0.0.1 Question 1a

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

Hint: Examine all variables present in the dataset carefully before answering this question!

Each row represents a record of bike-sharing usage information for each hour of the day from 2011 to 2012.

0.0.2 Question 1b

For this assignment, we'll be using this data to study bike usage in Washington, DC. 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?

A location variable may be helpful to analyze the concentration of bike-sharing popularity. The amount of casual users versus registered users may vary a good amount from one area to another. The location variable may also introduce reasonable certainty over the type of usage being done, whether that be going directly from work to home or heading to work from the train station. This data can inform future expansions of bike-sharing allocation for the city. Assuming that casual users is defined by non-registered users, there may actually be an overcounting problem here. Some regular bike-sharing users may simply never register as an official user, so there is no way to decipher whether these casual users are indeed just casual users. If the data collector designed this set of categories in order to retrieve information on user retention or even user behavior, then there is the problem that some casual users may actually behave like registered users but for one reason or another, simply chose to not register. Another problem I see is that the normalized metrics offers little insight into how the "divided by 41" or "divided by 50" was decided. A descriptor column delineating some interpretation of these seemingly arbitrary numbers may be helpful.

0.0.3 Question 3a

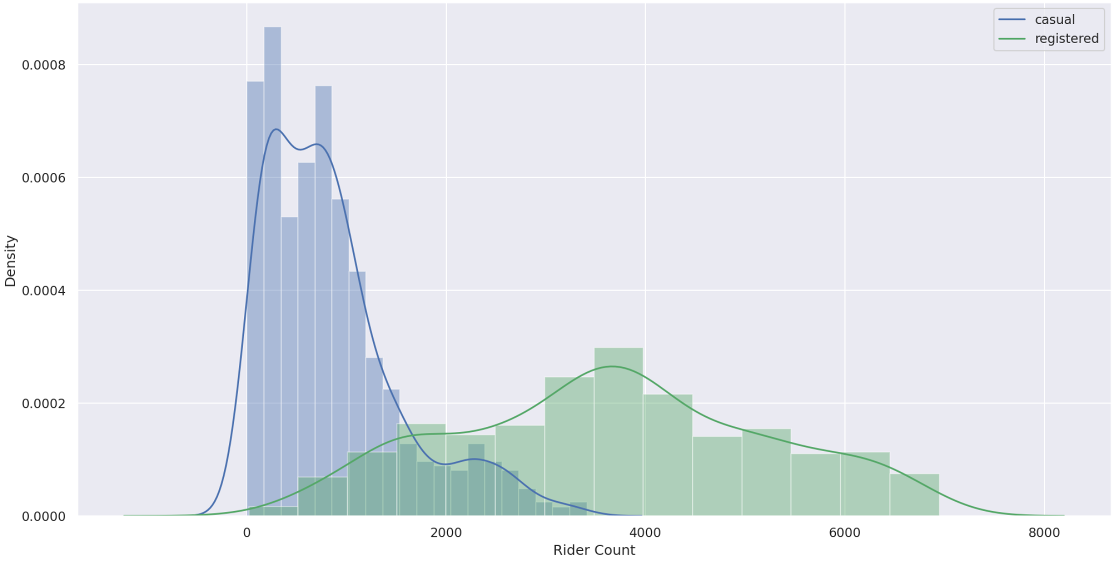
Use the `sns.histplot`(documentation) function to create a plot that overlays the distribution of the daily counts of bike users, using blue to represent `casual` riders, and green to represent `registered` riders. The temporal granularity of the records should be daily counts, which you should have after completing question 2.c. In other words, you should be using `daily_counts` to answer this question.

Hints: - You will need to set the `stat` parameter appropriately to match the desired plot. - The `label` parameter of `sns.histplot` allows you to specify, as a string, how the plot should be labeled in the legend. For example, passing in `label="My data"` would give your plot the label "My data" in the legend. - You will need to make two calls to `sns.histplot`.

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.

