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 shows the data of number of bikes rented for each hour with weather variables- temp, atemp, hum etc.

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?

1. Location might be useful to determine rental demand for example places with more parks might have higer demand.
2. Bike condition, which can affect users willingness to rent the bike

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.

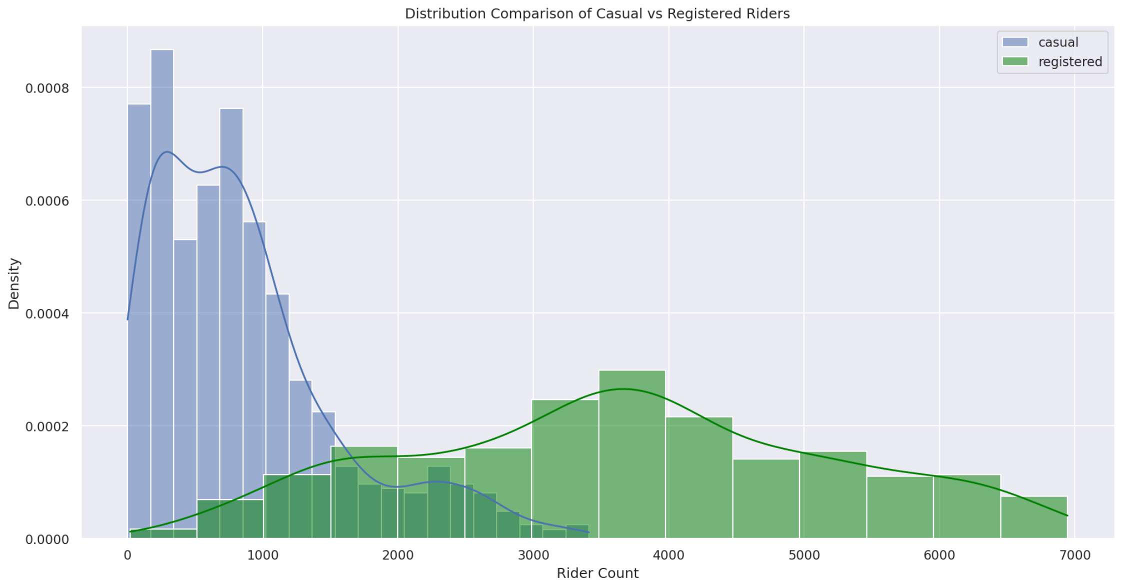
Hints: - You will need to set the `stat` parameter appropriately to match the desired plot. - The `label` parameter of `sns.histplot` allows you to specify, as a string, how the plot should be labeled in the legend. For example, passing in `label="My data"` would give your plot the label "My data" in 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 [58]: sns.histplot(daily_counts['casual'], label='casual', kde = True, stat = "density")
         sns.histplot(daily_counts['registered'],  label='registered', color = "green", kde = True, sta
         plt.xlabel("Rider Count")
         plt.title("Distribution Comparison of Casual vs Registered Riders")
         plt.legend()
```

