0.0.1 Question 0

Question 0A What is the granularity of the data (i.e. what does each row represent)?

Each row represents a ride taken on a specific day, each hour after the previous, and provides data about that current day (weather stats).

Question 0B For this assignment, we'll be using this data to study bike usage in Washington D.C. Based on the granularity and the variables present in the data, what might some limitations of using this data be? What are two additional data categories/variables that you can collect to address some of these limitations?

The data is really only about what the type of weather was like on a certain day (but only the temperature) and if the day was a holiday. Since we are trying to study bike usage, maybe we could also see what type of people make up the majority of bike users such as collecting age or gender be.

0.0.2 Question 2

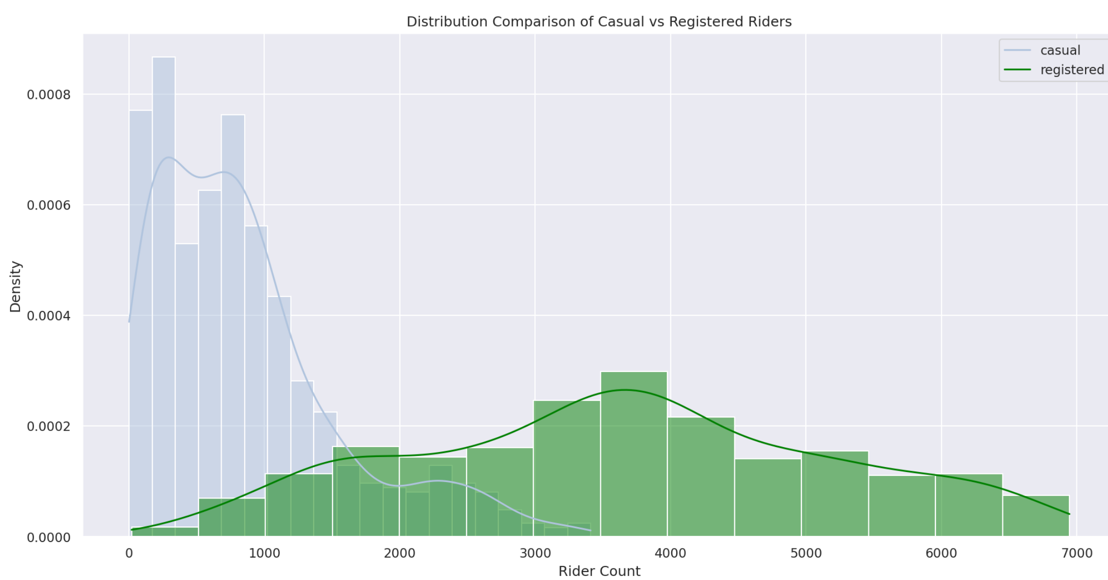
Question 2a Use the `sns.histplot` function to create a plot that overlays the distribution of the daily counts of bike users, using blue to represent `casual` riders, and green to represent `registered` riders. The temporal granularity of the records should be daily counts, which you should have after completing question 1c.

Hint: You will need to set the `stat` parameter appropriately to match the desired plot.

Include a legend, xlabel, ylabel, and title. Read the seaborn plotting tutorial if you're not sure how to add these. After creating the plot, look at it and make sure you understand what the plot is actually telling us, e.g on a given day, the most likely number of registered riders we expect is ~4000, but it could be anywhere from nearly 0 to 7000.

```
In [17]: sns.histplot(daily_counts, x="casual", color = "lightsteelblue", stat="density", kde=True)
         sns.histplot(daily_counts, x="registered", color = "green", stat="density", kde=True)
         plt.legend(labels=["casual", "registered"])

         plt.xlabel("Rider Count")
         plt.title("Distribution Comparison of Casual vs Registered Riders")
```

```
Out[17]: Text(0.5, 1.0, 'Distribution Comparison of Casual vs Registered Riders')
```



0.0.3 Question 2b

In the cell below, descibe the differences you notice between the density curves for casual and registered riders. Consider concepts such as modes, symmetry, skewness, tails, gaps and outliers. Include a comment on the spread of the distributions.

