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?*

Seasons:

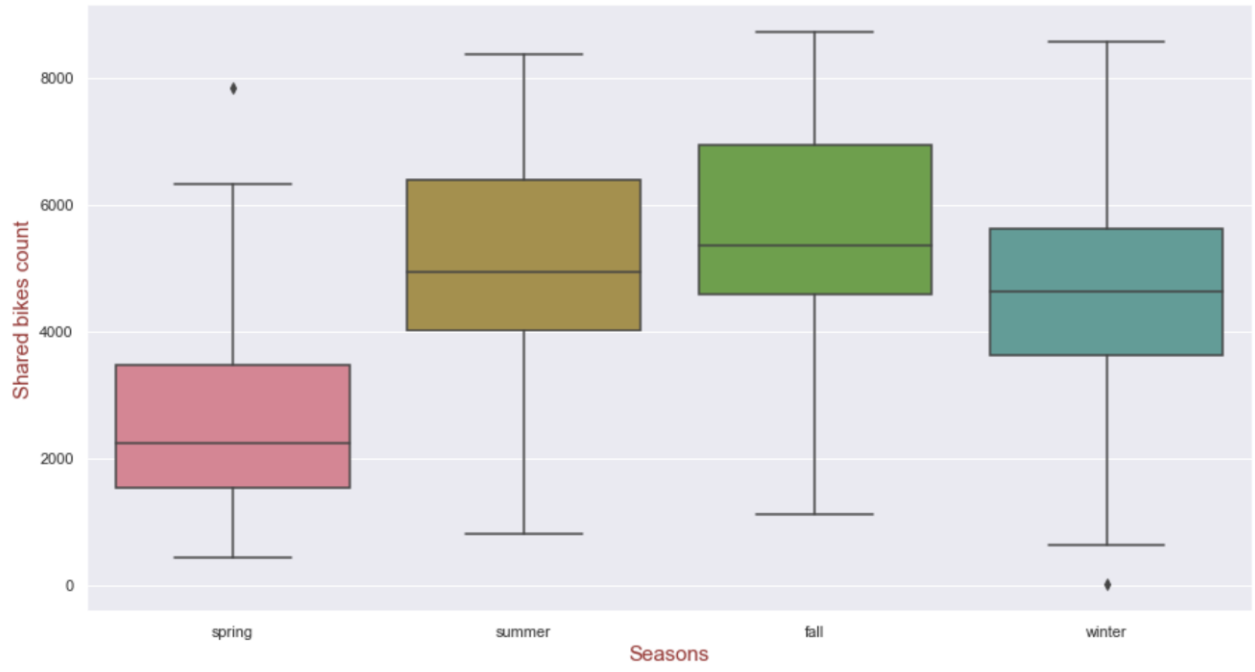
Equation of the model is:

Count = 1988.33 + 1981.38yr - 859.72holiday + 5195.64temp - 1500.5hum - 1646.97windspeed + 708.65summer + 1170.77winter -436.27Cloudy -2015.06Light Rain -415.6July + 835.94September

So summer and winter both are the valid predictors of the total rented bikes count. So season does play an important role.

Both summer and winter positively affects the dependent variable. One unit change in summer increases the dependent variable by 708.65 units if other predictors are kept constant. For winter, the dependent variable increases by 1170.77 units if other predictors are kept constant.

But if season is plotted exclusively with dependent variable, we see that bikes are rented most in fall, then summer, then winter and the least during spring, on average.



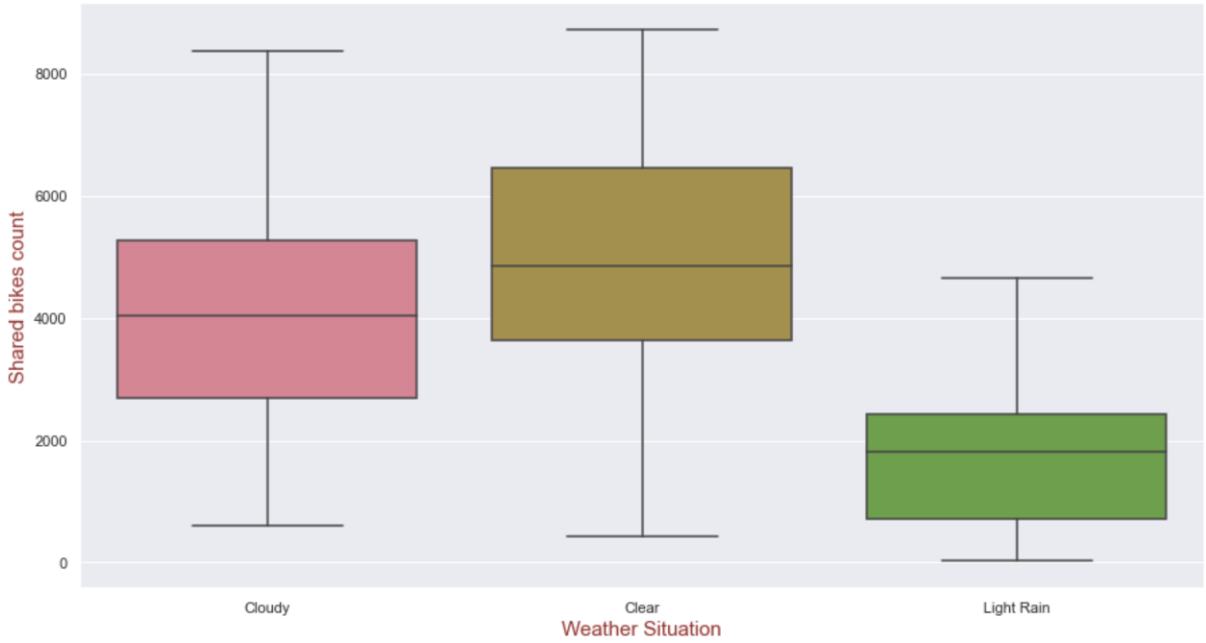
Weather Situation:

Weather situation also plays an important role in determining the dependent variable, since we can see that light rain and cloudy(*Mist + cloudy, Mist + broken clouds, Mist + few clouds, Mist*) both are present in our model equation.

