0.0.1 Question 0

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

Each row represents one hour of records in the bike sharing system in Washington D.C. through 2011 and 2012.

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?

One limitation of using this data is that it groups riders into the broad categories of "registered" and "casual". One problem with this is that this does not deal with the nature of these individuals' bike usage, e.g. registered riders may use this bikes for longer periods of time or for a greater distance because they actively use bikesharing as a form of commute. Meanwhile, casual riders may use these bikes for a shorter period of time and for shorter distances because it is merely an experience for them, something they are trying out. Eventually, if they enjoy their experience enough, they may convert to registered riders, which leads to overlap between the categories because this individual was earlier counted as a casual rider and is now counted as a registered rider.

Hence, additional data categories that would be useful is the distance travelled by registered riders vs casual riders. Additionally, keeping track of a Boolean i.e. did you used to be a casual rider? would allow us to use our data more efficiently.

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.**

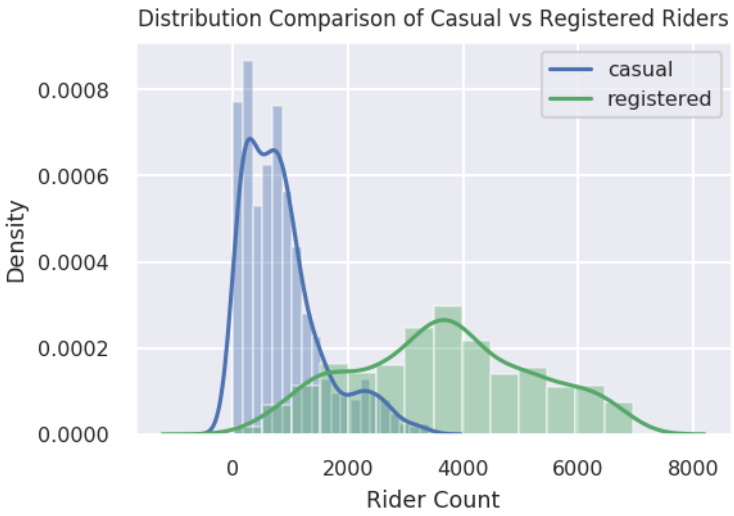
Include a legend, xlabel, ylabel, and title. Read the seaborn plotting tutorial if you're not sure how to add these. After creating the plot, look at it and make sure you understand what the plot is actually telling us, e.g on a given day, the most likely number of registered riders we expect is ~4000, but it could be anywhere from nearly 0 to 7000.

In [394]:
```
plt.figure(figsize=(4, 2.7))
sns.set(font_scale =0.7)

sns.distplot(daily_counts['casual'], color='b')
sns.distplot(daily_counts['registered'], color='g')
plt.legend(('casual', 'registered'), loc='upper right')
plt.title('Distribution Comparison of Casual vs Registered Riders', fontsize = 8)
plt.xlabel('Rider Count')
```

```
/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]
```

Out[394]:
```
Text(0.5, 0, 'Rider Count')
```


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

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 density curve for casual riders appears to have more modes than the density curve for registered riders. As a result, the casual density curve ends up looking more like a bimodal distribution and the registered density curve ends up looking more like a unimodal distribution. Furthermore, the mode of the casual density curve is higher than the mode of the registered density curve.

The casual density curve has a right tail, and is hence right-skewed. On the other hand, the registered density curve does not have a (visible) tail on either side and is hence symmetric, roughly about x=4000. The registered distribution has a greater spread as well.

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).

