0.0.1 Question 0

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

Bike rentals and respective statistics (temperature, humidity, windspeed, number of casual and registered bikers) at a given hour.

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?

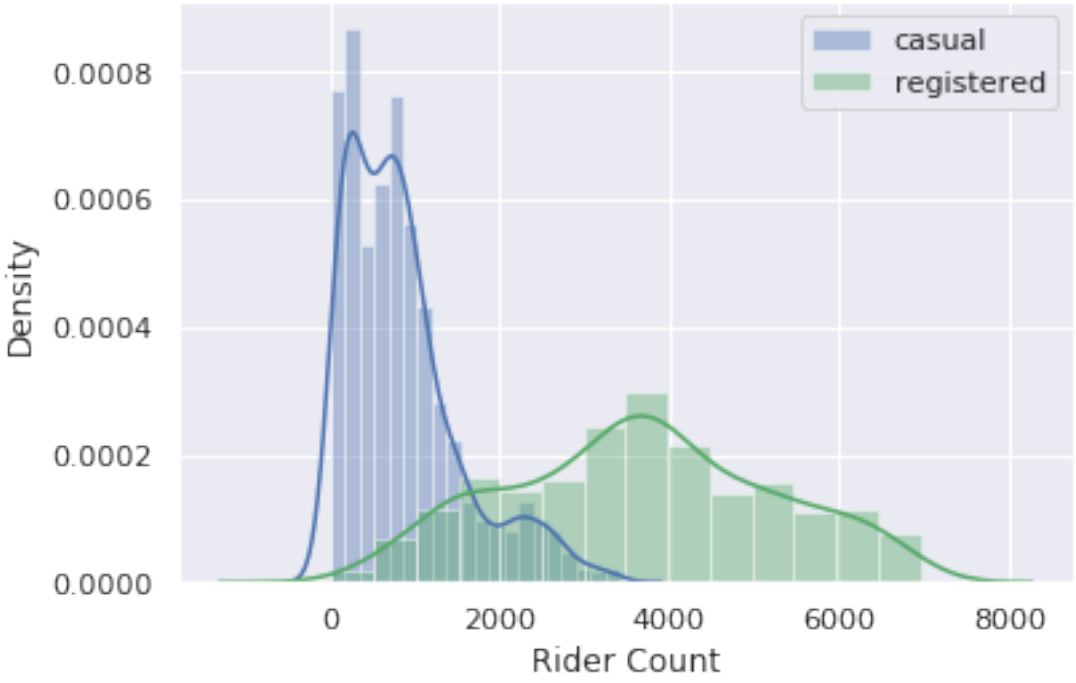
One limitation is that casual bikers can't be distinguishable so it's likely that they are overcounted. Another limitation may be that lack of information on location, thus it may be benefitial to add a variable on where the bikes are being rented and where they end up. Additionally, the numeric values for some of these columns may be hard to interpret, thus it may be benefitial to add a field translating these numbers into string values depending on what analysis is needed. Lastly, values like windspeed do not have units and other values like temp are divided by different numbers so it may be harder to understand.

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.

