Bicycle Rental Quantities and their Relation to Daily Conditions by Jalen Moore

August 8, 2022

1 Introduction

In cities, it is commonplace to see bicycle commuters and travelers going about their daily lives. A good amount of these bikes are rentals. The relevant dataset for this is a compilation from Kaggle[1] of the quantity of bike rental data in Washington D.C. from the years 2011 and 2012, and the relevant weather.

The dataset contains a large number of different points of information pertaining to the weather on the given day, and what kind of day was it. The data is structured in the following fields:

- instant: The same idea as an id. The instant/id is not used in this analysis.
- dteday: The full date (YYYY-MM-DD) of which the rest of the row's data refers to.
- season: Designated 1-4 starting from winter (1) up to fall (4).
- yr: The year of which the data pertains to, relative to the start of the dataset. Year 0 designates to 2011, and year 1 designates 2012.
- mnth: The relevant month for the row of data, ranging from 1-12.
- holiday: True of false depending on if that day was a holiday.
- weekday: Designates which day of the week it is, where Sunday is 0 and Saturday is 6.
- workingday: Ture of false depending on if that day was a workday.
- weathersit: The condition of the weather, where 1 means the weather is clear with few clouds, 2 is a misty or cloudy day, and increasing in intensity as the number increases.
- temp: The maximum temperature in Celsius of the day, normalized by a division of 41.
- atemp: The average temperature of the day, in Celsius and normalized.
- hum: The humidity of the given day, ranging from not humid (0) to very humid (1).
- windspeed: The windspeed of the day, normalized with an unknown divisor, and ranging from 0 to 1.
- casual: The number of casual rental bicycles rented.
- registered: The number of registered bicycles of the day.
- cnt: The total number of bicycles on a given day.

In analyzing all this data, the Python (v3.9.7) programming language will be utilized within the Jupyter Notebook environment. Throughout this report, code snippets will show how a number or model was derived. A method will only be defined once in this report, although the method may appear in other code snippets later in the report from which they were declared. The code snippet below shows the Python libraries and packages used throughout the report. These libraries **will not be explained** in this report unless necessary. The snippet also declares the relavent dataset

[1] https://www.kaggle.com/datasets/archit9406/bike-sharing

within a Pandas DataFrame object variable.

[ ]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math as math
import statistics as s
from scipy.stats import norm
import seaborn as sns

# the variable that is storing the data set.
data = pd.read_csv("dataset/day.csv")

# un-normalize temperature (to better understand data)
data['temp'] = data['temp'] * 41
data['atemp'] = data['atemp'] * 41

# rename some columns from original dataset to display better in plots
data.rename(columns = {
    'mnth': 'Month',
    'holiday': 'Holiday',
    'weekday': 'Week Day',
    'workingday': 'Working Day',
    'temp': 'Temperature',
    'atemp': 'Average Temperature',
    'hum': 'Humidity',
    'windspeed': 'Wind Speed',
    'cnt': 'Rental Count'
}, inplace=True)

# add years and seasons with proper values to data
years = [2011, 2012]
season = ["", "Winter", "Spring", "Summer", "Fall"]
df = pd.DataFrame(data={"Year": data["yr"].map(lambda x: years[x]), "Season":␣
  ↪data["season"].map(lambda x: season[x])})
data = data.join(df)
```

2 Research Questions

Throughout this report, the above-described dataset will be analyzed utilizing a plethora of different statistical models and methods. The goal is to determine whether daily conditions such as weather or season impact the quantity of bicycle rentals in Washington D.C., and if a correlation or significance can be determined, how the analysis can be utilized to predict future rental quantities for businesses in the industry.

3 Descriptive Statistics

The dataset will first be analyzed as a whole to better understand the overall characteristics of bicycle renting behaviors in Washington DC. The dataset contains data for 731 concecutive days.

3.1 Basic Overall Description of Bicycle Rental Count

The simplest way of viewing some basic characteristics of this dataset is to view the mean, median, and quartiles of the rented bicycle count.

[ ]:

```python
# computes the mean of a list.
def findmean(data, average=True):
    sum = 0
    for x in data:
        sum = sum + x
    return sum / len(data) if average else sum

# given a mean and list, computes the variance.
def variance(mean, list):
    sum = 0
    total = 0
    for i in range(len(list)):
        diff = list[i] - mean
        s = diff * diff
        sum = sum + s
        total = total + 1
    return sum / total