The spread of the registered riders is actually much larger than the spread of the casual riders which is probably due to the fact that non casual riders might register without actually biking. The casual riders is skewed left with there being most riders in the 0-1000 rider count showing there actually is not a lot of casual riders.

0.0.4 Question 2c

The density plots do not show us how the counts for registered and casual riders vary together. Use `sns.lmplot` to make a scatter plot to investigate the relationship between casual and registered counts. This time, let's use the `bike` DataFrame to plot hourly counts instead of daily counts.

The `lmplot` function will also try to draw a linear regression line (just as you saw in Data 8). Color the points in the scatterplot according to whether or not the day is a working day (your colors do not have to match ours exactly, but they should be different based on whether the day is a working day).

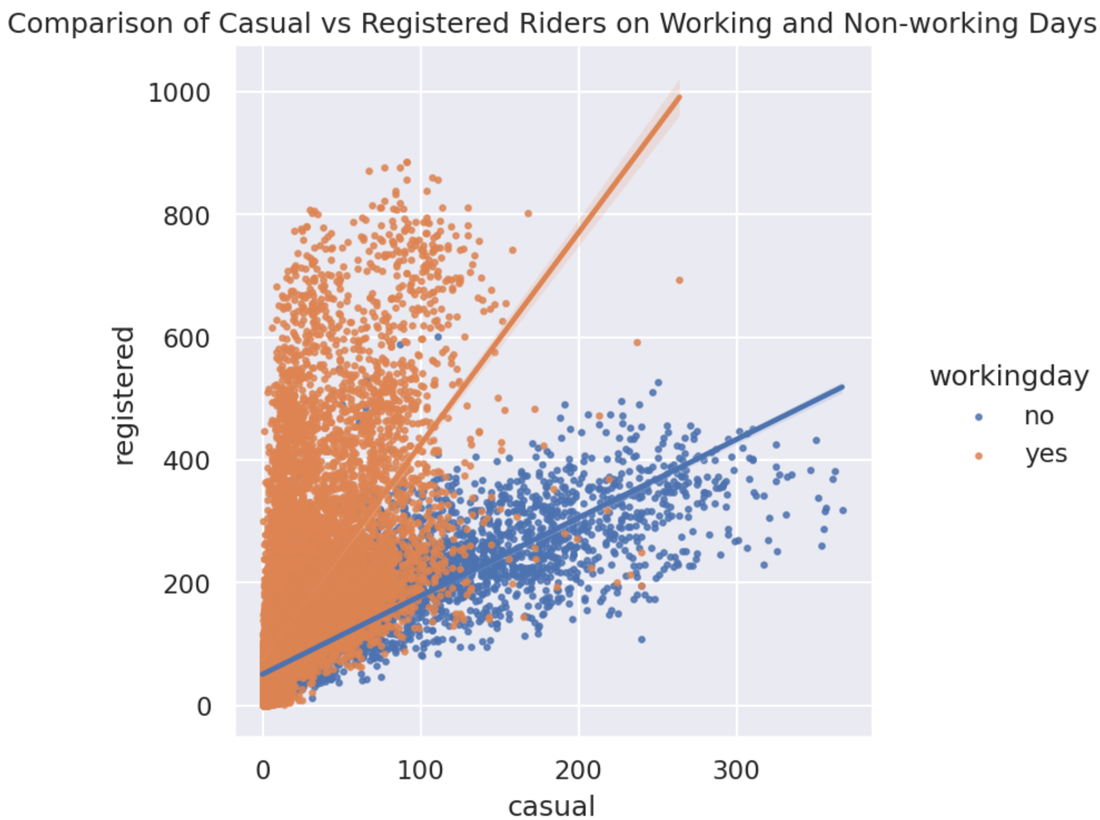
There are many points in the scatter plot, so make them small to help reduce overplotting. Also make sure to set `fit_reg=True` to generate the linear regression line. You can set the `height` parameter if you want to adjust the size of the `lmplot`.