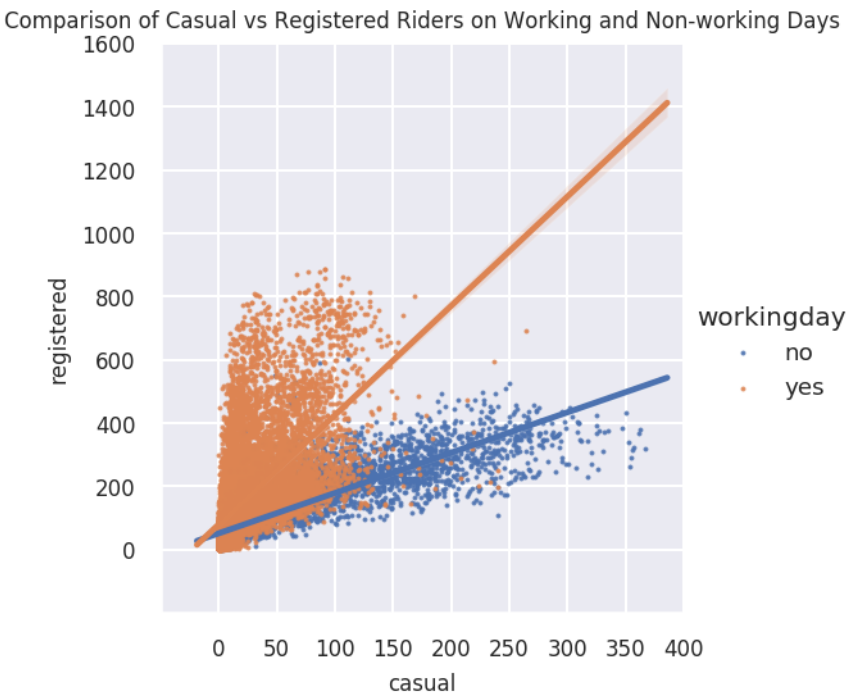
There are many points in the scatter plot, so make them small to help reduce overplotting. Also make sure to set `fit_reg=True` to generate the linear regression line. You can set the `height` parameter if you want to adjust the size of the `lmplot`.

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

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

In [403]:
```
# Make the font size a bit bigger
sns.set(font_scale=0.8)
sns.lmplot('casual','registered', bike, hue='workingday', fit_reg=True, scatter_kws={'s':0.8}
plt.xticks(np.arange(0, 430, 50), fontsize=8)
plt.yticks(np.arange(0, 1750, 200), fontsize=8)
plt.xlim(-50, 400)
plt.ylim(-200, 1600)
plt.xlabel('casual', fontsize=8)
plt.ylabel('registered', fontsize=8)
plt.title('Comparison of Casual vs Registered Riders on Working and Non-working Days', fontsi
plt.show()
```

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


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
1600
0
50
100
150
200
250
300
350
400

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 relationship between casual and registered riders on both weekends and weekdays. There is a stronger, more positive relationship between casual and registered riders on working days than on non-working days (line is steeper). Overplotting makes it difficult to see those points that correspond to both working days and non-working days because they are on top of eachother. As a result, it makes it difficult to describe the relationship at the points where the overlap occurs.

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). 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`.

