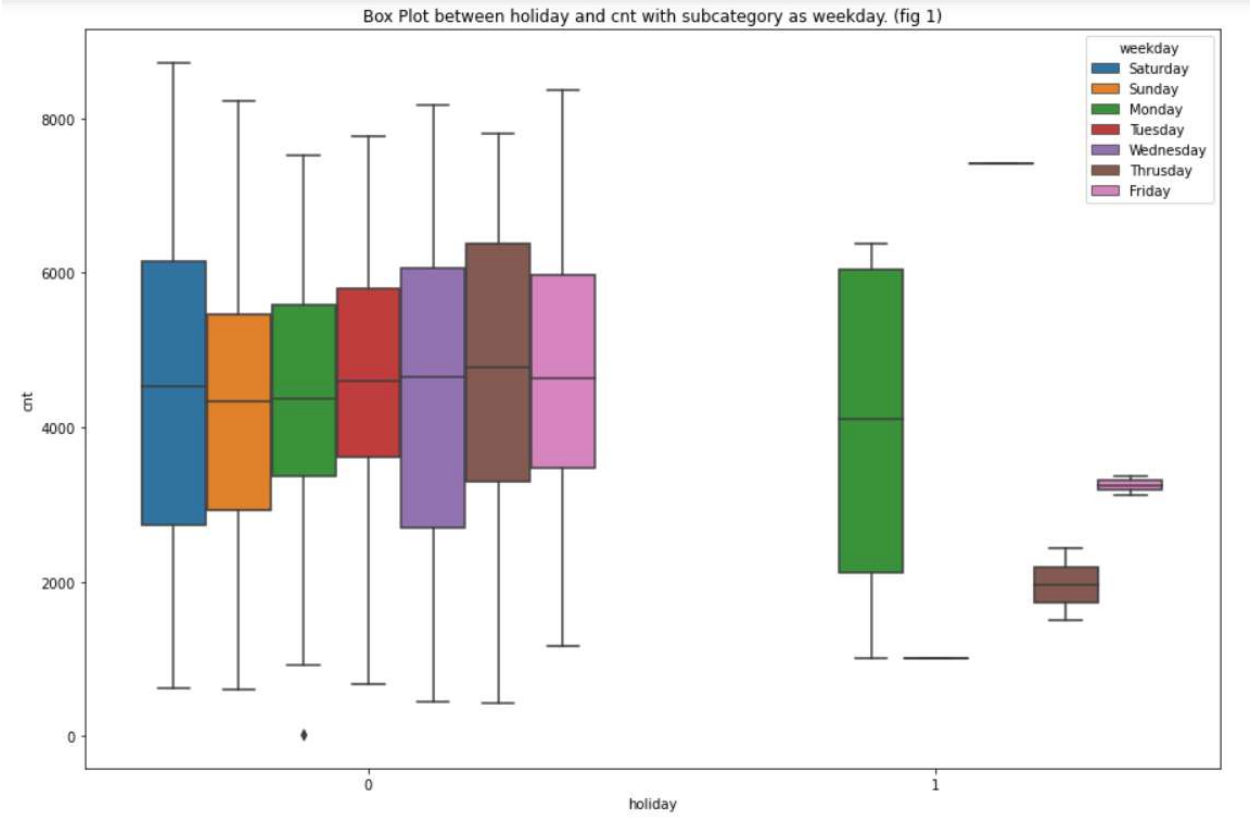
Weathersit

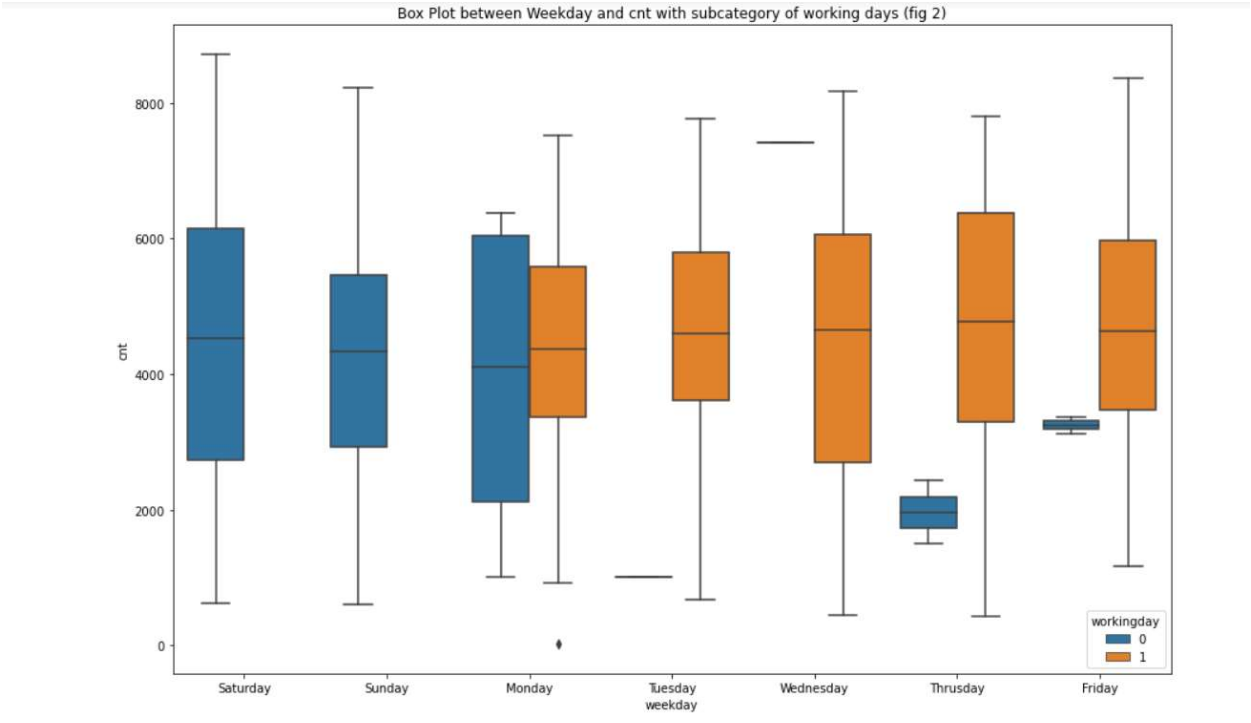
- Clear Weather has highest count, which means bike sharing count is high in clear weather.
- While its quite low in Light and Snow Rain weather, which is quite understandable.
- Mist and Cloudy weather has fairly high count and higher median but it still less than Clear weather.