# find the desired smallest/largest value
def find_bound(list, low=True):
    val = 200000 if low else 0
    for i in range(len(list)):
        cond = (list[i] < val) if low else (list[i] > val)
        if cond:
            val = list[i]
    return val

x_bar = findmean(data["Rental Count"]) # mean
sigma_sq = variance(x_bar, data["Rental Count"]) # variance
sigma = sigma_sq ** 0.5 # standard deviation
low = find_bound(data["Rental Count"]) # lowest rental count
high = find_bound(data["Rental Count"], low=False) # highest rental count
quartiles = s.quantiles(data["Rental Count"], n=4) # array of all quartiles
```

Statistic	Value
Mean	4506.78
Standard Deviation	1935.89
First Quartile	3141
Median	4548
Third Quartile	5976
Lowest Rentals	22
Highest Rentals	8714

Table 1: Statistical description of the total rental count.

Using *Table 1*, a few things can be determined. First, the found average/mean and the standard deviation show that most of the time, the rental quantity will be within the range of $2,571$ and 6443. Second, since the standard deviation is so large, the difference between daily rental quantities could wildly vary. To better visualize this data, a histogram is very useful. A histogram is a plot that describes a given array of data versus the probability of that data happening. Below, a probability histogram is generated using the Seaborn Python data package.

```
[ ]: # plot histogram
     sns.displot(data=data, kind="hist", x="Rental Count", stat="probability",␣
      ↪height=4)
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fad805da4c0>
```

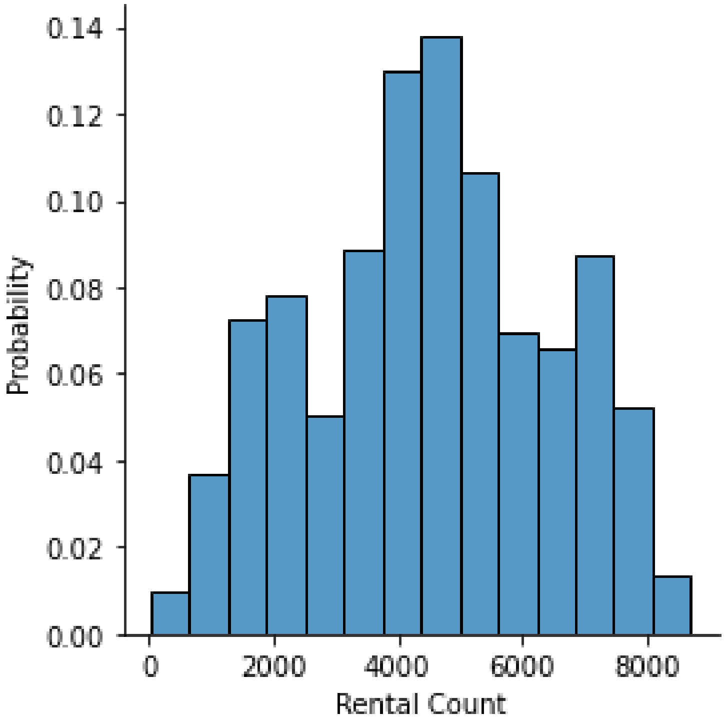

Figure 1: A probability histogram model for the rental count, showing peaks and declines in probability.

Figure 1 re-affirms the idea that the most likely rental count on a given day would be within the $2,000$ to $7,000$ range. The reason this histogram uses a probability y-axis as opposed to a quantity based one has to do with useful-ness. Since the dataset is quite large, a quantity count would not mean much. A probability gives a better feel for how likely a given rental count is to occur over this two year period. For example, the likely hood that a day will have a bike rental count of roughly $5,000$ is 14%. This also means that within the span of two years, 14% of that time have a daily bike rental quantity of around $5,000$. With this model, a good idea is given of what a daily behavior for rental quantity will be.

3.2 Finding Relevant Features

The statistical description given in the previous section is great and all, but it would be better to find a correlation between the rental count and other features in this dataset. If other daily features can be determined to correlate with the bicycle rentals per day, then the company and investors who profit off the service will be able to better predict future profit margins. In some cases, if the correlated feature is controllable, then the business can change the feature accordingly to increase their profits.

A useful model that will be utilized to aid in finding these correlated features of the dataset is a heat map. This heat map will take each given feature in the dataset and visualize it in a grid, much like a multiplication table. Each grid value in the heat map will show as a color hue representing the correlation coefficient of the row and column features. A heat map is useful to see which points of the dataset are worth comparing to others. An example of this can be seen below in *Figure 4*.