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 [17]: sns.distplot(daily_counts['casual'], color = 'b')
         sns.distplot(daily_counts['registered'], color = 'g')
         plt.xlabel('Rider Count')
         plt.ylabel('Density')
         plt.legend(['casual', 'registered'])
         plt.show()
```



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 center of the casual density curve is much higher than that of the registered. On the other hand, the registered curve is more spread (larger standard deviation) than that of the casual curve. The casual density curve is skewed to the right while the registered curve is more or less symmetrical. While it seems like the registered curve seems to be bimodal and the registered curve more unimodal, it's hard to distinguish the modes.

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).

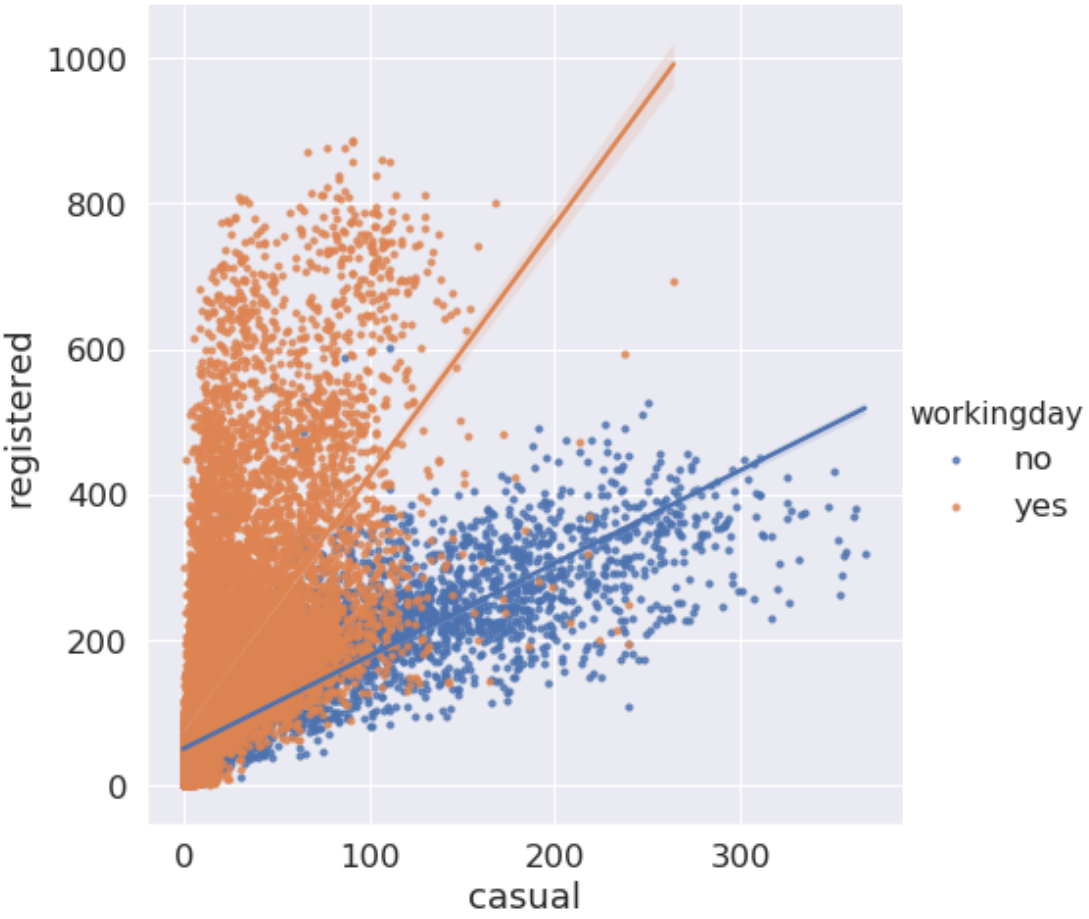
There are many points in the scatter plot, so make them small to help reduce overplotting. Also make sure to set `fit_reg=True` to generate the linear regression line. You can set the `height` parameter if you want to adjust the size of the `lmplot`.

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=1.5)
sns.lmplot(x = 'casual', y = 'registered', data = bike, size = 7, hue='workingday', scatter_kws

plt.show();
```

```
/srv/conda/envs/data100/lib/python3.7/site-packages/seaborn/regression.py:574: UserWarning: The `size` p
  warnings.warn(msg, UserWarning)
```


1000
800
600
400
200
0
registered
0
100
200
300
casual
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?