Hints: * Checkout this helpful tutorial on `lmplot`.

- You will need to set `x`, `y`, and `hue` and the `scatter_kws` in the `sns.lmplot` call.
- You will need to call `plt.title` to add a title for the graph.

In [18]:
```
# Make the font size a bit bigger
sns.set(font_scale=1)
sns.lmplot(data=bike, x="casual", y="registered", scatter_kws={"s": 5}, hue="workingday")
plt.title("Comparison of Casual vs Registered Riders on Working and Non-working Days")
plt.ylabel("registered")
plt.xlabel("casual")
```

Out[18]:
```
Text(0.5, 50.99999999999999, 'casual')
```


Comparison of Casual vs Registered Riders on Working and Non-working Days
registered
casual
0
200
400
600
800
1000
100
300
workingday
no
yes

0.0.5 Question 2d

What does this scatterplot seem to reveal about the relationship (if any) between casual and registered riders and whether or not the day is on the weekend? What effect does overplotting have on your ability to describe this relationship?

Casual riders seem to ride more on the weekends while there seems to be more riders registered to ride on the workingdays. The fact that there are so many points clustered up together makes this plot a little hard to describe as I cant really see any outliers and in some ways I feel as if I am just looking at a clump of colors.

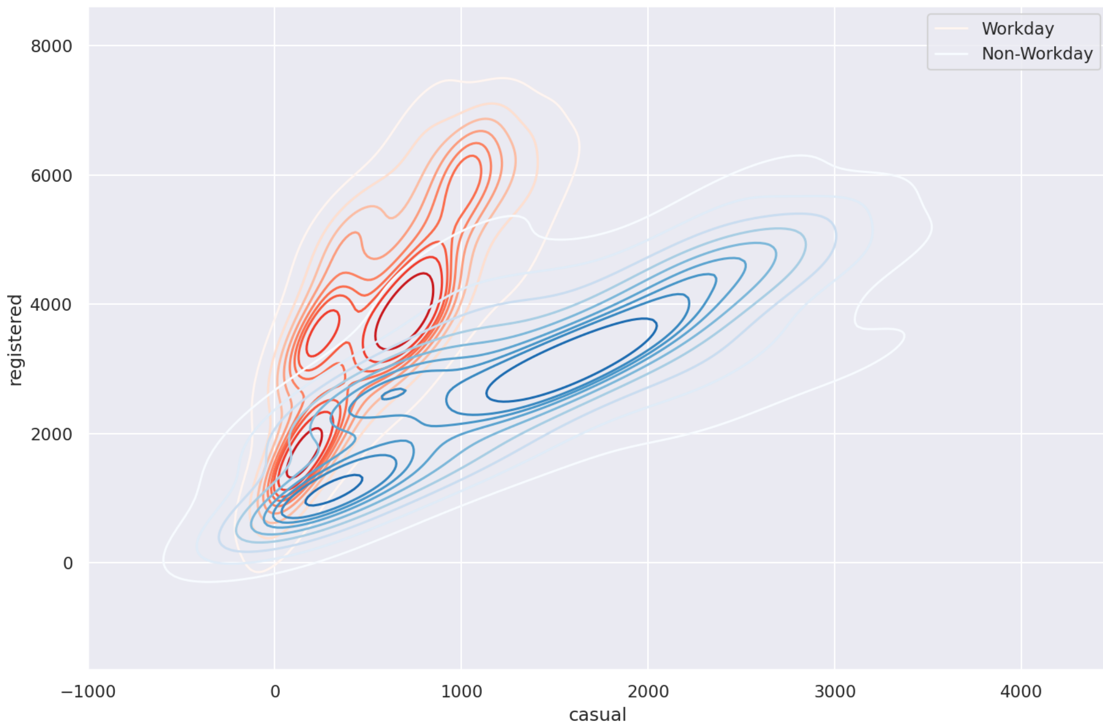
Generating the plot with weekend and weekday separated can be complicated so we will provide a walkthrough below, feel free to use whatever method you wish if you do not want to follow the walkthrough.

