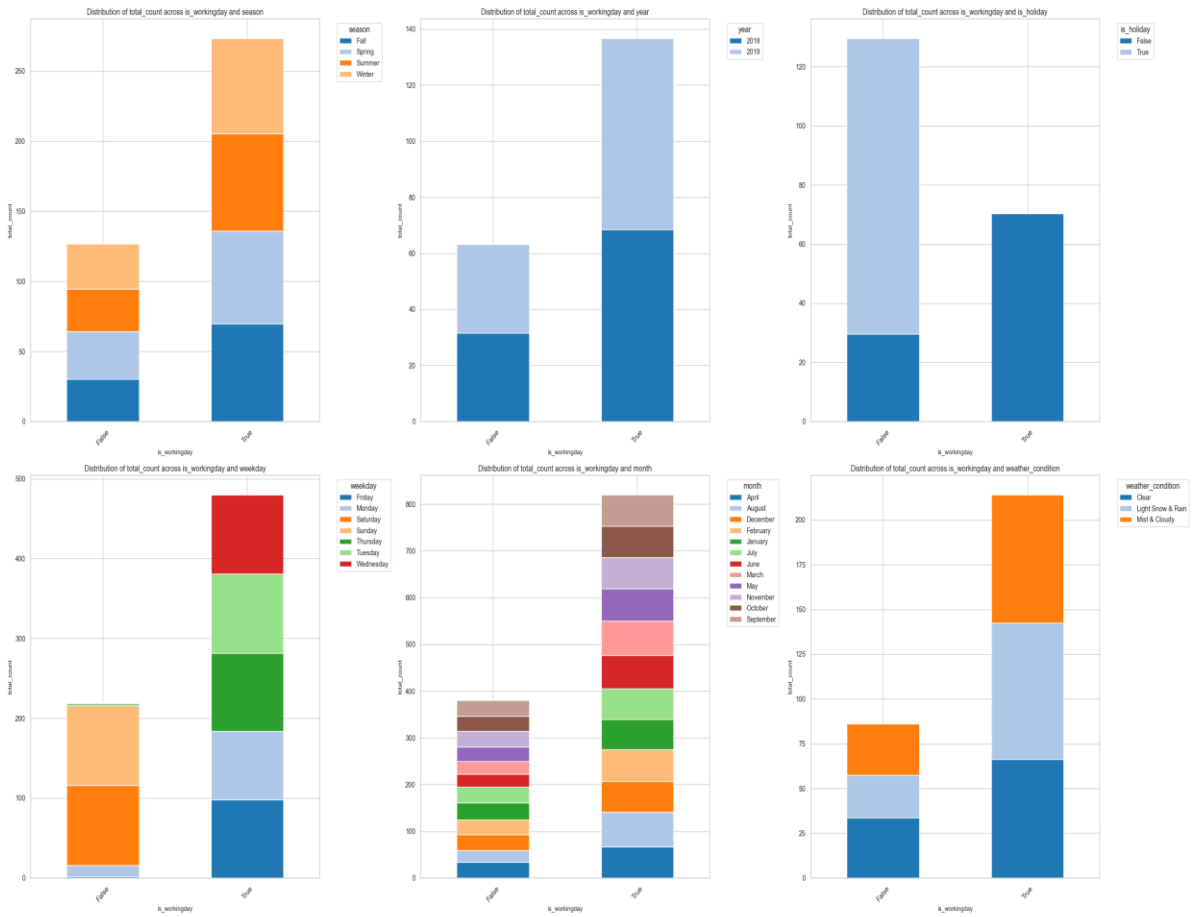
Assignment-Based Subjective Questions

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

- **Season vs total_count**: Rentals are highest in the fall and summer seasons. Winter has the lowest rentals, and spring shows moderate usage.
- **Month vs total_count:** Rentals increase from January, peaking around May and October, and then decline towards the end of the year.
- **Year vs total_count:** There is a significant increase in rentals from 2018 to 2019.
- **is_holiday vs total_count:** Rentals on is_holidays are slightly higher than on non-is_holidays, but the difference is minimal.
- **Weekday vs total_count:** Rentals are relatively even across the weekdays, with no significant drop during weekends.
- **is_workingday vs total_count:** Rentals on working days are slightly higher compared to non-working days.
- **weather_condition vs total_count**: Clear weather conditions (good) have the highest rentals, while bad and severe weather conditions see a significant drop.

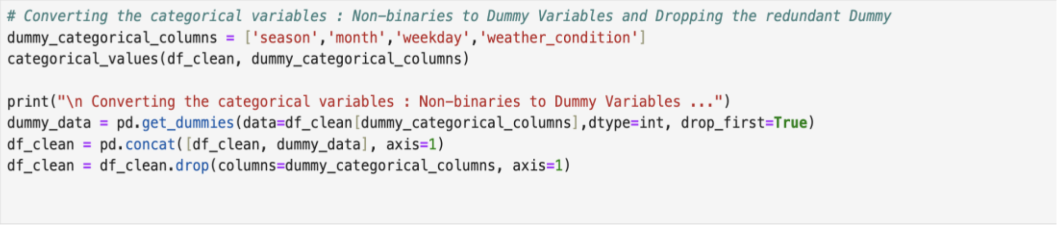


2. Why is it important to use drop_first=True during dummy variable creation?

Using drop_first=True during dummy variable creation is important to avoid multicollinearity in regression models and to ensure proper interpretation of the coefficients.

