0.0.1 Question 0

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

Each row represents an hour of bike rental activity.

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?

Because our granularity is on a per-hour basis, we wouldn't be able to investigate bike rental activity on a narrower basis, eg. if we wanted to investigate the bike rental activity between 4:30am and 5:30am or between 2:15pm and 2:45pm. In addition, it seems like we don't have a record of the total number of bikes available, or any data on the checking-in or checking-out of bikes; this implies we wouldn't be able to investigate, say, rental bike theft. Remedying either of these issues would come down to increasing the granularity of the data or adding additional columns with information about the total bicycle capacity.

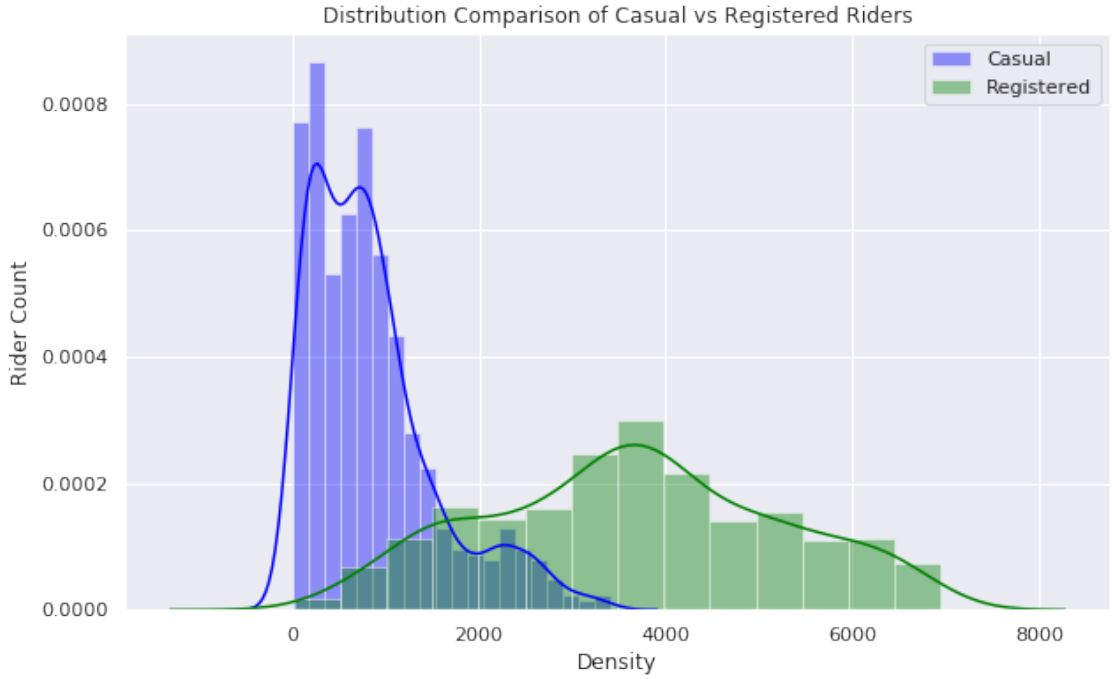
0.0.2 Question 2

Question 2a Use the `sns.distplot` 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.

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 [16]:
```
plt.figure(figsize=(10, 6))
sns.distplot(daily_counts["casual"], color="blue", kde=True, label="Casual")
sns.distplot(daily_counts["registered"], color="green", kde=True, label="Registered")
plt.legend()
plt.xlabel("Density")
plt.ylabel("Rider Count")
plt.title("Distribution Comparison of Casual vs Registered Riders")
```

