Notebook

July 9, 2019

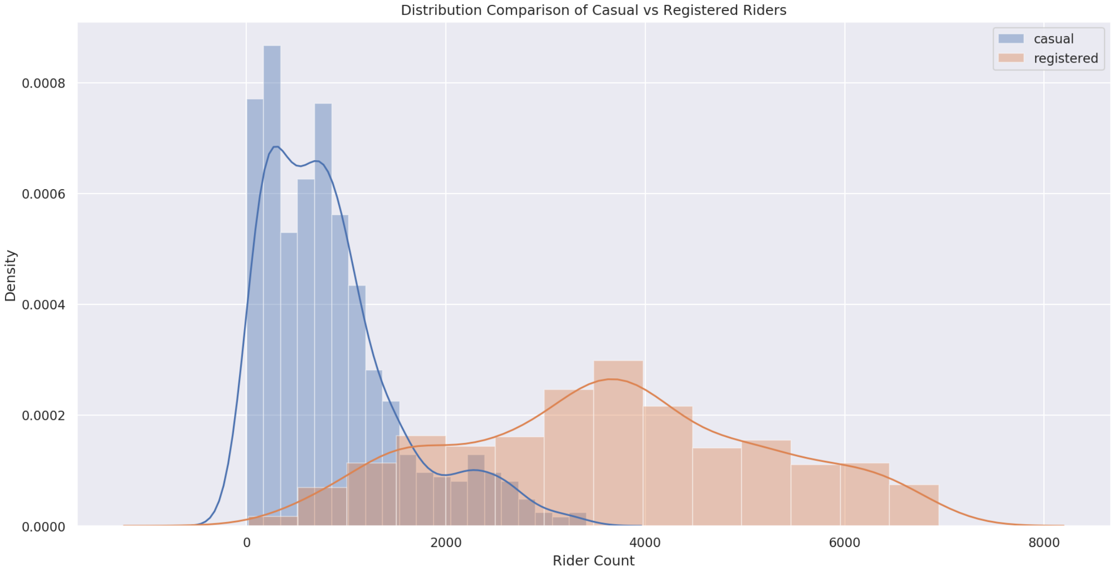
0.0.1 Question 2

Question 2a Use the sns.distplot function to create a plot that overlays the distribution of the daily counts of casual and registered users. 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 [85]: sns.distplot(daily_counts['casual'], hist=True, rug=False, label='casual');
         sns.distplot(daily_counts['registered'], hist=True, rug=False, label ='registered');
         plt.legend()
         plt.xlabel('Rider Count')
         plt.ylabel('Density')
         plt.title('Distribution Comparison of Casual vs Registered Riders')
```

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



0.0.2 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.

1. modes: For the registered riders, the mode of its curve is around 3500 to 4000 riders per day, with a high density over 0.0008, while for the casual riders, the mode of its curve is around 166 per day, which is much less than the number of registered riders per day. Also, the density of its mode is much smaller than the one of the casual riders, only around 0.0003.
2. skewness and symmetry: The data of the registered riders are fairly symmetrical, while the distribution of causal riders is of positive skewness, where its tail on the right side of the distribution is longer.
3. Spread of the distributions: For the casual riders, the distribution of riders is mostly between 0 and 2000, while the distribution of registered riders is fairly between 0 and 8000.

0.0.3 Question 2c

In addition to the type of rider (casual vs. registered) and the overall count of each, what other kinds of demographic data would be useful (e.g. identity, neighborhood, monetary expenses, etc.)?

What is an example of a privacy or consent issue that could occur when accessing the demographic data you brought up in the previous question?

1. Neigborhood:By counting the number of riders in all neigborhood, we can seethe density of sharing bikes' usage of each, helping the city planners to find out the one with less access to sharing bikes.Also, the ones with the most riders are probably the busiest districts, where we can dig more details from it by looking its transporation expenses.