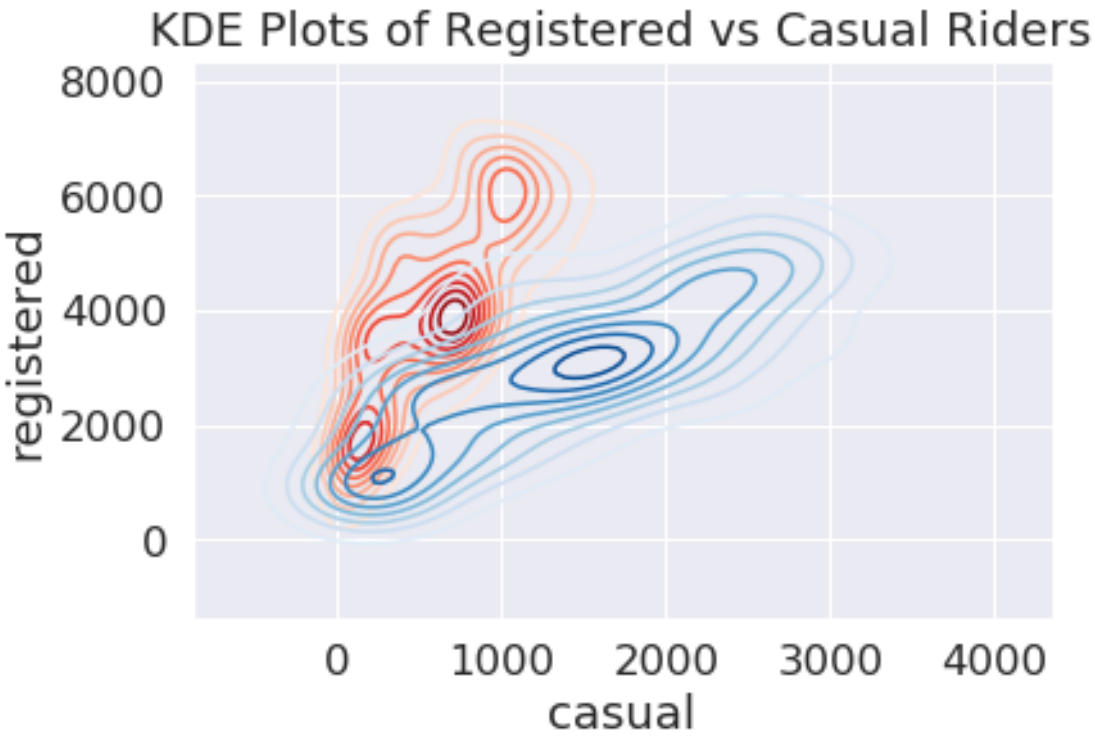
This scatterplot reveals that on workingdays, there are more casual bikers with respect to registered bikers and on nob workingdays there are more registered bikers with respect to casual bikers. There is a lot of overplotting in the bottom left hand corner of the graph and that makes it difficult to see the trend of non workingday points in that area (<100 casual).

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 however if you do not want to follow the walk-through.

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). 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]:
```
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[bike['workingday'] == 'yes'].set_index('dteday')

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

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

# Repeat the same steps above but for rows corresponding to non-workingdays
notworkingday = bike[bike['workingday'] == 'no'].set_index('dteday')
casual_non_workday = notworkingday['casual'].groupby('dteday').agg('sum')
registered_non_workday = notworkingday['registered'].groupby('dteday').agg('sum')
sns.kdeplot(casual_non_workday, registered_non_workday, cmap = 'Blues')

# Use sns.kdeplot on the two variables above to plot the bivariate KDE for non-workingday ride
plt.title('KDE Plots of Registered vs Casual Riders');
plt.show()
```


KDE Plots of Registered vs Casual Riders
8000
6000
4000
2000
0
registered
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?

We can see where most of the data points are concentrated on both plots and we can also see where the data points between the two plots overlap and how the data behaves in those areas (concentration of points at this overlap). It also makes the relationship between casual and registered bikers clearer on workindays/non-workingdays.

