0.0.1 Question 0a

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

By examining 48 rows of data, I think each row summarizes the date, time, weather condition and how many users are using bikes. Specifically, it contains date and time information down to year, date, and the season of the year, and hour of the day. It also classifies the day as one of 7 weekdays and labels the day as workingday or non-workingday. In terms of weather condition, it has a general description of the weather situation, temperature of the hour, “feels-like” temperature, humidity and windspeed. In terms of usage at the specific hour, both casual and registered user number, as well as the total count, are included.

0.0.2 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 one could collect to address some of these limitations?

I think the current data includes only how many users are using the bikes, but not how each bike is used, like how long each bike is used or the distance that each bike is used to cover. I would at least consider adding the following columns: 1) average time of bike usage; 2) average distance of bikes

0.0.3 Question 2a

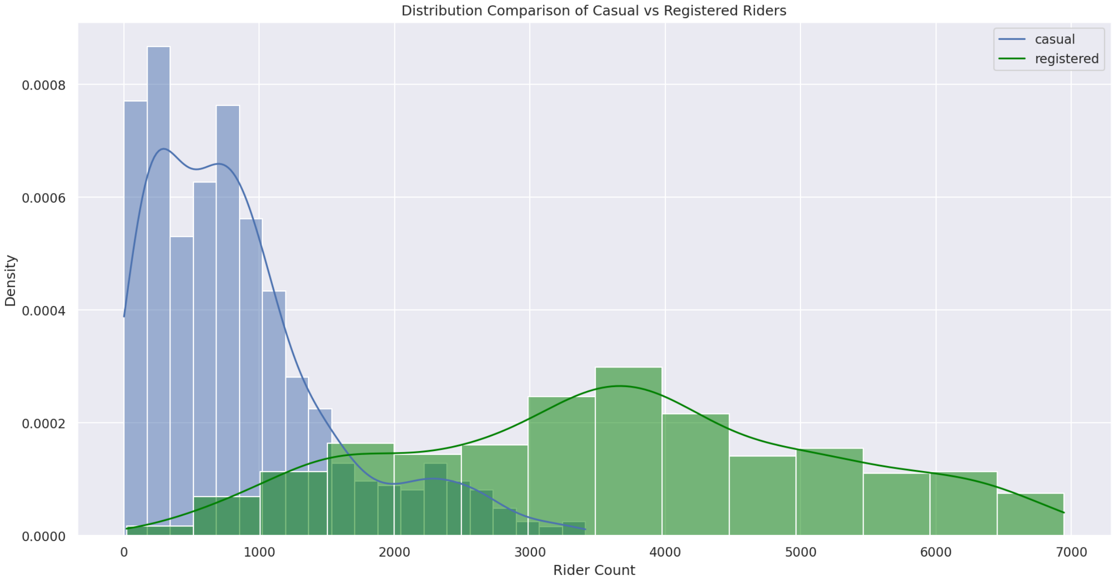
Use the `sns.histplot`(documentation) function to create a plot that overlays the distribution of the daily counts of bike users, using blue to represent `casual` riders, and green to represent `registered` riders. The temporal granularity of the records should be daily counts, which you should have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

Hint: You will need to set the `stat` parameter appropriately to match the desired plot, and may call `sns.histplot` more than one time.