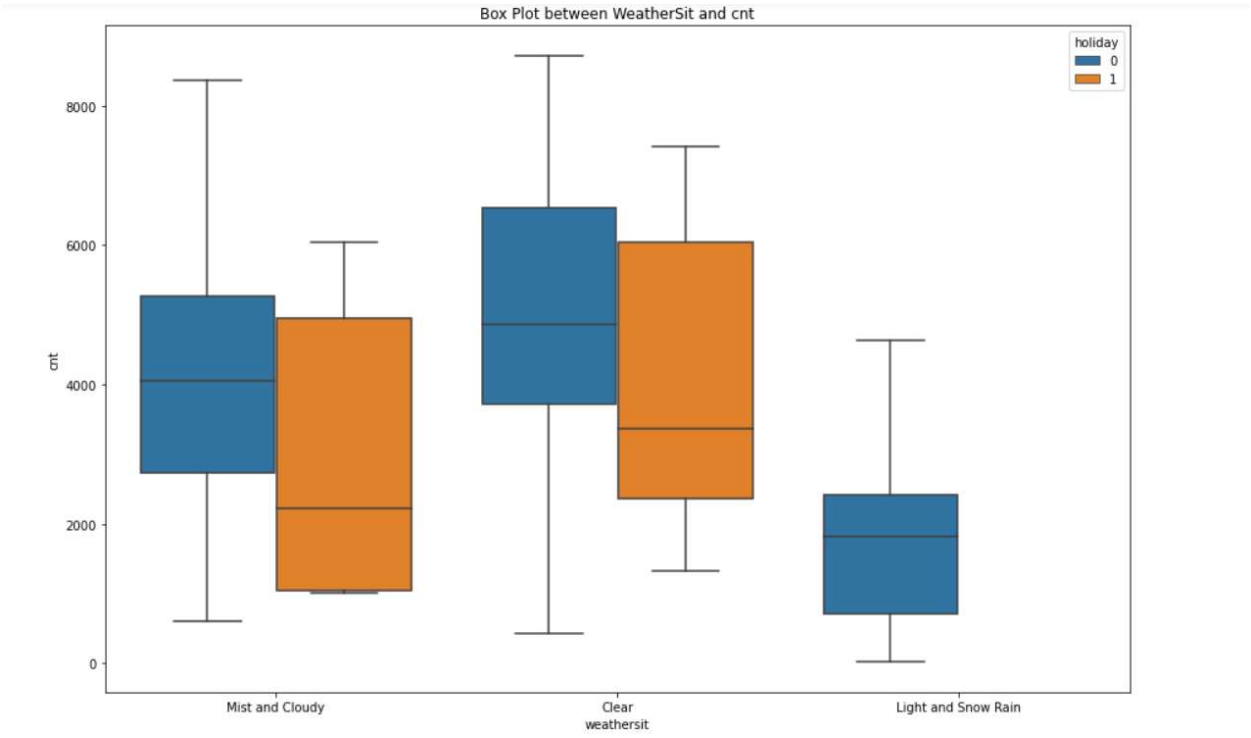


Further analysis - for workingday and holiday

- Monday appears to be a partial holiday from the analysis of fig 2 below and it has a higher range
- Thursdays and Fridays have some holidays too.
- The demand is higher during working days and highest demand happens when weather is nice. We can infer that people share bikes for commuting to work.