```
[ ]: # heat map
     sub_frame = data.filter(items=["Rental Count", "Temperature", "Average␣
      ↪Temperature", "Wind Speed", "Holiday", "Week Day", "Working Day",␣
      ↪"Humidity", "Wind Speed", "yr", "season"])
     sns.heatmap(round(sub_frame.corr(), 2), annot=False)
```

```
[ ]: <AxesSubplot:>
```

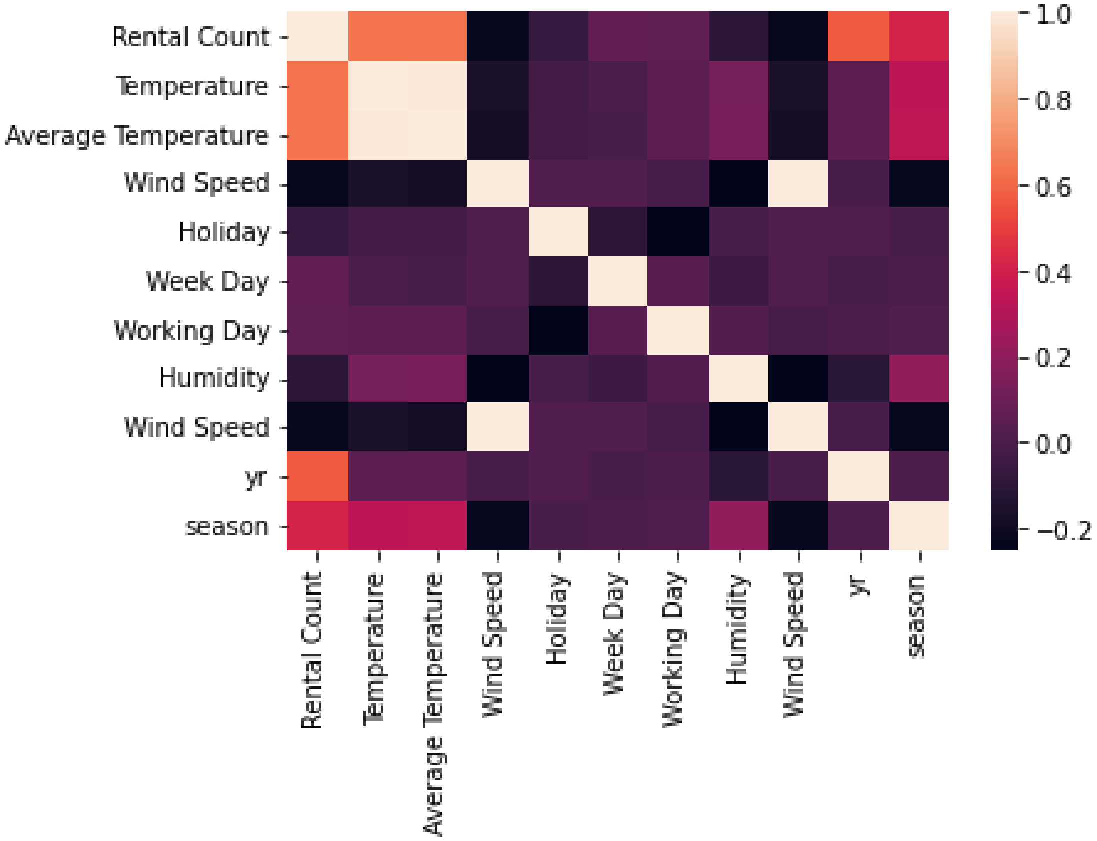

Figure 2: Heat Map of correlation for the primary features of the dataset.

From this heat map, a clear path is given for the rest of this statistical analysis. It is shown that the rental count for bicycles is most correlated with the temperature on a given day, with a correlation coefficient of 0.63. Furthermore, it is shown that the analysis previously done with the year and season is somewhat correct. Since both the year and season somewhat correlate with the rental count, the features can still be deemed relevant to the research question. More work in finding useful connections between the rental count and these features will be covered in the Inferential Statistics section. For now, however, these correlations will be described, rather than infered why hypotheses.

3.3 Analyzing Relevant Features

3.3.1 Daily Temperatures

The most important feature to analyze is the max and average temperature on each day in the data set. To better visualize how bike rental counts correlate with the average temperature, a scatter plot will be created. A scatter plot is simply a plot of each data point where (in this case) the x-axis is the rental count and the y-axis is the average temperature. The scatterplot is generated as:

[ ]:

```
# scatter plot of rental count vs average temperature
sns.scatterplot(y = "Rental Count",
                x = "Average Temperature",
                data = data,
                hue = "Season")
```

[ ]:

```
<AxesSubplot:xlabel='Average Temperature', ylabel='Rental Count'>
```

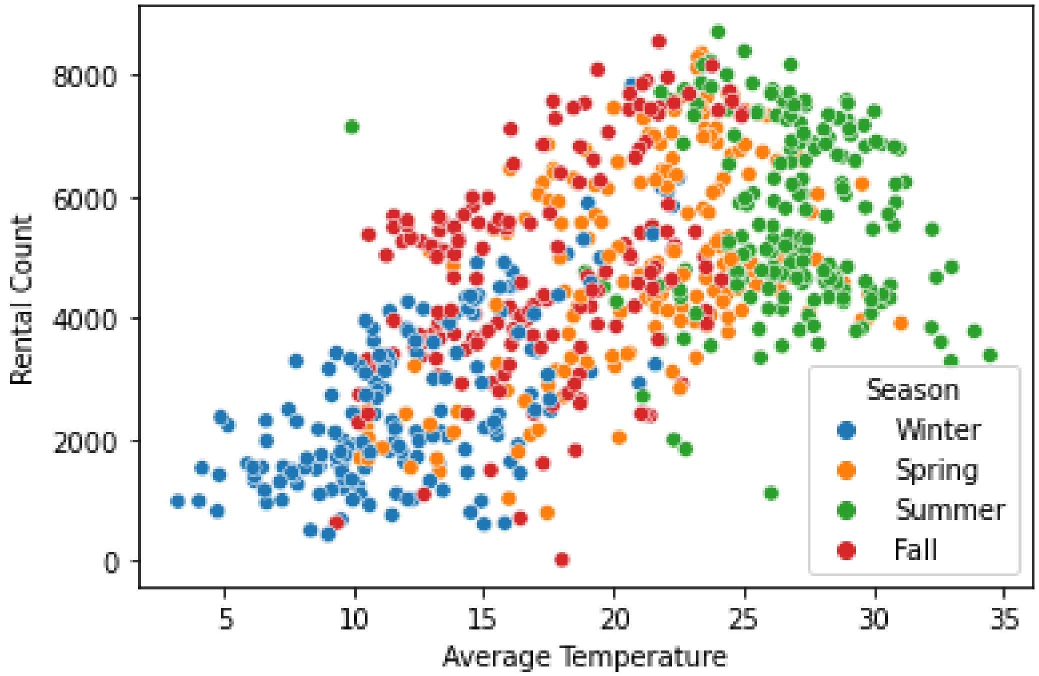

Figure 3: A scatter plot of the rental count versus the average temperature, hued to the different seasons.

The scatterplot in Figure 3 shows the positive linear correlation between rental count and the average temperature on a day. It can be determined from this plot that as the average temperature increases, so too does the rental count. This is only to a certain point, as the best performing days were the ones around 20 to 30 Celsius. Any temperature higher or lower than this range performs worse. To better view this linear correlation, a linear regression can be used to determine a likely rental count at a given temperature. A linear regression, also known as a line of best fit, is an attempt to approximize and find a linear relationship between two features of data within the linear equation $y = mx + b$. The linear regression for the temperature and rental count is generated below.

[ ]:

```
# linear regression for rental count vs average temperature with respect to␣
 ↪seasons.
sns.lmplot(x="Average Temperature", y="Rental Count", data=data, col="Season",␣
 ↪ci=False, col_wrap=2, height=4)
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fad807fa8b0>
```

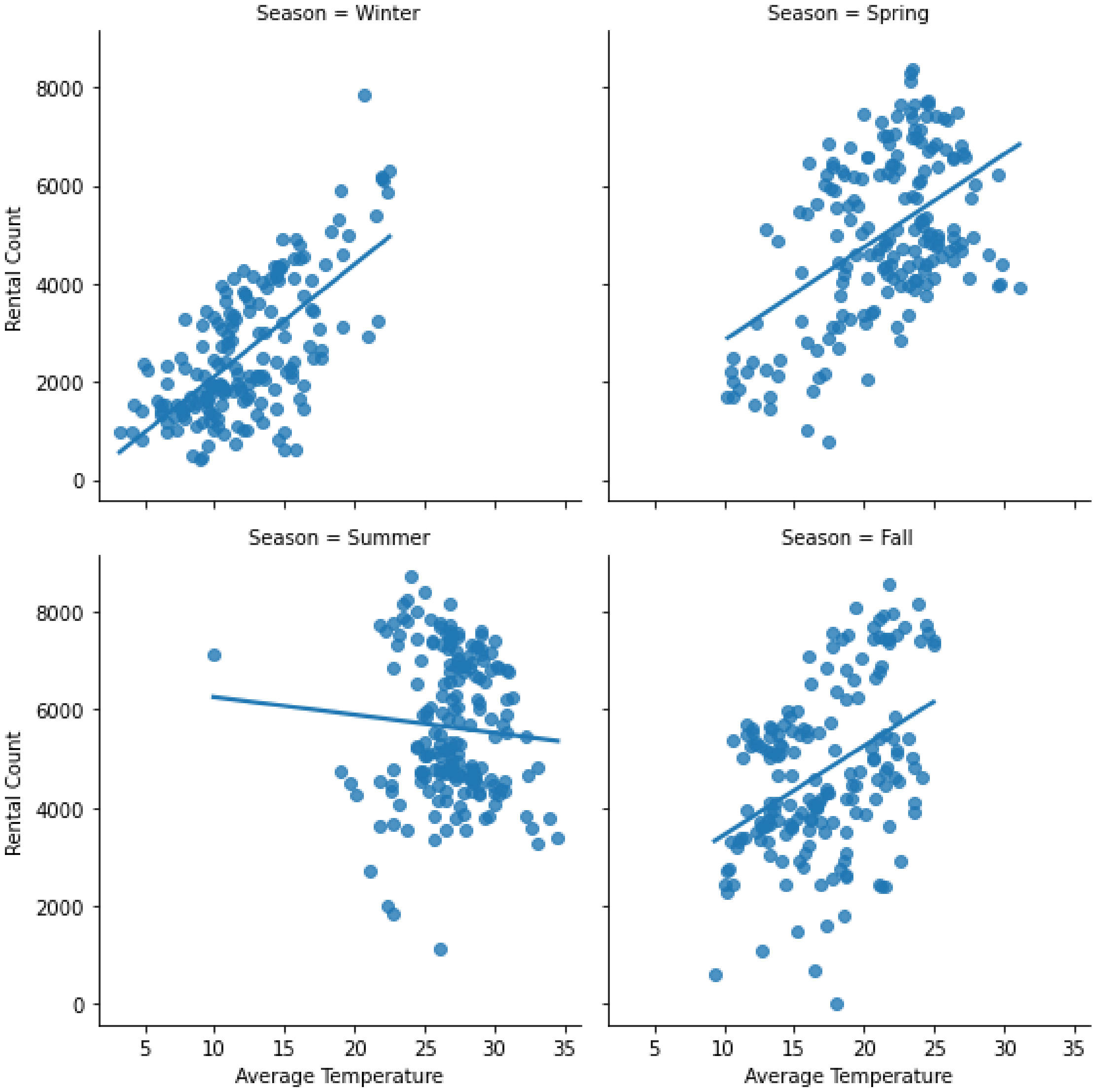

Figure 4: Linear regression for each season and the average temperature versus the rental count.

Using these linear regressions, each season's linear relation can be viewed. In nearly all of the seasons, the temperature has a positive correlation with the bicycle rental count. In contrast, the summer has a slightly negative correlation with the rental count. This further emphasizes what was found in the scatter plot. It is found that temperatures generally improve the rental count of bicycles, until the temperatures hit a threshold, at which the bicycles then start declining in count. If this seems more-so like a curve or polynomial, its because it is. Since the data follows the trend of growing until a peak before declining again, a binomial regression (order x^2) is the perfect way to visualize this relationship.