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]: hist1 = sns.histplot(data=daily_counts, x='casual', stat='density',legend=True, kde=True)
         hist2 = sns.histplot(data=daily_counts, x='registered', stat='density', color='green',legend=T
         plt.legend(labels = ['casual', 'registered'])
         plt.xlabel('Rider Count')
         plt.ylabel('Density')
         plt.title('Distribution Comparison of Casual vs Registered Riders')
```

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



0.0.4 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.

Factors	Casual riders	Registered Riders
Modes	Seems multimodality (3 modes), but could be outlier in the first two	Single peak
Symmetry	Asymmetric	nearly symmetric
Skewness	Positively skewed	Not skewed
Tails	Right tailed	Both left and right tailed
Gaps	One gap between <1000 and ~2300	No clear gap
Outliers	>3000 could be outlier	No clear outlier

0.0.5 Question 2c

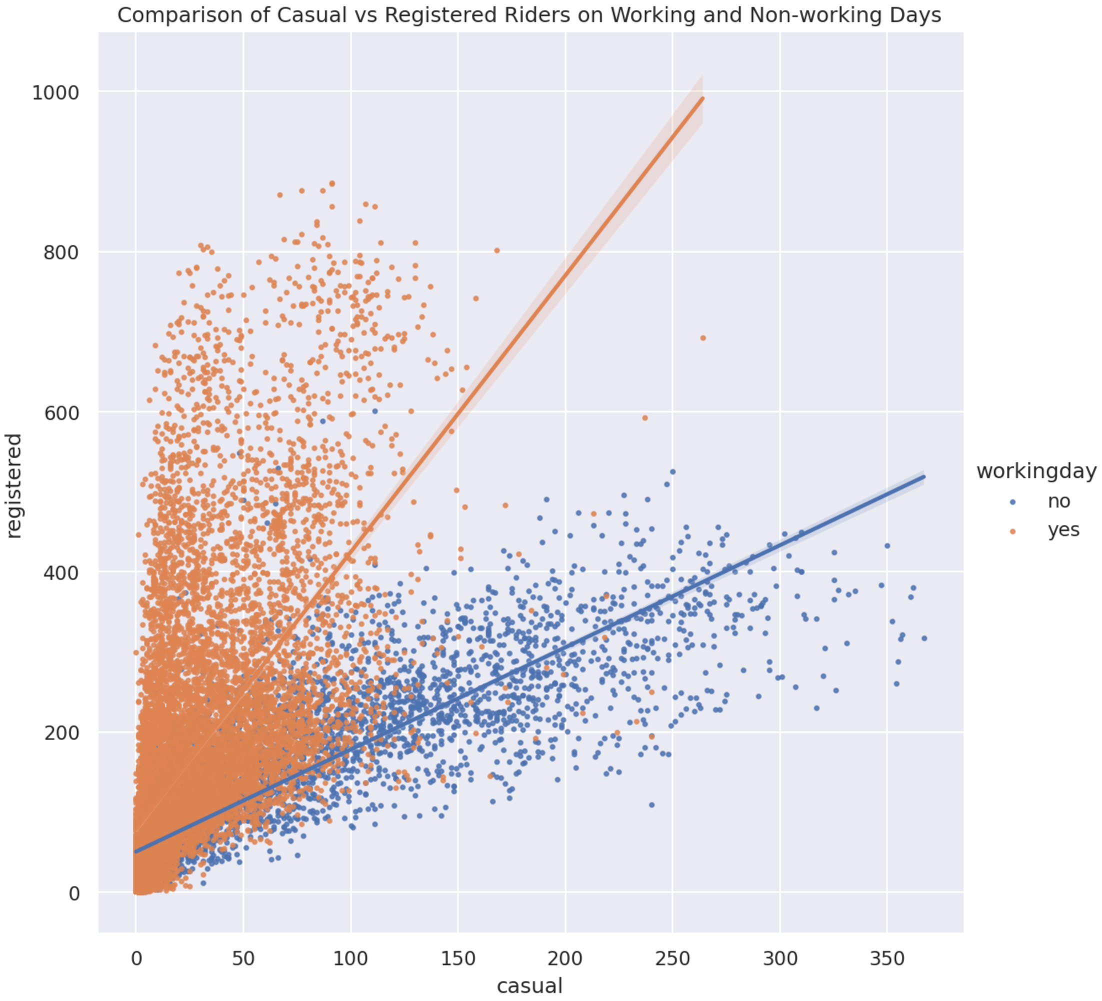
The density plots do not show us how the counts for registered and casual riders vary together. Use `sns.lmplot` (documentation) to make a scatter plot to investigate the relationship between casual and registered counts. This time, let's use the `bike DataFrame` to plot hourly counts instead of daily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

Hints: * Checkout this helpful tutorial on `lmplot`. * There are many points in the scatter plot, so make them small to help reduce overplotting. Check out the `scatter_kws` parameter of `lmplot`. * Generate and plot the linear regression line by setting a **paramter** of `lmplot` to `True`. Can you find this in the documentation? We will discuss what is linear regression is more details later. * You can set the `height` parameter if you want to adjust the size of the `lmplot`. * Add a descriptive title and axis labels for your plot.