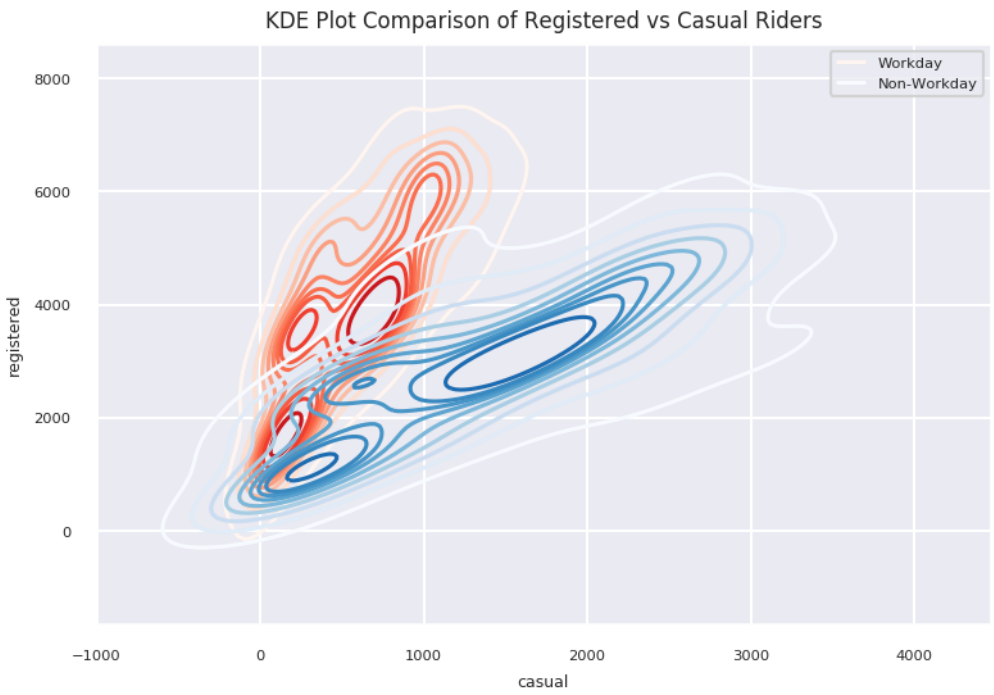
In [405]:
```
# Set 'is_workingday' to a boolean array that is true for all working_days
is_workingday = daily_counts['workingday'] == 'yes'

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

# Use sns.kdeplot on the two variables above to plot the bivariate KDE for weekday rides
plt.figure(figsize=(6, 4))
sns.set(font_scale=0.5)
sns.kdeplot(x=casual_workday, y=registered_workday, cmap='Reds', label='Workday', shade=False

# Repeat the same steps above but for rows corresponding to non-workingdays
casual_non_workday = daily_counts[~is_workingday]['casual']
registered_non_workday = daily_counts[~is_workingday]['registered']

# Use sns.kdeplot on the two variables above to plot the bivariate KDE for non-workingday rid
sns.kdeplot(x=casual_non_workday, y=registered_non_workday, cmap='Blues', label='Non-Workday'
plt.title('KDE Plot Comparison of Registered vs Casual Riders', fontsize=8);
plt.legend(['Workday', 'Non-Workday'])
plt.show()
```


KDE Plot Comparison of Registered vs Casual Riders
Workday
Non-Workday
registered
casual
8000
6000
4000
2000
0
−1000
0
1000
2000
3000
4000

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

We can see that there is a linear relationship between casual and registered riders on both working days, and non-working days (although the linear relationship is stronger on working days and than on non-working days). Although we could also gather this general relationship from the scatter plot, we could not firmly say that this was also true between x=0 to x=150 because there is too much overplotting to make a statement about what is going on with the 'blue' points. However, the linear relationship becomes more clear with the contour plot.

Areas with a high density of points are dark in color in the above plot. We see that the area of the highest density Non-Workday (i.e. blue) points occurs between x=1000 and x=2000. On the other hand, the area of the highest density Workday (i.e. red) points occurs between x=500 and x=1000. It seems that this highest density area is larger for Non-Workday points. We were not able to tell this in the scatter plot.