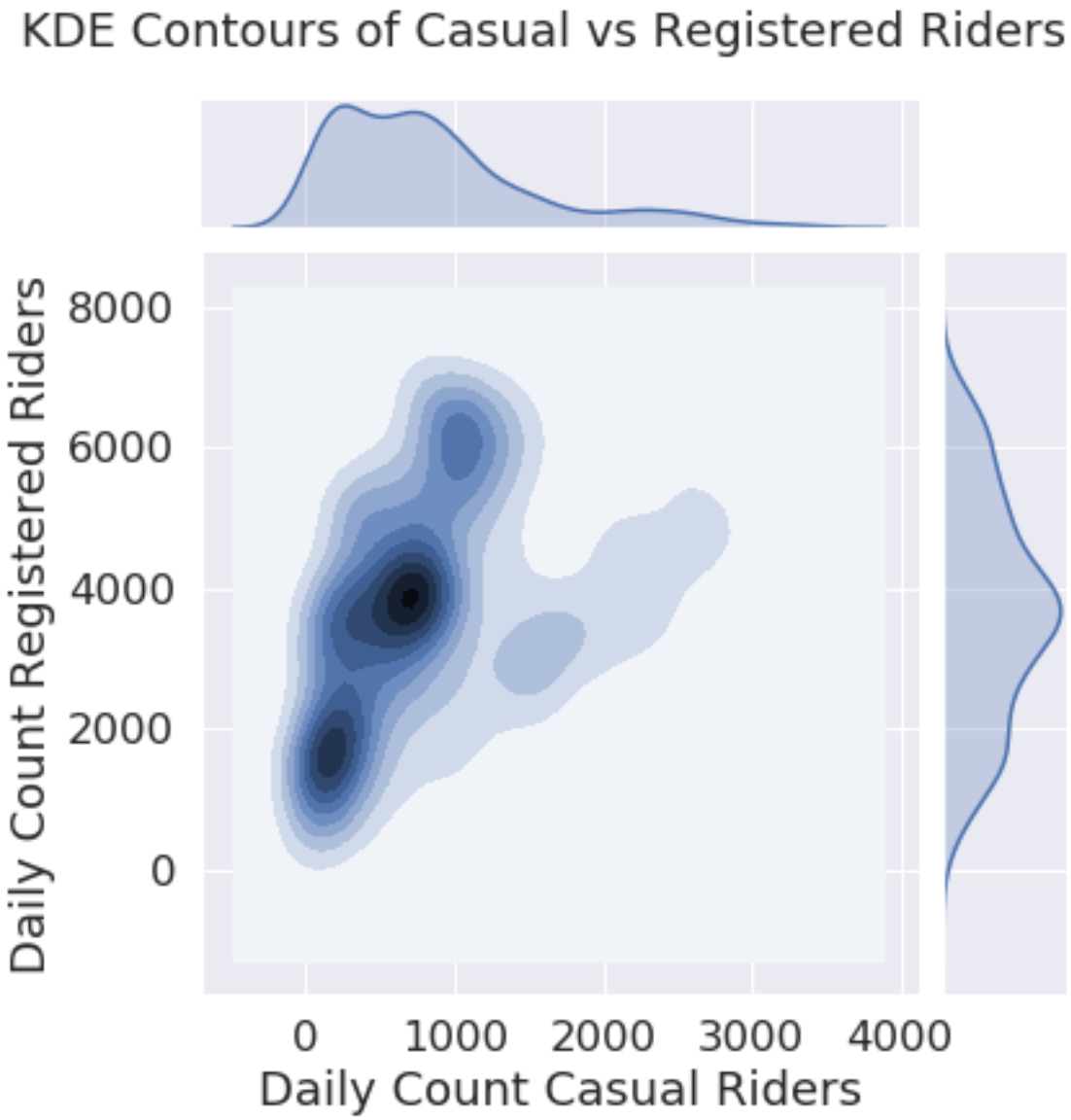
0.1 4: Joint Plot

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

We do not expect you to match our colors exactly, but the colors you choose should not distract from the information your plot conveys!

```
In [21]: sns.jointplot(x = 'casual', y = 'registered', data = daily_counts, kind = 'kde').set_axis_labe

         plt.title('KDE Contours of Casual vs Registered Riders', loc = 'right' , pad = 80)
         plt.show()
```