```
Out[58]: <matplotlib.legend.Legend at 0x7f2a34916d10>
```


Distribution Comparison of Casual vs Registered Riders
casual
registered
Density
Rider Count
0.0000
0.0002
0.0004
0.0006
0.0008
0
1000
2000
3000
4000
5000
6000
7000

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.

Casual rider has a bimodal distribution and it is skewed to the right. Whereas registered rider has a more symmetric distribution with peak at around 3750 and its spread is much wider than casual rider distribution.

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.

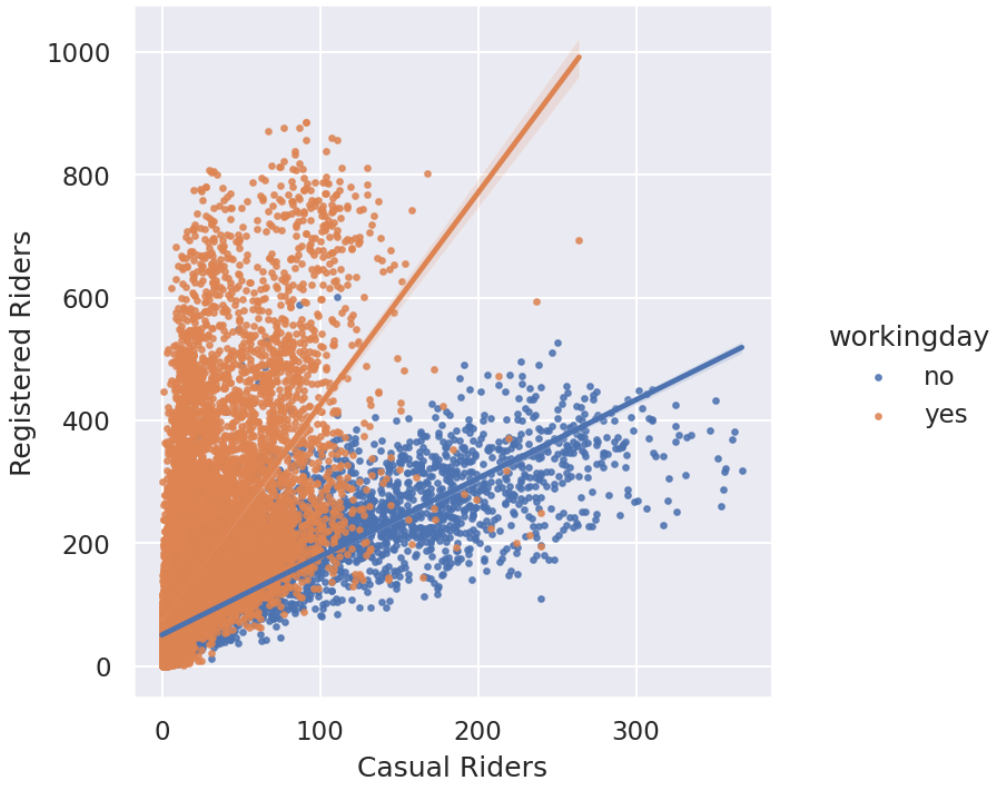
The `lmplot` function will also try and draw a linear regression line (just as you saw in Data 8). Color the points in the scatterplot according to whether or not the day is a working day (your colors do not have to match ours exactly, but they should be different based on whether the day is a working 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 [59]: sns.set(font_scale=1) # This line automatically makes the font size a bit bigger on the plot.
         sns.lmplot(data = bike, x = "casual", y = "registered", hue = "workingday", scatter_kws={"s": 
         plt.title("Comparison of Casual vs Registered Riders on Working and Non-working Days")
         plt.xlabel("Casual Riders")
         plt.ylabel("Registered Riders")
```

```
Out[59]: Text(81.0175416666667, 0.5, 'Registered Riders')
```

Comparison of Casual vs Registered Riders on Working and Non-working Days



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 is a positive correlation between casual riders and registered riders on weekend and weekdays. Overplotting will make it difficult to distinguish individual datapoints and thus making it difficult to see the main trend.

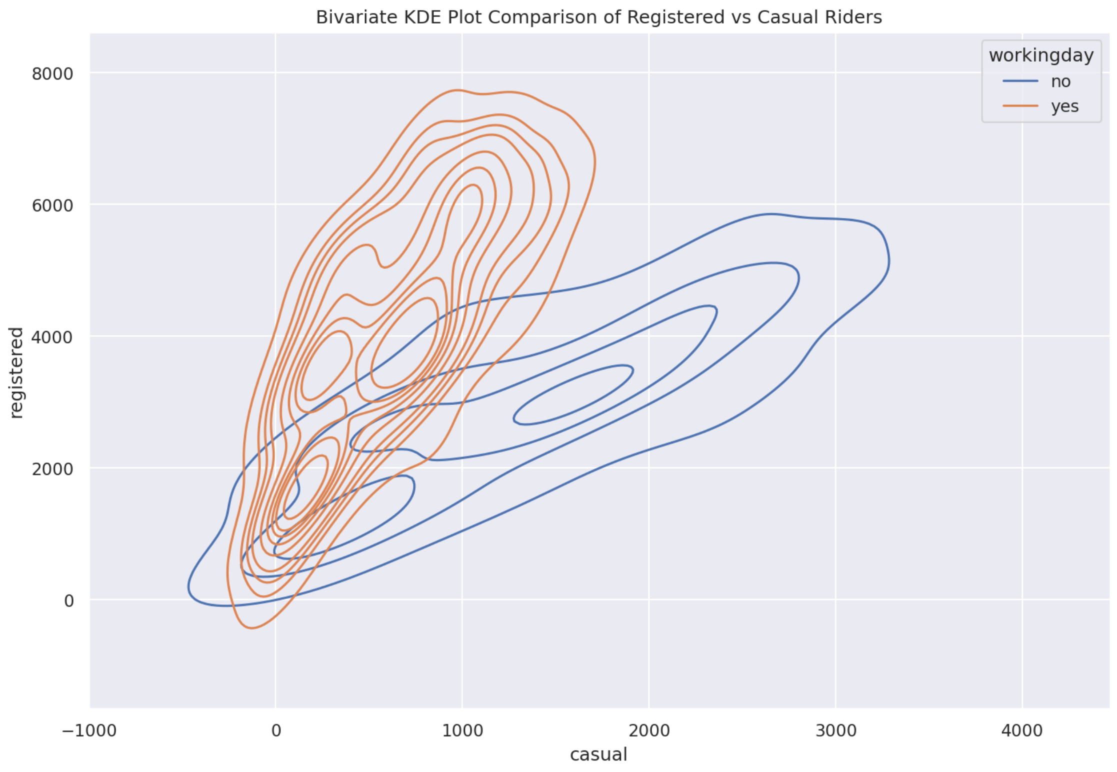
0.0.7 Question 4a (Bivariate Kernel Density Plot)