Out[16]:
```
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 distribution of registered users has a unimodal and fairly symmetric distribution with a mode of roughly 3500 riders. There do not appear to be any gaps or outliers at this granularity. The distribution as a whole has a range of about [0, 7000].

The distribution of casual users is asymmetric and skewed far to the right, with a bimodal distribution moded at roughly 175 and 875. The range of this distribution seems to be tighter at about [0, 3500]. No gaps or outliers appear at this granularity.

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).

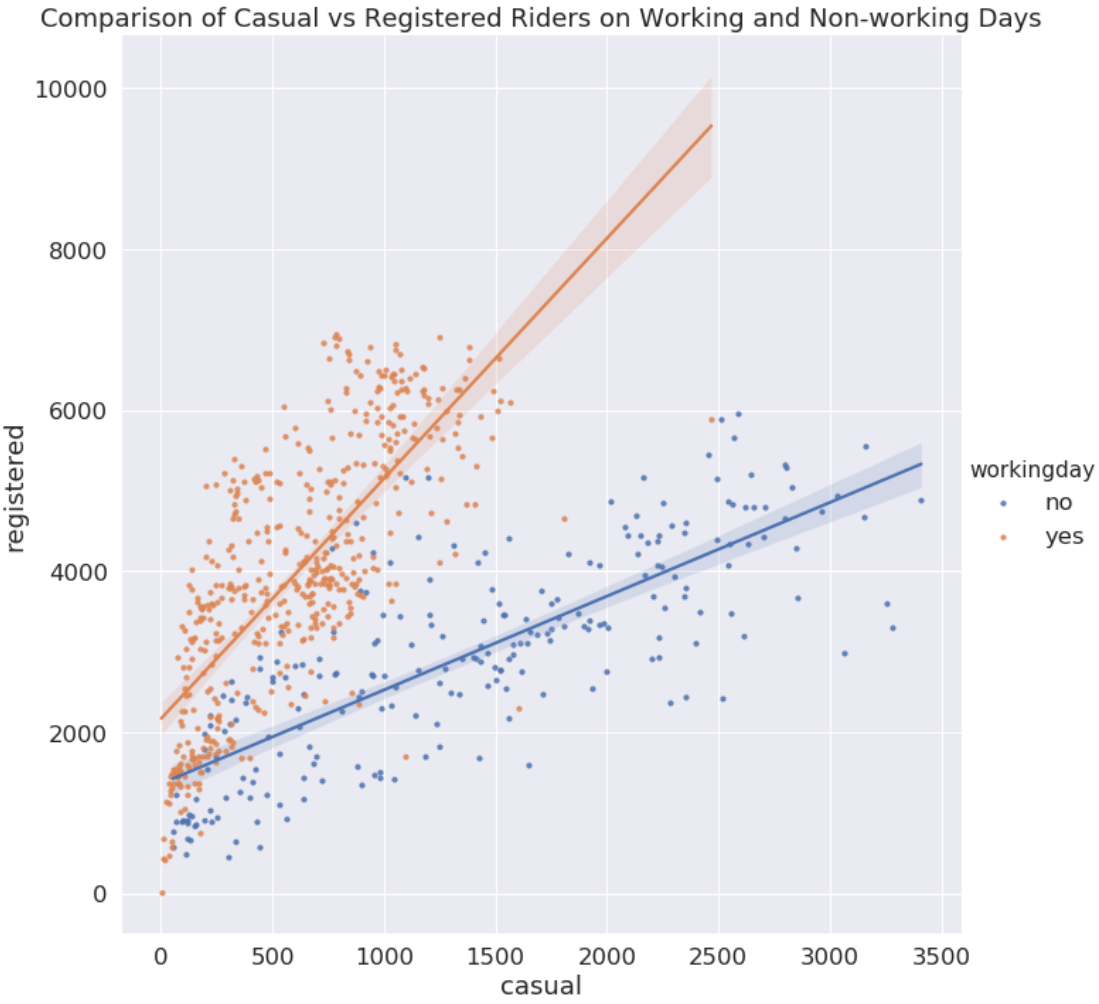
There are many points in the scatter plot, so make them small to help reduce overplotting. Also make sure to set `fit_reg=True` to generate the linear regression line. You can set the `height` parameter if you want to adjust the size of the `lmplot`.

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

- You will need to set `x`, `y`, and `hue` and the `scatter_kws`.

In [17]:
```
# Make the font size a bit bigger
sns.set(font_scale=1.5)
(sns.lmplot(x="casual", y="registered", hue="workingday",
            data=daily_counts, scatter_kws={"s": 10}, height=10, x_jitter=0.5, y_jitter=0.5)
    .set(title="Comparison of Casual vs Registered Riders on Working and Non-working Days"))
```

Out[17]:
```
<seaborn.axisgrid.FacetGrid at 0x7f97d8269ad0>
```


Comparison of Casual vs Registered Riders on Working and Non-working Days
registered
casual
workingday
no
yes
0
2000
4000
6000
8000
10000
0
500
1000
1500
2000
2500
3000
3500

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?

The scatter plot appears to show that far more casual users take out bikes on non-working days, while far more registered users take out bikes on working days. Overplotting does not seem to be a severe issue (except for in the bottom-left of the plot where point density appears to be greater) because uniform jitter of +/-0.5 was introduced across all points in both axes.

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

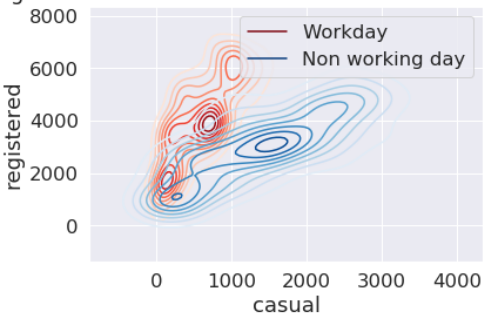
Hints: * You can use `loc` with a boolean array and column names at the same time * You will need to call kdeplot twice. * Check out this tutorial to see an example of how to set colors for each dataset and how to create a legend. The legend part uses some weird matplotlib syntax that we haven't learned! You'll probably find creating the legend annoying, but it's 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). 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 [19]:
```
import matplotlib.patches as mpatches  # see the tutorial for how we use mpatches to generate