Light rain and cloudy weather situation negatively affects the dependent variable.

If weather situation is 'light rain' and all other independent variables are kept constant, then the dependent variable drops by 2015.06 units, the same is 436.27 for cloudy weather.

Here is the plot of segmented univariate analysis of weather situation with dependent variable



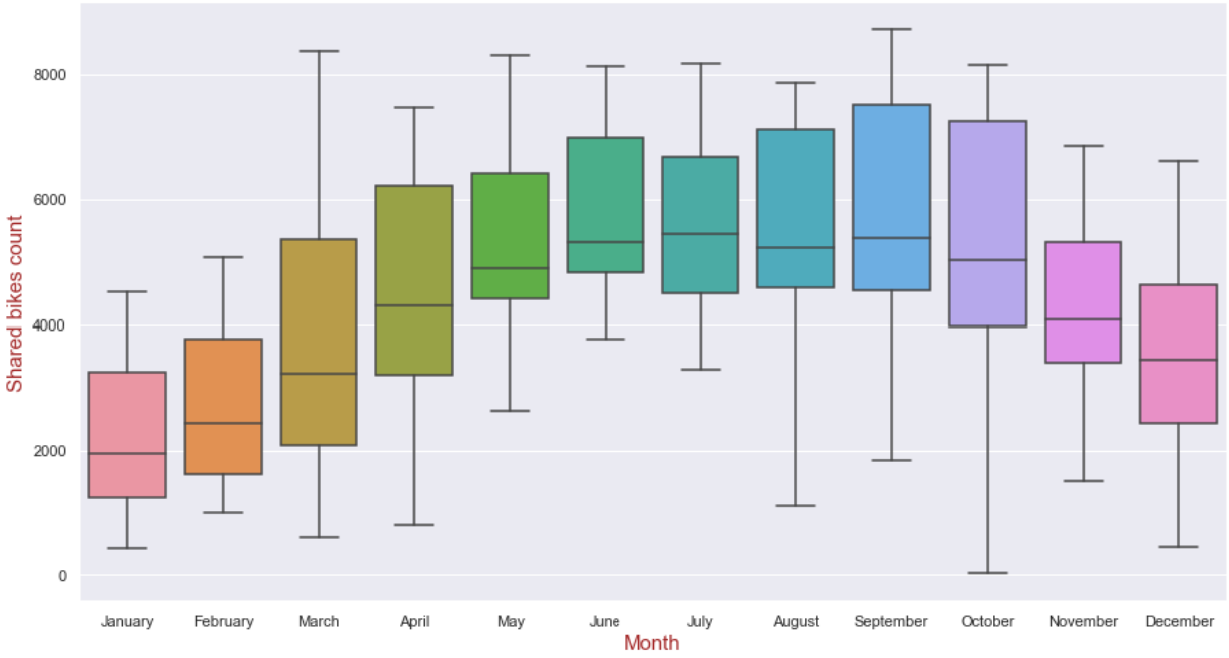
Months:

Month also affects dependent variable since 'July' and 'September' play a role in predicting the dependent variable in our model equation

If all other predictors are kept constant , then the dependent variable increases by 835.94 during the month of 'September'.

If all other predictors are kept constant , then the dependent variable decreases by 415.6 during the month of 'July.

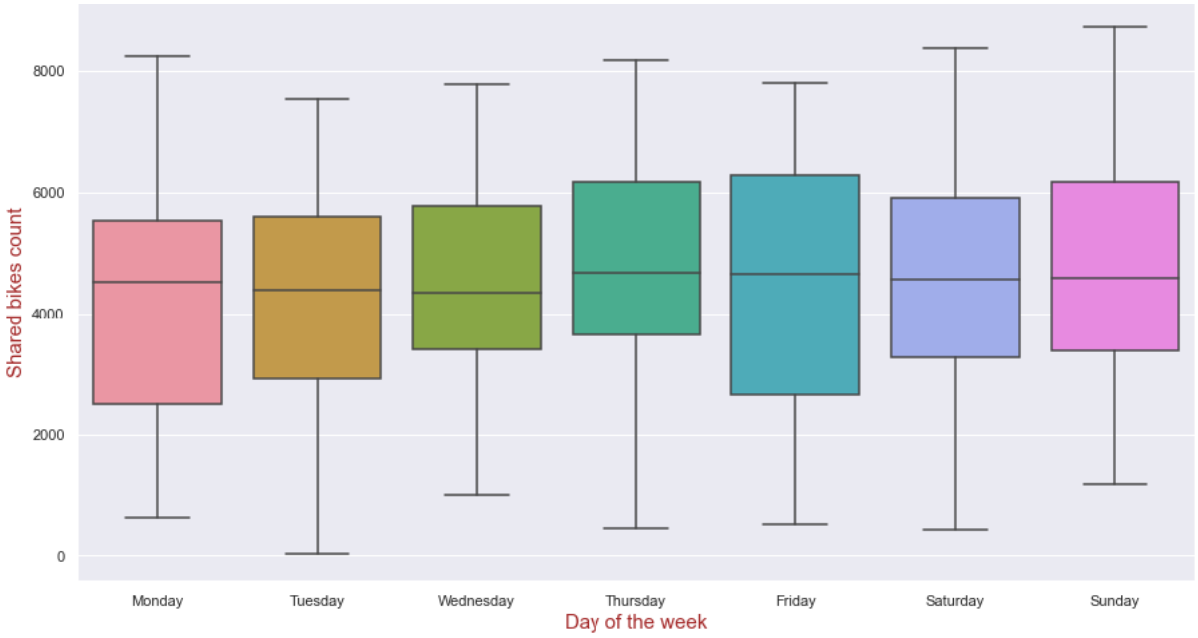
Below chart shows the result of segmented univariate analysis done on months and dependent variable



Day of the week:

Day of the week doesn't play any part in our model equation.