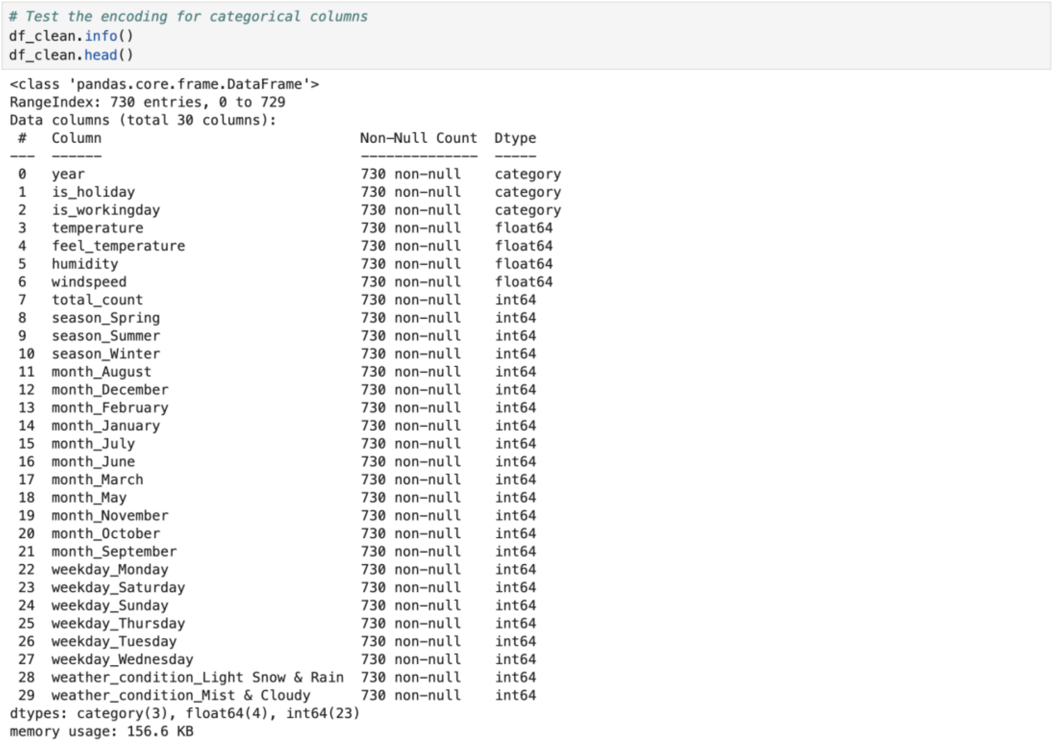
Example:

- Suppose we have a categorical variable "Season" with three categories: Spring, Summer, and Fall.
- **Dummy Variable Creation:**
  - If we create dummy variables without drop_first=True, we would have 3 variables: season_Spring, season_Summer, and season_Fall.
  - With drop_first=True, we create two variables: season_Summer and season_Fall.
  - Here, season_Spring is omitted as the baseline category.
- **Meaning:**
  - Coefficient for season_Summer: This tells you how much bike demand changes in summer compared to spring (baseline).
  - Coefficient for season_Fall: This tells you how much bike demand changes in fall compared to spring (baseline).



```
# Converting the categorical variables : Non-binaries to Dummy Variables and Dropping the redundant Dummy
dummy_categorical_columns = ['season','month','weekday','weather_condition']
categorical_values(df_clean, dummy_categorical_columns)

print("\n Converting the categorical variables : Non-binaries to Dummy Variables ...")