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

         sns.kdeplot(data=daily_counts, x='casual', y='registered', hue = "workingday")
         plt.title('Bivariate KDE Plot Comparison 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.

Each contour line represent area with the same density of data points and lighter color means less datapoints in the region.

0.0.9 Question 4c

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

Contour plots are more effective with high event counts, because the relative frequency of events is displayed in more detail than scatter plot.

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`.

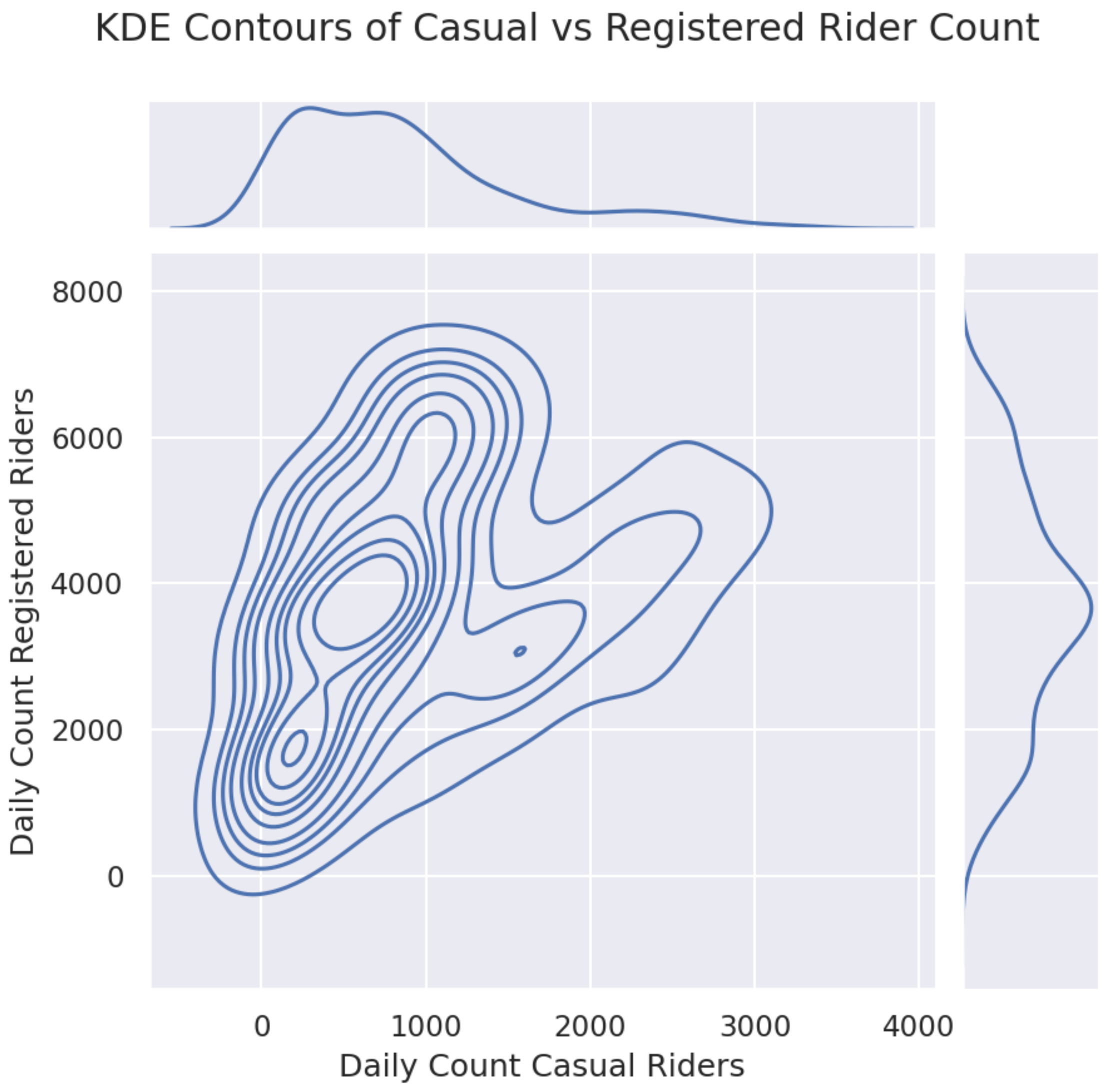
Hints: * The seaborn plotting tutorial has examples that may be helpful. * Take a look at `sns.jointplot` and its `kind` parameter. * `set_axis_labels` can be used to rename axes on a `seaborn` plot. For example, if we wanted to plot a scatterplot with 'Height' on the x-axis and 'Weight' on the y-axis 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 [67]: graph = sns.jointplot(data = daily_counts, x = "casual", y = "registered", kind="kde")
         plt.suptitle("KDE Contours of Casual vs Registered Rider Count")