Box Plot between holiday and cnt with subcategory as weekday. (fig 1)
weekday
Saturday
Sunday
Monday
Tuesday
Wednesday
Thrusday
Friday
cnt
8000
6000
4000
2000
0
0
1
holiday


Box Plot between Weekday and cnt with subcategory of working days (fig 2)
cnt
8000
6000
4000
2000
0
workingday
0
1
Saturday
Sunday
Monday
Tuesday
Wednesday
Thrusday
Friday
weekday



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

From the official documentation for pandas, we find the following definition

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get_dummies.html

drop_first: bool, default False

Whether to get k-1 dummies out of k categorical levels by removing the first level.

By default, the get_dummies() does **not** do dummy encoding, but one-hot encoding.

Some machine learning techniques require you to drop one dimension from the representation so as to avoid dependency among the variables. Use "drop_first=True" to achieve that.

If we don't drop the first column then your dummy variables will be correlated.

This may affect some models adversely and the effect is stronger when the cardinality is smaller. For example iterative models may have trouble converging and lists of variable importance may be distorted.

If you have a small number of dummies, we suggest removing the first dummy. For example, if you have a variable gender, you don't need both a male and female dummy. Just one will be fine. If male=1 then the person is a male and if male=0 then the person is female. However if you have a category with hundreds of values, we suggest not dropping the first column. That will make it easier for the model to "see" all the categories quickly during learning (and the adverse effects are negligible).

Example with Pandas –

	country
0	russia
1	germany
2	australia
3	korea
4	germany

To produce an actual dummy encoding from your data, use drop_first=True (not that 'australia' is missing from the columns)

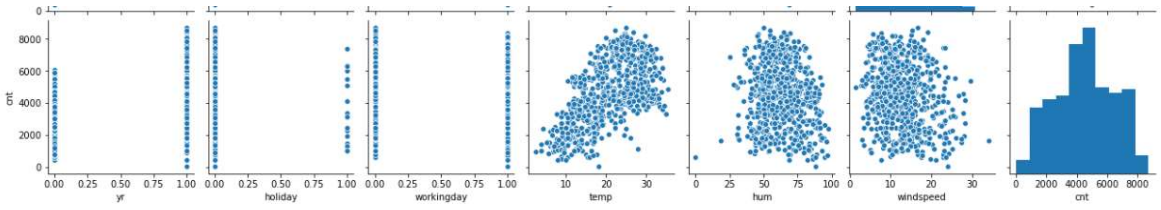
In [1]:
```
import pandas as pd
# using the same example as above
df = pd.DataFrame({'country': ['russia', 'germany', 'australia','korea','germany']})
pd.get_dummies(df["country"],prefix='country',drop_first=True)
```

Out[1]:

	country_germany	country_korea	country_russia
0	0	0	1
1	1	0	0
2	0	0	0
3	0	1	0
4	1	0	0

Now we see the Australia is gone!

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



Temp has the highest correlation with the target variable, followed by yr and windspeed.

Find the correlation with Target Variable

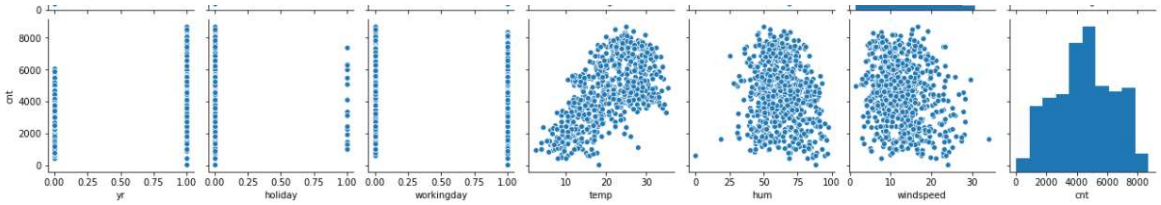
In [48]:
```
np.abs(df_bikes.corr().cnt).sort_values(ascending=False)
```

Out[48]:
```
cnt           1.000000
temp          0.627044
yr            0.569728
windspeed     0.235132
hum           0.098543
holiday       0.068764
workingday    0.062542
Name: cnt, dtype: float64
```

