0.0.1 Question 0

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

The granularity of the data is instances of bike rentals for every hour from year 2011-2012. Each row contains some information about the date, season, year, count etc. of bike rental/shares.

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 granularity and the variables present in the data might raise some limitations if we wanted to find out information and trends regarding the distance the bike traveled. Two additional data variables that could address this limitation is the average distance the bikes traveled and the average time the bikes were used.

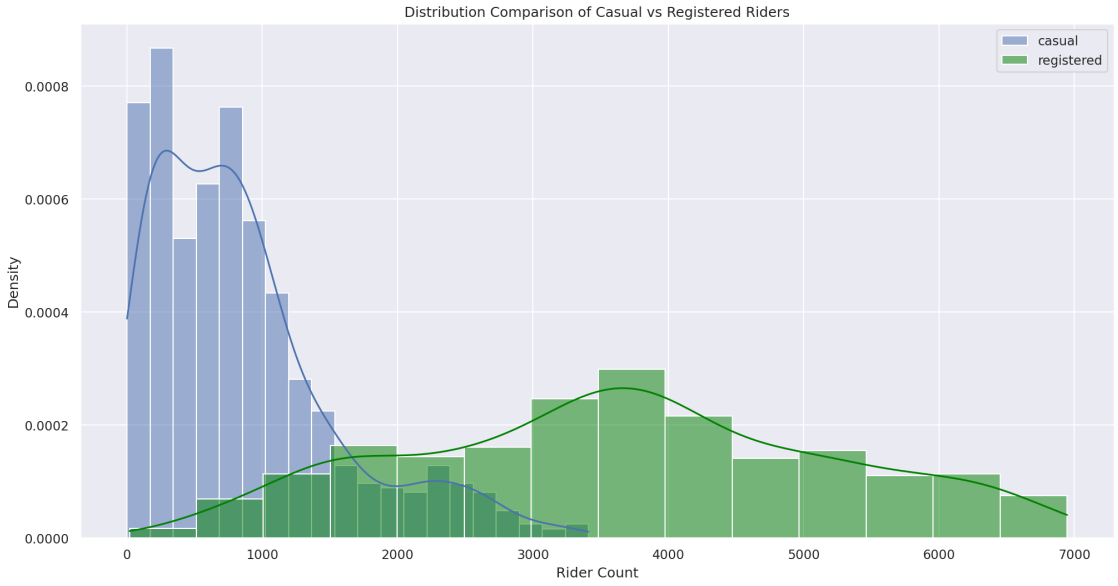
0.0.2 Question 2

Question 2a Use the `sns.histplot` 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.

Hint: You will need to set the `stat` parameter appropriately to match the desired plot.

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 [42]: sns.histplot(data = daily_counts, label='casual', x = 'casual', kde = True, stat = 'density')
         sns.histplot(data = daily_counts, x = 'registered', label='registered', kde = True, stat = 'den
         plt.legend()
         plt.title('Distribution Comparison of Casual vs Registered Riders')
         plt.xlabel('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 plot shows that for casual riders, the graph shows a right tail and is skewed to the right. There are no gaps nor any big outliers for casual riders. The mode for casual riders is the second bar from the left and appears on the very left side of the graph. For registered riders, the mode appears in the middle at around 3500-4000, and the graph is pretty symmetrical. There are no gaps, and doesn't seem to have an outlier.

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.