dummy_data = pd.get_dummies(data=df_clean[dummy_categorical_columns],dtype=int, drop_first=True)
df_clean = pd.concat([df_clean, dummy_data], axis=1)
df_clean = df_clean.drop(columns=dummy_categorical_columns, axis=1)
```



```
# Test the encoding for categorical columns
df_clean.info()
df_clean.head()
```

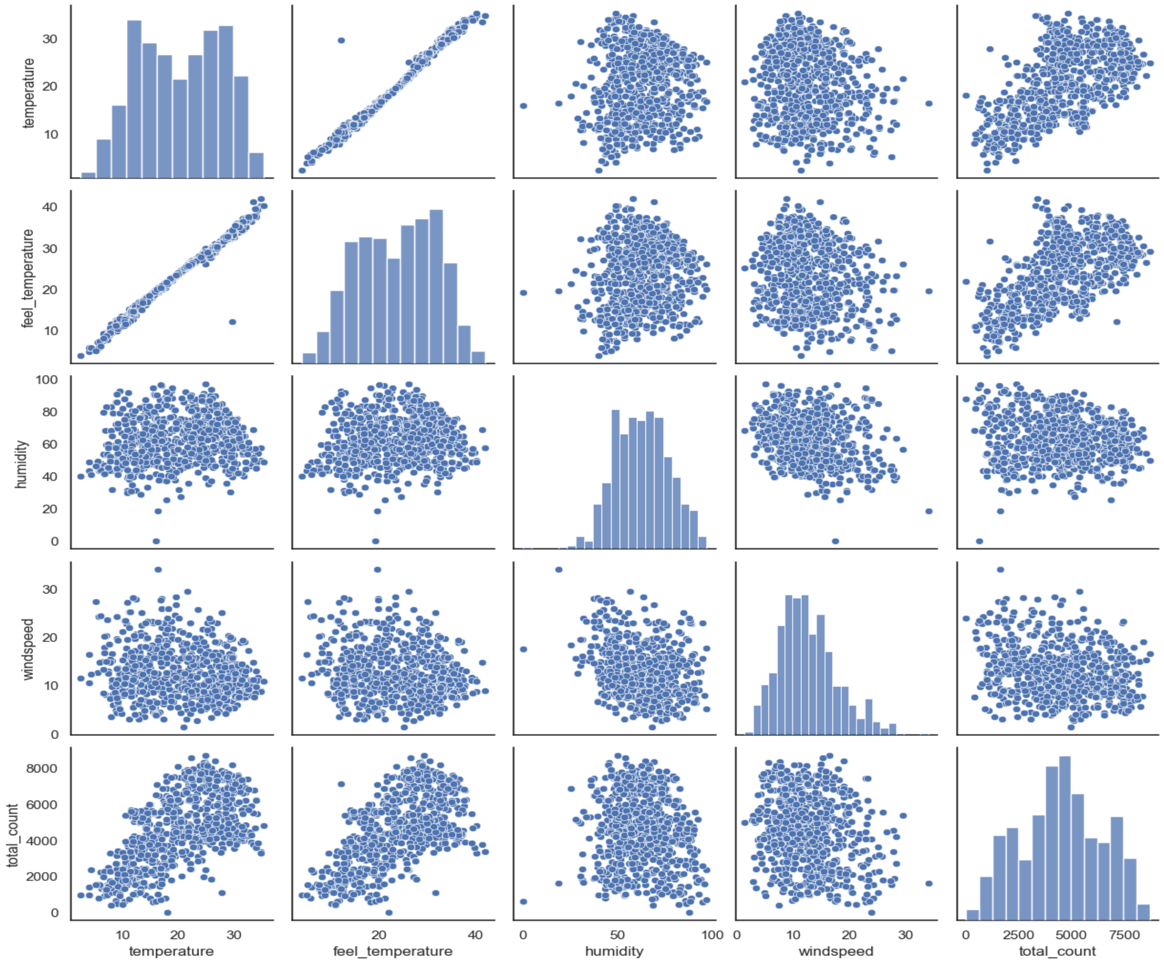
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 730 entries, 0 to 729
Data columns (total 30 columns):
 #   Column                            Non-Null Count  Dtype   
---  ------                            --------------  -----   
 0   year                              730 non-null    category
 1   is_holiday                        730 non-null    category
 2   is_workingday                     730 non-null    category
 3   temperature                       730 non-null    float64 
 4   feel_temperature                  730 non-null    float64 
 5   humidity                          730 non-null    float64 
 6   windspeed                         730 non-null    float64 
 7   total_count                       730 non-null    int64   
 8   season_Spring                     730 non-null    int64   
 9   season_Summer                     730 non-null    int64   
 10  season_Winter                     730 non-null    int64   
 11  month_August                      730 non-null    int64   
 12  month_December                    730 non-null    int64   
 13  month_February                    730 non-null    int64   
 14  month_January                     730 non-null    int64   
 15  month_July                        730 non-null    int64   
 16  month_June                        730 non-null    int64   
 17  month_March                       730 non-null    int64   
 18  month_May                         730 non-null    int64   
 19  month_November                    730 non-null    int64   
 20  month_October                     730 non-null    int64   
 21  month_September                   730 non-null    int64   
 22  weekday_Monday                    730 non-null    int64   
 23  weekday_Saturday                  730 non-null    int64   
 24  weekday_Sunday                    730 non-null    int64   
 25  weekday_Thursday                  730 non-null    int64   
 26  weekday_Tuesday                   730 non-null    int64   
 27  weekday_Wednesday                 730 non-null    int64   
 28  weather_condition_Light Snow & Rain  730 non-null    int64   
 29  weather_condition_Mist & Cloudy   730 non-null    int64   
dtypes: category(3), float64(4), int64(23)
memory usage: 156.6 KB
```

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

From the pair-plot, the variable that has the highest correlation with the target variable (total_count) appears to be **temperature**. The scatter plot between temperature and total_count shows a positive linear relationship, indicating that as the temperature increases, the total count of bike rentals also increases.