```
[ ]: # binomial regression for rental count vs average temperature with respect to␣
      ↪seasons.
     sns.lmplot(x="Average Temperature", y="Rental Count", order=2, data=data,␣
      ↪col="Season", ci=False, col_wrap=2, height=4)
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fad90612430>
```

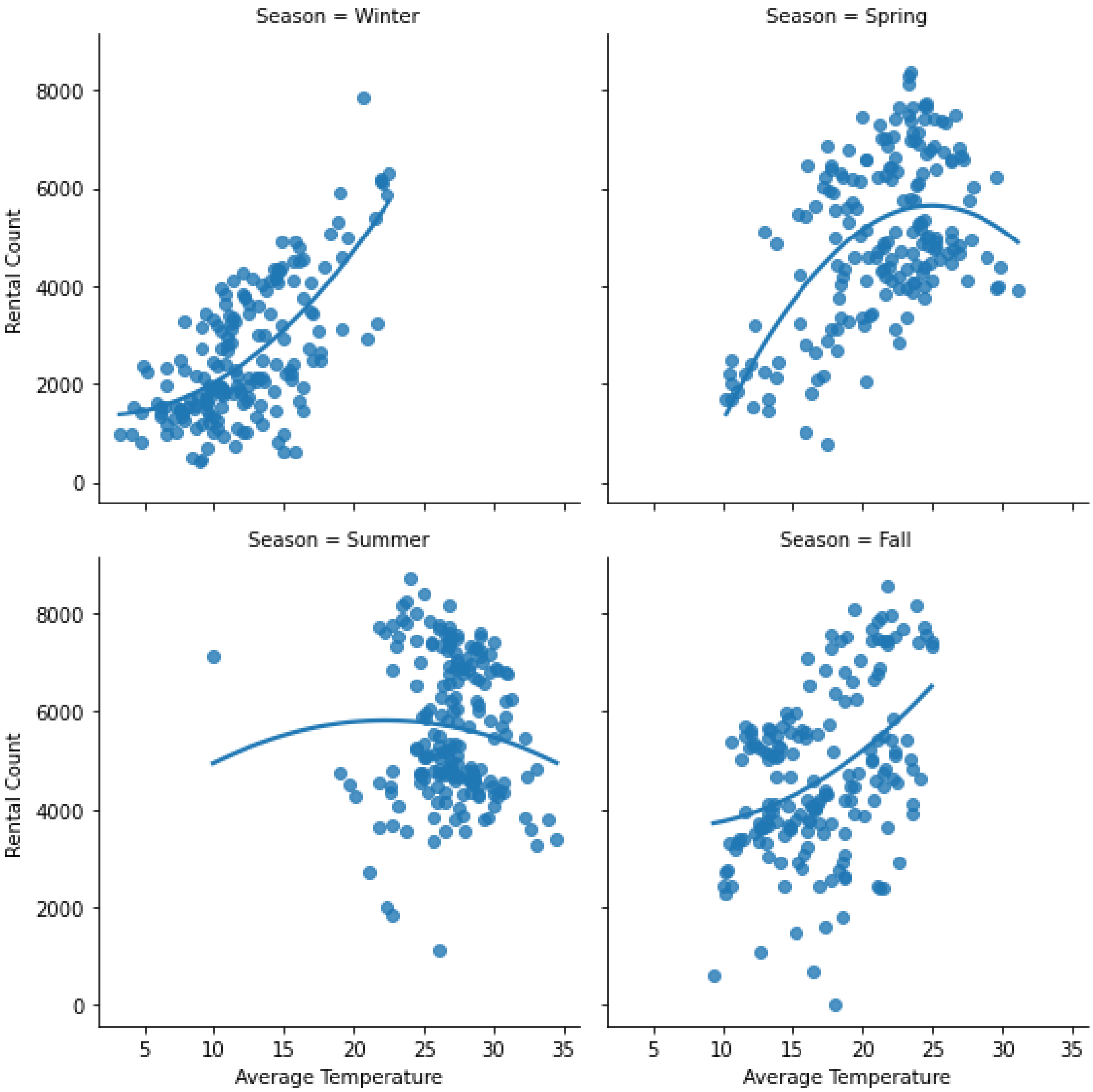

Figure 5: Binomial regression for each season and the average temperature versus the rental count.

The generated binomial regression is a much better representation of this data. It can be seen from each season that at as the average temperature grows, so too does the rental count in a more dynamic way. At the peak of 25 degrees Celsius, the rental count starts declining. If data was available up to 50 degrees, this decline would likely continue. It is therefore plausible to say that in the future, a day around 25 degrees will be most profitable without accounting for external business factors.

3.3.2 Years and Seasons

Now that a good understanding of the temperature data has been acquired, it is important to understand these numbers based on purely the season and year. Using the seasons as a grouping mechanism is an easy way to understand how each part of the behaves with respect to bicycle rental counts. A facet grid model will be utilized to quickly understand and view the differences between the four seasons. Facet grids are simply a grid layout of histograms. The grid columns and rows are designated as a data category, and each histogram takes the given independent variable x and shows its distribution. Below is the relevant facet grid for rental quantities with respect to the season and year.