2. Monetary expenses: By analyzing the data of monetary expenses in transporation of city residents, the city planners can find ways to reduce the transporation costs in districts or neigborhood with high heavy congestion and high transportation costs.
3. Identity: By grouping the data by riders' identities, such as occupations, age, races, and income, we can see more details of the distribution of the usage of sharing bikes among different social groups. In this way, the city planners can assess whether there are inequity in transportation .

0.0.4 Question 2d

What is an example of a privacy or consent issue that could occur when accessing the demographic data you brought up in the previous question?

-Picking the proposed recruitment methods: How are potential participants identified and contacted? It will violate the individuals' privacy if search through medical records for their demographic data,then have a researcher with no previous contact with potential subject recruit. Also, it's not acceptable to retain sensitive demographic information obtained at screening without the consent of people who either fail to qualify or refuse to participate for possible future studies.

0.0.5 Question 2e

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.

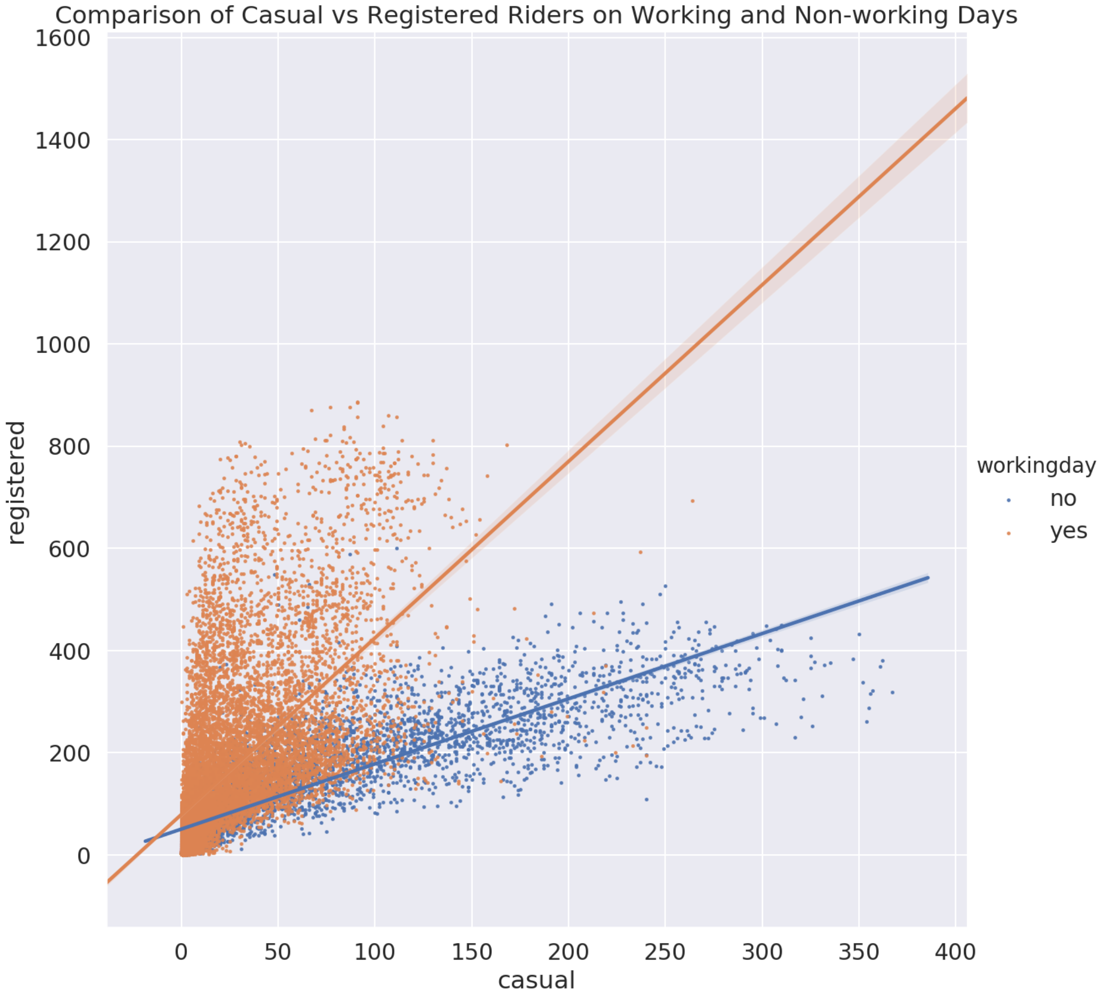
The `lmplot` function will also try to draw a linear regression line (just as you saw in Data 8). Color the points in the scatterplot according to whether or not the day is working day. There are many points in the scatter plot so make them small to help reduce overplotting. Also make sure to set `fit_reg=True` to generate the linear regression line. You can set the `height` parameter if you want to adjust the size of the `lmplot`. Make sure to include a title.

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

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

```
In [86]: # Make the font size a bit bigger
         sns.set(font_scale=1.3)
         sns.lmplot(x = 'casual', y = 'registered', data = bike, hue = 'workingday',fit_reg=True,height
         plt.title('Comparison of Casual vs Registered Riders on Working and Non-working Days')
```