Hints: * You can use `loc` with a boolean array and column names at the same time * You will need to call kdeplot twice, each time drawing different data from the `daily_counts` table. * Check out this guide to see an example of how to create a legend. In particular, look at how the example in the guide makes use of the `label` argument in the call to `plt.plot()` and what the `plt.legend()` call does. This is a good exercise to learn how to use examples to get the look you want. * You will want to set the `cmap` parameter of `kdeplot` to `"Reds"` and `"Blues"` (or whatever two contrasting colors you'd like), and also set the `label` parameter to address which type of day you want to plot. You are required for this question to use two sets of contrasting colors for your plots.

After you get your plot working, experiment by setting `shade=True` in `kdeplot` to see the difference between the shaded and unshaded version. Please submit your work with `shade=False`.

In [20]:
```
# Set the figure size for the plot
plt.figure(figsize=(12,8))

# Set 'is_workingday' to a boolean array that is true for all working_days
is_workingday = daily_counts["workingday"].str.contains("yes")

# Bivariate KDEs require two data inputs.
# In this case, we will need the daily counts for casual and registered riders on workdays
# Hint: consider using the .loc method here.
casual_workday = daily_counts.loc[is_workingday]["casual"]
registered_workday = daily_counts.loc[is_workingday]["registered"]

# Use sns.kdeplot on the two variables above to plot the bivariate KDE for weekday rides
sns.kdeplot(x=casual_workday, y=registered_workday, cmap="Reds")

not_workingday = daily_counts["workingday"].str.contains("no")
# Repeat the same steps above but for rows corresponding to non-workingdays
# Hint: Again, consider using the .loc method here.
casual_non_workday = daily_counts.loc[not_workingday]["casual"]
registered_non_workday = daily_counts.loc[not_workingday]["registered"]

# Use sns.kdeplot on the two variables above to plot the bivariate KDE for non-workingday ride
sns.kdeplot(x=casual_non_workday, y=registered_non_workday, cmap="Blues")
plt.legend(labels=["Workday", "Non-Workday"])
```

Out[20]:
```
<matplotlib.legend.Legend at 0x7f33c2110340>
```



Workday
Non-Workday
8000
6000
4000
2000
0
registered
−1000
0
1000
2000
3000
4000
casual

Question 3bi In your own words, describe what the lines and the color shades of the lines signify about the data.

The color shades that are the brightest have more data points in that specific area and in some ways seem almost more important than lines with light or even a white color. Colors represent the density at certain levels.

Question 3bii What additional details can you identify from this contour plot that were difficult to determine from the scatter plot?

Overall, casual riders tend to ride a lot more on non weekdays and the data is centered around 0-2000 riders on non workdays. Using the white lines we are able to see where the density is a lot lower compared to the higher density areas.