```
In [18]: # Make the font size a bit bigger
         sns.set(font_scale=1)
         sns.lmplot(data=bike, x='casual', y='registered', hue='workingday', legend_out=True, scatter_kw
         plt.title('Comparison of Casual vs Registered Riders on Working and Non-working Days')
```

```
Out[18]: Text(0.5, 1.0, 'Comparison of Casual vs Registered Riders on Working and Non-working Days')
```


Comparison of Casual vs Registered Riders on Working and Non-working Days
registered
1000
800
600
400
200
0
0
50
100
150
200
250
300
350
casual
workingday
no
yes

0.0.6 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?

In general, registered users use bikes more on workingday, whereas casual users are more often on non-workingdays. The effect is clearer on times when more users from each category are present, as one can differentiate workingday vs non-workingday when there are >150 casual users or >300 registered users. Below the threshold, it's difficult to tell. Overplotting reduces the ability to differentiate the data when sample size is small.

0.0.7 Question 3a (Bivariate Kernel Density Plot)

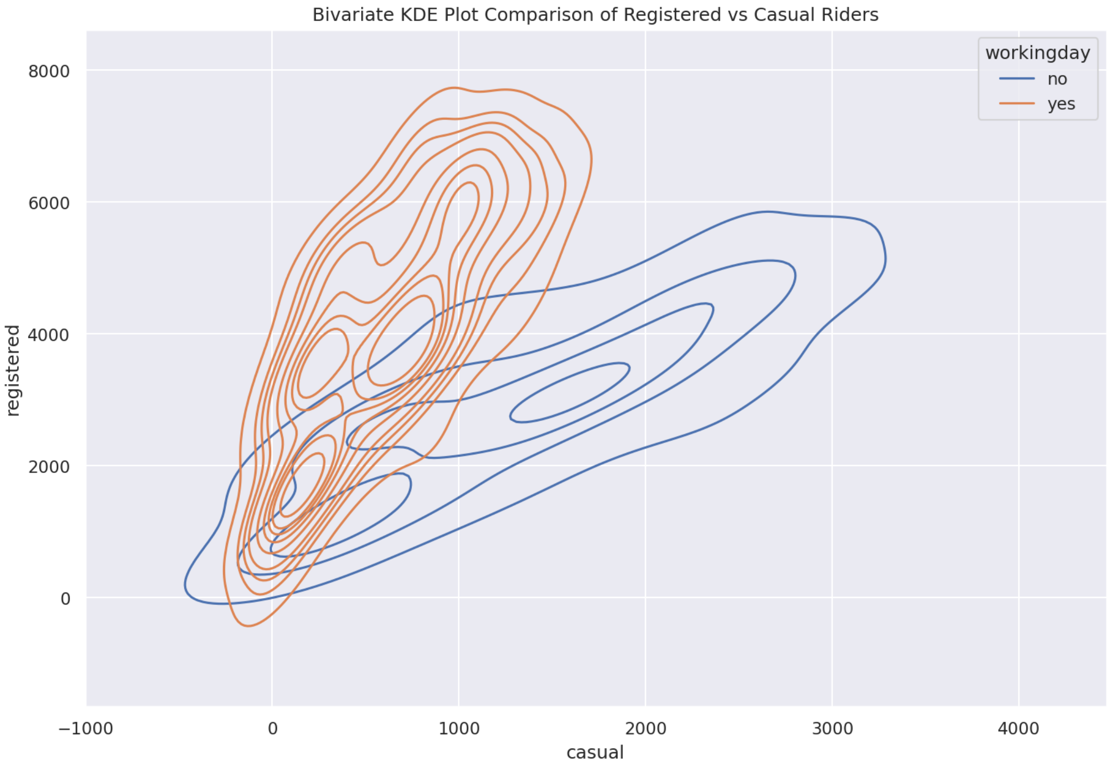
Generating a bivariate kernel density plot with workday and non-workday separated.

Hints: You only need to call `sns.kdeplot` once. Take a look at the `hue` paramter and adjust other inputs as needed.

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))

         sns.kdeplot(x=daily_counts['casual'], y=daily_counts['registered'], hue=daily_counts['workingda
         plt.title('Bivariate KDE Plot Comparison of Registered vs Casual Riders');
```