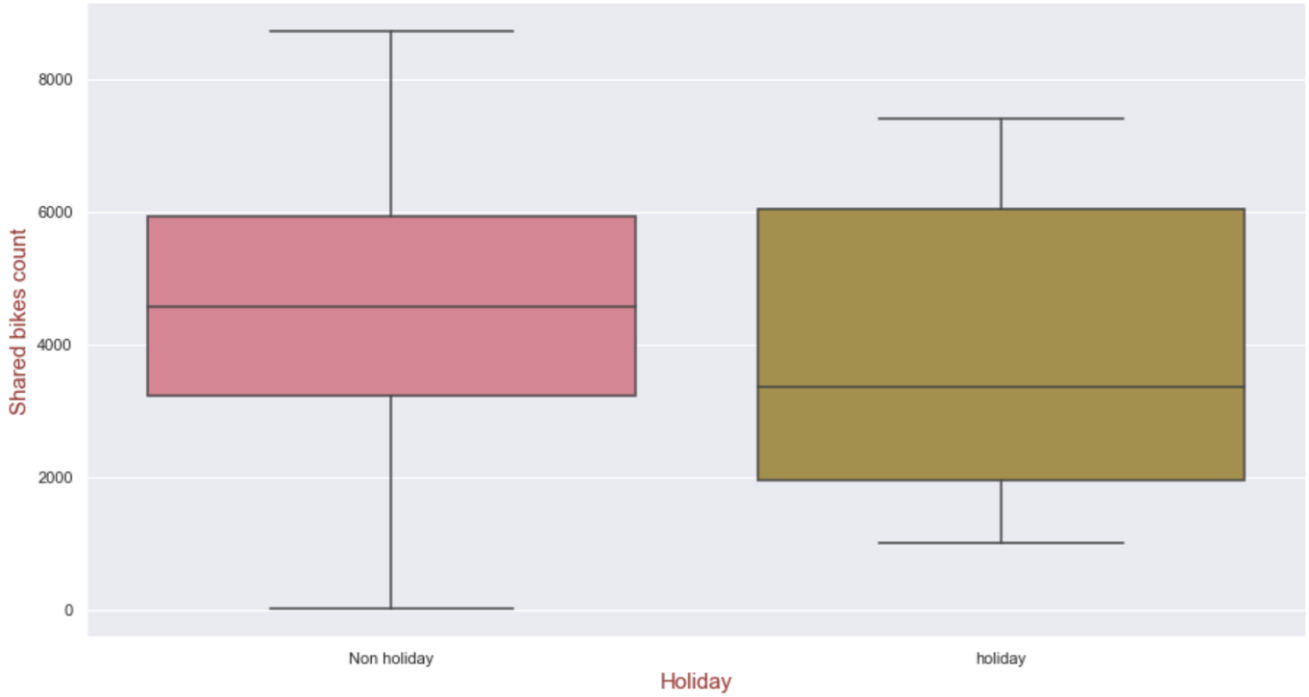
The plot below shows the result of segmented univariate analysis of weekday vs dependent variable



Holiday:

Holiday does play a part in predicting the dependent variable in the model equation. If other variables are kept constant, the dependent variable decreases by 859.72 units on a holiday.

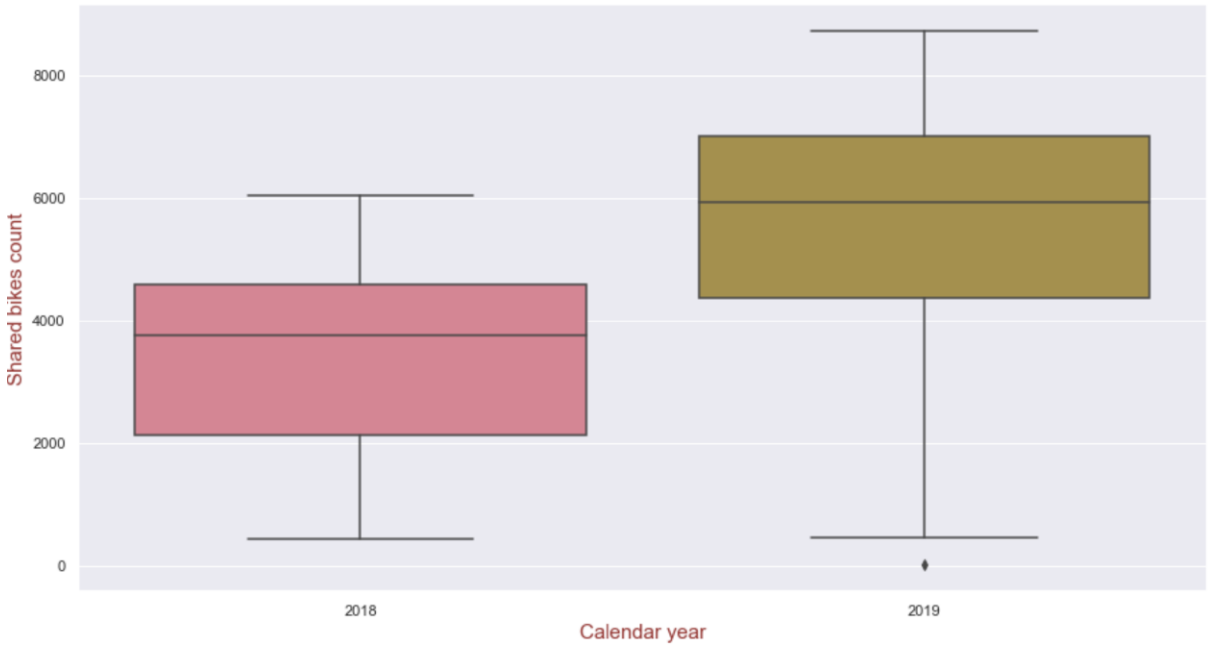
Below is the plot for segmented univariate analysis between holiday and count.



Year:

Year clearly played a part in predicting dependent variable. If all other things are kept constant then dependent variable increased by 1981.38 for year 2019

The segmented univariate analysis also shows a lot of difference for rented bikes count in 2018 vs 2019



Working day:

Working day doesn't play a part in predicting the dependent variable since it doesn't appear in our model equation.

