0.0.1 Question 0

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

The more detailed the data, the more granular it is considered. Overall, this dataset can be considered quite granular due to certain columns such as season, mnth, year, hr, weekday. On the other hand, columns such as dteday, holiday do not provide enough information or club too much information together and reduce the granularity of the dataset.

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 holiday column does not specify which column since the sales of bikes can increase during certain holidays such as Christmas and not change on holidays such as 4th of July. It also does not metion the point of pick-up and drop-off for the bike and the duration of ride.

In addition to the current variables, we can easily add the latitude and longitude of the pick-up and drop-off spots for the bikes. These can be in the form of 4 columns. In addition to these, we can also collect the time in hours, minutes and seconds between pick-up and drop-off of the bike.

0.0.2 Question 2

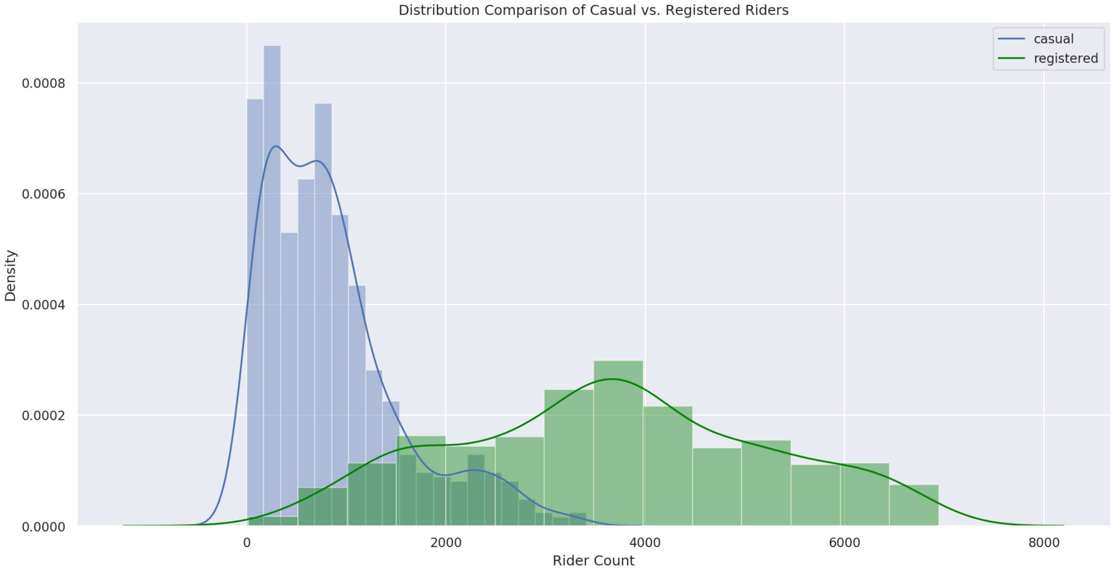
Question 2a Use the sns.distplot function to create a plot that overlays the distribution of the daily counts of bike users, using blue to represent `casual` riders, and green to represent `registered` riders. The temporal granularity of the records should be daily counts, which you should have after completing question 1c. **You can ignore all warnings that say `distplot` is a deprecated function.**

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 [33]: plt.figure(figsize=(5,3))
         fig, ax = plt.subplots()
         sns.distplot(daily_counts['casual'])
         sns.distplot(daily_counts['registered'], color="green")
         plt.xlabel("Rider Count")
         plt.ylabel("Density")
         plt.title('Distribution Comparison of Casual vs. Registered Riders')
         plt.legend(["casual","registered"], loc="upper right")
         plt.show()
```