0.0.8 Question 3b

With some modification to your 3a code (this modification is not in scope), we can obatined the plot above. In your own words, describe what the lines and the color shades of the lines signify about the data. What does each line and color represent?

The plot represents a contour plot of casual/registered riders in both working and non-working days. Each line represents a surface of equal density of riders within its own group. That is, on the circle line the amount of riders is the same. The darker the color shades (represented by lower value on the color bar), the higher density of rider population. The orange color plots the working day rider population whereas the blue color plots the non-workingday rider population.

0.0.9 Question 3c

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

I find it interesting that both KDE contour plots are multi-model. On workdays there are two high rider density peaks appearing whereas on non-workingdays there are three high rider density peaks. It would be interesting to analyze each group with more details like dates, weather to get more implications.

0.1 4: Joint Plot

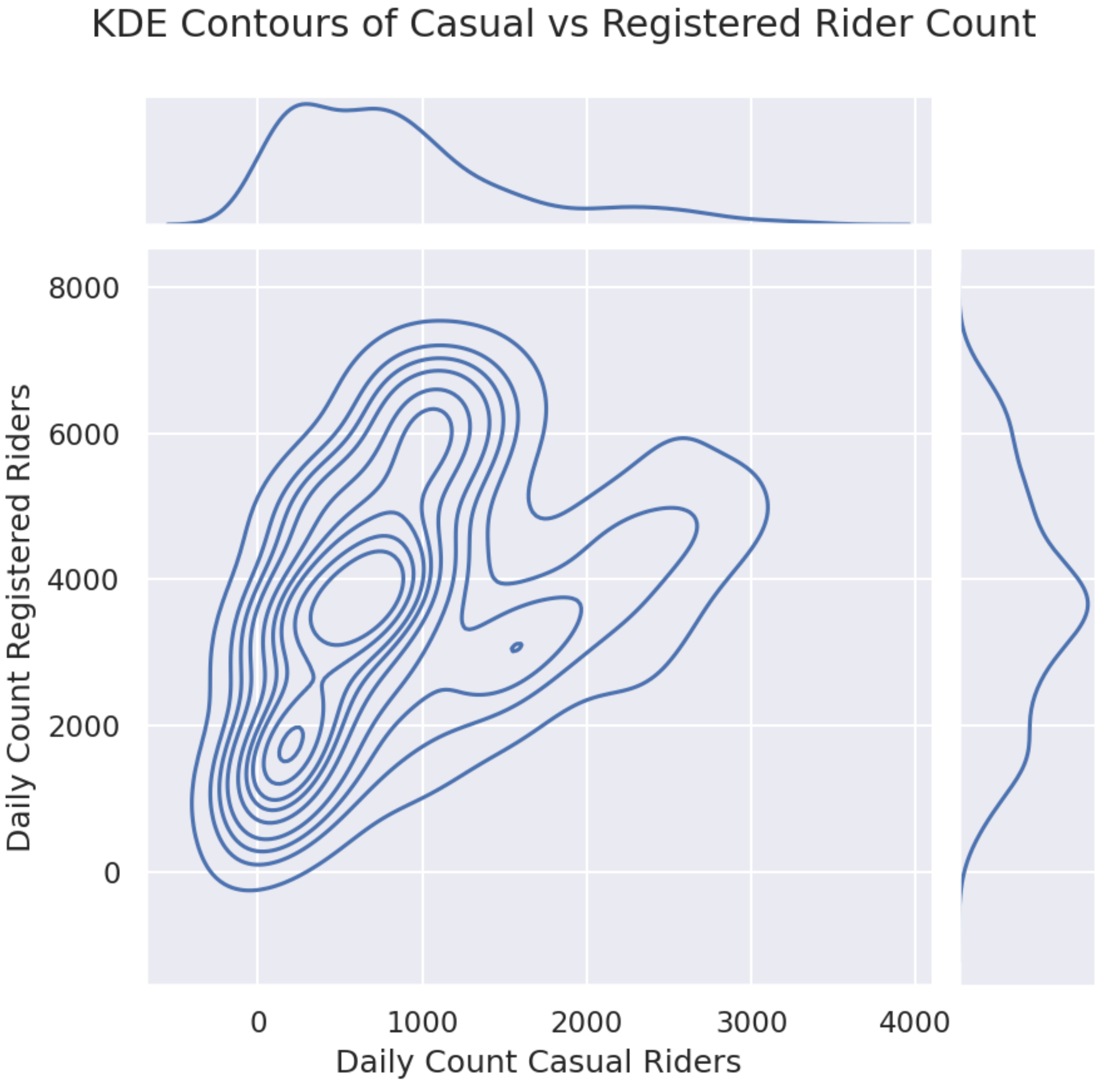
As an alternative approach to visualizing the data, construct the following set of three plots where the main plot shows the contours of the kernel density estimate of daily counts for registered and casual riders plotted together, and the two "margin" plots (at the top and right of the figure) provide 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]: plot = sns.jointplot(data=daily_counts, x='casual',y='registered', kind="kde")
         plt.suptitle("KDE Contours of Casual vs Registered Rider Count")
         plt.subplots_adjust(top=0.9);
         plot.set_axis_labels(xlabel='Daily Count Casual Riders', ylabel='Daily Count Registered Riders
```