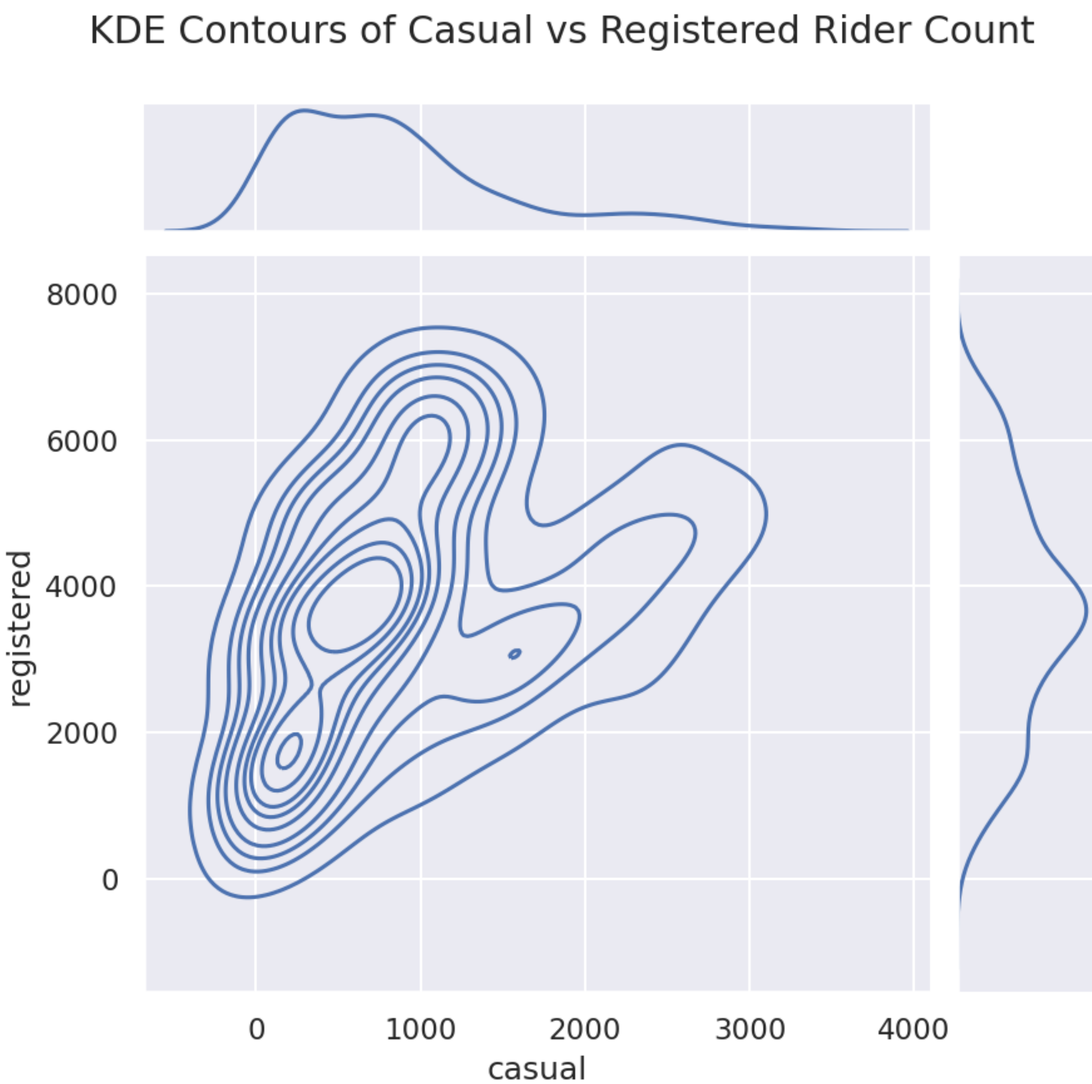
0.1 4: Joint Plot

As an alternative approach to visualizing the data, construct the following set of three plots where the main plot shows the contours of the kernel density estimate of daily counts for registered and casual riders plotted together, and the two "margin" plots (at the top and right of the figure) provide the univariate kernel density estimate of each of these variables. Note that this plot makes it harder see the linear relationships between casual and registered for the two different conditions (weekday vs. weekend).

Hints: * The seaborn plotting tutorial has examples that may be helpful. * Take a look at `sns.jointplot` and its `kind` parameter. * `set_axis_labels` can be used to rename axes on the contour plot.

Note: * At the end of the cell, we called `plt.suptitle` to set a custom location for the title. * We also called `plt.subplots_adjust(top=0.9)` in case your title overlaps with your plot.