```
/opt/conda/lib/python3.8/site-packages/seaborn/distributions.py:2551: FutureWarning: `distplot` is a dep
  warnings.warn(msg, FutureWarning)
/opt/conda/lib/python3.8/site-packages/matplotlib/cbook/__init__.py:1402: FutureWarning: Support for mul
  ndim = x[:, None].ndim
/opt/conda/lib/python3.8/site-packages/matplotlib/axes/_base.py:276: FutureWarning: Support for multi-di
  x = x[:, np.newaxis]
/opt/conda/lib/python3.8/site-packages/matplotlib/axes/_base.py:278: FutureWarning: Support for multi-di
  y = y[:, np.newaxis]
/opt/conda/lib/python3.8/site-packages/seaborn/distributions.py:2551: FutureWarning: `distplot` is a dep
  warnings.warn(msg, FutureWarning)
/opt/conda/lib/python3.8/site-packages/matplotlib/cbook/__init__.py:1402: FutureWarning: Support for mul
  ndim = x[:, None].ndim
/opt/conda/lib/python3.8/site-packages/matplotlib/axes/_base.py:276: FutureWarning: Support for multi-di
  x = x[:, np.newaxis]
/opt/conda/lib/python3.8/site-packages/matplotlib/axes/_base.py:278: FutureWarning: Support for multi-di
  y = y[:, np.newaxis]
```

```
<Figure size 750x450 with 0 Axes>
```


Distribution Comparison of Casual vs. Registered Riders
Density
0.0000
0.0002
0.0004
0.0006
0.0008
Rider Count
0
2000
4000
6000
8000
casual
registered

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.

In terms of mode: The curve for the registered riders ranges from 3500 to 4000 riders per day with the highest density above 0.0008. For casual riders, the mode of curve is much lower at around 160-170 per day and the density is also lower and around 0.0003.

In terms of skewness and symmetry: The curve for the registered riders is fairly symmetrical, but the distribution of casual riders is positively skewed with a longer tail to the right side.

In terms of distribution spread: The casual riders are distributed between 0 to 3000 while registered riders are distributed between 0 to 8000.

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).

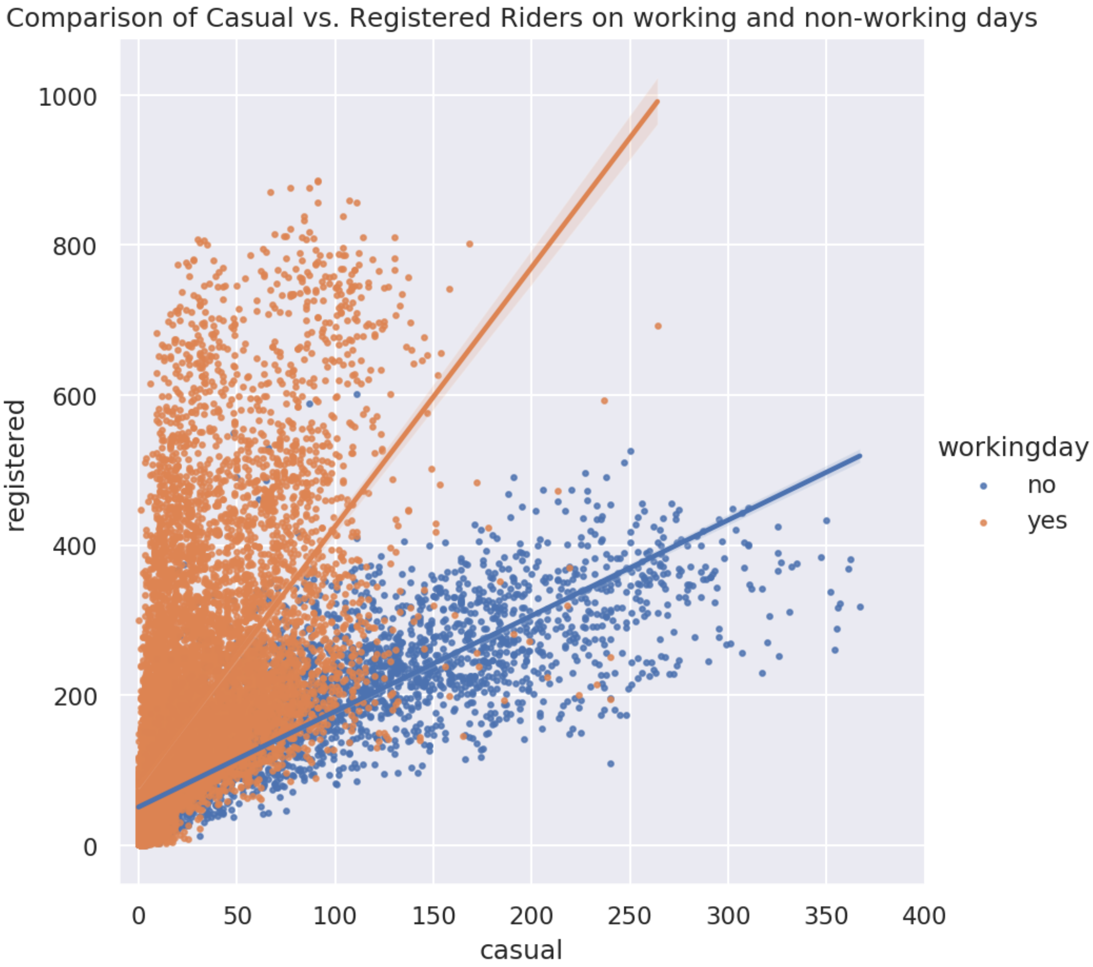
There are many points in the scatter plot, so make them small to help reduce overplotting. Also make sure to set fit_reg=True to generate the linear regression line. You can set the height parameter if you want to adjust the size of the lmplot.

Hints: * Checkout this helpful tutorial on lmplot.

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