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

# Bivariate KDEs require two data inputs.
# In this case, we will need the daily counts for casual and registered riders on workdays
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(casual_workday, registered_workday, cmap="Reds", label="Workday")

# Repeat the same steps above but for rows corresponding to non-workingdays
casual_non_workday = daily_counts.loc[~is_workingday, "casual"]
registered_non_workday = daily_counts.loc[~is_workingday, "registered"]

# Use sns.kdeplot on the two variables above to plot the bivariate KDE for non-workingday ride
sns.kdeplot(casual_non_workday, registered_non_workday, cmap="Blues", label="Non working day")
plt.legend()
plt.title("Bivariate distribution of registered vs casual users conditioned on whether data wa
```

Out[19]:
```
Text(0.5, 1.0, 'Bivariate distribution of registered vs casual users conditioned on whether da
```

Bivariate distribution of registered vs casual users conditioned on whether data was from a working day



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

Looking at the darkest contour lines helps us identify modes in the bivariate distributions. It follows that the contour plot more clearly shows locations in the distribution where modes were present, whereas overplotting in the scatter plot would have obscured these modes.

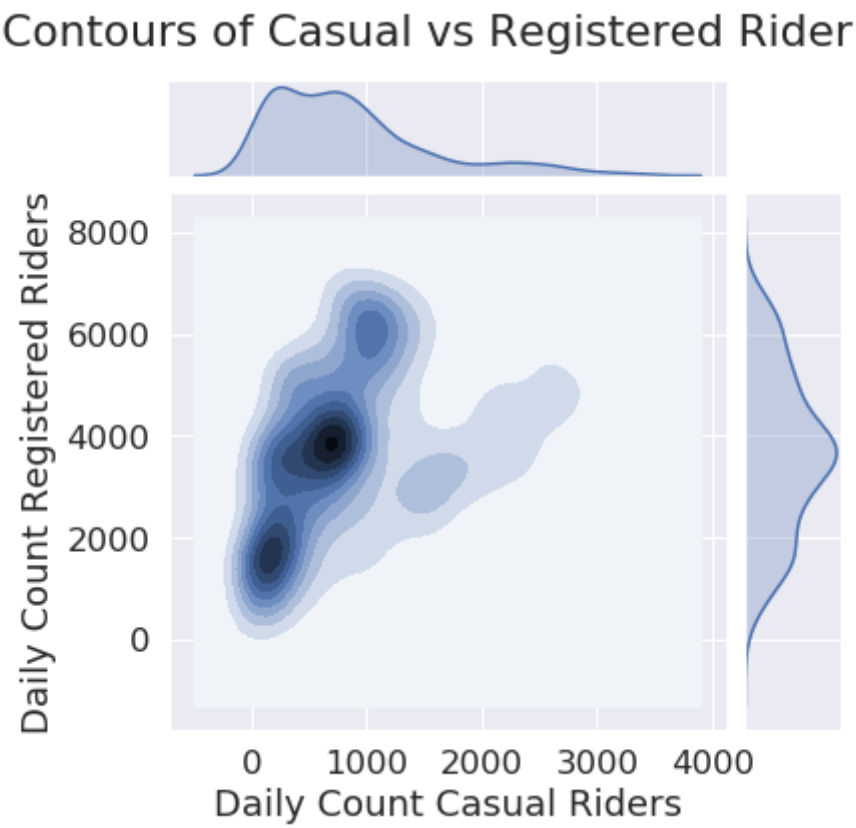
0.1 4: Joint Plot

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. * `plt.suptitle` from lab 1 can be handy for setting the title where you want. * `plt.subplots_adjust(top=0.9)` can help if your title overlaps with your plot

We do not expect you to match our colors exactly, but the colors you choose should not distract from the information your plot conveys!