Segmented univariate analysis as a plot is shown below:

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

Because it removes the redundant column, explanation:

Let's say a categorical variable has M levels. During the creation of dummy variables, M columns will be created if drop_first = False, where each column can have either 0 or 1 as values.

But it is completely possible to explain all M levels using M-1 columns, where if all M-1 column is 0, means the Mth level is 1.

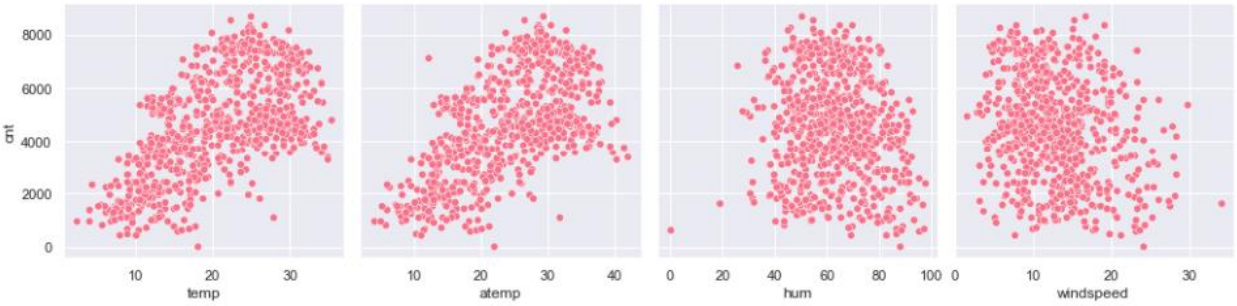
Example: Let's say a categorical variable 'player_action' has 3 levels – 'batting','bowling' and 'fielding'.

Now if use drop_first = True, only 2 columns are created say batting and bowling.

Batting	Bowling	
1	0	Player is batting
0	1	Player is bowling
0	0	Player is fielding

Hence drop_first = True is important, since it is possible to explain all levels using one column less than the levels of category and removes redundancy, and we would have 1 column less to deal with.

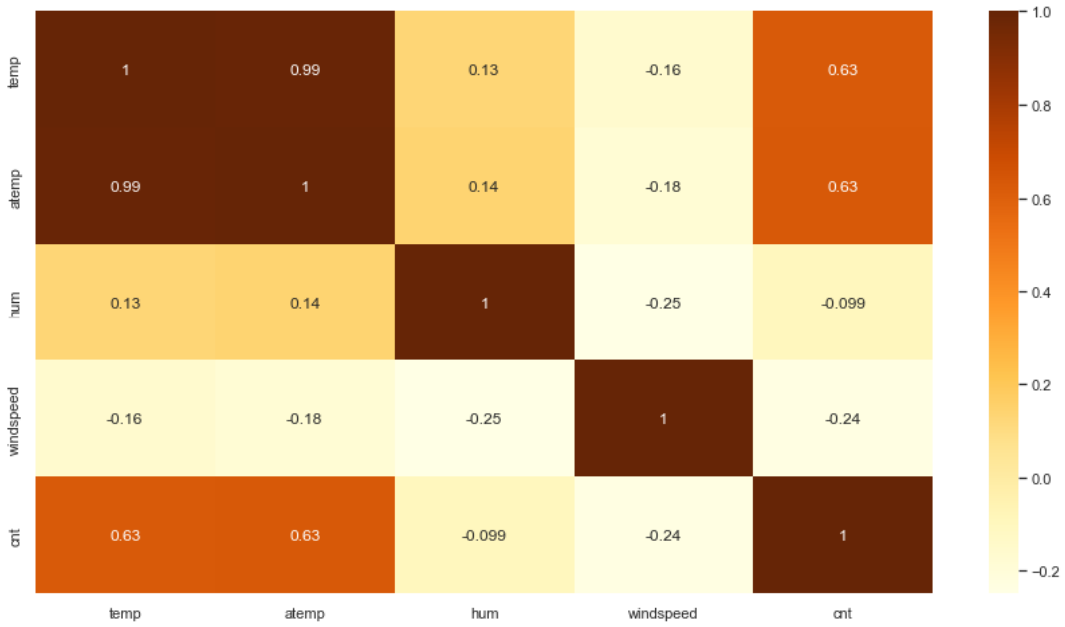
3. *Looking at the pair-plot among the numerical variables, which one has the highest correlation with the target variable?*



Temperature (temp) and feels like temperature (atemp) both looks highly correlated with the target variable.

But both temp and atemp are highly correlated between themselves.

We will use the heatmap as well to verify this:

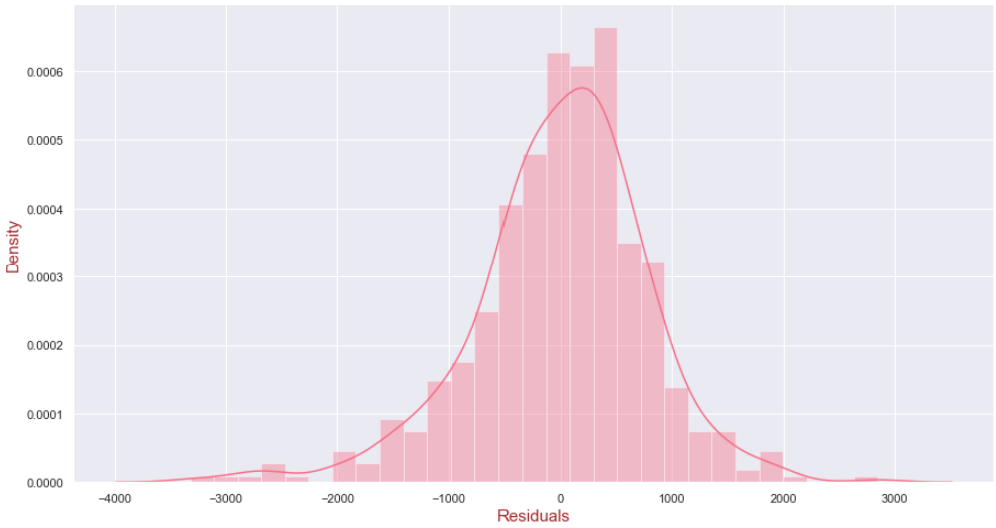


As we can see, temp and atemp are almost same, and their correlation with the target variable is same, and higher than other numerical variables – 0.63. So the answer is : temp, atemp