```
In [18]: # Make the font size a bit bigger
         sns.set(font_scale=1)
         plt.figure(figsize=(12,7))

         sns.lmplot(x="casual",y="registered", hue="workingday", data=bike, scatter_kws={"s": 5}, fit_r
         plt.xlim(-10,400)
         plt.xlabel("casual")
         plt.ylabel("registered")
         plt.title('Comparison of Casual vs. Registered Riders on working and non-working days')

         plt.show()
```

```
<Figure size 1800x1050 with 0 Axes>
```


Comparison of Casual vs. Registered Riders on working and non-working days
registered
1000
800
600
400
200
0
casual
0
50
100
150
200
250
300
350
400
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?

There is a positive correlation between number of casual riders and registered riders, so as one increases, the other increases as well.

As for whether the day is a workday, on workdays, the ratio of registered to casual is 4:1 and on non-workingday the ratio was 2:1. This shows that while number of casual riders stays constant, the number of registered riders on workingdays is double of that on non-workingday. Conversely, casual riders tend to increase on non-workingdays to twice the number on workingdays.

There is a fair amount of overlap between those riding on working and non-working days near the registered (y-axis). Due to this overlapping, it is hard to estimate the number of blue and orange points. It makes the graph hard to read, especially in the region close to the origin.

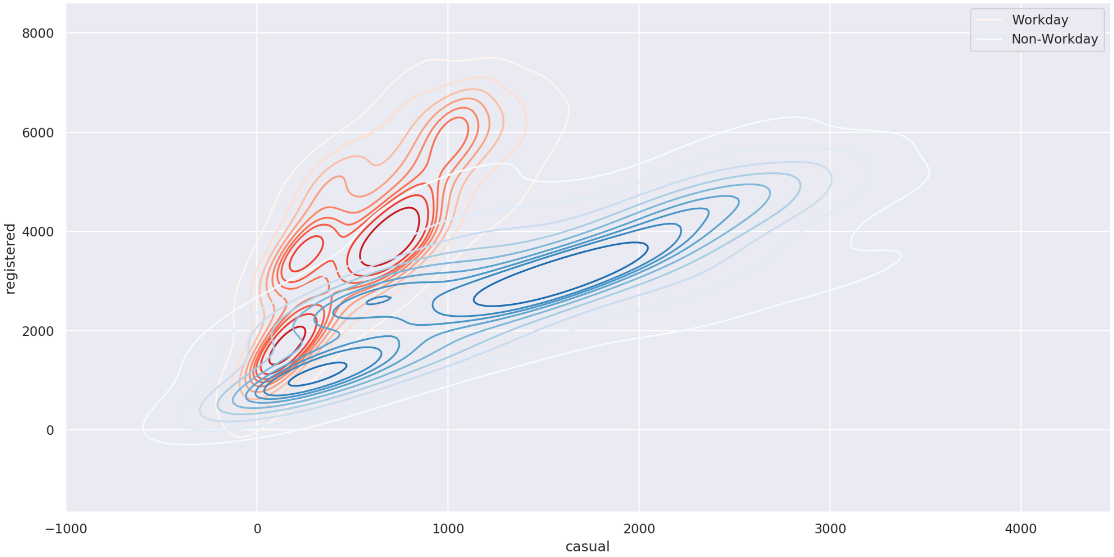
Generating the plot with weekend and weekday separated can be complicated so we will provide a walkthrough below, feel free to use whatever method you wish if you do not want to follow the walkthrough.

Hints: * You can use `loc` with a boolean array and column names at the same time * You will need to call kdeplot twice. * 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). 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 'is_workingday' to a boolean array that is true for all working_days
         is_workingday = np.asarray(daily_counts["workingday"].replace(["yes","no"],[True, False]).to_li

         # 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[daily_counts["workingday"]=="yes"]["casual"]
         registered_workday = daily_counts.loc[daily_counts["workingday"]=="yes"]["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",label="Workday")

         # Repeat the same steps above but for rows corresponding to non-workingdays
         casual_non_workday = daily_counts.loc[daily_counts["workingday"]=="no"]["casual"]
         registered_non_workday = daily_counts.loc[daily_counts["workingday"]=="no"]["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", label="Non-Workday")
         plt.legend(loc="upper right")
```

```
Out[20]: <matplotlib.legend.Legend at 0x7fb962b500d0>
```