Interpretation:

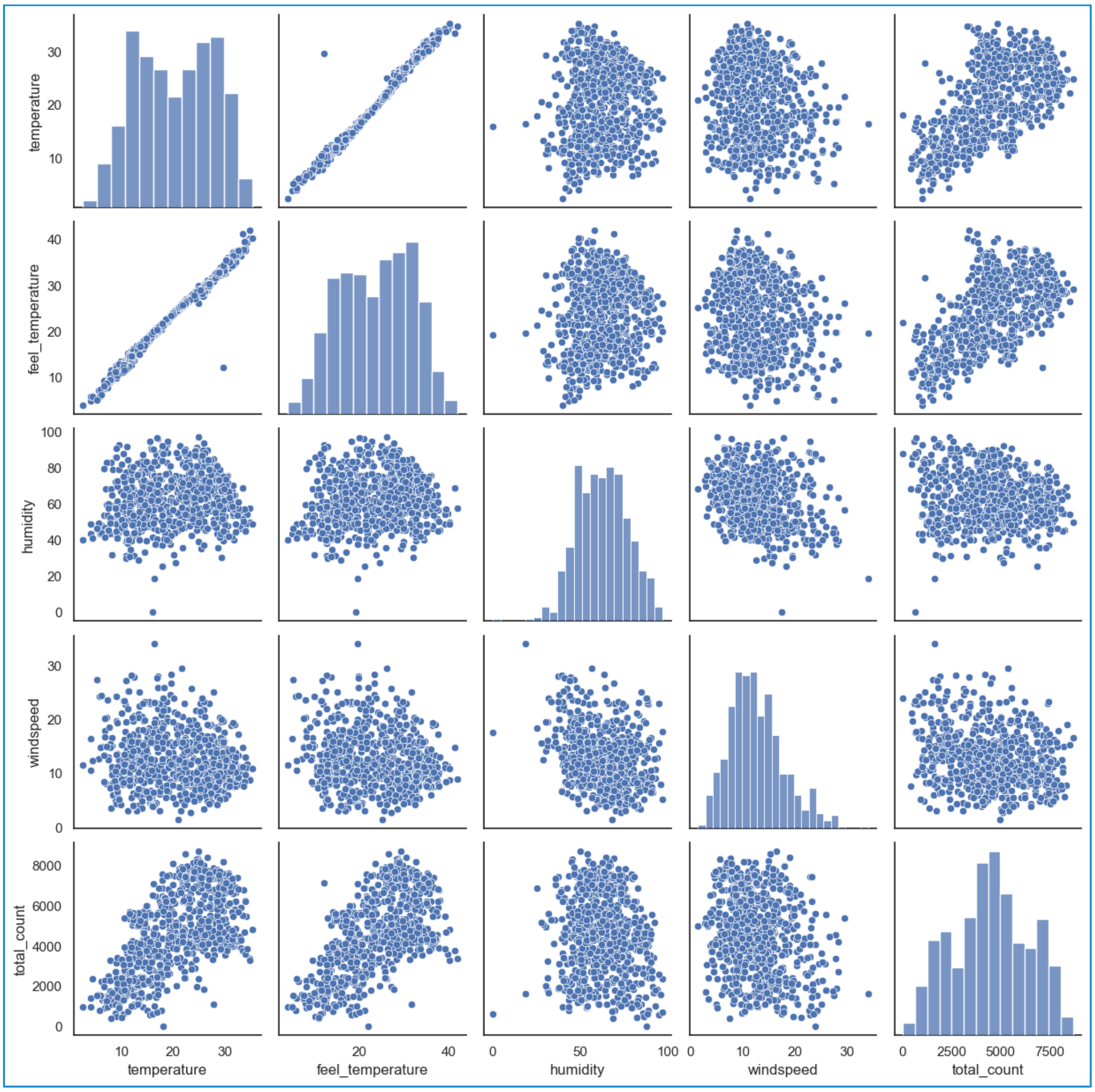
- **Temperature vs. Total Count:** Shows a clear positive trend, suggesting a strong correlation.
- **Feel Temperature vs. Total Count:** Shows a positive trend.
- **Humidity vs. Total Count:** Shows a less clear relationship, possibly indicating a weaker correlation.
- **Windspeed vs. Total Count:** Shows some negative relationship but not as strong as temperature.