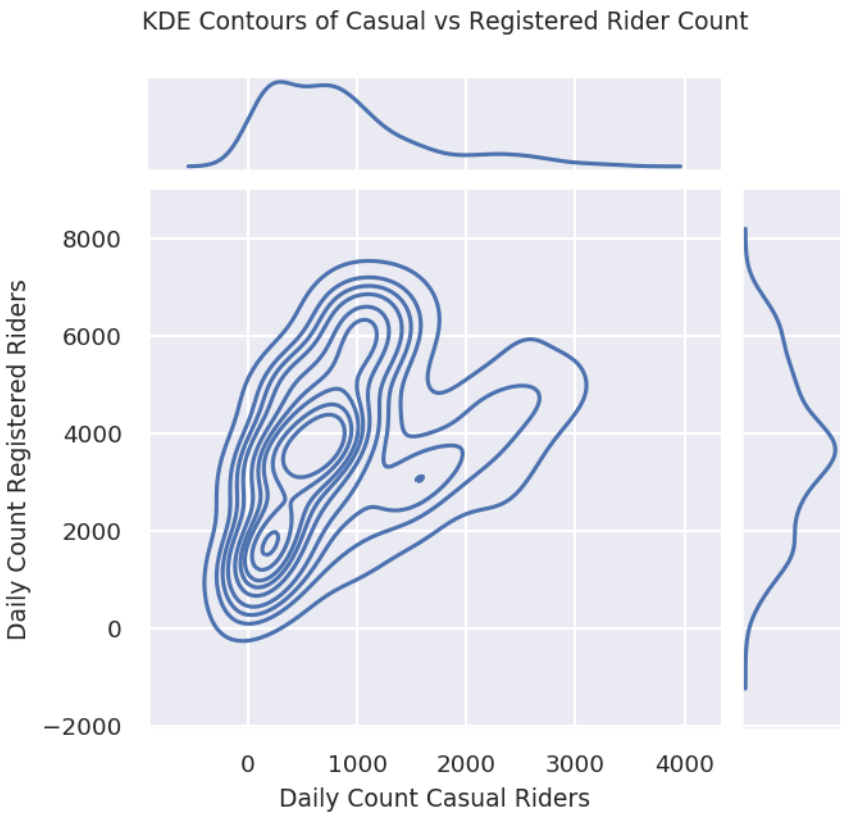
0.1 4: Joint Plot

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 [406]:
```
sns.set(font_scale=0.8)
counts_plot = sns.jointplot(x='casual', y='registered', data=daily_counts, kind='kde', height=
counts_plot.set_axis_labels('Daily Count Casual Riders', 'Daily Count Registered Riders', fon
plt.suptitle('KDE Contours of Casual vs Registered Rider Count', fontsize=9)
plt.subplots_adjust(top=0.9)
plt.show()
```

```
/opt/conda/lib/python3.8/site-packages/seaborn/distributions.py:1181: UserWarning: The following kwargs
  cset = contour_func(
/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 vs Registered Rider Count
Daily Count Registered Riders
8000
6000
4000
2000
0
−2000
0
1000
2000
3000
4000
Daily Count Casual Riders

0.2 5: Understanding Daily Patterns

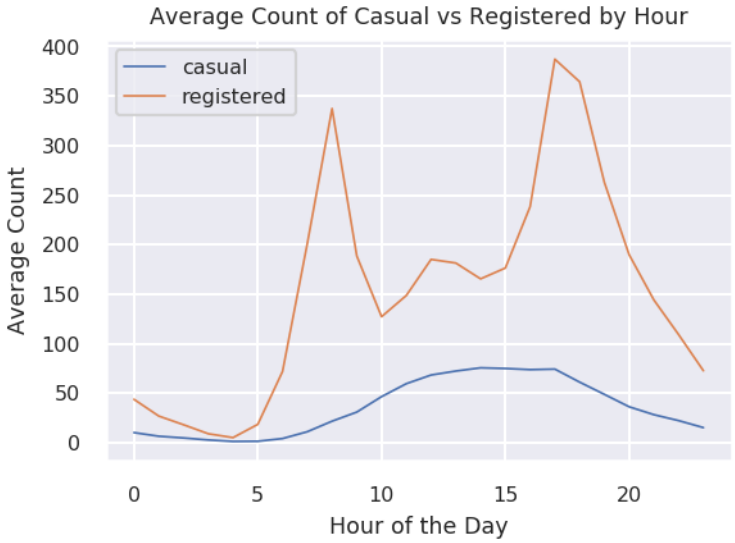
0.2.1 Question 5

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 [407]:
```
#group by hour and agg by avg (mean)
bikes_per_hour = bike.groupby('hr').agg('mean')
bikes_per_hour.reset_index()
#plot
sns.set(font_scale=0.7)
plt.figure(figsize=(4.2, 2.9))
sns.lineplot(x='hr', y='casual', data=bikes_per_hour, linewidth=0.8)
sns.lineplot(x='hr', y='registered', data=bikes_per_hour, linewidth=0.8)
plt.xlabel('Hour of the Day')
plt.ylabel('Average Count')
plt.title('Average Count of Casual vs Registered by Hour')
plt.legend(['casual', 'registered'])