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

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

In case of non-workingdays, some of the ranges of number of times casual riders used the bikes are between 0 and 300, 200 and 500, 500 to 900 and for registered bikers the ranges are 1000 to 2200, 3000 to 4000 and, 3000 to 5000.

In case of workingdays, some of the ranges of number of times casual riders used the bike are 0 to 800, 1100 to 2000 and for registered bikers the ranges are 800 to 1900, and 2500 to 4000.

This plot allows us to view these ranges more clearly.

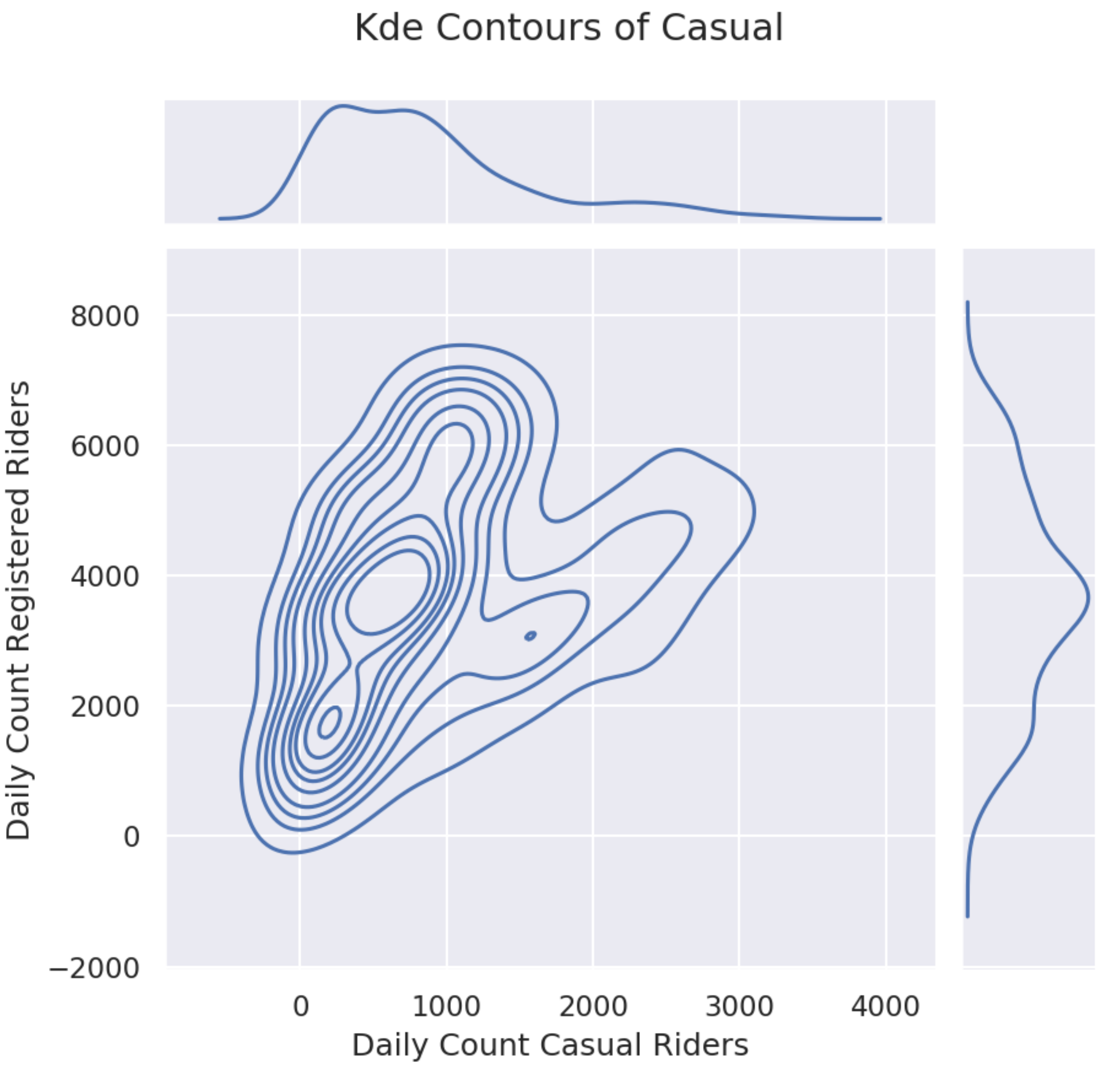
0.1 4: Joint Plot

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

```
In [21]: sns.jointplot(x="casual", y="registered", data=daily_counts, kind="kde").set_axis_labels("Daily
         plt.suptitle("Kde Contours of Casual ")
         plt.subplots_adjust(top=0.9)
```

```
/opt/conda/lib/python3.8/site-packages/matplotlib/cbook/__init__.py:1402: FutureWarning: Support for mul
  ndim = x[:, None].ndim
/opt/conda/lib/python3.8/site-packages/matplotlib/axes/_base.py:276: FutureWarning: Support for multi-di
  x = x[:, np.newaxis]
/opt/conda/lib/python3.8/site-packages/matplotlib/axes/_base.py:278: FutureWarning: Support for multi-di
  y = y[:, np.newaxis]
/opt/conda/lib/python3.8/site-packages/matplotlib/cbook/__init__.py:1402: FutureWarning: Support for mul
  ndim = x[:, None].ndim
/opt/conda/lib/python3.8/site-packages/matplotlib/axes/_base.py:276: FutureWarning: Support for multi-di
  x = x[:, np.newaxis]
/opt/conda/lib/python3.8/site-packages/matplotlib/axes/_base.py:278: FutureWarning: Support for multi-di
  y = y[:, np.newaxis]
```


Kde Contours of Casual
8000
6000
4000
2000
0
−2000
Daily Count Registered Riders
0
1000
2000
3000
4000
Daily Count Casual Riders

