Data 100/200 Homework 5 Written

Jackie Hu

TOTAL POINTS

33 / 34

QUESTION 1

1 Question 0A+0B 1 / 2

+ **2 pts** **0A + 0B** Fully correct

+ **1 pts** **0A** Correctly identifies granularity: each row represents bike sharing data per hour

✓ **+ 0 pts** **0A** Incorrect/Blank

✓ **+ 1 pts** **0B** Identifies limitations and at least 2 additional data categories/variables

+ **0.5 pts** **0B** Missing limitations or lists fewer than 2 additional data categories/variables

+ **0 pts** **0B** Incorrect/Blank

QUESTION 2

2 Question 2A+2B 4 / 4

Question 2a

✓ **+ 2 pts** **Correct plot, axes, labels, and title**

+ **1 pts** Correct plot

+ **1 pts** Correct axes/labels/title

+ **0 pts** Incorrect/Blank

Question 2b

✓ **+ 2 pts** **Accurate description of the distributions**

+ **0 pts** Incorrect/Blank

QUESTION 3

3 Question 2C+2D 4 / 4

+ **4 pts** All correct

✓ **+ 2 pts** **Addresses relationship and overplotting**

✓ **+ 2 pts** **Correct plot + axes labels and title**

+ **1 pts** Addresses only 1 of relationship or overplotting

+ **1 pts** Correct plot, incorrect axes labels and title

+ **1 pts** Correct axes labels and title. incorrect plot

+ **0 pts** Incorrect/Blank

QUESTION 4

4 Question 3 7 / 7

✓ **+ 7 pts** **Correct**

+ **5 pts** 3a: Correct plot and labels and title

+ **4 pts** 3a: Correct plot

+ **2.5 pts** 3a: Correct workday or non-workday plot only

+ **2 pts** 3a: Correct assignment of all given variables, but no correct plot

+ **1 pts** 3bi: Lines as similarity level of density for the data points, and color shades as the amount of the data points

+ **0.5 pts** 3bi: discussed either lines or color shades

+ **1 pts** 3bii: options: variability is higher on non-workdays; joint distribution of workday is trimodal; joint distribution of non-workday is bimodal

+ **0 pts** Incorrect/Blank

QUESTION 5

5 Question 4 2 / 2

✓ **+ 2 pts** **Correct plot + axis/title**

+ **1 pts** Correct axis/title

+ **1 pts** Correct points

+ **0 pts** Incorrect/Blank

QUESTION 6

6 Question 5A+5B 4 / 4

✓ **+ 2 pts** **(5a) Fully correct**

+ **0.5 pts** (5a) Correct axis labels

+ **0.5 pts** (5a) Distinct colors for different riders

+ **1 pts** (5a) Correct plot

✓ **+ 2 pts** **(5b) Accurate description/interpretation of answer to 5a**

+ **0 pts** (5a) Blank/incorrect

+ **0 pts** (5b) Blank/incorrect

QUESTION 7

7 Question 6B+6C 6 / 6

✓ **+ 6 pts Correct plot + axes labels and title and reasonable explanation**

+ 3 pts Correct points

+ 3 pts Correct axes, labels, and title

+ 0 pts Incorrect/blank

QUESTION 8

8 Question 7A+7B 5 / 5

Question 7a

✓ **+ 2 pts States an improvement with explanation**

+ 2 pts Gives sufficient explanation about equity in the dataset.

+ 1 pts States an improvement without explanation/insufficient explanation

+ 0 pts Incorrect/Blank

Question 7b

✓ **+ 3 pts Refers to a specific plot and states at least 2 reasons for explanation**

+ 2 pts Provides at least 2 reasons without reference to a specific plot.

+ 1 pts Refers to one specific plot

+ 1 pts Gives one reason

+ 0 pts Incorrect/Blank

0.0.1 Question 0

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

- each row is 1 bike use entry within 1 year.

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?