For all visualizations in Data 100, our grading team will evaluate your plot based on its similarity to the provided example. While your plot does not need to be *identical* to the example shown, we do expect it to capture its main features, such as the **general shape of the distribution**, the **axis labels**, the **legend**, and the **title**. It is okay if your plot contains small stylistic differences, such as differences in color, line weight, font, or size/scale.

```
In [82]: sns.distplot(daily_counts['casual'], color = 'b')
         sns.distplot(daily_counts['registered'], color = 'g')
         plt.xlabel('Rider Count')
         plt.ylabel('Density')
         plt.legend(['casual', 'registered'])
         plt.show()
```


casual
registered
Density
0.0008
0.0006
0.0004
0.0002
0.0000
0
2000
4000
6000
8000
Rider Count

0.0.4 Question 3b

In the cell below, describe 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 casual riders density curve center is higher than the registered riders curve and has a smaller spread in its distribution (likely meaning a smaller standard deviation). The latter curve is more symmetrical. The former curve has a left tail. It's not particularly clear what the mode is for the casual riders curve just by inspecting it visually, but the registered riders curve looks like it has a mode in the high 3000s.

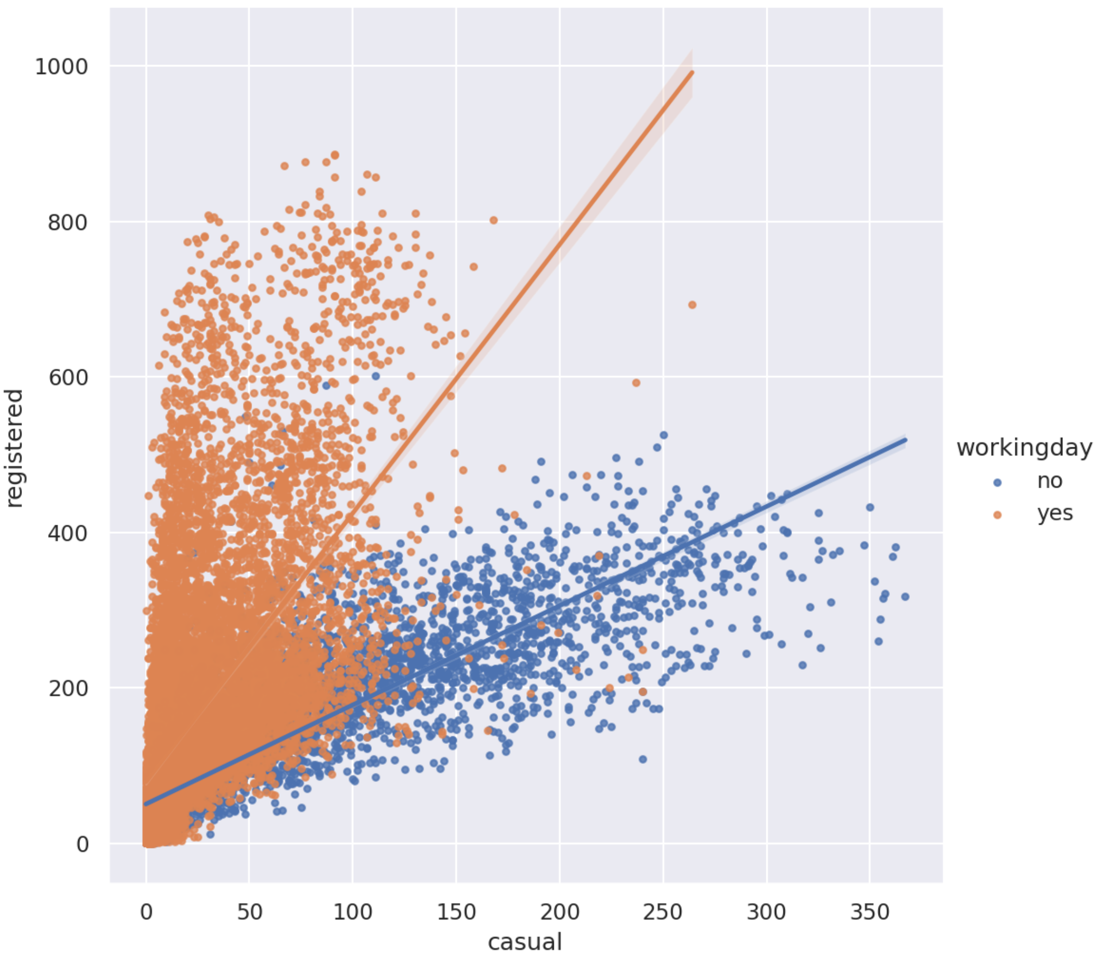
0.0.5 Question 3c