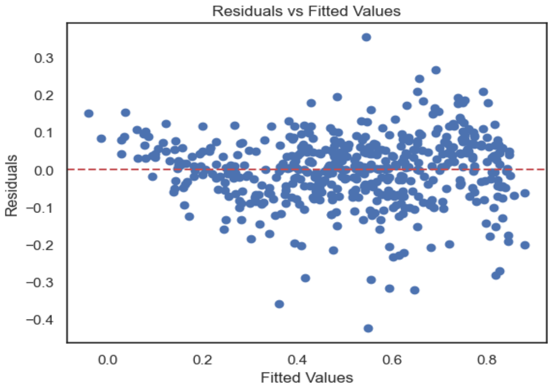
4. How did you validate the assumptions of Linear Regression after building the model on the training set?

We have validated the assumptions of linear regression using the following checks:

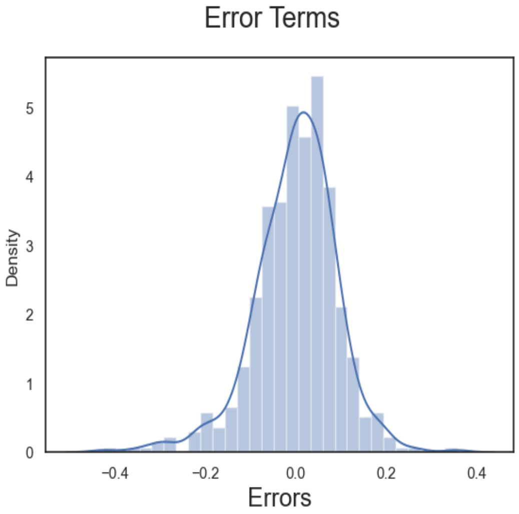
1. **Linearity**: Pair-Plot (We checked if the relationship between the predictors and the response variable is linear.)

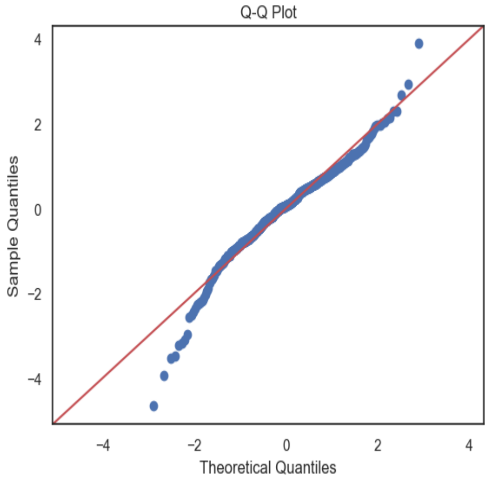


2. **Homoscedasticity (Constant Variance of Residuals)**: Residuals vs. Fitted Values Plot (We checked if the residuals have constant variance and the plot had a random scatter of points around the zero line, with no clear pattern.)

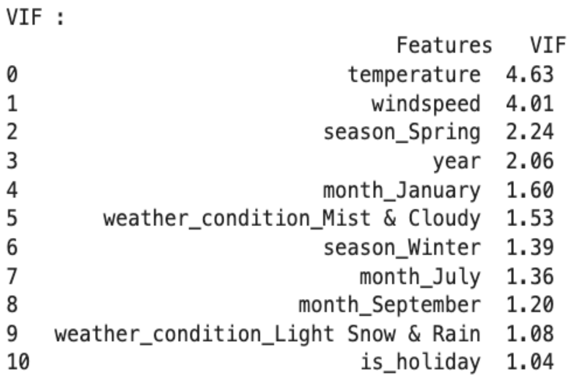


3. **Normality of Residuals**: Q-Q Plot (Quantile-Quantile Plot) and Error Terms (We checked if the residuals are normally distributed.)





4. **Multicollinearity**: Variance Inflation Factor (VIF) (We checked for multicollinearity using the Variance Inflation Factor (VIF). VIF values above 5-10 indicate multicollinearity.)



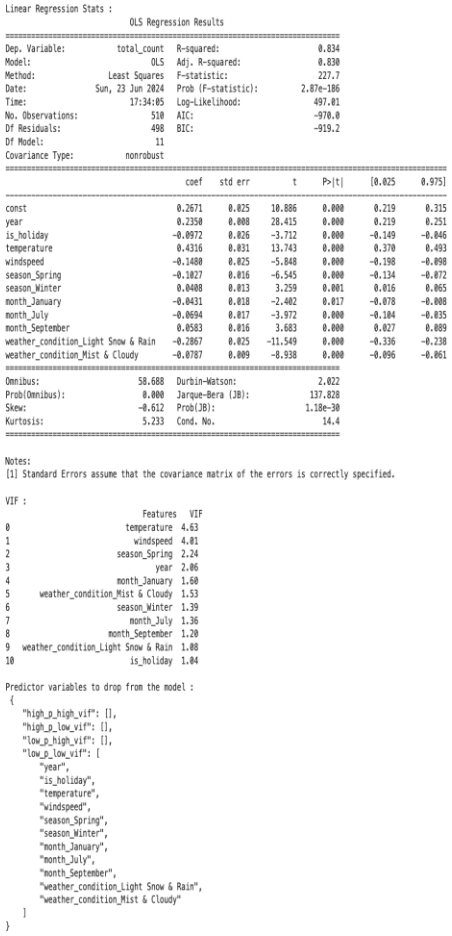
```
VIF :
                                Features   VIF
0                            temperature  4.63
1                              windspeed  4.01
2                          season_Spring  2.24
3                                   year  2.06
4                          month_January  1.60
5          weather_condition_Mist & Cloudy  1.53
6                          season_Winter  1.39
7                             month_July  1.36
8                        month_September  1.20
9    weather_condition_Light Snow & Rain  1.08
10                            is_holiday  1.04
```

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

Based on the final model's coefficients and their significance (p-values), the top 3 features contributing significantly towards explaining the demand for shared bikes are as follows:

1. **Temperature (coef: 0.4316, p-value: 0.000)**
   - Temperature has a strong positive impact on the demand for shared bikes. As the temperature increases, the demand for bikes also increases significantly.