- if we are curious about the geographic distribution about the bike usage then we want to collect where the bike is picked up and dropped off.
- if we are curious about the demographic of the bike users, we want to have some data categories for the users, that can be index on the instance, so we can annalyze user groups and tiers.

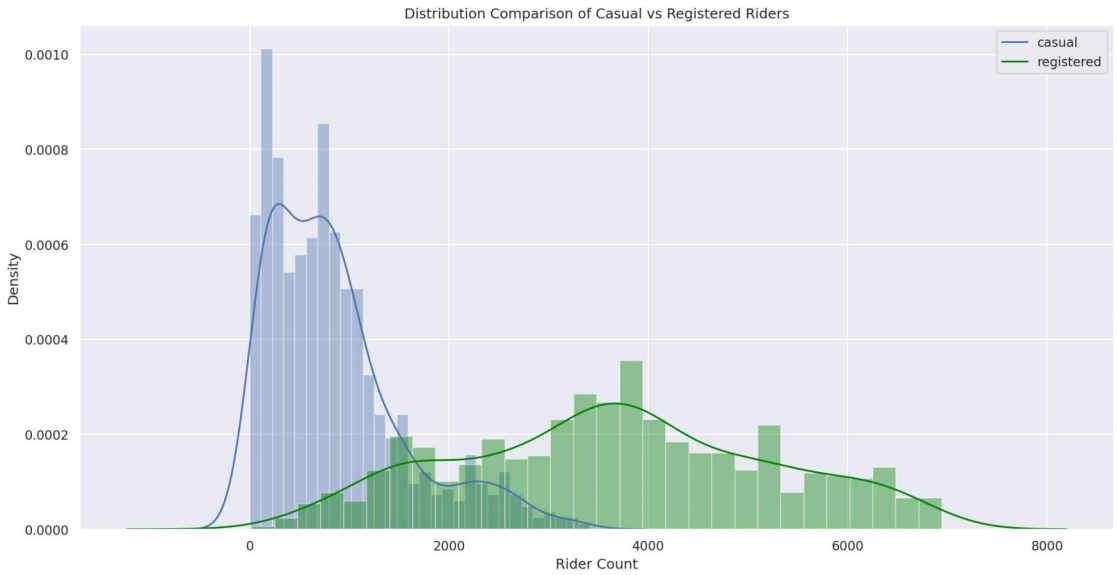
0.0.2 Question 2

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 [19]: sns.distplot(x= daily_counts['casual'], kde = True, bins = 30)
         sns.distplot(x= daily_counts['registered'], kde = True, bins = 30, color = 'green')

         plt.title('Distribution Comparison of Casual vs Registered Riders')
         plt.xlabel('Rider Count')
         plt.legend(['casual', 'registered'])
```

```
Out[19]: <matplotlib.legend.Legend at 0x7f1f2c3f0550>
```



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.

- The spread for registered riders are wider comparing to the casual riders.
- The modes of the casual rider is less than 1000, where the mode for resgistered rider is around 4000.
- The casual riders has a right skew (right tail), the registered use has more of a normal distribution shape.

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 the `sns.lmplot` call.
- You will need to call `plot.title` to add a title for the graph.

```
In [20]: x_min = bike['casual'].min()
         x_max = bike['casual'].max()

         y_min = bike['registered'].min()
         y_max = bike['registered'].max()
```

```
In [21]: # Make the font size a bit bigger

         sns.set(font_scale= 2)
         sns.lmplot(data= bike, x = 'casual', y = 'registered', ci = False, hue = 'workingday',height =

         plt.title('Comparison of Casual vs Registered Riders on working and non-working days')
         plt.xlabel('casual')
         plt.ylabel('registered')
```