```
Out[21]: <seaborn.axisgrid.JointGrid at 0x7fdca49e21c0>
```


KDE Contours of Casual vs Registered Rider Count
8000
6000
4000
2000
0
Daily Count Registered Riders
0
1000
2000
3000
4000
Daily Count Casual Riders

0.2 5: Understanding Daily Patterns

0.2.1 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.

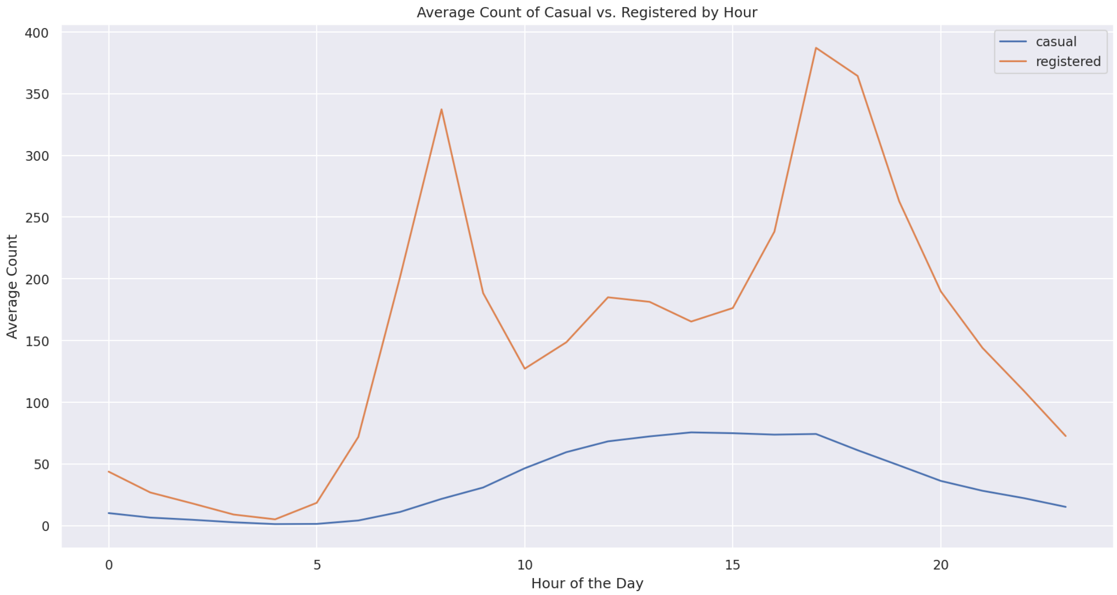
Your plot should look like the plot below. While we don't expect your plot's colors to match ours exactly, your plot should have legend in the plot and different colored lines for different kinds of riders.