In [49]:
```
## Let's draw a heatmap for the data visualization
```

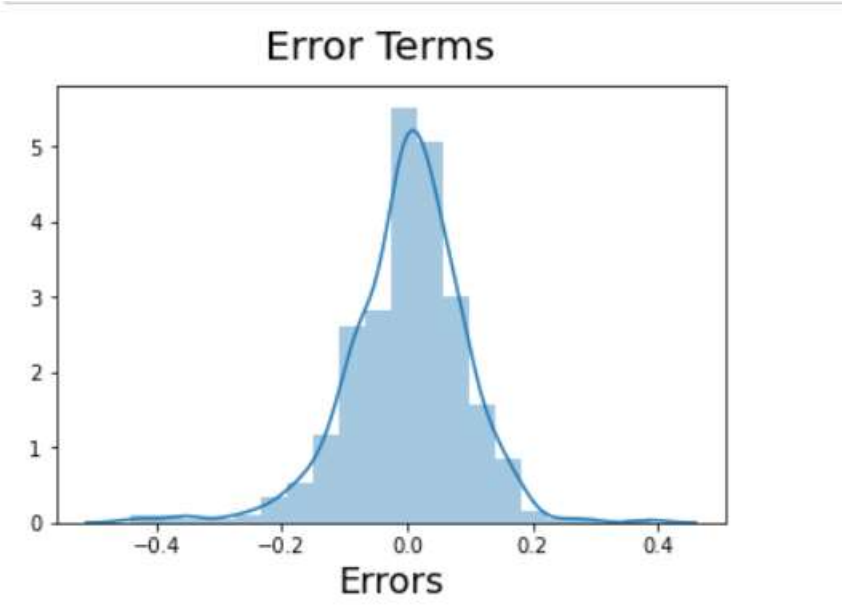
4. How did you validate the assumptions of Linear Regression after building the model on the training set?

Linearity: Linear regression needs the relationship between the independent and dependent variables to be linear. Let's check the pair plot between the cnt and the other independent variables



The demand is pretty much linearly correlated with temp, windspeed, humidity, year etc.

Check for Normality of Residuals: The error terms are normally distributed for the number of samples we took.

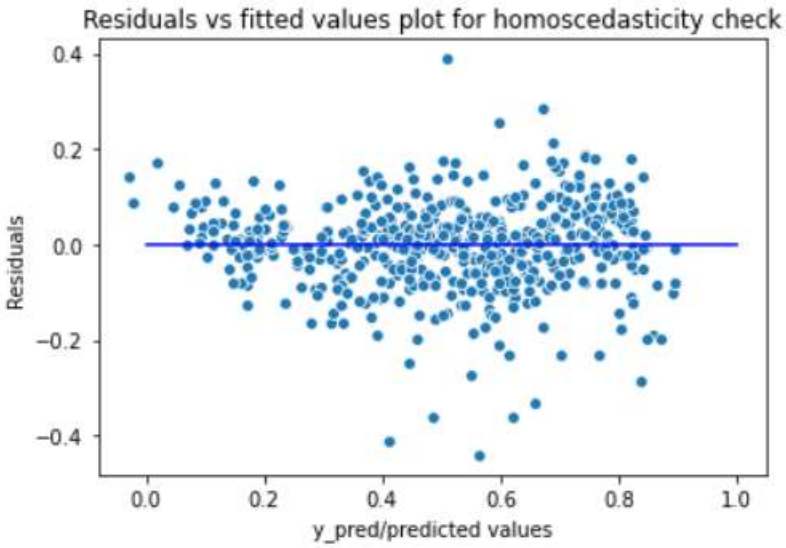


Mean of Residuals: Residuals as we know are the differences between the true value and the predicted value. One of the assumptions of linear regression is that the mean of the residuals should be zero.

In our case the Mean of Residuals are very close to 0.

```
]: res=y_train - y_train_cnt
   mean_residuals = np.mean(res)
   print("Mean of Residuals {}".format(mean_residuals))

   Mean of Residuals -3.2676622877421435e-16
```

Check for Homoscedasticity: Homoscedasticity means that the residuals have equal or almost equal variance across the regression line. By plotting the error terms with predicted terms we can check that there is no particular pattern for the error terms.



Hypothesis Testing for Homoscedasticity

Checking heteroscedasticity: Using Goldfeld Quandt we test for heteroscedasticity.

-Null Hypothesis: Error terms are homoscedastic

-Alternative Hypothesis: Error terms are heteroscedastic.