         graph.set_axis_labels("Daily Count Casual Riders", "Daily Count Registered Riders")
         plt.subplots_adjust(top=0.9);
```


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

0.2 6: Understanding Daily Patterns

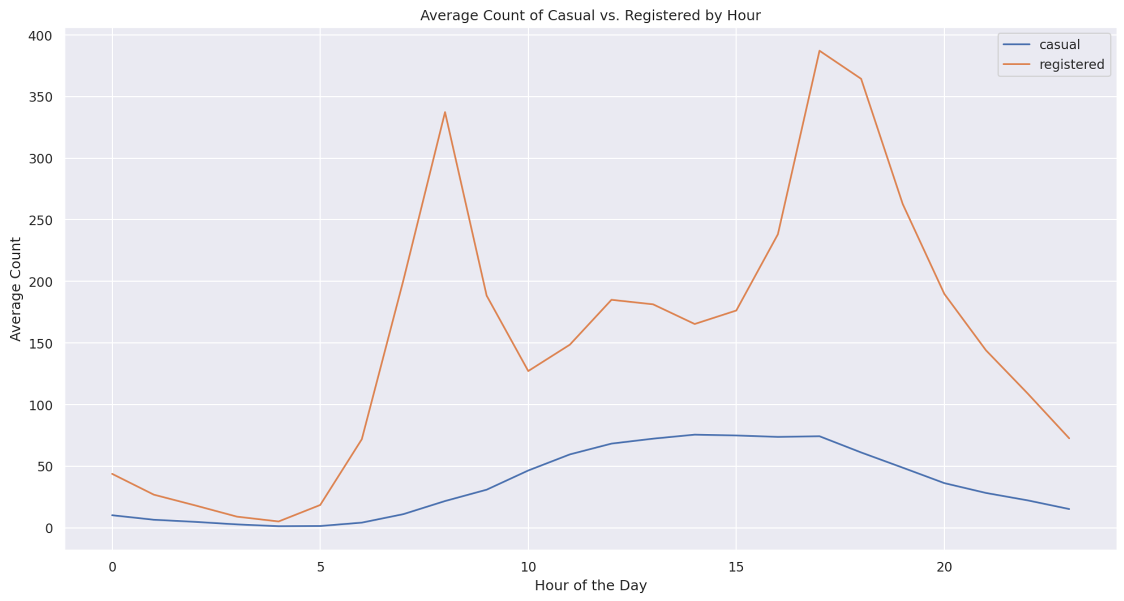
0.2.1 Question 6a

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 [79]: newdf = bike.groupby("hr").agg({"casual": "mean", "registered": "mean"})
         sns.lineplot(data = newdf, x = "hr", y = "casual", label = "casual")
         sns.lineplot(data = newdf, x = "hr", y = "registered", label = "registered")
         plt.xlabel("Hour of the Day")
         plt.ylabel("Average Count")
         plt.title("Average Count of Casual vs. Registered by Hour")
```

```
Out[79]: Text(0.5, 1.0, 'Average Count of Casual vs. Registered by Hour')
```



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 significantly more registered riders than casual riders. Casual rider has steady count througout the day. Registered riders' distribution peak at around 8am and around 5pm. It may be the time where returning registered rider ride consistently.

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.