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 and 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: * Check out 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 **parameter** of lmplot to True. Can you find this in the documentation? We will discuss the concept of linear regression later in the course. * 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. * You should be using the bike DataFrame to create your plot. * It is okay if the scales of your x and y axis (i.e., the numbers labeled on the two axes) are different from those used in the provided example.

```
In [83]: sns.set(font_scale=1) # This line automatically makes the font size a bit bigger on the plot.
         sns.lmplot(x = 'casual', y = 'registered', data = bike, height = 7, hue = 'workingday', scatte

         plt.show();
```


registered
1000
800
600
400
200
0
0
50
100
150
200
250
300
350
casual
workingday
● no
● yes

0.0.6 Question 3d

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?

There are more casual than registered riders on working days, and the contrary is true for non-working days. There exists overplotting in the bottom left of the scatter plot, which crowds the space visually and I cannot tell whether or not there is a pattern in relationships.

0.0.7 Question 4a (Bivariate Kernel Density Plot)

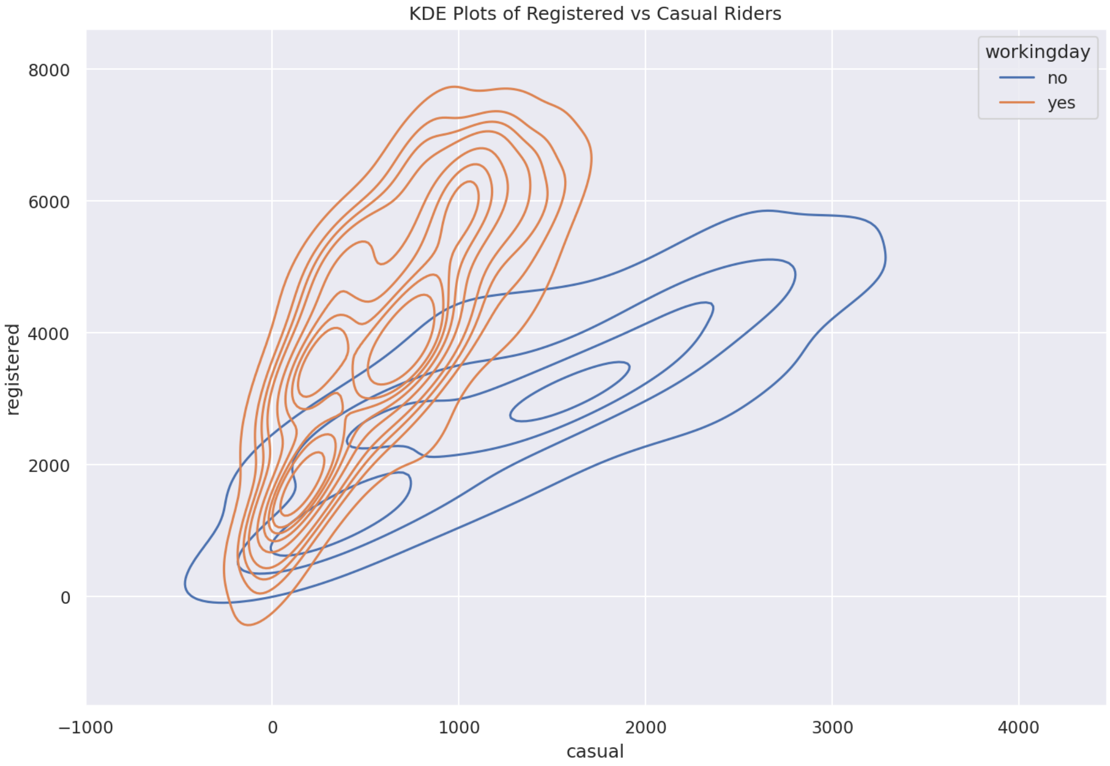
Generate a bivariate kernel density plot with workday and non-workday separated using the `daily_counts` `DataFrame`.