0.2 5: Understanding Daily Patterns

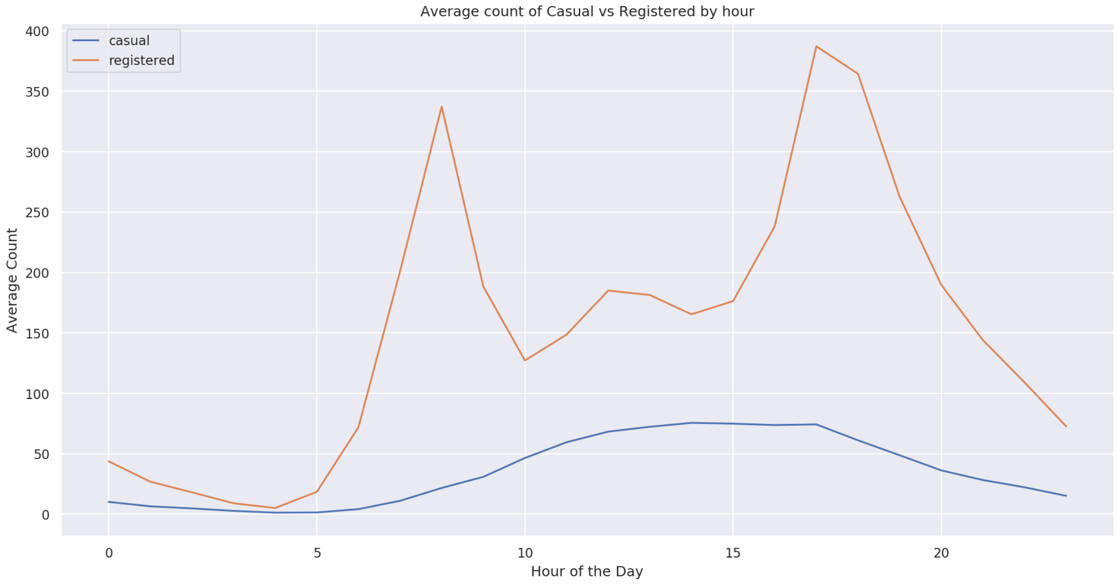
0.2.1 Question 5

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]: df2 = bike.groupby("hr", as_index=False).aggregate("mean")
         df2 = df2[["hr","casual","registered"]]
         df2.head()
         sns.lineplot(x="hr",y="casual", data=df2, label="casual")
         sns.lineplot(x="hr", y="registered", data=df2, label="registered")
         plt.xlabel("Hour of the Day")
         plt.ylabel("Average Count")
         plt.title("Average count of Casual vs Registered by hour")
         plt.legend(loc="upper left")
```

```
Out[22]: <matplotlib.legend.Legend at 0x7fb962b2e100>
```



0.2.2

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

For casual riders, the distribution curve is increasing till 12pm and then plateaus till around 4pm after which the number of casual riders begins to reduce. Overall, the curve is very smooth, which could mean there are no times when there's a sharp increase or decrease in number of casual riders. It reaches a low peak between 70 and 80 which is much lower than registered riders.