```
Out[21]: Text(73.34408179012344, 0.5, 'registered')
```


Comparison of Casual vs Registered Riders on working and non-working days
registered
casual
workingday
no
yes
0
200
400
600
800
1000
1200
1400
0
50
100
150
200
250
300
350

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?

- When it's working day, the reg line is almost diagonal, so there're almost an equal amount of registered user and casual users on working day; when it's not a working day, there are generally more casual users than registered ones.
- overplotting happens since there're so many pointes overlapping and it's hard to see the trend.

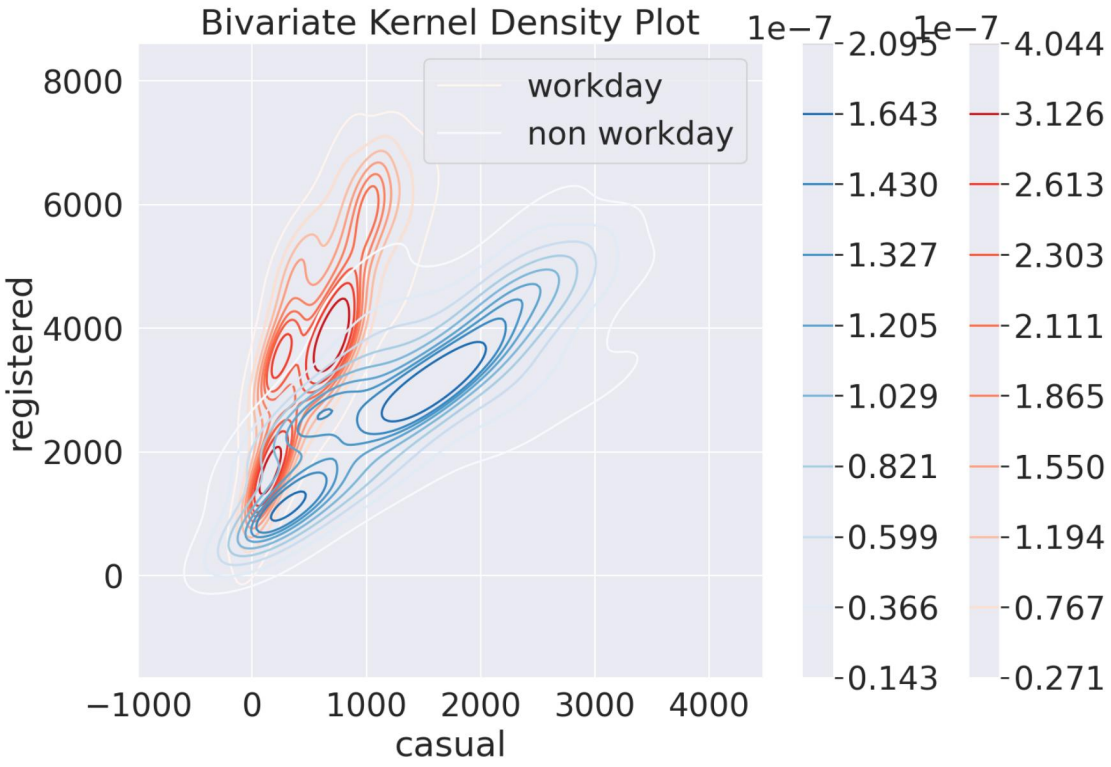
Generating the plot with weekend and weekday separated can be complicated so we will provide a walk-through below, feel free to use whatever method you wish if you do not want to follow the walkthrough.

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), and also set the `label` parameter to address which type of day you want to plot. 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 [23]:
```
# Set the figure size for the plot
plt.figure(figsize=(12,8))

# Set 'is_workingday' to a boolean array that is true for all working_days
is_workingday = daily_counts['workingday'] == 'yes'

# Bivariate KDEs require two data inputs.
# In this case, we will need the daily counts for casual and registered riders on workdays
# Hint: consider using the .loc method here.
casual_workday = daily_counts.loc[is_workingday,'casual']
registered_workday = daily_counts.loc[is_workingday,'registered']