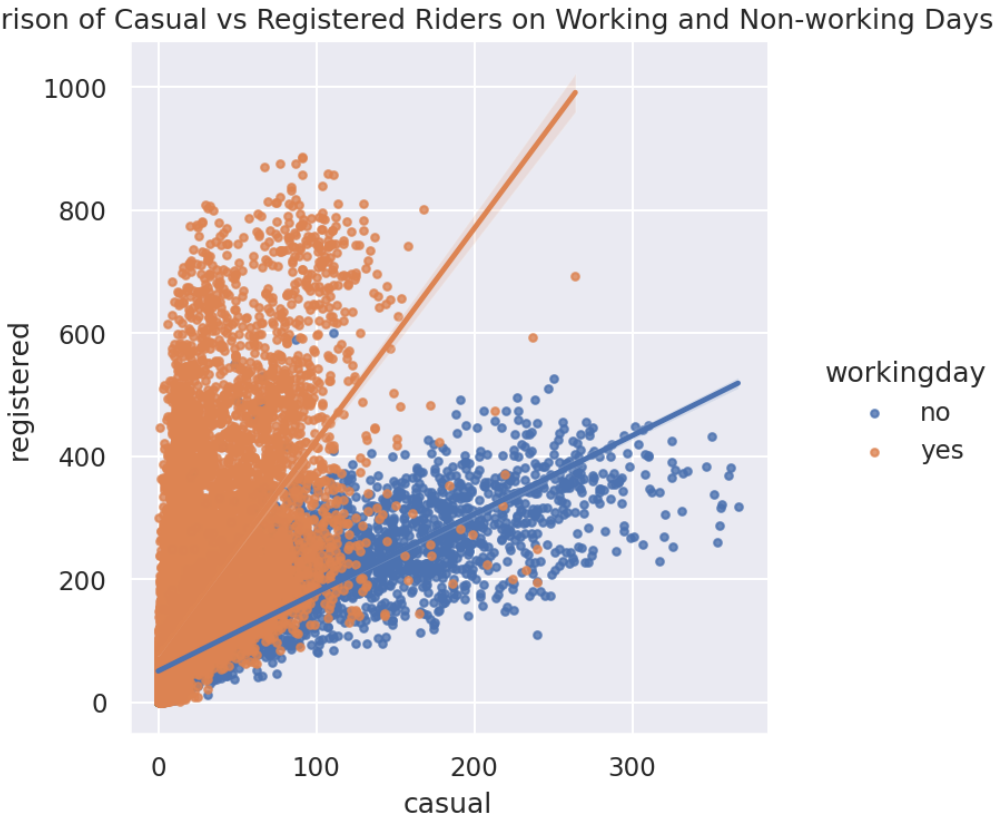
The `lmplot` function will also try to draw a linear regression line (just as you saw in Data 8). Color the points in the scatterplot according to whether or not the day is a working day (your colors do not have to match ours exactly, but they should be different based on whether the day is a working 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 `plt.title` to add a title for the graph.

```
In [57]: # Make the font size a bit bigger
         sns.set(font_scale=1)
         sns.lmplot(data = bike, x = 'casual', y = 'registered', hue = 'workingday', fit_reg = True, sca
         plt.title('Comparison of Casual vs Registered Riders on Working and Non-working Days');
```


Comparison of Casual vs Registered Riders on Working and Non-working Days
registered
casual
workingday
no
yes
0
200
400
600
800
1000
0
100
200
300

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?

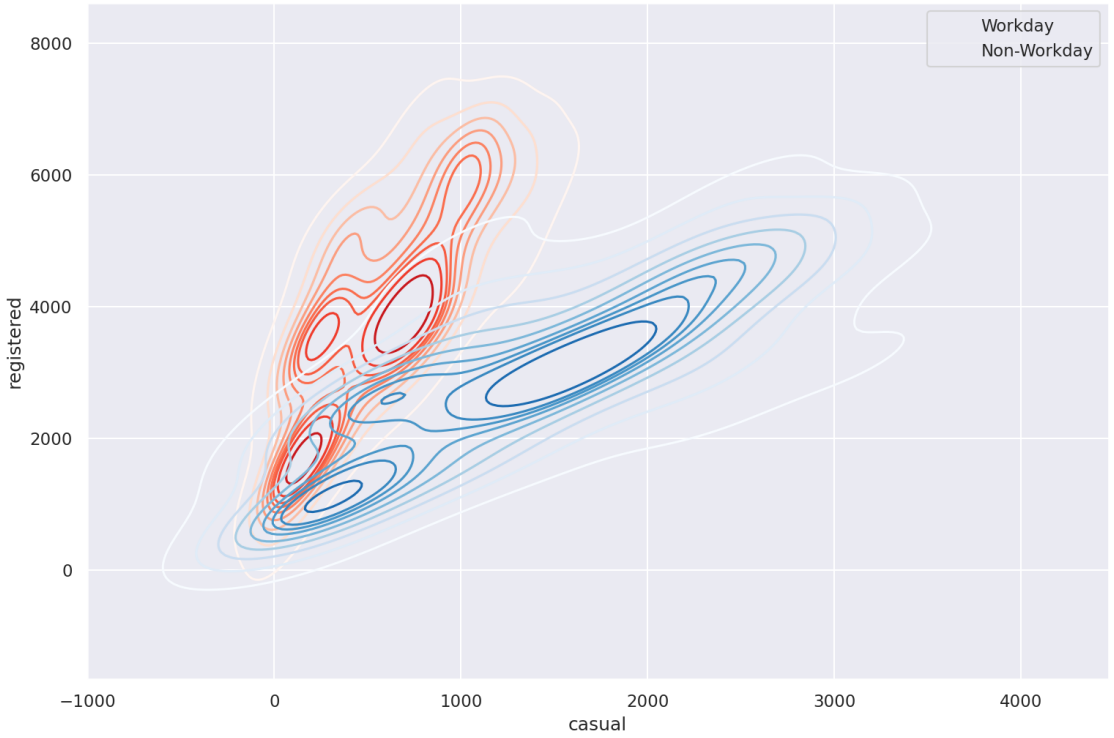
The scatterplot reveals linear relationships between casual riders and registered riders for both working days and weekends. The overplotting makes it hard or impossible to discern data where there are overlaps, which could cause the viewers to make decisions or conclusions without having the full knowledge.