```
[ ]: # facet grid of seasons
     sns.FacetGrid(data, col="Season", row="Year").map(sns.histplot, "Rental Count",␣
      ↪stat="probability")
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fad72d00130>
```

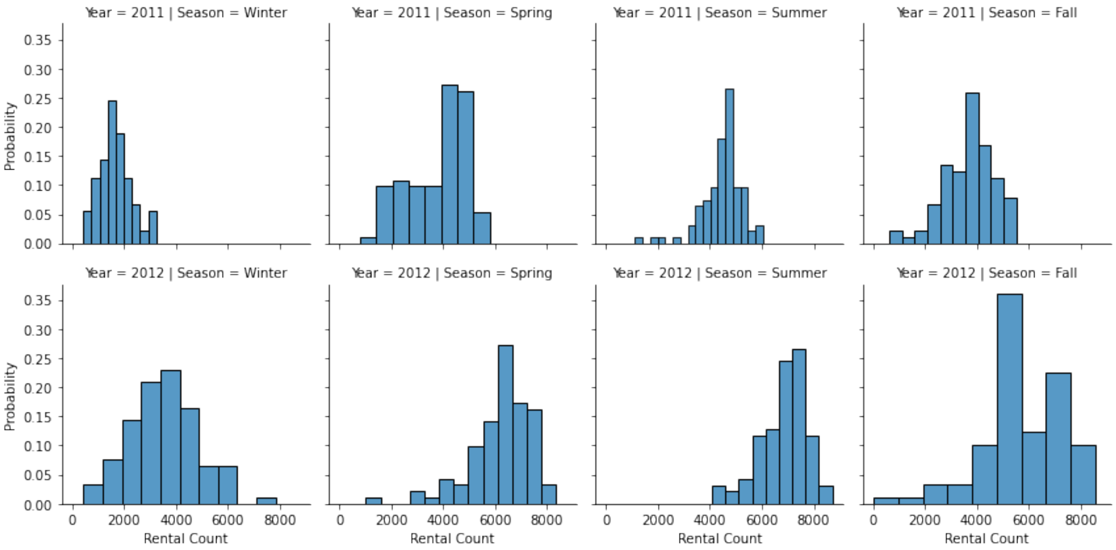

Figure 6: A facet grid for the bicycle rental count with respect to the season and year.

The above facet grid shows the histograms for a number of different conditions. This grid can be interpretted as follows. The rows designate the years, while the columns designate the season in which the count was made.

Utilizing the facet grid model, it can be noticed that winter typically has the least amount of rental counts, while summer and fall have the highest probable rental counts. It is also notice-able that the behavior wildly varies depending on the year in which the count was taken. The probable rental count in winter of 2011 was much lower than that of winter in 2012. The same can be said for each season, as 2012 had higher probable rental counts in general as opposed to in 2011.

4 Inferential Statistics

Now that the dataset has been described as a whole, the data will now be tested. The test used will be a chi-square test (test of independence) to determine whether different features are statisically related. This method will be utilized to prove the relationships between the rental count and season.

4.1 Relationship Between Rentals and Seasons

In order to test how statistically significant the correlation between rentals and temperatures is, both tests will be conducted. The first test is the test of independence. The null hypothesis (H_0) will be that rental count and the current season are not related. The alternative hypothesis (H_a) will be the opposite; The rental count and the current season are related and are statistically significant. Before finding whether these two are statisically significant, a contigency table should be made. To make this table, the rental count feature will be split into two categories: <= mean and > mean, where the mean is the mean/average of the rental count. With the categories now set, the following contigency table can be made:

[ ]:

```
# calculate rental and season contigency table
seasonal_data = data.filter(items=['season', 'Rental Count'])
# filter rental count by seasons
winter = seasonal_data[seasonal_data['season'] == 1].filter(items=['Rental␣
 ↪Count'])
spring = seasonal_data[seasonal_data['season'] == 2].filter(items=['Rental␣
 ↪Count'])
summer = seasonal_data[seasonal_data['season'] == 3].filter(items=['Rental␣
 ↪Count'])
fall = seasonal_data[seasonal_data['season'] == 4].filter(items=['Rental␣
 ↪Count'])

# get condition categories
greater = np.array([
    np.sum(winter.to_numpy() > x_bar),
    np.sum(spring.to_numpy() > x_bar),
    np.sum(summer.to_numpy() > x_bar),
    np.sum(fall.to_numpy() > x_bar),
])

less_eq = np.array([
    np.sum(winter.to_numpy() <= x_bar),
    np.sum(spring.to_numpy() <= x_bar),
    np.sum(summer.to_numpy() <= x_bar),
    np.sum(fall.to_numpy() <= x_bar),
])