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

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

```
In [85]: # Set the figure size for the plot
         plt.figure(figsize=(12,8))

         sns.kdeplot(data=daily_counts, x='casual', y='registered', hue='workingday', fill=False)
         plt.title('KDE Plots of Registered vs Casual Riders')
```

```
Out[85]: Text(0.5, 1.0, 'KDE Plots of Registered vs Casual Riders')
```



0.0.8 Question 4b

With some modification to your 4a code (this modification is not in scope), we can generate 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?

Hint: You may find it helpful to compare it to a contour or topographical map as shown here.

The darker the color, the more dense the data points are in that particular spot in the plot. The lines would signify the same "altitude." In this case, I think it means that the density of the data points along the same line would be the same density. The lines close together implies that there is more smooth of a change in density of data points in that spot.

0.0.9 Question 4c

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

The clutter on the bottom left of the scatter plot revealed little information other than the fact that there were a lot of data points near that spot. The working day data points in orange covered the non-working day data points in blue. This contour plot makes it easier to see the overlap since the orange is see-through. We now see the blue non-working day data points on top.

0.1 5: 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 to see the linear relationships between casual and registered for the two different conditions (weekday vs. weekend). You should be making use of `daily_counts`.

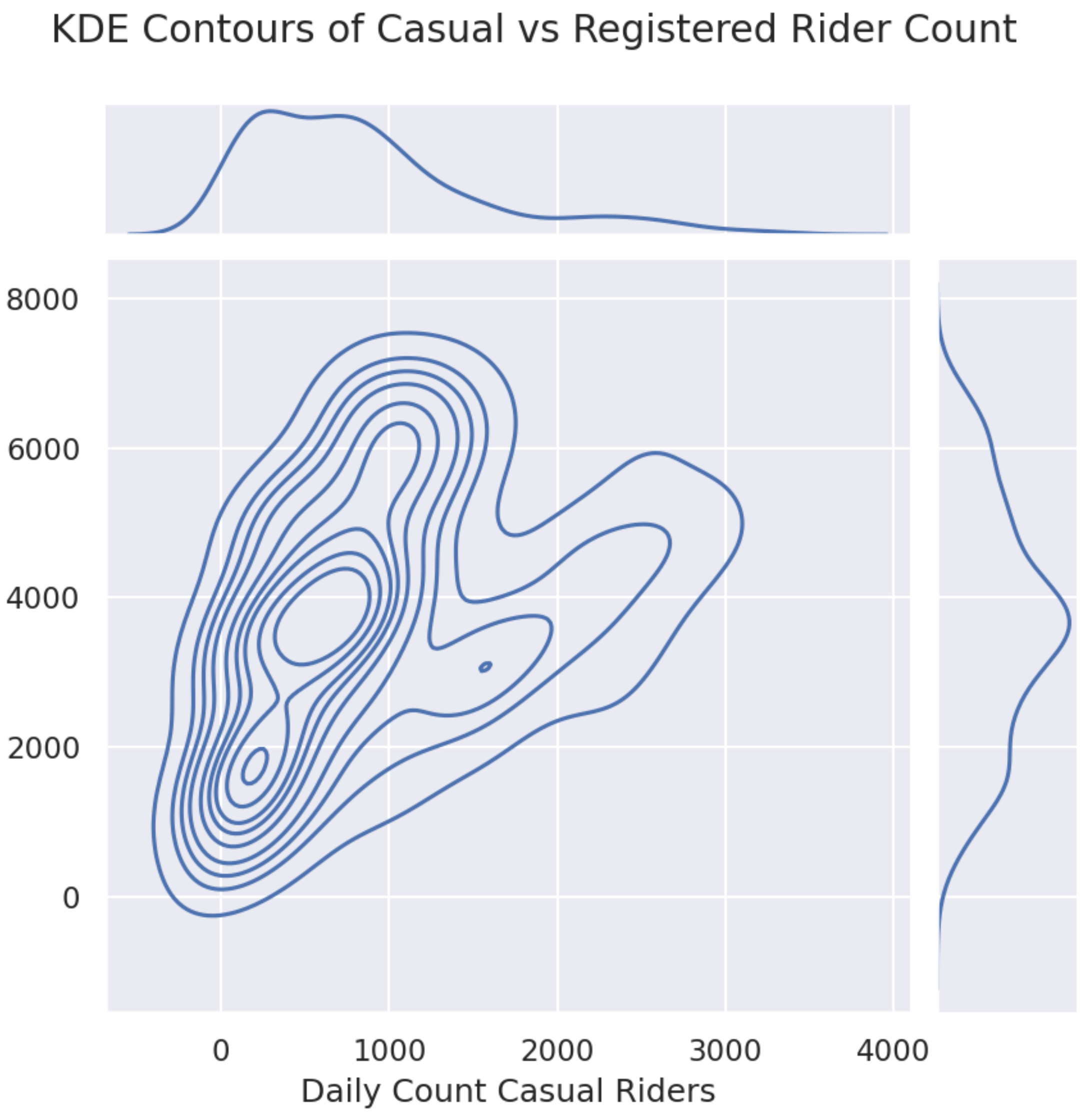
Hints: * The seaborn plotting tutorial has examples that may be helpful. * Take a look at `sns.jointplot` and its `kind` parameter. * `set_axis_labels` can be used to rename axes on a `seaborn` plot. For example, if we wanted to plot a scatterplot with 'Height' on the x-axis and 'Weight' on the y-axis from some dataset `stats_df`, we could write the following:

```
graph = sns.scatterplot(data=stats_df, x='Height', y='Weight')
```

```
graph.set_axis_labels("Height (cm)", "Weight (kg)")
```

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 [86]: sns.jointplot(x = 'casual', y = 'registered', data = daily_counts, kind = 'kde').set_axis_label
         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
0
1000
2000
3000
4000
Daily Count Casual Riders

0.2 6: Understanding Daily Patterns

0.2.1 Question 6a

Let's examine the behavior of riders by plotting the average number of riders for each hour of the day over the **entire dataset** (that is, `bike DataFrame`), 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 a legend in the plot and different colored lines for different kinds of riders, in addition to the title and axis labels.