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, each time drawing different data from the `daily_counts` table. * 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 [79]:
```
# 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(casual_workday, registered_workday, cmap = 'Reds', label = 'Workday')

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 = 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(casual_non_workday, registered_non_workday, cmap = 'Blues', label = 'Non-Workday')

plt.legend();
```


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

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

The lines signify that there are linear relationships between casual and registered bike riders in both weekdays and weekends. The color shades of the lines are the density of the data in that region, which signifies that the data suggests that both casual and registered riders ride on workdays, but much more registered bikers ride during workdays than casual riders do.

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

I can identify the density of overlapping regions for both weekdays and non-weekdays, which was difficult to determine from the scatter plot.

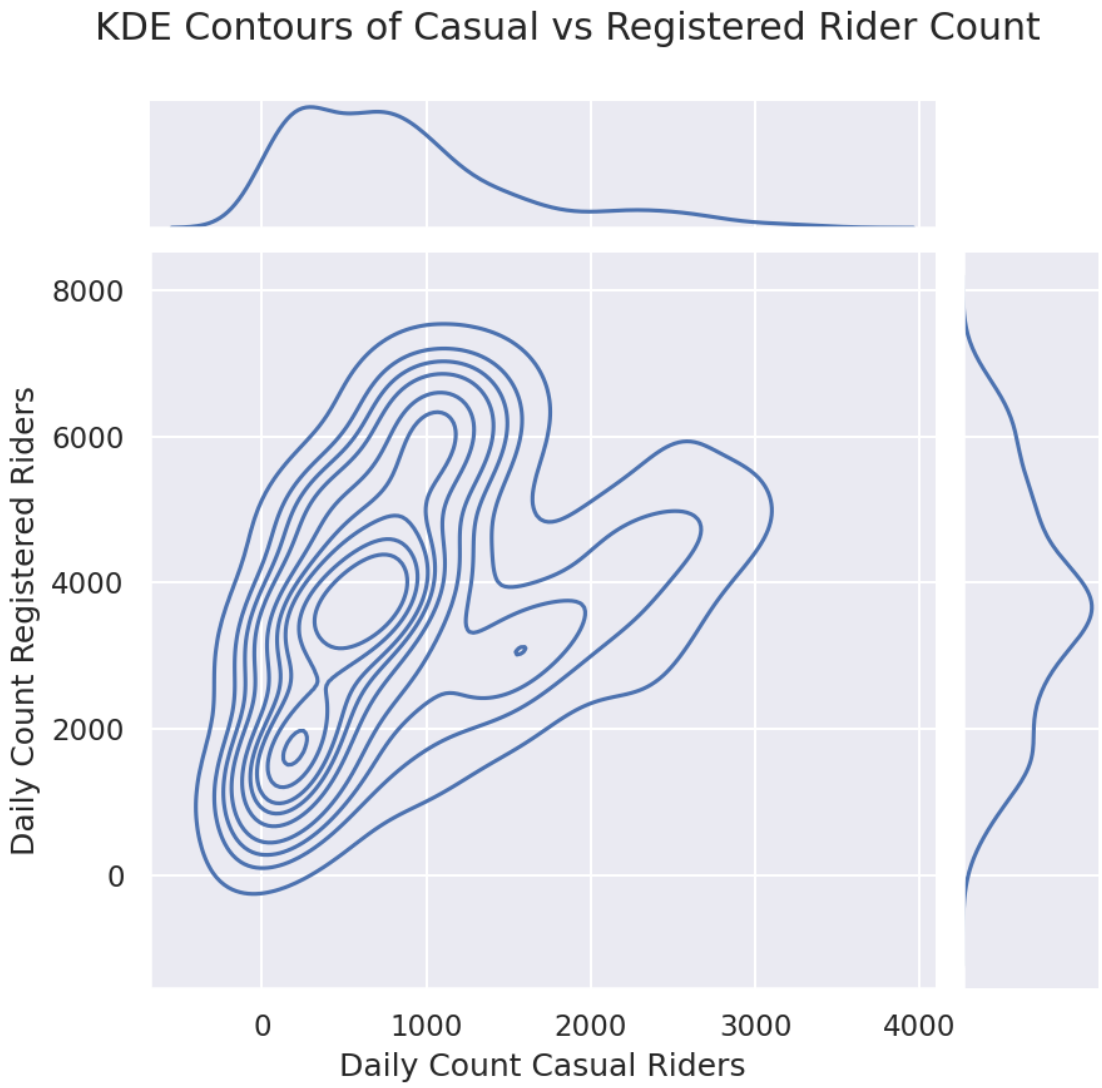
0.1 4: Joint Plot

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.