# Use sns.kdeplot on the two variables above to plot the bivariate KDE for weekday rides
sns.kdeplot(data=daily_counts, x=casual_workday, y=registered_workday, cmap="Reds", cbar=True)

not_workingday = daily_counts['workingday'] == 'no'
# Repeat the same steps above but for rows corresponding to non-workingdays
# Hint: Again, consider using the .loc method here.
casual_non_workday = daily_counts.loc[not_workingday,'casual']
registered_non_workday = daily_counts.loc[not_workingday,'registered']

# Use sns.kdeplot on the two variables above to plot the bivariate KDE for non-workingday ride
sns.kdeplot(data=daily_counts, x=casual_non_workday, y=registered_non_workday, cmap="Blues", cb

plt.title('Bivariate Kernel Density Plot')
plt.legend(['workday', 'non workday'])
```

Out[23]:
```
<matplotlib.legend.Legend at 0x7f1f1c32c190>
```


Bivariate Kernel Density Plot
workday
non workday
registered
casual
8000
6000
4000
2000
0
−1000
0
1000
2000
3000
4000
1e−7
2.095
1.643
1.430
1.327
1.205
1.029
0.821
0.599
0.366
0.143
1e−7
4.044
3.126
2.613
2.303
2.111
1.865
1.550
1.194
0.767
0.271

Question 3bi In your own words, describe what the lines and the color shades of the lines signify about the data.

- the lower the probability density, the line is more approaching to white.
- The more coloured contours correspond to region which contains the higher probability mass.

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

- From the contour plot we can get more information about the densities distribution between the cause and registered users. Therefore form the graph, we can see on work day, there're more registered users comparing to the casual users, where on non-working days, the density distribution for both types of users are more evenly distributed.
- We can also see the spread/variance of the casual user is wider comparing to the registered users.

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).

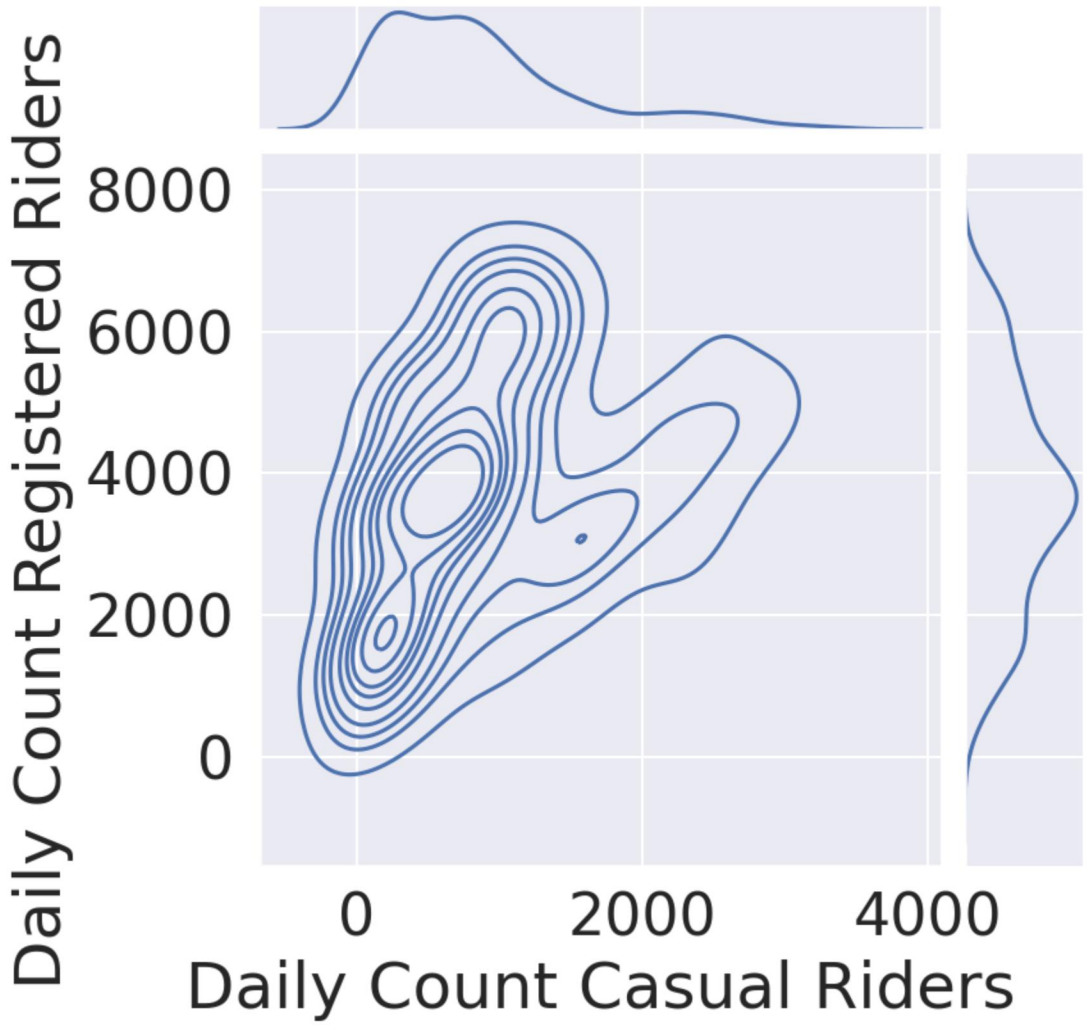
Hints: * The seaborn plotting tutorial has examples that may be helpful. * Take a look at `sns.jointplot` and its `kind` parameter. * `set_axis_labels` can be used to rename axes on the contour plot.

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 [24]:
```
...
plt.suptitle("KDE Contours of Casual vs Registered Rider Count")
plt.subplots_adjust(top=0.9);