```
In [33]: count_hr=bike.groupby('hr')[['casual','registered']].agg(np.mean)
         count_hr.head()
```

```
Out[33]:        casual  registered
         hr
         0   10.158402   43.739669
         1    6.504144   26.871547
         2    4.772028   18.097902
         3    2.715925    9.011478
         4    1.253945    5.098996
```

```
In [43]: plt.plot(count_hr['casual'])
         plt.plot(count_hr['registered'])
         plt.legend(['casual', 'registered'])
         plt.xlabel('Hour of the Day')
         plt.ylabel('Average Count')
         plt.title('Average Count of Casual vs. Registered by Hour')
```

```
Out[43]: Text(0.5, 1.0, 'Average Count of Casual vs. Registered by Hour')
```


Average Count of Casual vs. Registered by Hour
casual
registered
Average Count
400
350
300
250
200
150
100
50
0
0
5
10
15
20
Hour of the Day

0.2.2 Question 5b

What can you observe from the plot? Discuss your obseravtion and hypothesize about the meaning of the peaks in the registered riders' distribution.

Registered users peak at hours 8-9 and 17-18. It's possible that registered users ride bikes mostly for commute. Casual users peak from hours 13-18. It seems they use the bikes mostly for purposes outside work, like touring or dinner, etc.

0.2.3 Question 6b

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.

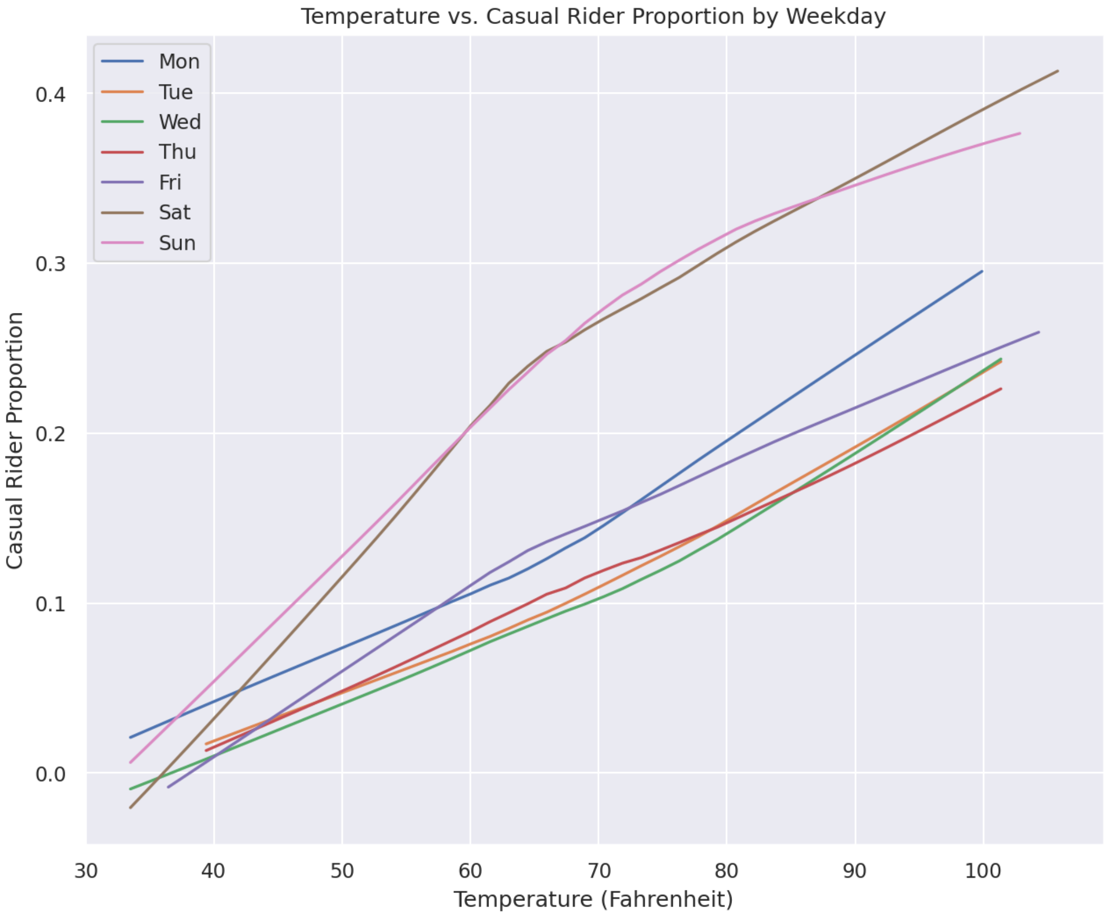
Hints: * Start by just plotting only one day of the week to make sure you can do that first.

- 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 (normalized) 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 [62]: from statsmodels.nonparametric.smoothers_lowess import lowess
         plt.figure(figsize=(10,8))
         for day in ['Mon','Tue','Wed','Thu','Fri','Sat','Sun']:
             xobs = (bike[bike['weekday']==day]['temp'] * 41) * 9 / 5 + 32
             ysmooth = lowess(bike[bike['weekday']==day]['prop_casual'], xobs, return_sorted=False)
             sns.lineplot(xobs, ysmooth, label=day)
         plt.xlabel('Temperature (Fahrenheit)')