```
Out[86]: Text(0.5, 1, 'Comparison of Casual vs Registered Riders on Working and Non-working Days')
```



0.0.6 Question 2f

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?

1. 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? 1)The number of casual riders are positively correlated with the number of registered riders, which means as the number of the casual riders rises up, the number of the registered riders will also rise up. 2)In the case of workdaythe ratio of registered riders to casual riders is around 4, while during the weekends, the ratio is around 4. This represents that with the same number of casual riders, the demand for sharing bikes of registered riders in workdays is bigger than the one in weekends. On the contrary, with the same number of registered riders, the demand for sharing bikes of casual riders during weekends is bigger than the one during the workdays. 3)During the workdays, there are more data points lying at the bottom left of the plot, which means that the times of registered using sharing bikes are mostly around 0~200, while during the weekends, the distribution of data points are more evenly distributed within all casual riders.
2. What effect does overplotting have on your ability to describe this relationship? Overlapping will make the dots unreadable, because there are too many data points with similar values. If overlapping, at the bottom left all the data points merge together to form a blue region and there is no way to know how many point are there.

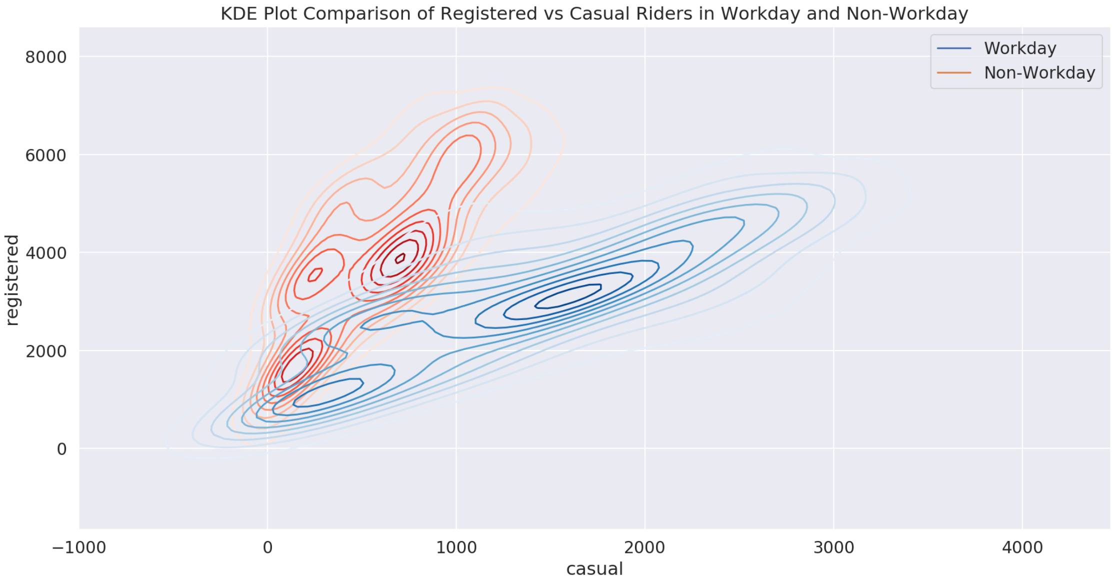
Generating the plot with weekend and weekday separated can be complicated so we will provide a walkthrough below, feel free to use whatever method you wish however if you do not want to follow the walkthrough.

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 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).

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 [88]: 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 = (bike['workingday']=='yes').as_matrix()

         # Bivariate KDEs require two data inputs.
         # In this case, we will need the for casual and registered riders on weekdays
         # Hint: use loc and is_workingday to splice out the relevant rows and column (casual/registere
         a=bike.loc[is_workingday]
         casual_weekday = a['casual'].groupby(a['dteday']).agg('sum')
         registered_weekday = a['registered'].groupby(a['dteday']).agg('sum')

         # Use sns.kdeplot on the two variables above to plot the bivariate KDE for weekday rides
         sns.kdeplot(casual_weekday,registered_weekday,label='Workday',cmap='Reds')

         # Repeat the same steps above but for rows corresponding to non-workingdays
         b=bike.loc[~is_workingday]
         casual_weekend = b['casual'].groupby(b['dteday']).agg('sum')
         registered_weekend = b['registered'].groupby(b['dteday']).agg('sum')

         # Use sns.kdeplot on the two variables above to plot the bivariate KDE for weekday rides
         sns.kdeplot(casual_weekend,registered_weekend,label='Non-Workday',cmap='Blues')

         plt.legend()
         plt.title('KDE Plot Comparison of Registered vs Casual Riders in Workday and Non-Workday');
         sns.set(rc={'figure.figsize':(20,10)})
         sns.set(font_scale=3)
```

```
/srv/conda/envs/data100/lib/python3.6/site-packages/ipykernel_launcher.py:4: FutureWarning: Method .as_
  after removing the cwd from sys.path.
```