2. **Year (coef: 0.2350, p-value: 0.000)**
   - The year variable indicates a general upward trend in bike demand over time. With each passing year significantly increases the demand for bikes.
3. **Weather Condition - Light Snow & Rain (coef: -0.2867, p-value: 0.000)**
   - Adverse weather conditions like light snow and rain have a strong negative impact on bike demand. When adverse weather conditions are present, the demand for bikes decreases significantly.

The above features are chosen because of their large coefficient values and very low p-values, indicating that they have a strong and statistically significant impact on the target variable (total count of bike rentals).



```
Linear Regression Stats :
                            OLS Regression Results
==============================================================================
Dep. Variable:            total_count   R-squared:                       0.834
Model:                            OLS   Adj. R-squared:                  0.830
Method:                 Least Squares   F-statistic:                     227.7
Date:                Sun, 23 Jun 2024   Prob (F-statistic):          2.87e-186
Time:                        17:34:05   Log-Likelihood:                 497.01
No. Observations:                 510   AIC:                            -970.0
Df Residuals:                     498   BIC:                            -919.2
Df Model:                          11
Covariance Type:            nonrobust
=======================================================================================================
                                          coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------------------
const                                   0.2671      0.025     10.886      0.000       0.219       0.315
year                                    0.2350      0.008     28.415      0.000       0.219       0.251
is_holiday                             -0.0972      0.026     -3.712      0.000      -0.149      -0.046
temperature                             0.4316      0.031     13.743      0.000       0.370       0.493
windspeed                              -0.1480      0.025     -5.848      0.000      -0.198      -0.098
season_Spring                          -0.1027      0.016     -6.545      0.000      -0.134      -0.072
season_Winter                           0.0408      0.013      3.259      0.001       0.016       0.065
month_January                          -0.0431      0.018     -2.402      0.017      -0.078      -0.008
month_July                             -0.0694      0.017     -3.972      0.000      -0.104      -0.035
month_September                         0.0583      0.016      3.683      0.000       0.027       0.089
weather_condition_Light Snow & Rain    -0.2867      0.025    -11.549      0.000      -0.336      -0.238
weather_condition_Mist & Cloudy        -0.0787      0.009     -8.938      0.000      -0.096      -0.061
==============================================================================
Omnibus:                       58.688   Durbin-Watson:                   2.022
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              137.828
Skew:                          -0.612   Prob(JB):                     1.18e-30
Kurtosis:                       5.233   Cond. No.                         14.4
==============================================================================

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

VIF :
                               Features   VIF
0                           temperature  4.63
1                             windspeed  4.01
2                         season_Spring  2.24
3                                  year  2.06
4                         month_January  1.60
5       weather_condition_Mist & Cloudy  1.53
6                         season_Winter  1.39
7                            month_July  1.36
8                       month_September  1.20
9   weather_condition_Light Snow & Rain  1.08
10                           is_holiday  1.04

Predictor variables to drop from the model :
 {
    "high_p_high_vif": [],
    "high_p_low_vif": [],
    "low_p_high_vif": [],
    "low_p_low_vif": [
        "year",
        "is_holiday",
        "temperature",
        "windspeed",
        "season_Spring",
        "season_Winter",
        "month_January",
        "month_July",
        "month_September",
        "weather_condition_Light Snow & Rain",
        "weather_condition_Mist & Cloudy"
    ]
}
```

General Subjective Questions

1. Explain the linear regression algorithm in detail.

Linear regression is a statistical method that is used to model the relationship between a dependent variable (target) and independent variables (predictors) by fitting a linear equation to the data points.

Steps of Linear Regression:

1. **Define the Problem:** Identify the dependent variable (the variable we want to predict or explain) and the independent variables (variables that may influence the dependent variable also known as predictors).
2. **Collect Data:** Gather the dataset containing observations of the dependent and independent variables. Ensure that the data is clean and does not contain any missing values or outliers that could skew the results.
3. **Explore the Data:** Perform exploratory data analysis (EDA) to understand the distribution, relationships, and summary statistics of the data. Visualize relationships between variables using scatter plots or correlation metrices.

4. **Split Data:** Split the dataset into training and testing sets. The training set is then used to train the model, while the testing set is used to evaluate the model's performance.
5. **Choose a Model:** Select the linear regression as the modeling technique, assuming there is a linear relationship between the independent and dependent variables.
6. **Formulate the Model:** The linear regression model is represented as:

   $y = \beta^0 + \beta^1 x^1 + \beta^2 x^2 + \cdots + \beta_\mathrm{n} x_\mathrm{n} + \epsilon$, where:

   - y is the dependent variable (target).
   - $x1, x2, \dots, xn$ are the independent variables (predictors).
   - $\beta 0$ is the intercept (where the line crosses the y-axis).
   - $\beta 1, \beta 2, \dots, \beta n$ are the coefficients (slopes) of the independent variables.