Please refer to

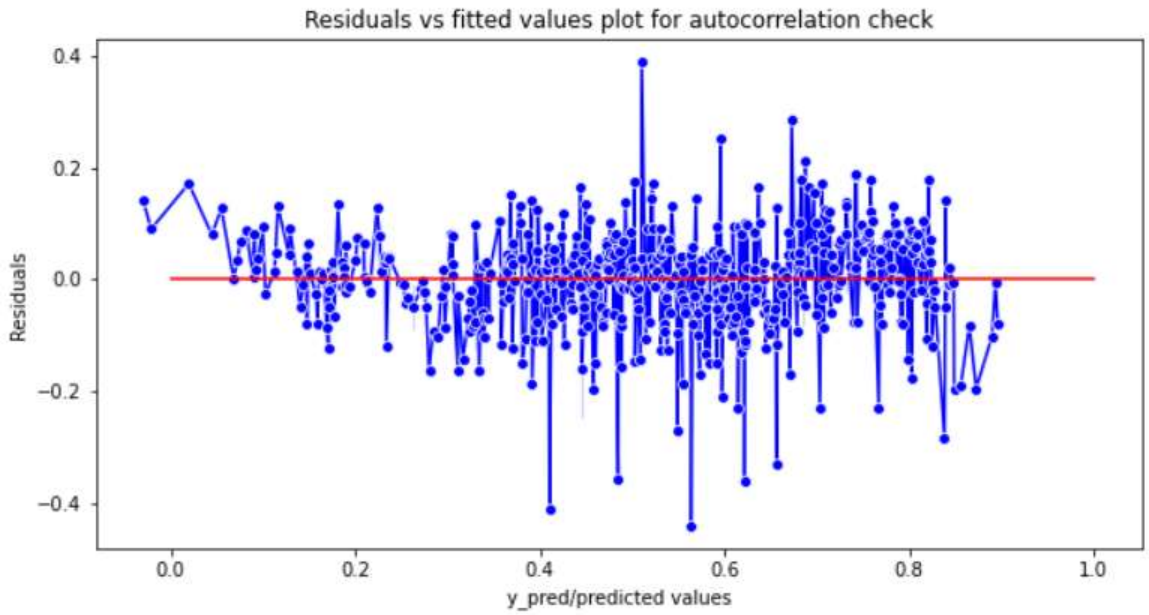
- https://www.statsmodels.org/devel/examples/notebooks/generated/regression_diagnostics.html
- https://www.youtube.com/watch?v=yb4CIJzftjc

```
In [92]: import statsmodels.stats.api as sms
         from statsmodels.compat import lzip
         name = ['F statistic', 'p-value']
         test = sms.het_goldfeldquandt(res, X_train_rfe1)
         lzip(name, test)

Out[92]: [('F statistic', 1.1070554515955917), ('p-value', 0.21351777812673153)]
```

From the Hypothesis testing we cannot reject the null hypothesis as the p-value is higher than 0.05. So this model is Homoscedastic

No autocorrelation of residuals: When the residuals are auto correlated, it means that the current value is dependent of the previous (historic) values and that there is a definite unexplained pattern in the target variable that shows up in the error terms.

We can clearly see that from the plot there is not definite pattern exists among the residuals.



Checking for autocorrelation to ensure the absence of autocorrelation we use durbin_watson test.

Please have a look at
https://www.statsmodels.org/stable/generated/statsmodels.stats.stattools.durbin_watson.html

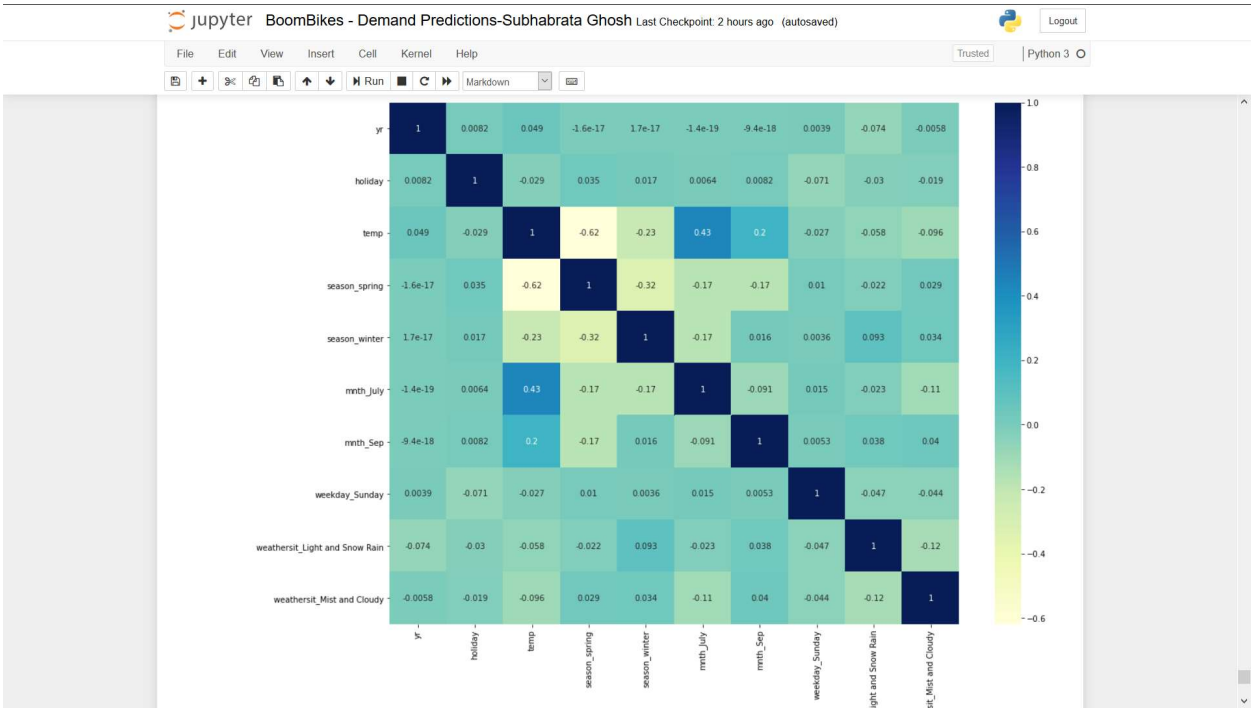
```
In [108]: from statsmodels.stats import diagnostic as diag
          from statsmodels.stats import stattools
          ##min(diag.acorr_ljungbox(res , lags = 40)[1])
          stattools.durbin_watson(res)

Out[108]: 2.025414880189178
```

The test statistic is approximately equal to 2*(1-r) where r is the sample autocorrelation of the residuals. Thus, for r == 0, indicating no serial correlation, the test statistic equals 2. This statistic will always be between 0 and 4. The closer to 0 the statistic, the more evidence for positive serial correlation. The closer to 4, the more evidence for negative serial correlation.