         plt.ylabel('Casual Rider Proportion')
         plt.title('Temperature vs. Casual Rider Proportion by Weekday')
```

```
Out[62]: Text(0.5, 1.0, 'Temperature vs. Casual Rider Proportion by Weekday')
```


Temperature vs. Casual Rider Proportion by Weekday
Mon
Tue
Wed
Thu
Fri
Sat
Sun
Casual Rider Proportion
0.0
0.1
0.2
0.3
0.4
Temperature (Fahrenheit)
30
40
50
60
70
80
90
100

0.2.4 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?

In general, casual rider proportion increase with temperature. Interesting observations: 1. There's nearly no casual rider when the temperature approaches 40F. 2. More casual riders on Saturdays and Sundays, followed by Mondays and Fridays. The middle three days are generally overlapping. 3. Above 70F, casual riders - temperature increase slower compared to below 70F.

0.2.5 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 think the current dataset can't assess equity, because it lacks geographical data. Based on the current dataset, I would at least add two new columns: `Starting Point` & `Returning Point`. It will likely split the rows of the current table. E.g. depending on how many geophical locations we'd like to assess, one row will be split into corresponding number of new rows. The columns will need to update accordingly. The new table will allow us accessibility to neighborhoods related information. Other socio-economic classes including genders, races will likely be acquired from a table that records usage information of individual bikes.

0.2.6 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!

I would not recommend the company's expansion unless there's ample amount of data suggesting that in each single city there's profit. From Q5a, we can notice that on average of two-year period, there are only a few hundreds of riders at peak hours, in a city with population close to 1 million. It indicates the speading ratio is less than 1% population. There's a lot the company needs to improve, such as collecting more data, ens