KDE Plot Comparison of Registered vs Casual Riders in Workday and Non-Workday
Workday
Non-Workday
8000
6000
4000
2000
0
registered
−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?

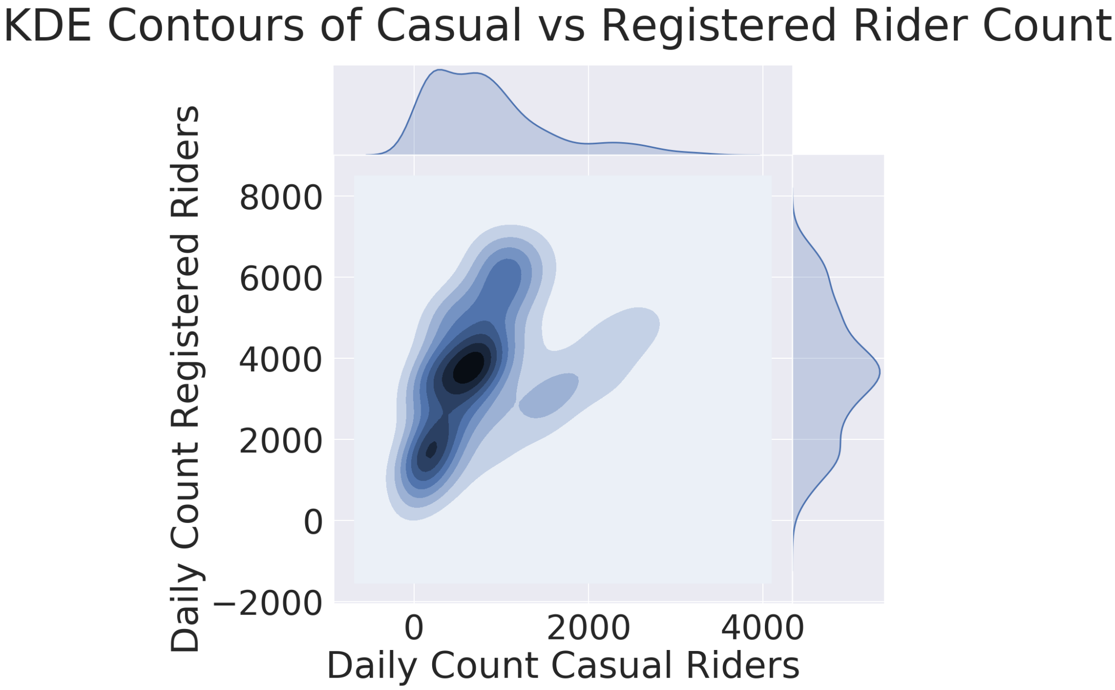
During the workdays,The times of using sharing bikes of casual riders is roughly between 0 and 800,and between 1100 and 2000, while times of using sharing bikes of registered riders is between 800 and 1900, and between 2500 and 4000. In non-workday, The times of using sharing bikes of casual riders is roughly between 0 and 300,between 200 and 500, and between 500 to 900, while times of using sharing bikes of registered riders is between 1000 and 2200, between 3000 and 4000, and between 3000 and 5000.