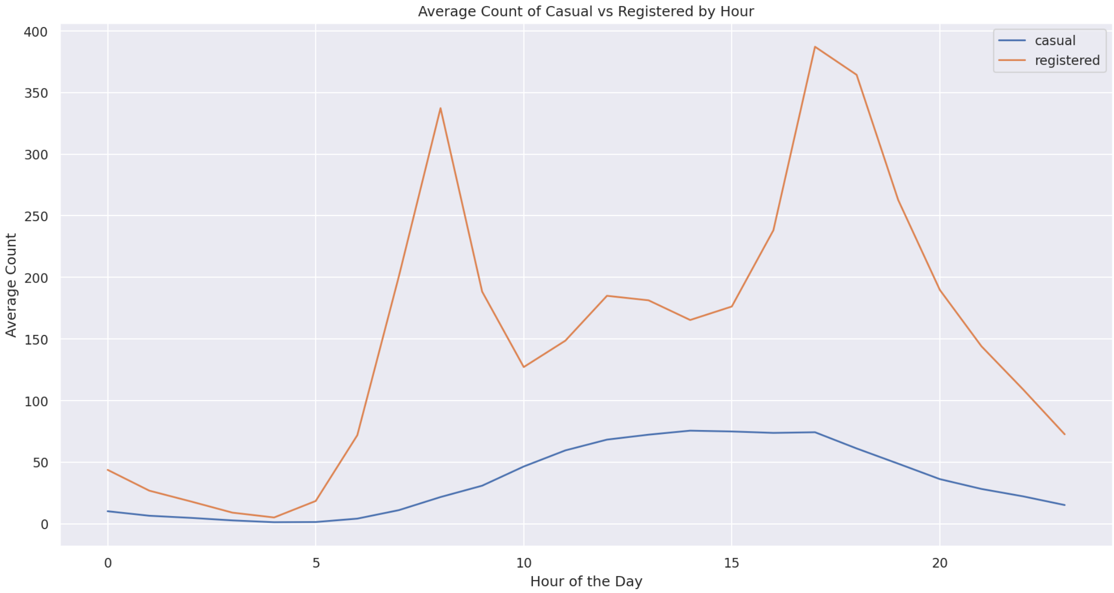
```
In [87]: bike.head()
```

```
Out[87]:    instant      dteday  season  yr  mnth  hr holiday weekday workingday  \
         0        1  2011-01-01       1   0     1   0      no     Sat         no
         1        2  2011-01-01       1   0     1   1      no     Sat         no
         2        3  2011-01-01       1   0     1   2      no     Sat         no
         3        4  2011-01-01       1   0     1   3      no     Sat         no
         4        5  2011-01-01       1   0     1   4      no     Sat         no

           weathersit  temp   atemp   hum  windspeed  casual  registered  cnt
         0      Clear  0.24  0.2879  0.81        0.0       3          13   16
         1      Clear  0.22  0.2727  0.80        0.0       8          32   40
         2      Clear  0.22  0.2727  0.80        0.0       5          27   32
         3      Clear  0.24  0.2879  0.75        0.0       3          10   13
         4      Clear  0.24  0.2879  0.75        0.0       0           1    1
```

```
In [88]: hourly_means = bike.pivot_table(index='hr', values=['casual', 'registered'], aggfunc='mean')
         hourly_means.head()
```

```
Out[88]:        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 [89]: hourly_means = bike.pivot_table(index='hr', values=['casual', 'registered'], aggfunc='mean')
         sns.lineplot(x = hourly_means.index, y = hourly_means['casual'], label = 'casual')
sns.lineplot(x = hourly_means.index, y = hourly_means['registered'], label = 'registered')

plt.title('Average Count of Casual vs Registered by Hour')
plt.xlabel("Hour of the Day")
plt.ylabel("Average Count")
plt.show()
```



0.2.2 Question 6b

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

There are more registered riders than casual riders throughout all hours of the day. The times when more usage happens seem to concentrate or peak are around normal business hours where people go to and from work. After the standard getting off from work hour at around 6PM, the count of both casual and registered riders start to plummet.

0.2.3 Question 7b

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 (as given in the 'temp' column), 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.