   - ϵ is the error term (residuals), representing the difference between observed and predicted values.
7. **Fit the Model:** Use the training data to estimate the coefficients $\beta 0, \beta 1, \dots, \beta n$ that minimize prediction errors using methods like Ordinary Least Squares (OLS).
8. **Evaluate the Model:** Assess the model's performance using evaluation metrics such as R-squared, on the testing dataset. R-squared measures how well the model explains the variance in the dependent variable.
9. **Make Predictions:** Use the fitted model to make predictions on testing data by applying the learned coefficients to the independent variables.
10. **Validate and Iterate:** Validate the model by comparing predicted values with actual values in the testing set. Iterate by refining the model, adjusting variables.

Example: Suppose a bike sharing company wants to predict daily bike rentals based on weather conditions and time-related factors. Here's how they would apply linear regression:

Step 1: Define the problem—predict daily bike rentals based on variables like temperature, humidity, windspeed, and time-related factors (hour, day of week).
Step 2: Collect data—gather historical data on daily bike rentals, weather conditions (temperature, humidity, windspeed), time indicators (hour, day of week), and other relevant factors.
Step 3: Explore data—plot variables like temperature against bike rental count to see if there's a relationship.
Step 4: Split data—divide the dataset into training (e.g., 80%) and testing (e.g., 20%) sets.
Step 5: Choose model—select linear regression because it's effective for predicting a continuous outcome based on multiple predictors.
Step 6: Formulate model—

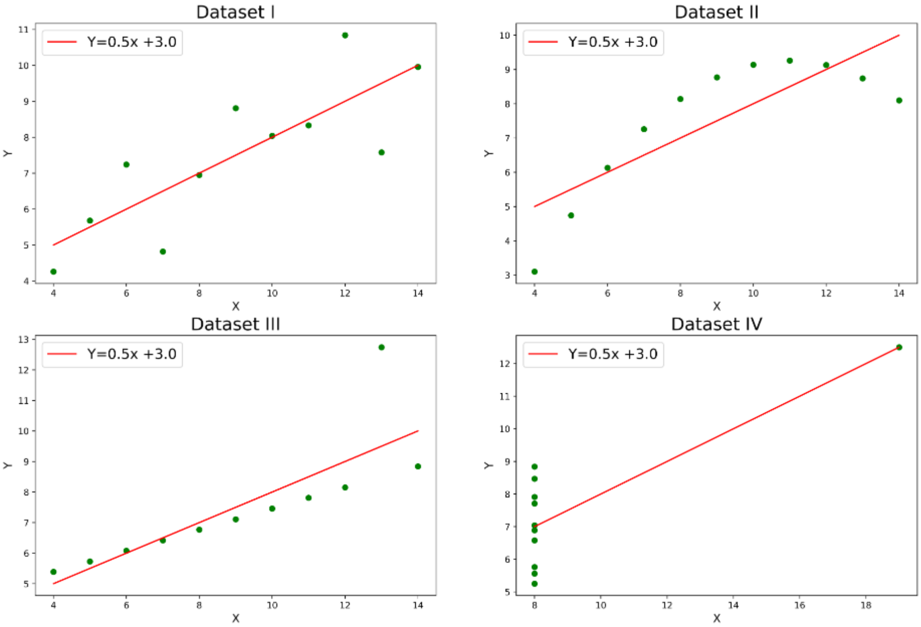
$$total\square\; count = \beta^0 + \beta^1 \times temperature + \beta^2 \times humidity + \beta^3 \times windspeed + \cdots + \epsilon$$

Step 7: Fit model—use training data to estimate coefficients β0, β1, β2, ... that minimize prediction errors.
Step 8: Evaluate model—use metrics like R-squared and RMSE to measure how well the model predicts bike rental counts on the testing set.
Step 9: Make predictions—apply the model to new data (upcoming days or months) to forecast bike rental demand.
Step 10: Validate and iterate—compare predicted rentals with actual counts to refine the model, possibly adding new variables or adjusting parameters for better predictions.

2. Explain the Anscombe's quartet in detail.

Anscombe’s quartet consists of four datasets that have nearly identical simple descriptive statistics but differ when plotted. It demonstrates the importance of visualizing data and checking assumptions before applying statistical analysis.

- Each dataset consists of 11 (x, y) pairs that have the same mean, variance, correlation coefficient, and linear regression line.
- Despite identical mean, variance, correlation coefficient, and linear regression line, the datasets vary widely in terms of scatter plots, regression lines, and correlations.
- Anscombe’s quartet highlights the limitations of summary statistics alone and emphasizes the need for graphical exploration and visualization in data analysis.
- It underscores the downside of relying solely on summary statistics.



Explanation of this plots above:

- In the first one (top left) , there seems to be a linear relationship between x and y.
- In the second one (top right), there is a non-linear relationship between x and y.
- In the third one (bottom left), there is a perfect linear relationship for all the data points except one which seems to be an outlier which is indicated be far away from that line.
- Finally, the fourth one (bottom right) shows an example of when one high-leverage point is enough to produce a high correlation coefficient.