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

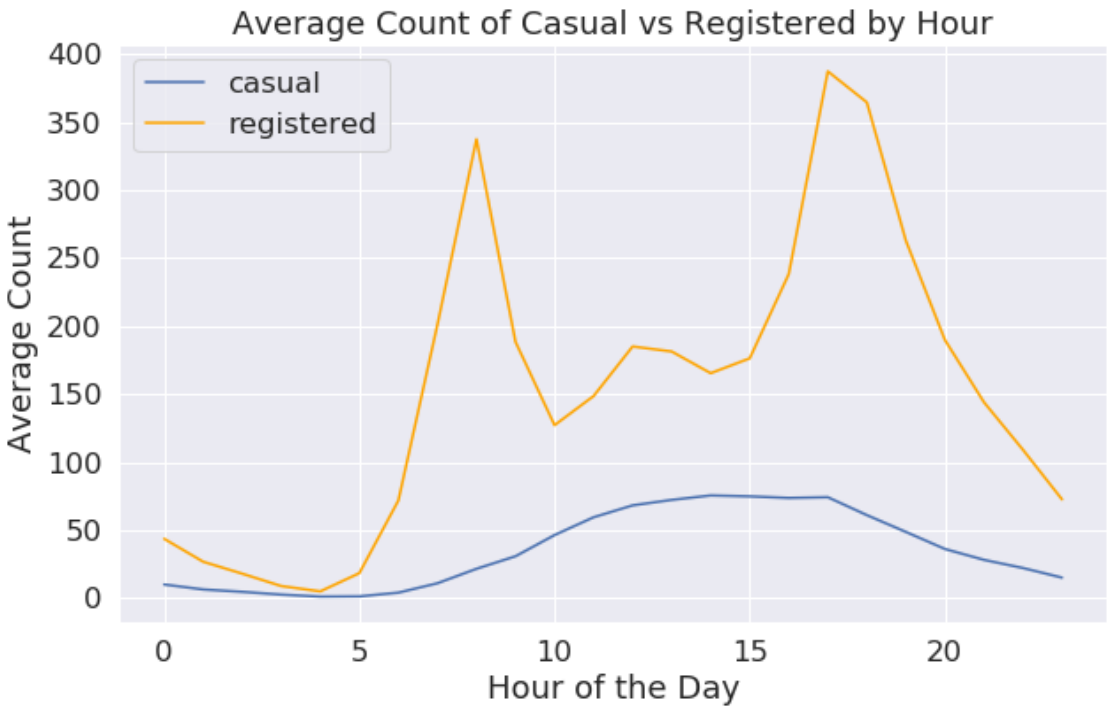
0.2 5: Understanding Daily Patterns

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]: mean = bike.groupby('hr').mean()[['casual', 'registered']]
         mean['hour'] = mean.index

         plt.rcParams['figure.figsize'] = (10,6)
         sns.lineplot(x='hour', y='casual',data= mean, color='b')
         sns.lineplot(x='hour', y='registered',data= mean, color='orange')

         plt.title('Average Count of Casual vs Registered by Hour')
         plt.xlabel("Hour of the Day")
         plt.ylabel("Average Count")
         plt.legend(['casual','registered'])
         plt.show()
```


Average Count of Casual vs Registered by Hour
casual
registered
Average Count
400
350
300
250
200
150
100
50
0
0
5
10
15
20
Hour of the Day

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

We can observe that overall, the average count of reistered bikers is higher than that of casual bikers and that's consistent throughout the day. There are several peaks on the curve of the registered bikers and I assume that's because of the usgae of bikes during certain hours. During rush hours, when people are trying to get to or from work, there should be a spike in usage. This is consitent with the graph, where we see peaks at around 8AM (when people are trying to go to work) and at 6PM (when people are leaving work). There is a drop in usage between or after those times because less people would be renting bikes at that 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.

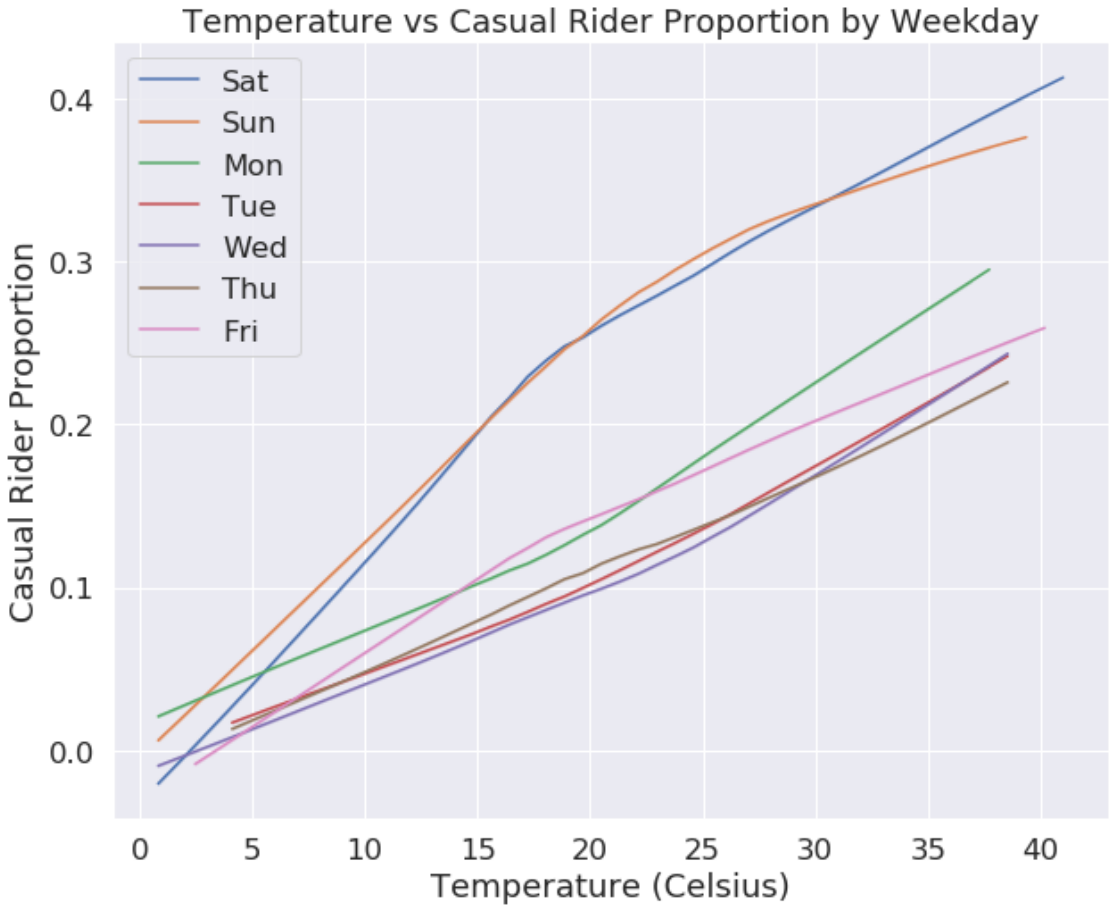
You do not need to match the colors on our sample plot as long as the colors in your plot make it easy to distinguish which day they represent.

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 i in ['Sat','Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri']:
    sm = bike[bike['weekday'] == i]
    sm = lowess(sm['prop_casual'].values, sm['temp'].values*41, return_sorted = False)
    sns.lineplot(bike[bike['weekday'] == i]['temp']*41, sm, label = i)
plt.title("Temperature vs Casual Rider Proportion by Weekday")
plt.xlabel('Temperature (Celsius)')
plt.ylabel('Casual Rider Proportion')
plt.show()
```