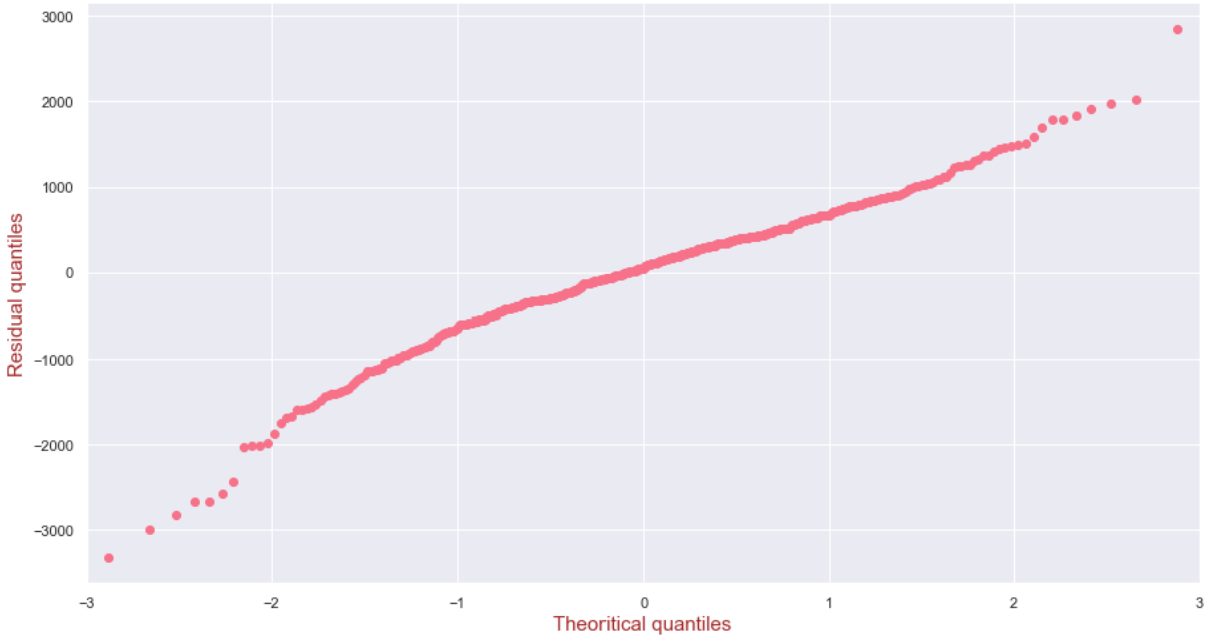
4. *How did you validate the assumptions of Linear Regression after building the model on the training set?*

There are multiple checks that we can do to validate the assumptions of Linear Regression:

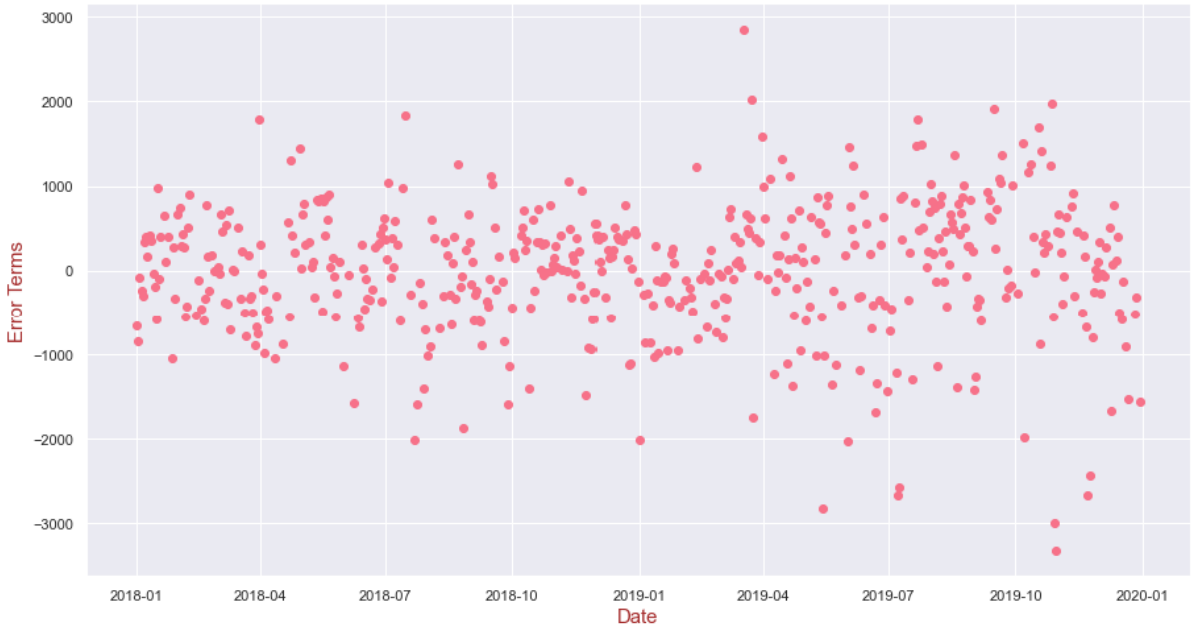
1) Checking if the error terms are normally distributed with zero mean
   Plot the distribution plot of residuals (error terms) i.e. trained y – predicted y



   We can also plot of Q-Q plot of error terms against a theoretical normal distribution to confirm if they follow the normal distribution or not