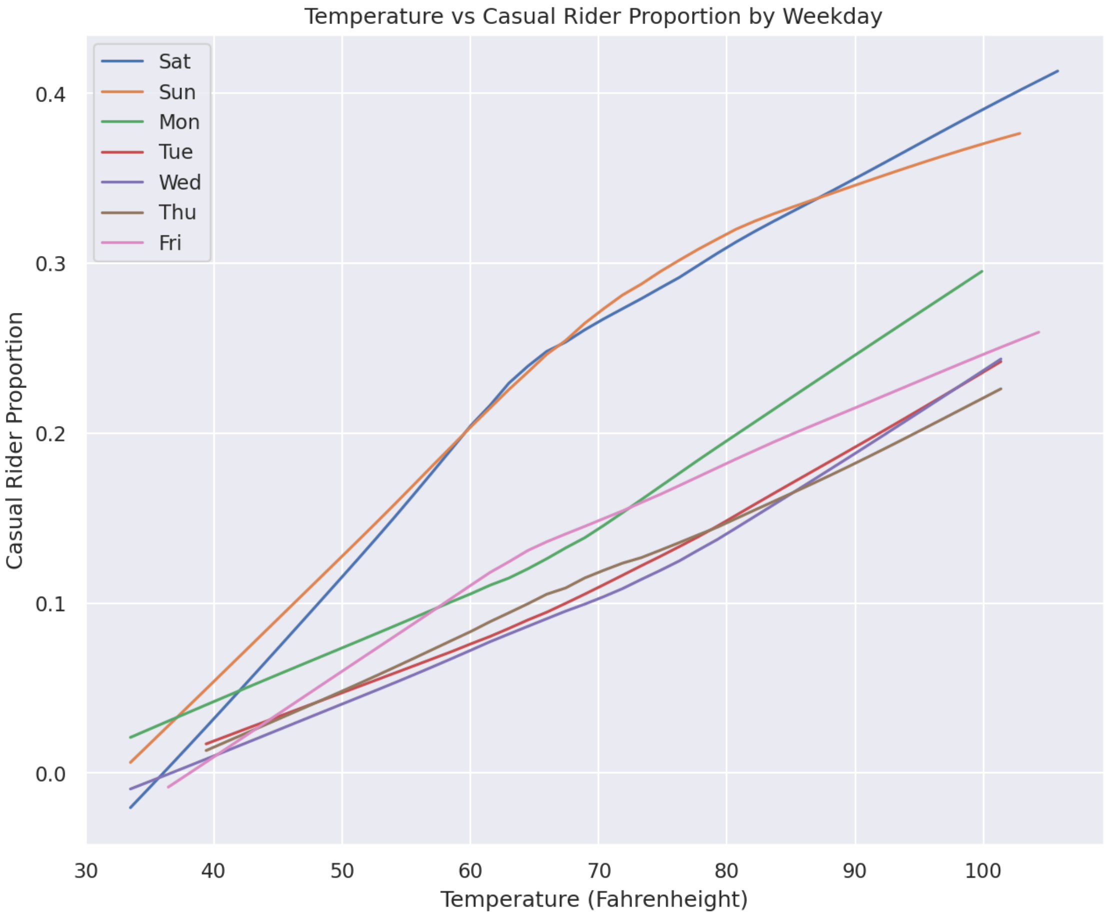
Hints: * Start by plotting only one day of the week to make sure you can do that first. Then, consider using a `for` loop to repeat this plotting operation for all days of the week.

- The `lowess` function expects the `y` coordinate first, then the `x` coordinate. You should also set the `return_sorted` field to `False`.
- **You will need to rescale the normalized temperatures stored in this dataset to Fahrenheit values.** Look at the section of this notebook titled 'Loading Bike Sharing Data' for a description of the (normalized) temperature field to know how to convert back to Celsius first. After doing so, convert it to Fahrenheit. By default, the temperature field ranges from 0.0 to 1.0. In case you need it, Fahrenheit $=$ Celsius $\times \frac{9}{5} + 32$.

Note: If you prefer plotting temperatures in Celsius, that's fine as well! Just remember to convert accordingly so the graph is still interpretable.

```
In [95]: from statsmodels.nonparametric.smoothers_lowess import lowess

         plt.figure(figsize=(10,8))

         for day in bike['weekday'].unique():
             curr_day = bike[bike['weekday'] == day]
             curr_day['temp'] = curr_day['temp'] * 41 * 9 / 5 + 32
             y = lowess(curr_day['prop_casual'], curr_day['temp'], return_sorted = False)
             sns.lineplot(data = curr_day, x = curr_day['temp'], y = y, label = day)

         plt.title("Temperature vs Casual Rider Proportion by Weekday")
         plt.xlabel('Temperature (Fahrenheight)')
         plt.ylabel('Casual Rider Proportion')
```

```
Out[95]: Text(0, 0.5, 'Casual Rider Proportion')
```


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

0.2.4 Question 7c

What do you observe in the above plot? How is `prop_casual` changing as a function of temperature? Do you notice anything else interesting?