We have found that the durbin_watson value close to 2. Hence we can assume that there is not autocorrelation.

No perfect multicollinearity: In regression, multicollinearity refers to the extent to which independent variables are correlated. Multicollinearity affects the coefficients and p-values, but it does not influence the predictions, precision of the predictions, and the goodness-of-fit statistics

We don't have any perfect multicollinearity in our model.



The Variance Inflation Factor (VIF**) is 1/Tolerance, it is always greater than or equal to 1. There is no formal** VIF value **for determining presence of** multicollinearity**.** Values **of** VIF **that exceed 10 are often regarded as** indicating multicollinearity**, but in weaker models** values **above 5 may be a cause for concern.**

All the VIF values are <5

	Features	VIF
2	temp	3.09
0	yr	2.05
9	weathersit_Mist and Cloudy	1.51
4	season_winter	1.35
5	mnth_July	1.33
3	season_spring	1.27
6	mnth_Sep	1.19
7	weekday_Sunday	1.17
8	weathersit_Light and Snow Rain	1.07
1	holiday	1.05

All the assumptions for the Linear Regression have been satisfied by our model.

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

Our final model lm7

Final Model - lm7

```
print(lm7.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    cnt   R-squared:                       0.828
Model:                            OLS   Adj. R-squared:                  0.824
Method:                 Least Squares   F-statistic:                     239.9
Date:                Thu, 27 Aug 2020   Prob (F-statistic):          1.19e-183
Time:                        02:01:05   Log-Likelihood:                 487.32
No. Observations:                 511   AIC:                            -952.6
Df Residuals:                     500   BIC:                            -906.0
Df Model:                          10
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                            0.2026      0.022      9.227      0.000       0.159       0.246
yr                               0.2336      0.008     27.742      0.000       0.217       0.250
holiday                         -0.1088      0.027     -4.064      0.000      -0.161      -0.056
temp                             0.4689      0.031     15.182      0.000       0.408       0.530
season_spring                   -0.1113      0.015     -7.186      0.000      -0.142      -0.081
season_winter                    0.0569      0.013      4.537      0.000       0.032       0.082
mnth_July                       -0.0669      0.018     -3.766      0.000      -0.102      -0.032
mnth_Sep                         0.0639      0.016      3.974      0.000       0.032       0.095
weekday_Sunday                  -0.0483      0.012     -4.027      0.000      -0.072      -0.025
weathersit_Light and Snow Rain  -0.3052      0.025    -12.100      0.000      -0.355      -0.256
weathersit_Mist and Cloudy      -0.0806      0.009     -8.979      0.000      -0.098      -0.063
==============================================================================
Omnibus:                       69.328   Durbin-Watson:                   2.025
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              200.361
Skew:                          -0.648   Prob(JB):                     3.11e-44
Kurtosis:                       5.781   Cond. No.                         13.3
==============================================================================
```

The top 3 features contributing towards explaining bike demands are

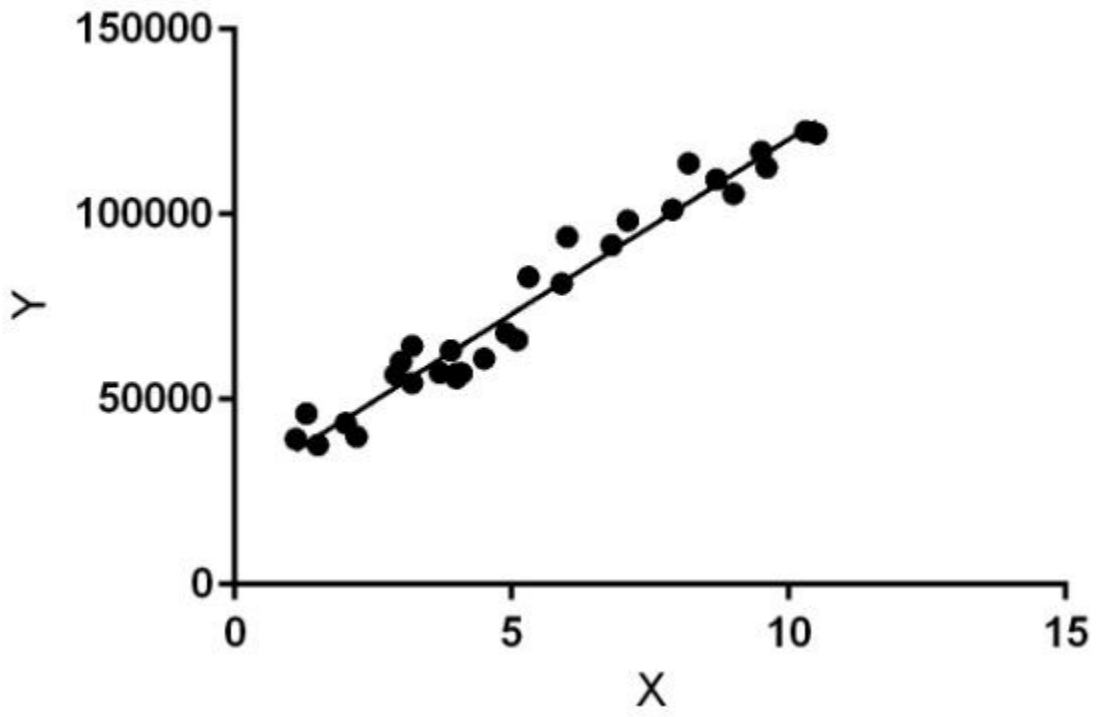
- temp : temperature in Celsius
- weathersit_Light and Snow Rain (Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds)
- yr : year (0: 2018, 1:2019)