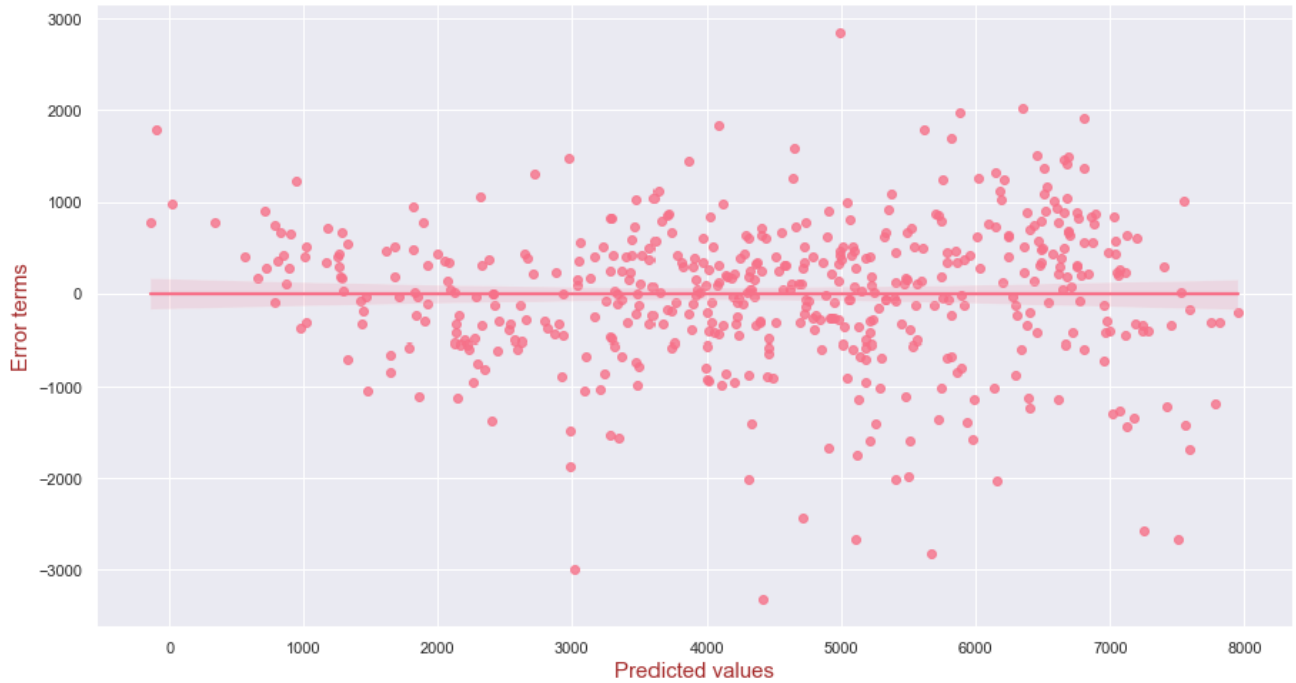
2) We can check if error terms are independent of one another by plotting a scatter plot of error terms against time (date in this case)



3) Check if independents variables and dependent variable follows a linear relationship. Since we have multiple independent variables, we can use the predicted values of y (dependent variable), since it comes from a linear combination of independent variables

So, basically we can plot of y predicted vs y. Here I am attaching a snapshot of y_predicted from test data against y of test data

4) We can check for 'homoscedasticity' by plotting error terms against the predicted values of dependent variable – the residuals should not grow as a function of predicted values



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

Count = 1988.33 + 1981.38yr - 859.72holiday + 5195.64temp - 1500.5hum - 1646.97windspeed + 708.65summer + 1170.77winter -436.27Cloudy -2015.06Light Rain -415.6July + 835.94September

Going by this model, the three top features are

- Temperature (count increases by 5195.64 units for every unit increase in temperature if other predictors are kept constant)
- Light Rain (count decreases by 2015.06 units for every unit increase in humidity if other predictors are kept constant)
- 2019 year (count increases by 1981.38 units for every unit when year changed to 2019 if other predictors are kept constant)

GENERAL SUBJECTIVE QUESTIONS

1. *Explain the linear regression algorithm in detail.*

When there is a linear relationship between the predictors and the dependent variable, the process of predicting or forecasting the value of dependent variable is called linear regression.

Basically we plot the given datapoints between predictors(say X) and dependent variable (say Y), and try to plot the best fit line to explain the relationship between them.

The equation of the line is $y = mx + c$, where m and c are slope and intercept respectively

For multiple predictors: $y = b_0 + b_1x_1 + b_2x_2 .... b_nx_n$

As discussed, the goal of the algorithms is to find the best values of b_0, b_1 … b_n to fit the best possible to explain the relationship between predictors and y.

Now how do we plot the best fit line between them – we tried to reduce the errors between the actual value and the predicted value.

Let's assume we have a single predictor variable x for the dependent variable y.

Say actual value is y, and predicted value yp = mx +c

For x_1, y_1 : $yp_1 = mx_1 + c$. We basically take square of the error

So error square for x_1, y_1 = $(y_1 – yp_1)^2$ = $(y_1 – mx_1-c)^2$

When we sum the error square for all the points , we get $\sum (y_i – mx_i -c)^2$ where i runs from 1 to N

The above sum is called RSS, or residual sum of squares(RSS). This is the cost function that needs to be minimised using an optimization method.

We tried to minimise this sum, either by using differentiation with respect to m and c, or using gradient descent algorithm.

Once we have the optimized m and c available using above methods, we can find the predicted y.

If the error terms are defined as sum of squares of difference between average value of y and predicted y, then the sum is called TSS.

R^2= 1 – RSS/TSS, that we calculate to check the quality of our predictions

If R^2 ~ 1, then the model is assumed to be good.

We need to ensure that the error terms are normally distributed, doesn't follow any pattern and have the constant variance with respect to values of x. And the main thing, y and x should be linearly related.

We need to ensure, in case of multiple linear regression, that there shouldn't be any multicollinearity among the predictors in the final model, and we can use VIF to eliminate such predictors.