Temperature vs Casual Rider Proportion by Weekday
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Casual Rider Proportion
0.0
0.1
0.2
0.3
0.4
0
5
10
15
20
25
30
35
40
Temperature (Celsius)

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?

I see that for every day, as the temperature increases, so does prop_casual (or the casual rider proportion). Another interesting thing I notice is the huge gap between the weekend (saturday and sunday) curves and the weekday curves. The weekend curves have a much higher proportion of casual riders compared to weekdays.

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 do not think the current bike dataset has information that can help assess equity. Assessing equity means looking at certain classes, genders, races, and neighborhoods to see if there is equal access in all areas. For this reason, it would be benefitial to include these demographics in the dataset. This dataset should include renter information (race, gender, socio-economic class) to assess differences in rental demographics (do certain groups rent more than others?), as well as rental location to see if bike rental services are equally as available in poorer neighborhoods (maybe even more so than affluent neighborhoods). In this case, the granularity of the data set would be the individual renting to include their demographics as well as the location they rented from. It would also be benefitial to survey certain neighborhoods to see, if there is low usage in certain areas, why certain people don't use bike rental services (cost, other modes of transportation, inconvinience, etc.) and include this in the dataset as well.

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 would recommend expanding to cities that are typically warmer and also have a high working population (metropolitan areas). I make the recommendation on warmer cities because as we can see from the "Temperature vs Casual Rider Proportion by Weekday" graph, that as temperature increases, so do the proportion of casual riders. Warmer cities will have individuals more willing to go outside to bike while colder cities will hinder individuals from going outdoors to bike. I make the second recommendation (high working population) because the "Average Count of Casual vs Registered by Hour" graph shows us that the highest registered bike usage is during typical "rush hours," implying that working individuals make up the majority of registered users. Expanding to metropolitan areas will definitely be benefitial because these areas are usually congested and having bike rentals would ensure that the working population has another way of commuting without having to deal with driving through traffic, which in turn also decreases congestion.