In [21]:
```
sns.jointplot(data=daily_counts, x="casual", y="registered", kind="kde")
plt.ylabel("Daily Count Registered Riders")
plt.ylabel("Daily Count Casual Riders")

plt.suptitle("KDE Contours of Casual vs Registered Rider Count")
plt.subplots_adjust(top=0.9);
```


KDE Contours of Casual vs Registered Rider Count
8000
6000
4000
2000
0
registered
0
1000
2000
3000
4000
casual

0.2 5: Understanding Daily Patterns

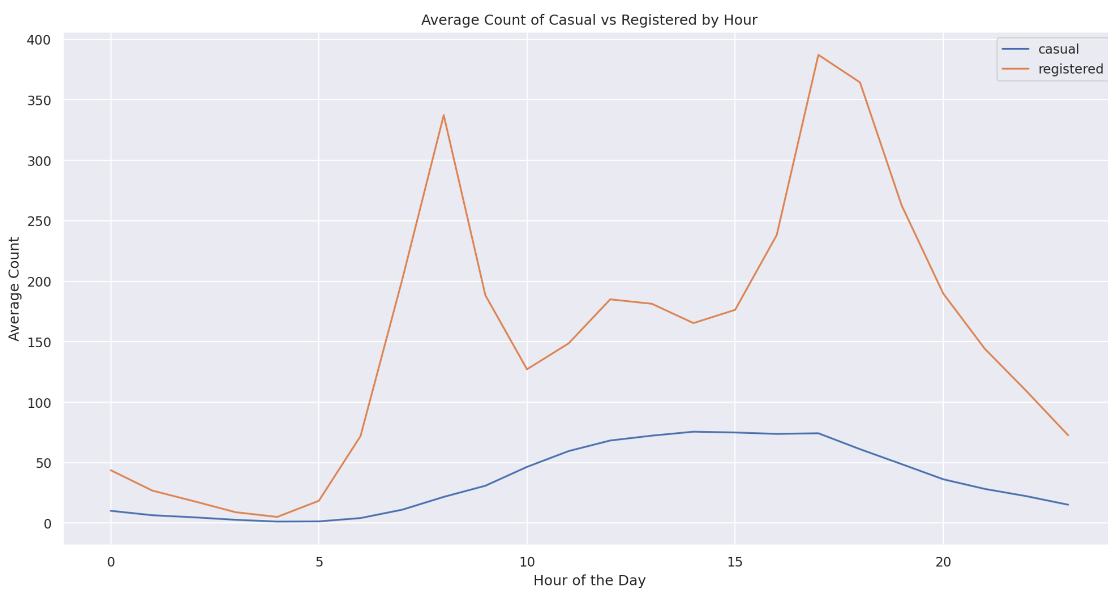
0.2.1 Question 5

Question 5a Let's examine the behavior of riders by plotting the average number of riders for each hour of the day over the **entire dataset**, stratified by rider type.

Your plot should look like the plot below. While we don't expect your plot's colors to match ours exactly, your plot should have different colored lines for different kinds of riders.

```
In [22]: sns.lineplot(data=bike, x="hr", y="casual", err_style=None)
         sns.lineplot(data=bike, x="hr", y="registered", err_style=None)
         plt.xlabel("Hour of the Day")
         plt.ylabel("Average Count")
         plt.title("Average Count of Casual vs Registered by Hour")
         plt.legend(labels=["casual", "registered"])
```

```
Out[22]: <matplotlib.legend.Legend at 0x7f33c214fd90>
```



Question 5b What can you observe from the plot? Hypothesize about the meaning of the peaks in the registered riders' distribution.

Registered riders peak around hour 7-8 and around hour 17. This is probably due to riders registering to ride to work or back from work. Meanwhile the casual riders is much more uniform with more ridership during the day and less during the night and early morning hours.

In our case with the bike ridership data, we want 7 curves, one for each day of the week. The x-axis will be the temperature and the y-axis will be a smoothed version of the proportion of casual riders.