3. What is Pearson’s R?

Pearson’s R, also known as the Pearson correlation coefficient, is a measure of the linear relationship between two variables. It quantifies how well the variation in one variable predicts the variation in another variable. The value of Pearson’s R ranges from -1 to 1:

- **+1** indicates a perfect positive linear relationship.
- **-1** indicates a perfect negative linear relationship.
- **0** indicates no linear relationship.

Characteristics:

- Positive values indicate a positive relationship, where one variable increases, the other also increases.
- Negative values indicate a negative relationship, where one variable increases, the other decreases.
- Values closer to +1 or -1 indicate a stronger linear relationship.
- Values closer to 0 indicate a weaker linear relationship.

Example:

- **r = 0.7 to 1.0 or -0.7 to -1.0**: Strong relationship
- **r = 0.3 to 0.7 or -0.3 to -0.7**: Moderate relationship
- **r = 0.0 to 0.3 or -0.0 to -0.3**: Weak relationship

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

Scaling is a technique used in machine learning to make data points closer together and standardize independent features in data. It's usually performed during data pre-processing to handle highly varying values or units. Scaling is important because it can help the model learn and understand the problem more easily, and it can also help the algorithm train faster.

For example, algorithms like SVM and KNN are sensitive to the scale of features hence scaling can be used to speed up convergence in optimization algorithms and ensure fair regularization across features. It's generally applied before using these algorithms to improve their effectiveness.

Normalized Scaling: Normalization (also known as min-max scaling) rescales the feature values to a range of [0, 1] (or sometimes [-1, 1]).

$$X' = \frac{X_{max} - X_{min}}{X - X_{min}}$$

Where,

- X : Original value.
- X': Normalized value.
- X_{min}: Minimum value of the feature.
- X_{max} : Maximum value of the feature.

Standardized Scaling: Standardization *(also known as Z-score normalization) rescales the feature values to have a mean of 0 and a standard deviation of 1.*

$$X' = \frac{X - \mu}{\sigma}$$

Where,

- X : Original value.
- X': Standardized value.
- μ : Mean of the feature.
- σ : Standard deviation of the feature.

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

Variance Inflation Factor (VIF) is a measure used to detect multicollinearity in a regression analysis. VIF quantifies how much the variance of a regression coefficient is inflated due to collinearity with other predictors.

$$VIF(X_i) = \frac{1}{1 - R_i^2}$$

Where R_i^2 is the coefficient of determination of the regression of predictor X_i on all other predictors.

VIF can become infinite when perfect multicollinearity is present. This happens when one predictor variable is an exact linear combination of one or more other predictor variables.

Causes of Infinite VIF

1. If a predictor can be perfectly predicted by a linear combination of other predictors, R_i^2 becomes 1. This makes the denominator $1 - R_i^2$ equal to 0, resulting in an infinite VIF.
2. Including the same variable more than once in the regression model can lead to perfect collinearity.
3. When variables are linearly dependent, such $X_3 = 2 \times X_1 + 3 \times X_2$, perfect multicollinearity occurs, causing infinite VIF.

Example: Consider a regression model with the following variables:

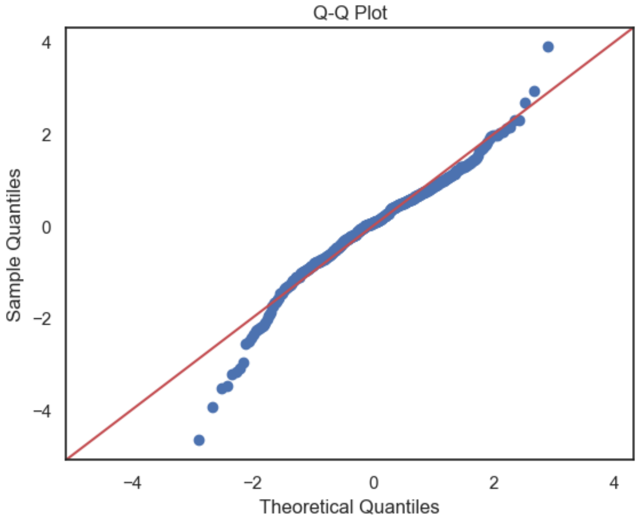
- X_1, X_2 and X_3 where $X_3 = 2 \times X_1 + 3 \times X_2$

Here, X_3 is a perfect linear combination of X_1 and X_2. This lead to $R_3^2 = 1$ and $VIF(X_3) = \frac{1}{1-1} = \infty$

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

A Q-Q plot (Quantile-Quantile plot) is a scatter plot that helps to see if the data follows a normal distribution pattern. In linear regression, we assume that the residuals (differences between observed and predicted values) are normally distributed. A Q-Q plot helps to validate this assumption:

- **Straight Line**: Residuals are normally distributed.
- **Curved Line**: Residuals are not normally distributed meaning that the model has some problems.


Q-Q Plot
Sample Quantiles
4
2
0
−2
−4
Theoretical Quantiles
−4
−2
0
2
4