Here is a step by step process to create the model using linear regression:

1) Convert the categorical variables into numeric ones
2) Split the dataset into train and test set
3) Scale the train set's numeric variables using normalization/standardization
4) Split y_train and X_train from training set
5) Select top N predictors from X_train using RFE
6) Add constant to X_train if using statsmodels library
7) Train data and analyse the probability values based on tscore
8) Check R square and Adjusted R square values
9) Remove predictor with beta coefficient with high p value
10) Calculate VIF and check predictor with VIF > 5
11) Remove unwanted predictors based on step 9 and 10
12) Repeat steps 7 to 11 until a good model is created
13) Check distribution of error terms using distplot or qq plot
14) Check if the error terms are random and variance is not varying across values of predictors
15) Split test set into y_test and X_test
16) Scale the values of numerical variables with same scale as that in step 3
17) Add constant to X_test
18) Predict y_test from X_test using the model from step 7
19) Calculate r2_score for y_test_predict and X_test
20) Plot the scatter plot between y_test and y_test_predict

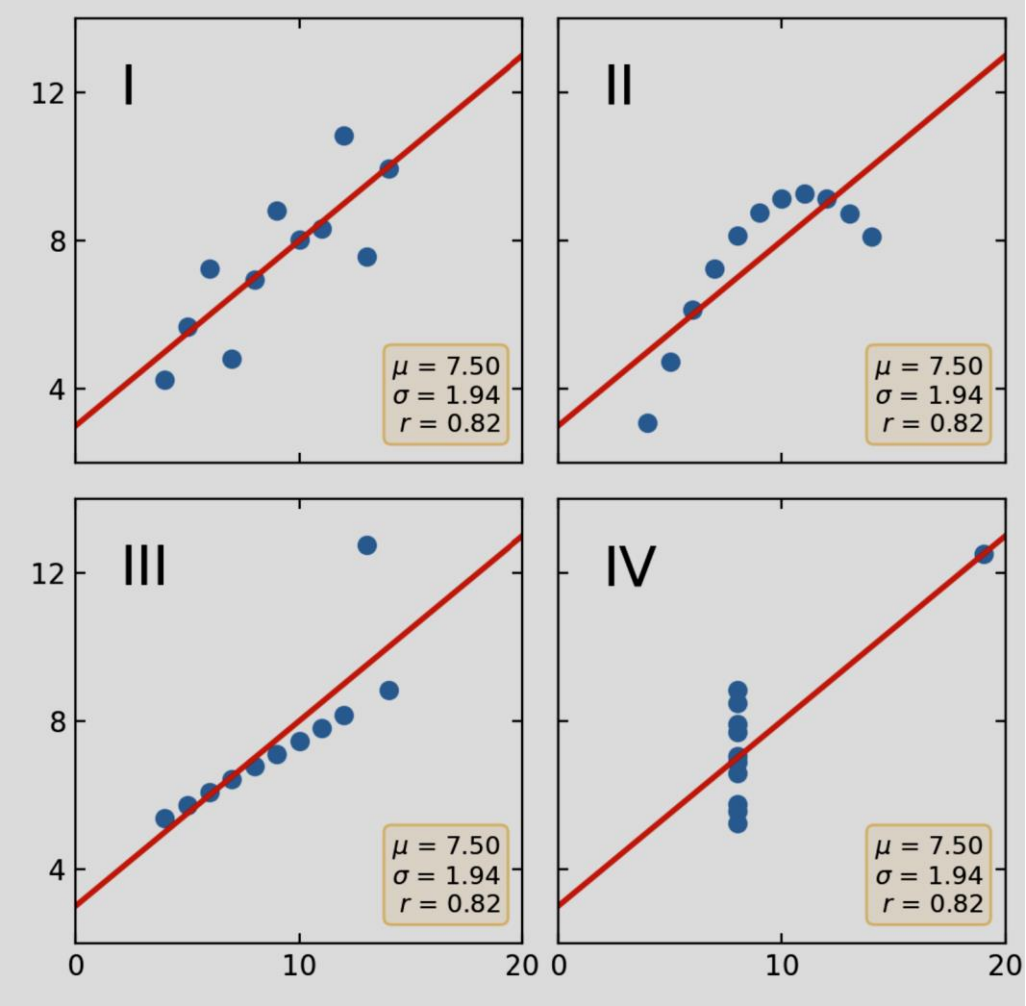
2. *Explain the Anscombe's quartet in detail.*

Anscombe's quartet is a set of four datasets, that have identical statistical properties (mean, standard deviation, regression line) but looks very different when plotted in a graph. The purpose is to emphasize on graphical representation as well, since relying only on basic statistical properties could paint a wrong picture about the dataset.

```
Datasets: x = [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5]
y1 = [8.04, 6.95, 7.58, 8.81, 8.33, 9.96, 7.24, 4.26, 10.84, 4.82, 5.68]
y2 = [9.14, 8.14, 8.74, 8.77, 9.26, 8.10, 6.13, 3.10, 9.13, 7.26, 4.74]
y3 = [7.46, 6.77, 12.74, 7.11, 7.81, 8.84, 6.08, 5.39, 8.15, 6.42, 5.73]
x4 = [8, 8, 8, 8, 8, 8, 8, 19, 8, 8, 8]
y4 = [6.58, 5.76, 7.71, 8.84, 8.47, 7.04, 5.25, 12.50, 5.56, 7.91, 6.89]
```

the four datasets are 1: (x,y1), 2: (x,y2),3: (x,y3) and 4: (x4,y4)

When plotted together, the graph looks like this:



The picture is taken from https://matplotlib.org/

As we can see, all statistical properties are same for all the datasets (described in legends) but the graphs look very different from each other.

3. *What is Pearson's R?*

Pearson's R is a way to measure how two datasets are linearly correlated with each other.

To understand Pearson's R, we need to understand covariance.

Covariance is the measurement of how two random variables are varying with respect to each other. Let's simplify it more, how do we measure variability of one random variable? We measure the variance. If we extend this concept to two different random variables, we would arrive at covariance.