ax = sns.jointplot(
    data= daily_counts,
    x="casual", y="registered",
    kind="kde",
)
ax.set_axis_labels('Daily Count Casual Riders', 'Daily Count Registered Riders')
```

Out[24]:
```
<seaborn.axisgrid.JointGrid at 0x7f1f1c24e340>
```

```
<Figure size 2400x1200 with 0 Axes>
```


Daily Count Registered Riders
8000
6000
4000
2000
0
0
2000
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 [25]: df1 = bike[['casual', 'registered', 'hr']]
         #df.head(30)
         #df.set_index('dteday')
         hourly_counts = df1.groupby('hr').agg({'casual': 'mean', 'registered': 'mean'})
         hourly_counts
```

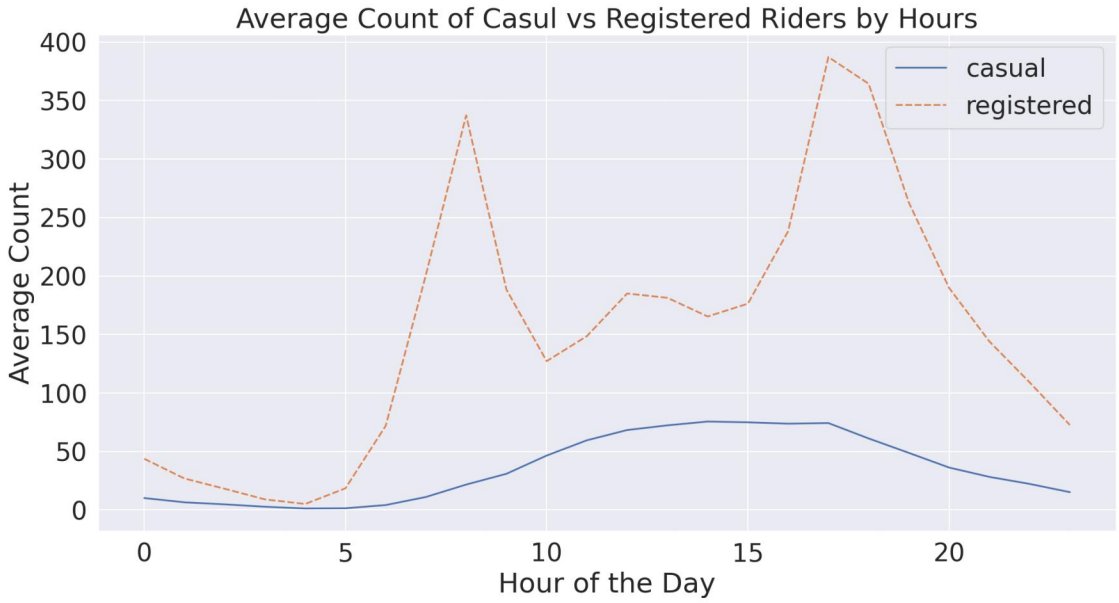
Out[25]:

hr	casual	registered
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
5	1.411437	18.478382
6	4.161379	71.882759
7	11.055021	201.009629
8	21.679505	337.331499
9	30.891334	188.418157
10	46.477304	127.191197
11	59.540578	148.602476
12	68.293956	185.021978
13	72.308642	181.352538
14	75.567901	165.381344
15	74.905350	176.327846
16	73.745205	238.238356
17	74.273973	387.178082
18	61.120879	364.390110
19	48.770604	262.752747
20	36.233516	189.796703
21	28.255495	144.059066
22	22.252747	109.082418
23	15.199176	72.631868

```
In [26]: sns.lineplot(data=hourly_counts, ci = False);
```

```
plt.title('Average Count of Casul vs Registered Riders by Hours')
plt.xlabel('Hour of the Day')
plt.ylabel('Average Count')
```

```
Out[26]: 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.

- Registered users has 2 noticable peaks, my assumption is these peaks uses are correlate to the working hours, when most people commute to work and back home.
- Casual users has a smoother curve, however the total use is much less than the registered users.

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 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 [53]: df_temp_f = bike[['weekday', 'prop_casual', 'temp_f']]
         df_temp_f
```

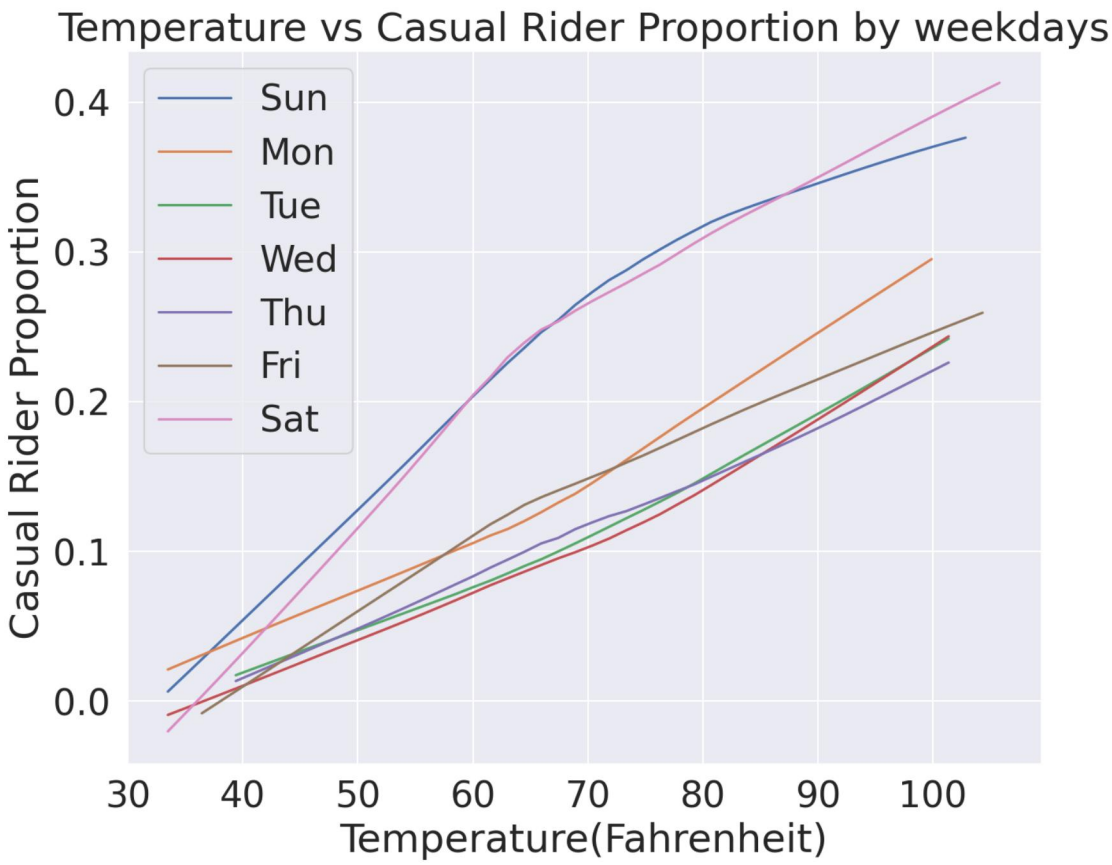
Out[53]:

	weekday	prop_casual	temp_f
0	Sat	0.187500	49.712
1	Sat	0.200000	48.236
2	Sat	0.156250	48.236
3	Sat	0.230769	49.712
4	Sat	0.000000	49.712
…	…	…	…
17374	Mon	0.092437	51.188
17375	Mon	0.089888	51.188
17376	Mon	0.077778	51.188
17377	Mon	0.213115	51.188
17378	Mon	0.244898	51.188

[17379 rows x 3 columns]

```
In [54]: from statsmodels.nonparametric.smoothers_lowess import lowess

         dow = ['Sun', 'Mon', 'Tue', "Wed", 'Thu', 'Fri', 'Sat']

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

```
for i in dow:
    xobs = df_temp_f[df_temp_f['weekday'] == str(i)]['temp_f']
    yobs = df_temp_f[df_temp_f['weekday'] == str(i)]['prop_casual']

    dow_smooth = lowess(yobs, xobs, return_sorted=False)
    sns.lineplot(xobs, dow_smooth)


plt.title('Temperature vs Casual Rider Proportion by weekdays')
plt.xlabel('Temperature(Fahrenheit)')
plt.ylabel('Casual Rider Proportion')
plt.legend(dow);
```



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, the hotter the weather, the higher the proportion of the casual riders.
- On weekend, the casual rider proportion is higher then the workdays.

0.2.2 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.

- I don't think so, because the only categories we have on users data is casual/registered users, which dose not provide other demographic information about the user, therefore cannot be used as the identifier to annalyse equity.
- We need to collect user data that consists basic user informations like age, race, genders, and other personalized infromation such as education level, active level, etc.
- To annalyze the geographic trend, we might want to collect the route for each usage. So that we get the total distance, and start/drop off location. Then we are able to classify long/short distance usage, in order to set up mobility score for each use.

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!

- From the temperature plots (supporting graph 1 below), for cities that has similar climates and temperature trend, we can predict the general daily usage based on the temperature. The trend can be use as a metrics to determine whether these cities will be a prefered choice for expantion.
- From the hourly usage of registered and casual users graph (supporting graph 2 below), we can deduce that most of the registered users are working commuters, they also composites most of the uses and are more consistent users (from the comparison graph, which shows normally distributed) (supporting graph 3), therefore if the city has a large population of working commuters, than the expansion would be preferable.
- From the demographic data we collect from the previous steps, we can have a rough demographic users data base, we can compare the composition of age tiers, racial constitution, active level, to predict whether a similar demographic trend exists in some other cities that can be expanding.