General Subjective Question:

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 being used.



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 θ_1 and θ_2 values.

θ_1: intercept
θ_2: coefficient of x

Once we find the best θ_1 and θ_2 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.

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 θ_1 and θ_2 values, to reach the best value that minimize the error between predicted y value (pred) and true y value (y).

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

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

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:

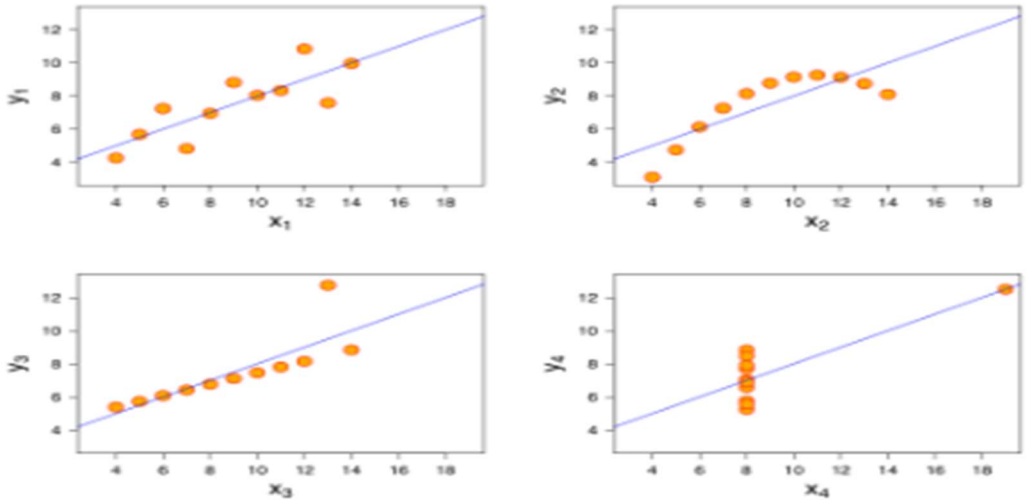
To update θ_1 and θ_2 values in order to reduce Cost function (minimizing RMSE value) and achieving the best fit line the model uses Gradient Descent. The idea is to start with random θ_1 and θ_2 values and then iteratively updating the values, reaching minimum cost.

Explain the Anscombe's quartet in detail.

It's a group of four datasets that appear to be similar when using typical summary statistics, yet tell four different stories when graphed. Each dataset consists of eleven *(x, y)* pairs.

All the summary statistics we think to compute are close to identical:

1. The average *x* value is 9 for each dataset
2. The average *y* value is 7.50 for each dataset
3. The variance for *x* is 11 and the variance for *y* is 4.12
4. The correlation between *x* and *y* is 0.816 for each dataset
5. A linear regression (line of best fit) for each dataset follows the equation $y = 0.5x + 3$.
6. So far these four datasets appear to be pretty similar. But when we plot these four data sets on an x/y coordinate plane, we get the following results



Now we see the real relationships in the datasets start to emerge. Dataset I consists of a set of points that appear to follow a rough linear relationship with some variance. Dataset II fits a neat curve but doesn't follow a linear relationship (maybe its quadratic?). Dataset III looks like a tight linear relationship between x and y, except for one large outlier. Dataset IV looks like x remains constant, except for one outlier as well.

Computing summary statistics or staring at the data wouldn't have told us any of these stories. Instead, it's important to visualize the data to get a clear picture of what's going on.

What is Pearson's R?

Pearson's correlation coefficient (r) is a measure of the strength of the association between the two variables.