Covariance (X,Y) = E ((X – E(X)) (Y-E(Y)) where E is the expected value function

Now, Pearson's R is Covariance(X,Y) / SD(X)*SD(Y) where SD is standard deviation

Pearson's R belongs to [-1, 1]

If Pearson's R is 0 then X and Y are not linearly correlated at all

If Pearson's R belongs to [-1 ,0), then X and Y negatively correlated

If Pearson's R belongs to (0, 1], then X and Y are positively correlated

In graphical terms, Pearson's R is the angle between Y = f(X) and X = f(Y), which is regressing X on Y and Y on X

We can find Pearson's R in

Numpy as => np.corrcoef(X, Y)

Pandas as => X.corr(Y)

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

Scaling is the process of mapping a dataset to a small range.

Scaling is performed so that beta coefficients of independent variables that comes out after training the dataset doesn't vary hugely and should be easy to interpret. Say an independent variable X1 belongs to [1,10000] and other variable X2 belongs to [1,10], then their beta coefficients could vary wildly in the magnitude, also it would be very difficult to interpret the coefficients.

Normalized scaling set the range of independent variables to [0,1] while Standardized scaling brings the data to normal distribution with mean 0 and standard deviation 1.

Standardization : (x – mean(s) / sd(x))

Normalization: x – min(x) / max(x) – min(x)

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

Let's revisit the formula of VIF

VIF(i) = 1 / (1 – $R(i)^2$) where R(i) is the R square value of the ith predictor variable

If VIF is infinite, means the denominator is 0

1 – R^2 = 0

R^2 = 1

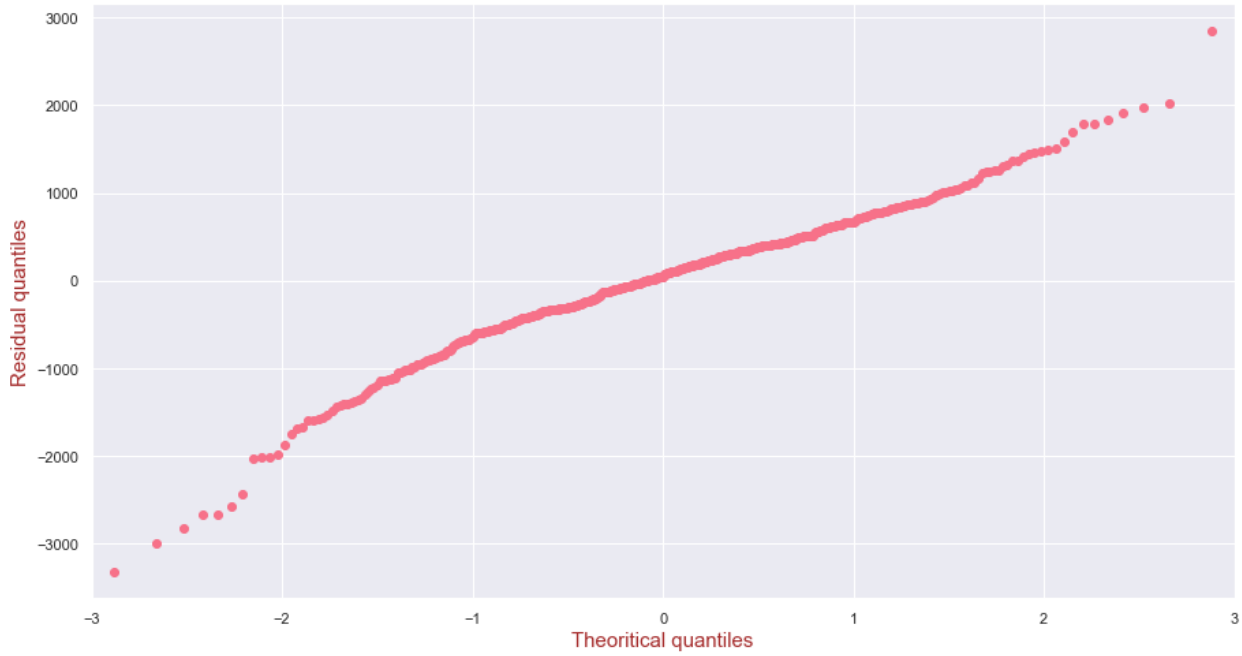
Therefore, the predictor variable here is perfectly correlated with other predictor variables or it can be perfectly expressed linearly by the combination of other predictor variables, and it would be a good idea to remove this variable.

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

Q- Q plot, or quantile-quantile plot, is a plot used to determine whether the plot is a normal distribution, exponential distribution among other kind of distributions. In q-q plot the quantiles of two probability distributions are plotted against each other.

We can check how much data lies within 1,2 or 3 standard deviations of mean. On the X -axis we plot a theoretical normal distribution with standard deviation of 1 unit. And on the Y -axis we plot the sorted ordered values of the dataset for which we are checking the distribution type.

If the data is normally distributed, we get a straight line. If the plotted line is deviating from the ideal straight line, then we can conclude that the data is not normally distributed.



Above is the Q-Q plot for residuals that we got in our model. It is normally distributed, and most data points lies towards the centre. The ends of the line are wearing off and the line is straight at the centre, which implies that there is not much deviation, hence it's a pretty good normal distribution.

Now we can deduce the type of skewness and deviation in data as well by looking at the plot as mentioned below:

1. If left end of the Q-Q plots deviates from the line which is straight otherwise, means the left tail is longer which in turn implies the data is left skewed
2. If the right end of the Q-Q plots deviates from the line which is straight otherwise, means the right tail is longer which in turn implies that the data is right skewed.
3. If both end deviates from the line which is straight at the centre, means there the data has a large deviation

We can use statsmodel.api to plot the Q-Q plot

```
import statsmodel.api as sm

qqplot(res)
```

In the context of linear regression, when we want to check the assumptions of linear regression – specifically if the error terms are normally distributed or not, then we can plot their Q-Q plot against a theoretical distribution with mean 0 and standard deviation of 1 unit.