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 [89]: sns.jointplot("casual", "registered", data=daily_counts,
                       kind="kde", height=10, space=0, color="b").set_axis_labels('Daily Count Casual
         plt.suptitle('KDE Contours of Casual vs Registered Rider Count')
         plt.subplots_adjust(top=0.9)
```



0.2 5: Understanding Daily Patterns

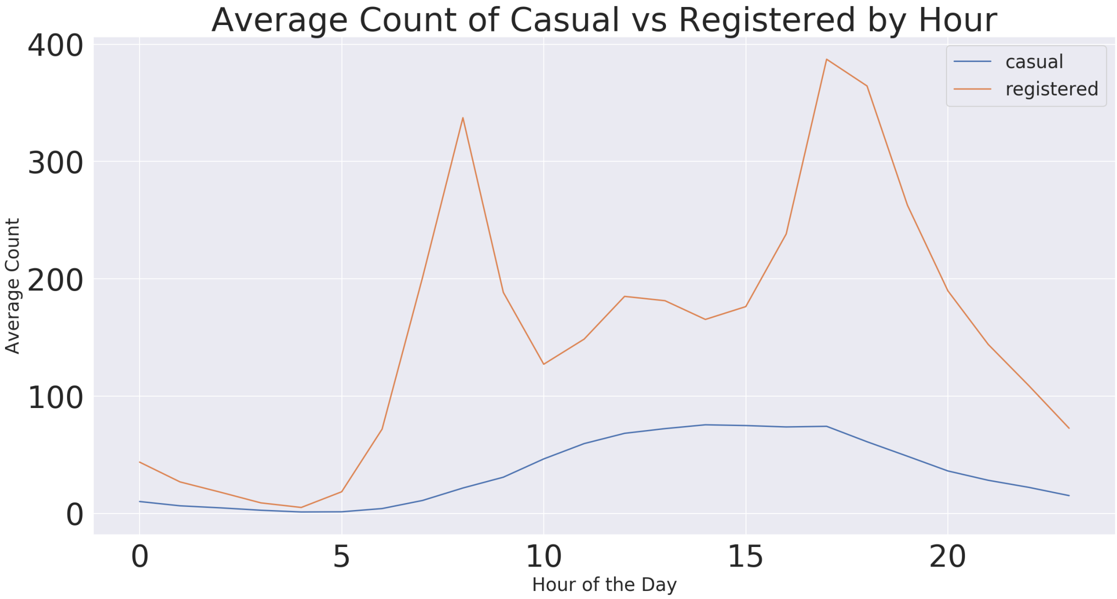
0.2.1 Question 5

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 following:

```
In [90]: regis=bike['registered'].groupby(bike['hr']).agg('mean')
         cas = bike['casual'].groupby(bike['hr']).agg('mean')
         plt.plot(cas,label='casual')
         plt.plot(regis,label='registered')
         plt.legend(prop={'size': 20})
         plt.title('Average Count of Casual vs Registered by Hour')
         plt.xlabel('Hour of the Day',fontsize=20)
         plt.ylabel('Average Count',fontsize=20)
```

```
Out[90]: Text(0, 0.5, 'Average Count')
```



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

1. What can you observe from the plot? 1)For the casual riders, the slope of its curve is quite smooth and small, which means the number of casual riders during the day doesn't has a large change. Its peak of riders is between 13:00 and 15:00 during the day. However, its peak is lower than 100 riders per day, roughly around 75 riders. From around 5:00 am to 15:00, the number of casual riders rises up slowly, and from 17:00 to the late night, the number goes down again and reaches zero at around 4:00 am. 2)For the registered riders, there are three modes during the day, with two great modes and one small mode. At around 8:00 am and at around 17:00, the number of registered riders reachs the peak, with 340 times and 390 times. Another small peak is at around 13:00 with roughly 190 registered riders. From 4:00am to 8:00am, the number of registered riders rises up rapidly, but from 8:00am to 10:00am, the number of registered rider goes down rapidly from around 340 riders to around 140 riders. Later on, the number of registered rider slightly goes up to the small peak at 13:00 from 140 to 190, and it goes down until 15:00, when it begins to experience another rapid increase of riders, from 180 at 15:00 to 390 at 17:00. After that, the number of registered riders keeps going down to zero at 4:00am.
2. Hypothesize about the meaning of the peaks in the registered riders' distribution. It represents the largest number of registered riders during a certain period of day.

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.

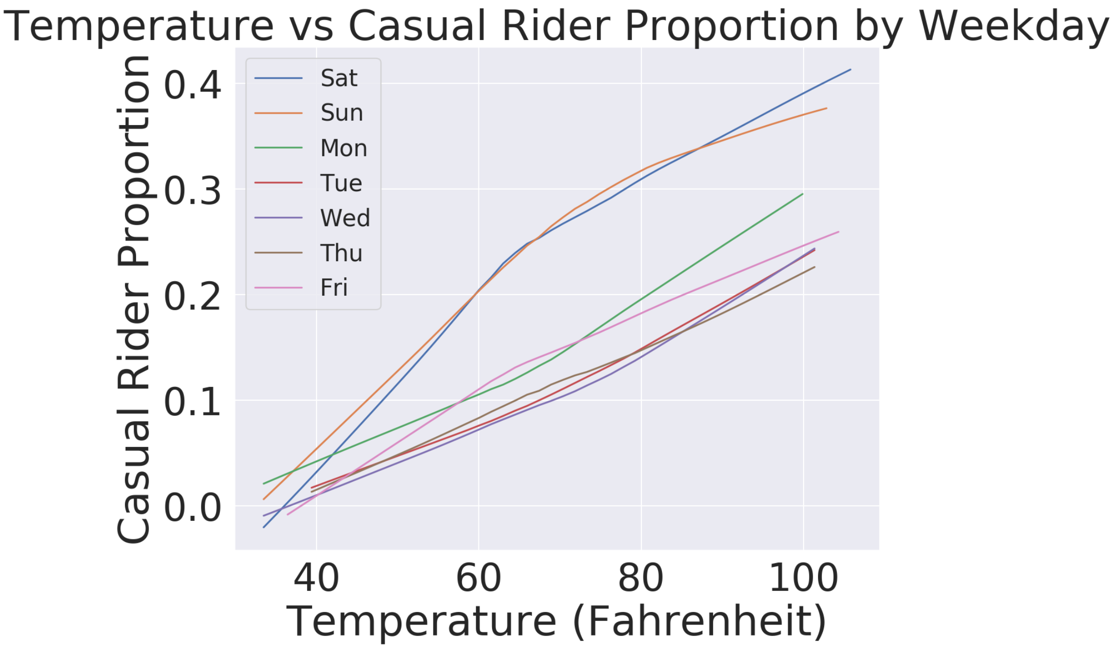
Hints: * Start by just plotting only one day of the week to make sure you can do that first.

- 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, $\text{Fahrenheit} = \text{Celsius} * \frac{9}{5} + 32$.

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