```
In [20]: (sns.jointplot(x=daily_counts["casual"], y=daily_counts["registered"], kind="kde")
              .set_axis_labels("Daily Count Casual Riders", "Daily Count Registered Riders"))
         plt.suptitle("KDE Contours of Casual vs Registered Rider Count")
         plt.subplots_adjust(top=0.9)
```



0.2 5: Understanding Daily Patterns

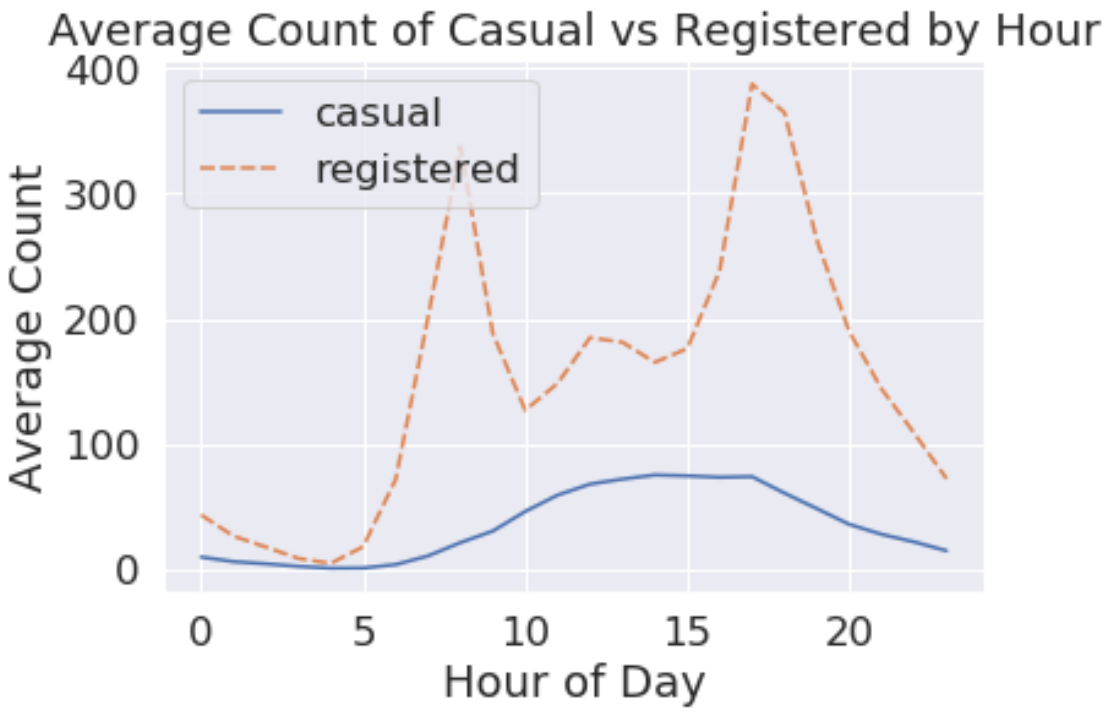
0.2.1 Question 5

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 [21]: hourly = bike.groupby("hr").agg({"casual": np.mean, "registered": np.mean})
         (sns.lineplot(data=hourly).set(
             xlabel="Hour of Day", ylabel="Average Count", title="Average Count of Casual vs Registered
```

```
Out[21]: [Text(0, 0.5, 'Average Count'),
          Text(0.5, 0, 'Hour of Day'),
          Text(0.5, 1.0, 'Average Count of Casual vs Registered by Hour')]
```



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

The plot of casual riders doesn't have a distinct mode, with a fairly flat distribution peaking around 12-17, implying casual riders tend to ride in the afternoon. The plot of registered riders has two very prominent modes at 8 and 17, which is likely when employees get off work and commute home using bikes.

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.