‘prop_casual’ increases as temperature increases. The blue and orange lines representing Saturday and Sunday have dramatically higher proportions of casual riders than the weekdays do. Another interesting thing is that although the weekend proportion of casual riders seem to taper off past 70 degrees, the weekday proportions are generally linear and grow at a constant slope. This makes sense, because casually riding to and from work is non-negotiable and less dependent on the temperature as weekend casual riding is.

0.2.5 Question 8a

Imagine you are working for a bike-sharing company that collaborates with city planners, transportation agencies, and policymakers 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 informs 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 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.

Note: There is no single "right" answer to this question – we are looking for thoughtful reflection and commentary on whether or not this dataset, in its current form, encodes information about equity.

```
In [96]: bike.head()
```

```
Out[96]:    instant      dteday  season  yr  mnth  hr holiday weekday workingday  \
         0        1  2011-01-01       1   0     1   0      no     Sat         no
         1        2  2011-01-01       1   0     1   1      no     Sat         no
         2        3  2011-01-01       1   0     1   2      no     Sat         no
         3        4  2011-01-01       1   0     1   3      no     Sat         no
         4        5  2011-01-01       1   0     1   4      no     Sat         no

           weathersit  temp   atemp   hum  windspeed  casual  registered  cnt  \
         0      Clear  0.24  0.2879  0.81        0.0       3          13   16
         1      Clear  0.22  0.2727  0.80        0.0       8          32   40
         2      Clear  0.22  0.2727  0.80        0.0       5          27   32
         3      Clear  0.24  0.2879  0.75        0.0       3          10   13
         4      Clear  0.24  0.2879  0.75        0.0       0           1    1

            prop_casual
         0     0.187500
         1     0.200000
         2     0.156250
         3     0.230769
         4     0.000000
```

As stated before, I would like to see a 'location' variable so that the dataset can inform us where the trips using the bike-sharing program is starting and ending. It would also be helpful to know which area of D.C. bike-sharing is most acccessible by seeing where these bikes are returned to and most available in number.

It may also be informative to add a ‘gender’ variable to see whether there is a difference in usage. If there is, then gender equity may be of concern to the company as it factors in inclusitivity.

We found that the usage pattern of registered riders peak during rush hours of going to and from work, so we can make a reasonable assumption that the majority demographic of people using this ride-sharing program is of the working class. This is something that the current data tells us, but with the introduction of the ‘location’ variable, which can come in the form of separate ‘latitude’ and longitude’ variables, would give us a lot more to work with in the pursuit of equity. This would likely require the granularity to be at the individual level, where each user would be a record. Because of this, other demographic attributes such as race, gender, and socio-economic class could also be surveyed.

0.2.6 Question 8b

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 in the US.

Based on your plots in this assignment, would you recommend expanding bike sharing to additional cities in the US? If so, what cities (or types of cities) would you suggest? Please list at least two reasons why, and mention which plot(s) you drew your 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 like to see how price fluctuates throughout the hours of the day. I presume that there has to be a surge charge for rush hours in order to maximize profit and monitor bike supply. Based on on what we found earlier in the "Temperature vs Casual Rider Proportion by Weekday" plot, I would suggest cities with decently warm weather to implement bike-sharing expansion. It makes even more sense if the cities chosen have a dense working population, because the "Average Count of Casual vs Registered by Hour" plot demonstrated peaks in usage amongst registered riders around rush hours. Other considerations include whether the city has a robust public transportation system with a lack of walkability in the "last mile". This type of city would benefit a lot from a ride-sharing program since it would make the last mile to and from work a lot easier. Another consideration would be the level of car congestion and bike-ability. If a city has heavy car congestion but has well-structured bike lanes, a bike-sharing expansion there makes a lot of sense.