The first step in studying the relationship between two continuous variables is to draw a scatter plot of the variables to check for linearity. The correlation coefficient should not be calculated if the relationship is not linear. For correlation only purposes, it does not really matter on which axis the variables are plotted. However, conventionally, the independent (or explanatory) variable is plotted on the x-axis (horizontally) and the dependent (or response) variable is plotted on the y-axis (vertically).

The nearer the scatter of points is to a straight line, the higher the strength of association between the variables. Also, it does not matter what measurement units are used.

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

Scaling is the process of bringing all the variables to a same scale. Scaling is performed mostly during model building processes to bring everything to the same scale.

In normalized scaling, we use the maximum and the minimum values of a particular column to perform the scaling. For any data point 'X' in a column 'C', this scaling is performed using the formula: X – min(C)/max(C) – min(C).

Standardized scaling, on the other hand, brings all the data points in a normal distribution with mean zero and standard deviation one. It is performed using the formula: X – mean(C)/SD(C)

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

VIF = Variance Inflation Factor

VIF = 1/ (1-R^2)

When we are calculating the VIF for one independent variable using all the other independent variables, if the R^2 value comes out to be 1, the VIF will become infinite. This happens when an independent variable is strongly correlated with many other independent variables.

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

Quantile-Quantile (Q-Q) plot, is a graphical tool to help us assess if a set of data plausibly came from some theoretical distribution such as a Normal, exponential or uniform distribution. Also, it helps to determine if two data sets come from populations with a common distribution.

This helps in a scenario of linear regression when we have training and test data set received separately and then we can confirm using Q-Q plot that both the data sets are from populations with same distributions.

Few advantages:

a) It can be used with sample sizes also

b) Many distributional aspects like shifts in location, shifts in scale, changes in symmetry, and the presence of outliers can all be detected from this plot.

It is used to check following scenarios:

If two data sets —

- come from populations with a common distribution
- have common location and scale
- have similar distributional shapes
- have similar tail behavior

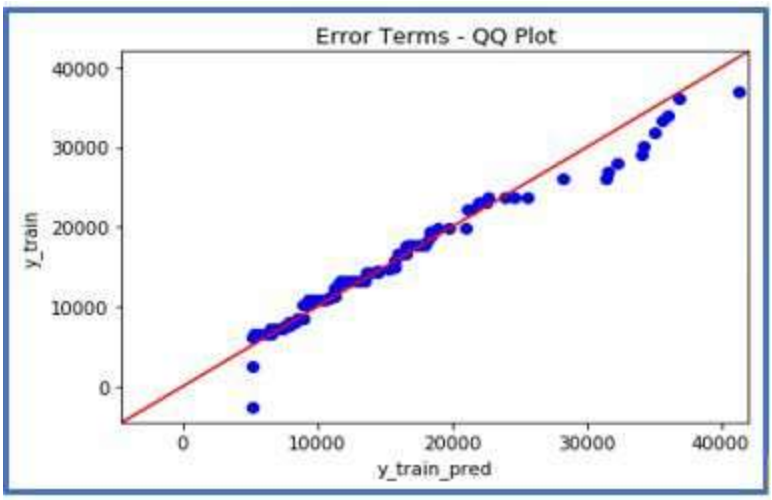
Interpretation:

A q-q plot is a plot of the quintiles of the first data set against the quintiles of the second data set.

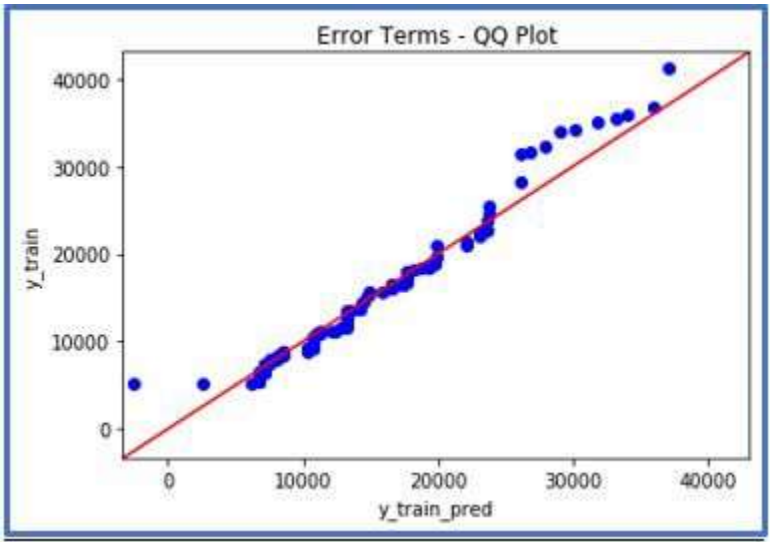
Below are the possible interpretations for two data sets.

a) Similar distribution: If all point of quintiles lies on or close to straight line at an angle of 45 degree from x -axis

b) Y-values < X-values: If y-quintiles are lower than the x-quintiles.



c) X-values < Y-values: If x-quintiles are lower than the y-quintiles.



d) Different distribution: If all point of quintiles lies away from the straight line at an angle of 45 degree from x -axis