```

Out[407]:
```
<matplotlib.legend.Legend at 0x7f01a0d766a0>
```



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

On the whole, the average count of registered riders is greater than the average count of casual riders at any time of the day. However, the maximum and minimum average for registered riders has a large difference which shows that their averages vary more. On the other hand, the maximum and minimum average for casual riders do not vary much by time of the day.

The peaks for the registered riders occur at roughly Hour 8 and Hour 17 of the day i.e 8am and 5pm. These hours correspond to the times most people generally leave for work and the times they come back. There is also a peak at Hour 12 i.e. 12pm which may correspond to their lunch break. However, this peak isn't as big as the other 2 peaks which means that not everyone leaves for lunch.

The peaks for the casual riders occurs at hour roughly Hour 15 of the day i.e. 3pm and this may be because riding a bike is merely a recreational activity for casual riders so they usually go at this time because this is usually the warmest time of the day, which makes it ideal for those wanting to make the most of summer.

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 $* \frac{9}{5} + 32$.

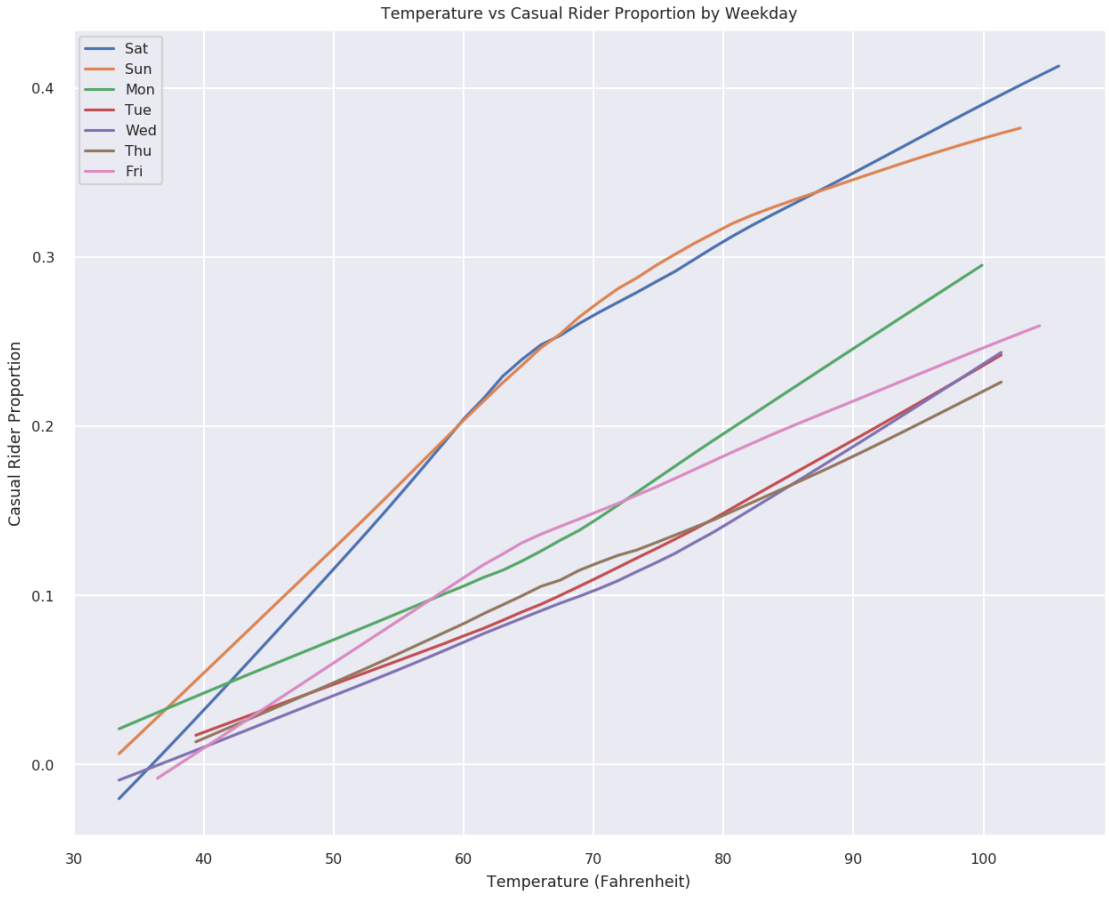
Note: If you prefer plotting temperatures in Celsius, that's fine as well!