```
In [100]: from statsmodels.nonparametric.smoothers_lowess import lowess

          plt.figure(figsize=(10,8))
          bike['fatemp']=bike['temp']*41*(9/5)+32
          weekday = ['Sat','Sun','Mon','Tue','Wed','Thu','Fri']
          for d in weekday:
              x = bike[bike['weekday']==d]['fatemp'].array
              y =  bike[bike['weekday']==d]['prop_casual'].array
              y_smooth = lowess(y, x, return_sorted=False)
              sns.lineplot(x, y_smooth, label=d)
          plt.title('Temperature vs Casual Rider Proportion by Weekday')
          plt.xlabel('Temperature (Fahrenheit)')
          plt.ylabel('Casual Rider Proportion')
          plt.legend(prop={'size':20})
```

```
Out[100]: <matplotlib.legend.Legend at 0x7f8cac4e8748>
```



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?

For every weekday, as the temperature rises up, the prop_casual goes up as well, which is most obvious in Saturday and Sunday, as the slopes of these two days'curves are the biggest. The curve of each weekday is not linear, but the curves of Tuesday, Wednesday and Thursday are very close to linear line. For Friday,Saturday and Sunday, their curves are slightly concave, while the curve of Monday is slightly convex.

Question 6d Based on the data you have explored (distribution of orders, daily patterns, weather, additional data/information you have seen), do you think bike sharing should be realistically scaled across major cities in the the US in order to alleviate congestion, provide geographic connectivity, reduce carbon emissions, and promote inclusion among communities? Why or why not? Provide a visualisation and justify how it supports your answer

I think it should be realistically scaled across major cities in the the US depending on each city's state. From this image we can see that the registered riders are the majority and the number of them is around 7500 per day at most in the US, which is only a tiny part of the whole population, meaning the user group of bike sharing is not big enough to alleviate congestion and provide enough geographic connectivity, although it can reduce carbon emissions and promote inclusion among communities in some way.

The demand to sharing bikes of casual riders are mostly distributed in the weekends rather than the workdays, and the proportion of casual riders is postively associated with the weather. These informs us that sharing bikes would be most popular among cities with warm weather, where casual riders would like to spend more time riding in the cities during the weekend and the workdays. Also, there are certain peaks during the day for bike sharing, which are the morning, the noon and the afternoon. During these peak times, it might be the busiest time in the cities, where the congestion is the most severe and the transportation cost is the highest. Therefore, bike sharing should be allocated according to time and districts that requires bike sharing the most, but not spreading them randomly. Finally, from this map, we can tell sharing bikes are mostly distributed in the east and the west coast, where some of their cities experience the most severe transportation congestion and require the most amount of bike sharing accordingly. Therefore, bike sharing should be allocated more in these districts.