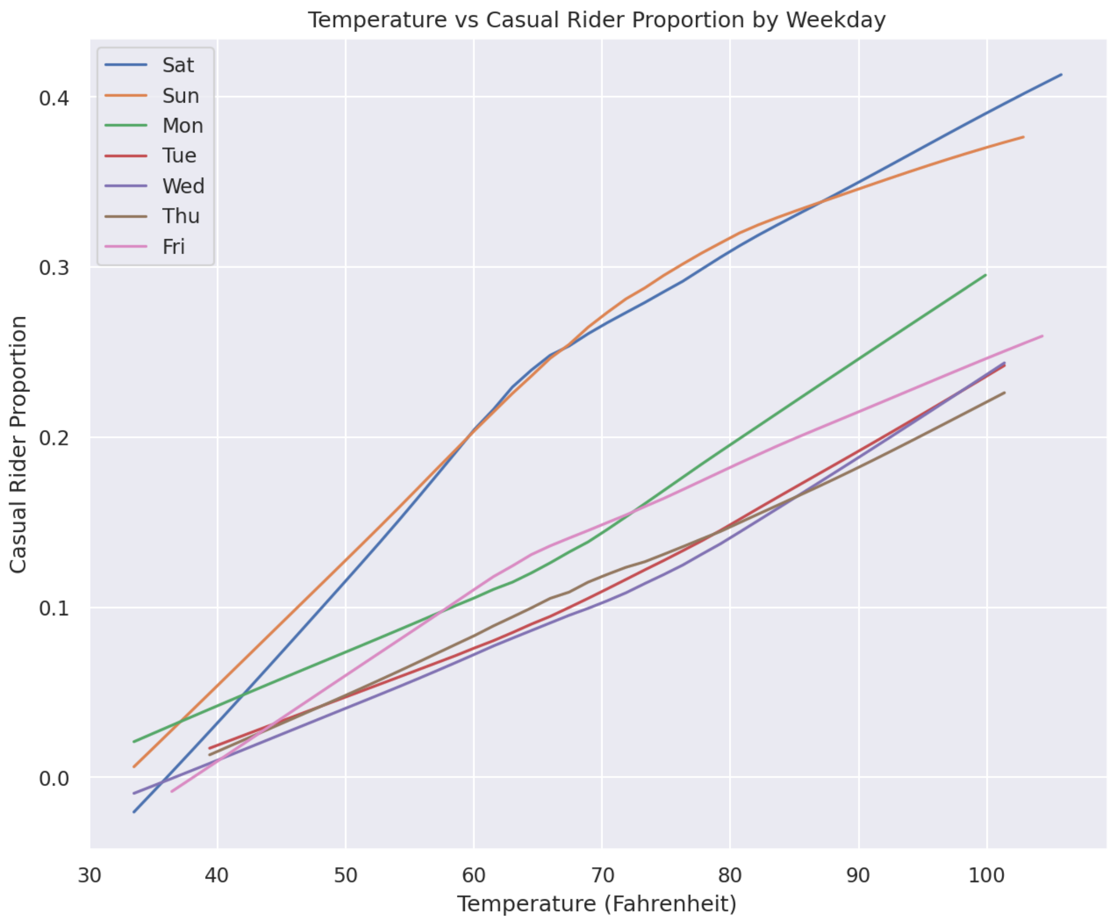
- The `lowess` function expects the `y` coordinate first, then the `x` coordinate. You should also set the `return_sorted` field to `False`.
- **You will need to rescale the normalized temperatures stored in this dataset to Fahrenheit values.** Look at the section of this notebook titled 'Loading Bike 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 [119]: from statsmodels.nonparametric.smoothers_lowess import lowess

          plt.figure(figsize=(10,8))
          days_series = []
          bike["temp in F"] = bike["temp"]* 41 * (9 / 5) + 32
          day_arr = bike["weekday"].unique()

          for days in day_arr:
              days_series.append(bike[bike["weekday"] == days])


          for day_ser in days_series:
              y_smooth2 = lowess(day_ser['prop_casual'], day_ser['temp in F'], return_sorted=False)
              sns.lineplot(x = day_ser['temp in F'], y = y_smooth2, label = day_ser.iloc[1, 7])
```

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

```
Out[119]: Text(0.5, 1.0, 'Temperature vs Casual Rider Proportion by Weekday')
```



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?

The proportion of casual rider increases as temperature increases. Weekends has steeper slopes- more riders.

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.

There is no information on pricing and demographic data. Pricing information would be useful to understand the demand at different price point to different people from different income/social classes.

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 recommend expanding bike sharing to SF. Based on the plot in 6a we see that the number of riders peaks around work start and end time which is also the time when peak hour congestion occurs in SF. Also, based on the plot in 7b we see that the proportion of casual riders increases as temperature increases which is optimum for cities with mild climate like SF.