You should use `statsmodels.nonparametric.smoothers_lowess.lowess` just like the example above. Unlike the example above, plot ONLY the lowess curve. Do not plot the actual data, which would result in overplotting. For this problem, the simplest way is to use a loop.

You do not need to match the colors on our sample plot as long as the colors in your plot make it easy to distinguish which day they represent.

Hints: * Start by just plotting only one day of the week to make sure you can do that first.

- The `lowess` function expects y coordinate first, then x coordinate. You should also set the `return_sorted` field to `False`.
- Look at the top of this homework notebook for a description of the temperature field to know how to convert to Fahrenheit. By default, the temperature field ranges from 0.0 to 1.0. In case you need it, Fahrenheit = Celsius $* \frac{9}{5} + 32$.

Note: If you prefer plotting temperatures in Celsius, that's fine as well!

In [ ]:
```
from statsmodels.nonparametric.smoothers_lowess import lowess

plt.figure(figsize=(10,8))
#bike.loc[bike["weekday"] == "Fri"]["prop_casual"]
xs = bike["weekday"].unique()
for x in xs:
    prop = bike.loc[bike["weekday"] == x]["prop_casual"]
    temp = bike.loc[bike["weekday"] == x]["temp"] * 100
    ysmooth = lowess(prop, temp, return_sorted=False)
    sns.lineplot(temp, ysmooth, label=x)

plt.ylabel("Casual Rider Proportion")
plt.xlabel("Temperature (Celsius)")
plt.title("Temperature vs Casual Rider Proportion per weekday")
```

Question 6c What do you see from the curve plot? How is `prop_casual` changing as a function of temperature? Do you notice anything else interesting?

We see that as the temperature increases so does the proportion of casual riders. Also as the temperature goes up there are a lot more riders (casual) on the weekend.

0.2.2 Question 7

Question 7A Imagine you are working for a Bike Sharing Company that collaborates with city planners, transportation agencies, and policy makers in order to implement bike sharing in a city. These stakeholders would like to reduce congestion and lower transportation costs. They also want to ensure the bike sharing program is implemented equitably. In this sense, equity is a social value that is informing the deployment and assessment of your bike sharing technology.

Equity in transportation includes: improving the ability of people of different socio-economic classes, genders, races, and neighborhoods to access and afford the transportation services, and assessing how inclusive transportation systems are over time.

Do you think the `bike` data as it is can help you assess equity? If so, please explain. If not, how would you change the dataset? You may discuss how you would change the granularity, what other kinds of variables you'd introduce to it, or anything else that might help you answer this question.

I don't believe the bike data as it is could help in assessing equity because it really doesn't give any data on the users of the bikes other than how or when they ride and trying to make inferences based off of that might be too much of a stretch. I would add new variables such as gender and income level to see who actually is riding the bikes. And, I would keep granularity the same because I would want to see the variance of ridership per hour as to and from work hours might be something worth looking at.

Question 7B Bike sharing is growing in popularity and new cities and regions are making efforts to implement bike sharing systems that complement their other transportation offerings. The goals of these efforts are to have bike sharing serve as an alternate form of transportation in order to alleviate congestion, provide geographic connectivity, reduce carbon emissions, and promote inclusion among communities.

Bike sharing systems have spread to many cities across the country. The company you work for asks you to determine the feasibility of expanding bike sharing to additional cities of the U.S.

Based on your plots in this assignment, what would you recommend and why? Please list at least two reasons why, and mention which plot(s) you drew you analysis from.

Note: There isn't a set right or wrong answer for this question, feel free to come up with your own conclusions based on evidence from your plots!

Based on the plot that assesses how ridership changes as temperature changes, we saw that as temperature increases and becomes warmer, ridership (at least for casual riders) increased. Therefore, I do believe this would be a very feasible for certain cities that have warm and pleasant weather and less feasible for cities without this weather. We also saw that there were more riders on the weekday that were registered this might be very feasible for cities with large working populations.