For registered riders, we see the 2 distinct modes roughly around 9am and at 6pm which shows the 2 timings when office-goers ride to office and ride back home. There is another, smaller mode around 12pm which could be the time when some people choose to ride to a different place for their lunch break. The number of riders decreases sharply at around 7pm which keeps decreasing till 4am on the next day after which the cycle continues as the number of riders starts increasing till 9am.

The peaks represent the times when the number of riders sharply increase and decrease within a short span of time.

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.

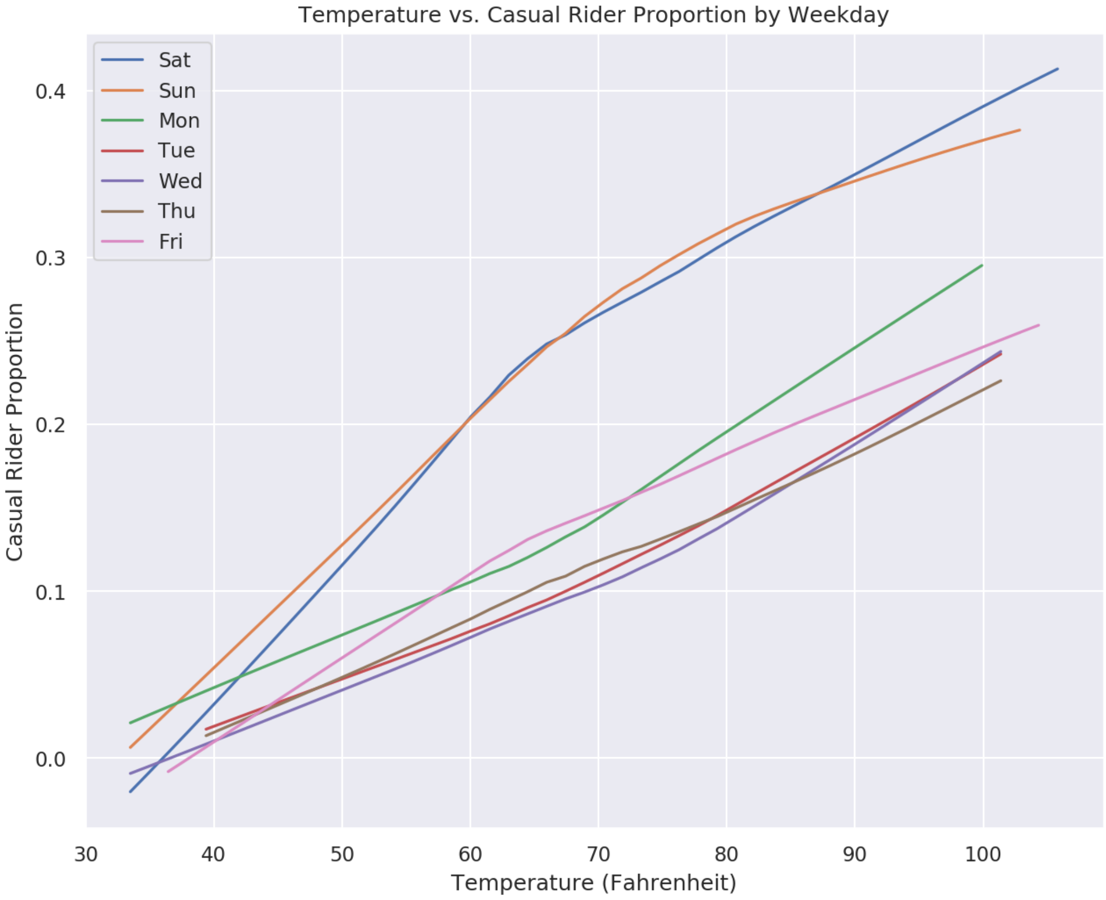
- The `lowess` function expects y coordinate first, then x coordinate.
- Look at the top of this homework notebook for a description of the temperature field to know how to convert to Fahrenheit. By default, the temperature field ranges from 0.0 to 1.0. In case you need it, Fahrenheit $=$ Celsius $*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

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