# get totals
greater = np.append(greater, sum(greater))
less_eq = np.append(less_eq, sum(less_eq))
```

```
d = {
    'Season': ['Winter', 'Spring', 'Summer', 'Fall', 'Total'],
    'Greater Than Mean': greater,
    'Less Than/Equals Mean': less_eq,
    'Total': greater + less_eq,
}
```

Season	Greater Than Mean	Less Than/Equals Mean	Total
Winter	18	163	181
Spring	114	70	184
Summer	147	41	188
Fall	93	85	178
Total	372	359	731

Table 2: The contingency table for seasons versus the number of days above or below the rental mean.

The contingency table computed using the code snippet above shows that as seasons change, so too does the days above the rental average. On its own, however, it cannot be determined whether this relationship is statistically significant or not. A test statistic must be made using the *expected value* of each of these seasonal and average conditions. The expected value is calculated by multiplying the obsereved and total values of a season and for its condition, and dividing by the total number of days. With this in mind, the new table of expected values becomes:

Season	Expected Greater Than Mean	Expected Less Than/Equals Mean	Total
Winter	92.11	88.89	181
Spring	93.64	90.36	184
Summer	95.67	92.32	188
Fall	90.58	87.42	178
Total	372	359	731

Table 3: The contingency table for the seasons versus the number of extepecteddays above or below the rental mean.

Now that the expected values are known, the test statistic can be calculated as the summation of the quantity: observed minus expected value squared over the expected value. This summation is to be done for both conditional categories (greater or less), as the test statistic will be the same regardless. The resulting table from calculating the test statistic can be seen below.

Season	Outcome	Observed - Expected	$(O-E)^2$	$(O-E)^2/E$
Winter	Greater	-74.11	5492.29	59.63
	Less/Equal	74.11	^	61.79
Spring	Greater	20.36	414.53	4.43
	Less/Equal	-20.36	^	4.59

Season	Outcome	Observed - Expected	$(O-E)^2$	$(O-E)^2/E$
Summer	Greater	51.33	2634.77	27.54
	Less/Equal	-51.33	$\frown$	28.54
Fall	Greater	2.42	5.86	0.06
	Less/Equal	-2.42	$\frown$	0.07

Table 4: The calculation table for the chi-square test statistic. $\frown$ *means the same values as the above cell.*

By finding the summation of last column of the table, the test statistic can be found. The test statistic therefore is 186.65. The last step of the chi-square test is to find the critical value to compare the test statistic to. To find the critical value, the degrees of freedom and significance level must be found or defined. The degrees of freedom is found as the quantity number of seasons minus one multiplied by the quantity number of rental groups minus one. In this case, there are four seasons and two rental groups, so there are 3 degrees of freedom. By setting the significance level to 0.05, the final critical value can be found using a chi-square value table. The resulting critical value is 7.815. Finally, since the test statistic of 186.65 is much larger than that of the critical value of 7.815, the null hypothesis can be rejected.

Since the null hypothesis is rejected, this means that the rental count and the season are statistically significant to one another. This means that a season impacts the likely hood that the daily rental count is above average.

5 Conclusion

From the analysis above, a few features were found to be useful in predicting bicycle rental quantities per day. The first feature that was, at first glance not very useful was the seasons feature. Viewing the Figure 2 heat map, it is shown that the correlation between seasons and rental count is there but quite small. It is not until Figure 4 and 5that a difference between seasons becomes apparent. It is shown in these facet grids that as seasons change, so too does the rental count, if only by a small margin. Using the binomial regression, it can be seen the in the spring and summer, the rental count is most likely to peak. It can also be seen that the winter rental count will likely be smaller than the rest of the year. This relationship is proven to be true through the chi-square test. The test showed that on a given day, the rental count being above/below average is related to the current season. Tied with the binomial regression, it is found that the spring and summer are the best times for above average bicycle renting. This information is useful for businesses in this industry as future rental counts can be roughly predicted. This count impacts the business' expense schedule and profit margin, and allows them to better understand how their business will survive under the same conditions.

The second feature that was shown to be relevant from the Figure 2 heat map was average temperature of the day. With a higher correlation coefficient, it was the first feature to be analyzed. Using both a linear and binomial regression, it was found that the relationship between the rental count and temperature was a binomial that peaks at around 25 degrees Celsius. This data is relevant, as a more short term prediction can be made on the rental count for the next day or week based on external weather forecasts. A business can determine that if tomorrow is 30 degrees Celsius, the number of bicycles rented would be roughly 5000 with an error of around 2000. While the error is quite large, it is certainly the start of a more accurate statistic that a business could create. The relationship between rented bicycles and the average temperature is there however.

Throughout this report, the effects of the average temperature and the current season on the number of rented bicycles in a day have been analyzed. The analysis shows that these two features do impact the rent quantity. Using this information, businesses can make rough long and short term predictions on their bicycles. More importantly however, businesses can determine their profits that would be of use to both the company and their investors. This report is not definitive, as the relationship between these features and the bicycle quantity is rough at best, but it is a start to a much more robust analysis that could be used for business predictions.