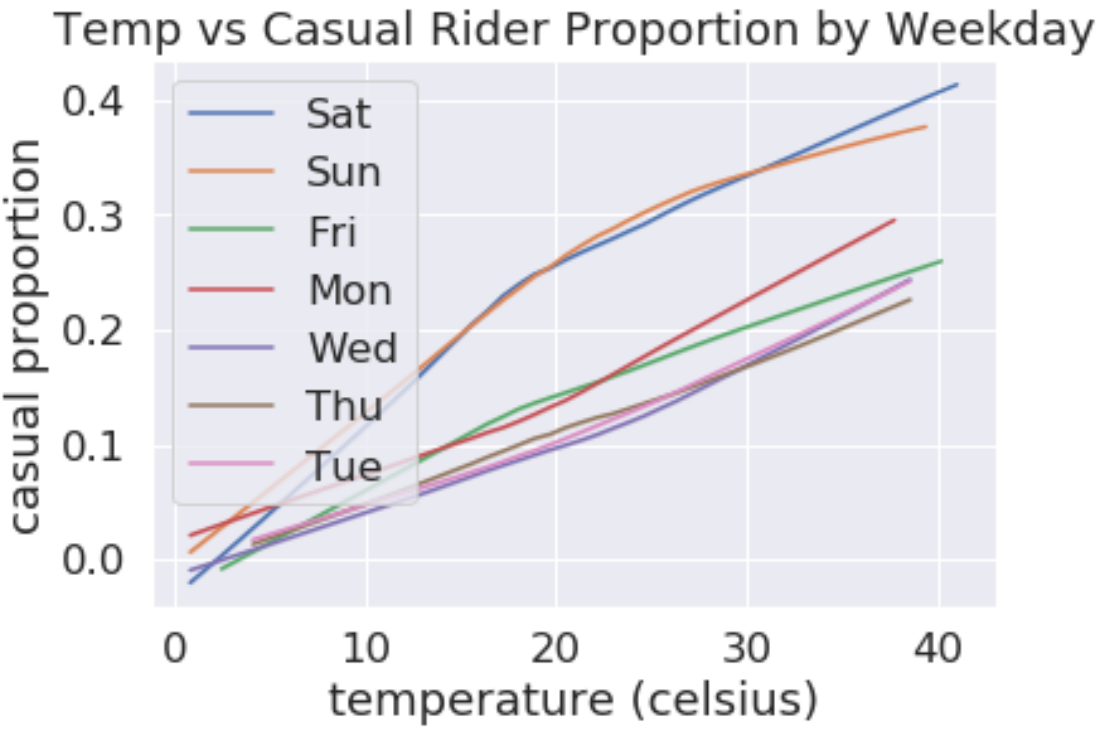
You do not need to match the colors on our sample plot as long as the colors in your plot make it easy to distinguish which day they represent.

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.
- 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 [27]:
```
from statsmodels.nonparametric.smoothers_lowess import lowess

for i in bike["weekday"].value_counts().index:
    temp = bike.loc[bike["weekday"] == i, "temp"] * 41
    smoothed = lowess(bike.loc[bike["weekday"] == i]["prop_casual"], temp, return_sorted=False)
    (sns.lineplot(x=temp, y=smoothed, label=i)
        .set(ylabel="casual proportion", xlabel="temperature (celsius)", title="Temp vs Casual
```


Temp vs Casual Rider Proportion by Weekday
casual proportion
0.0
0.1
0.2
0.3
0.4
Sat
Sun
Fri
Mon
Wed
Thu
Tue
0
10
20
30
40
temperature (celsius)

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?

It seems like as the temperature warms up, the proportion of casual riders increase steadily. In addition, the casual proportion is markedly higher on weekends.

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 think the current bike data is helpful for assessing equity. Geographical location of a certain bike-sharing station (especially in DC) heavily informs the socioeconomic breakdown of the residents it serves. Consequently, I'd like to see qualitative location data for locations of bike hubs that could be cross-referenced with socioconomic census data to better assess equitable access. With the current information, we can only really draw tenuous conclusions about the activity of the bikesharing system as a whole, which is not entirely helpful in assessing its impact on certain populations.

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!

For a city as small as DC to experience as much as 400 riders per hour at peak times (Average Count of Reg/Casual by Hour) even with its developed metro and public transit system says a lot about the readiness with which the population seems to adopt bike-sharing programs. In addition, there were a surprisingly high number of casual riders (same plot) which seems to me that the public sees bike-sharing to be more than just a transport tool. With that said, I think DC is a city where bike-sharing would work very well, since it's highly metropolitan but not exceedingly dense. Thus I think that in implementing similar bike-sharing programs elsewhere there should be a consideration for the actual civic layout.