```
In [28]: from statsmodels.nonparametric.smoothers_lowess import lowess

         plt.figure(figsize=(10,8))
         #for sunday, need to multiply by 41 for celsius
         #now try for all 7 days:
         days = bike["weekday"].unique().tolist()
         for d in days:
             x_val = bike[bike["weekday"]==d]["temp"].to_list()
             x_to_c = [((i*41)*(9/5))+ 32 for i in x_val]
             y_val = bike[bike["weekday"]==d]["prop_casual"].to_list()

             ysmooth = lowess(y_val, x_to_c, return_sorted=False)
             sns.lineplot(x_to_c, ysmooth, label=d)
             plt.legend();

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

```
/opt/conda/lib/python3.8/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following var
  warnings.warn(
/opt/conda/lib/python3.8/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following var
  warnings.warn(
/opt/conda/lib/python3.8/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following var
  warnings.warn(
/opt/conda/lib/python3.8/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following var
  warnings.warn(
/opt/conda/lib/python3.8/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following vari
  warnings.warn(
/opt/conda/lib/python3.8/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following vari
  warnings.warn(
/opt/conda/lib/python3.8/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following vari
  warnings.warn(
```

```
Out[28]: Text(0.5, 1.0, 'Temperature vs. Casual Rider Proportion by 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?

Prop_casual is increasing as temperature increases so there is a positive correlation between the two. Regardless of temperature, the proportion of casual riders is always higher on weekends (Saturday and Sunday).

For the weekdays, the lines come very close to linear for Tuesday, Wednesday and Thursday. The line for Monday is convex while that for Friday is concave.

0.2.3 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.

The bike dataset contains a lot fo important information regarding the weather or the number of people overall who use the bike on a particular day. Unofrtunately it lacks other information specific to the rider, such as gender of the rider, or the location where the rider picks up and drops off the bike. So I would increase the granularity of the count column, and create column for number of riders for different genders. Also, since we are focusing on the Washington DC area, I would break the area down by postal code to get a better idea of the distribution of bikes across the city as well understand the demand for bikes in different areas.

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 think whether or not we should expanding into a new city depends on multiple factors which cannot be determined by looking at plots of Washington DC. Things such as the alternative forms of transportation such as bus or train heavily influence whether people will switch to bikes. Also, we need to ensure the roads have bike routes, which might not be the case for some states.

According to plot in 5a, we can see that time of the day really affects the number of riders, casual or registered. So I would recommend looking into how the people choose to travel at those time and if there is a lot of congestion/traffic on the roads they may choose not to use bikes around peak times.

Also we saw in 6b that there is a positive correlation between temperature and number of riders. So I would give priority to those areas with higher temperatures and more pleasant weather conditions.