```
In [413]: from statsmodels.nonparametric.smoothers_lowess import lowess

          plt.figure(figsize=(10,8))
          bike['temp_fahren']=(bike['temp']*41)*(9/5)+32
          days = ['Sat', 'Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri']
          for day in days:
              x_axis = bike[bike['weekday']==day]['temp_fahren']
              y_axis = bike[bike['weekday']==day]['prop_casual']
              ysmooth_axis = lowess(y_axis, x_axis, return_sorted=False)
              sns.lineplot(x_axis, ysmooth_axis, label=day)
          plt.xlabel('Temperature (Fahrenheit)')
          plt.ylabel('Casual Rider Proportion')
          plt.title('Temperature vs Casual Rider Proportion by Weekday')
          plt.show()
```

```
/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 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(
```



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?

We see a general positive relationship between Temperature and Casual Rider Proportion for every day of the week. For every day of the week, prop_casual increases with temperature and this makes sense logically because casual riders mostly only use bikesharing for recreational purposes and people tend to enjoy recreational activities more on warmer days to make the most of the good weather, particularly in states like Washington D.C. where winter months are cold and snowy.

On top of this, another interesting factor is that the highest Casual Rider Proportions are on the weekends, perhaps due to the same reason as above i.e. casual riders mostly only use bikesharing for recreational purposes and people tend to enjoy recreational activities more on weekends. On a sidenote, it is interesting to see that the lowest Casual Rider Proportions are on Wednesdays, perhaps when the mid-week work-load hits the hardest, making it harder for people to enjoy recreational activities.

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 think the bike data in its current form would not help me assess equity, because it tells us nothing about who is using the bikesharing services in Washington D.C. except "casual" and "registered" riders. Perhaps "registered" riders belong to lower socio-economic backgrounds because they bike to work as a regular form of commute (which may/may not be cheaper than driving to work) and perhaps "casual" riders belong to higher socio-economic backgrounds because they use this as a recreational activity because usually lower socio-economic classes do not have the time, opportunity and privilege to do so, but these are just speculations. To make a strong statement about equity in the Bike Sharing Company, we would need more information about the riders: the income bracket they belong to, whether they are male, female, or other, what race they belong to, and what neighborhood they reside in. This would allow us to truly guage whether the bikesharing services contain a good proportion of individuals from all the categories mentioned above, and if so, then we would be able to say that the bike sharing program is implemented equitably.

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!

My first recommendation would be to implement this in cities with warmer temperatures because my plot from 6b clearly indicates a positive relationship between temperature and the proportion of casual riders for every day of the week. While I understand that this relationship is between proportion of casual riders and temperature, not the total count of casual riders and temperature, it is important to note that the number of registered riders roughly stays constant because these people use it as a form of commute which is essential, so the count of these riders is less prone to fluctuation with a change in temperature. This then implies that the count of casual riders increases with temperature, so implementing this in warmer cities e.g. Los Angeles be a good idea.

My second, and more important recommendation would be to implement this in bigger cities because as seen from my plot in 5a, the average count of registered users is higher than the average count of casual users for every hour of the day. This means that most of the riders are those that commute to work, as interpreted in 5b. Bigger cities usually have large downtowns, where all the buildings are closer together, a "bikeable" distance away from eachother. Furthermore, they usually have lots of traffic so more people would prefer to bike to work rather than drive.

The ideal scenario would be to implement this in a large city with warm temperatures.