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 [83]: joint = sns.jointplot(data = daily_counts, x = 'casual', y = 'registered', kind = 'kde')
         joint.set_axis_labels('Daily Count Casual Riders', 'Daily Count Registered Riders')
         plt.suptitle("KDE Contours of Casual vs Registered Rider Count")
         plt.subplots_adjust(top=0.9);
```


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

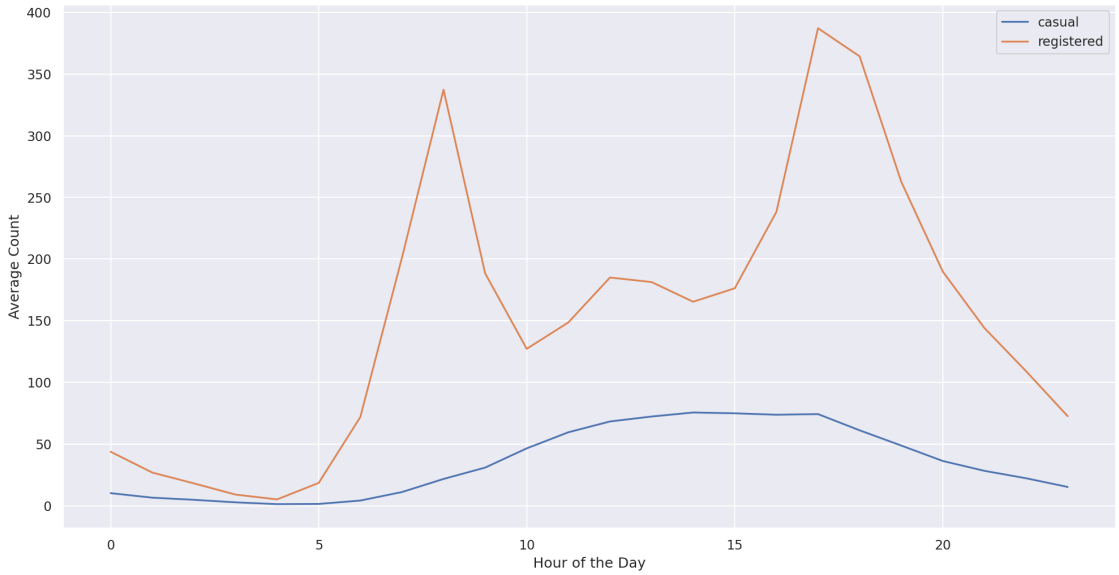
0.2 5: Understanding Daily Patterns

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 [92]: average_hours = bike.groupby('hr').mean()
         sns.lineplot(data = average_hours, x = 'hr', y = 'casual', label='casual')
         sns.lineplot(data = average_hours, x = 'hr', y = 'registered', label='registered')
         plt.xlabel('Hour of the Day')
         plt.ylabel('Average Count')
         plt.legend();
```



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

The plot shows peaks for registered riders around hours 8,17, and 18. We can point out that these are the times where most work start (8am) and end (5,6pm), and therefore we can hypothesize that many registered bikers ride their bike to work. For both types of riders, there is a very low average count from hours 0 to 5, which we can hypothesize that neither types of riders ride late at night or really early in the morning, where there is no sun in the sky.

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.

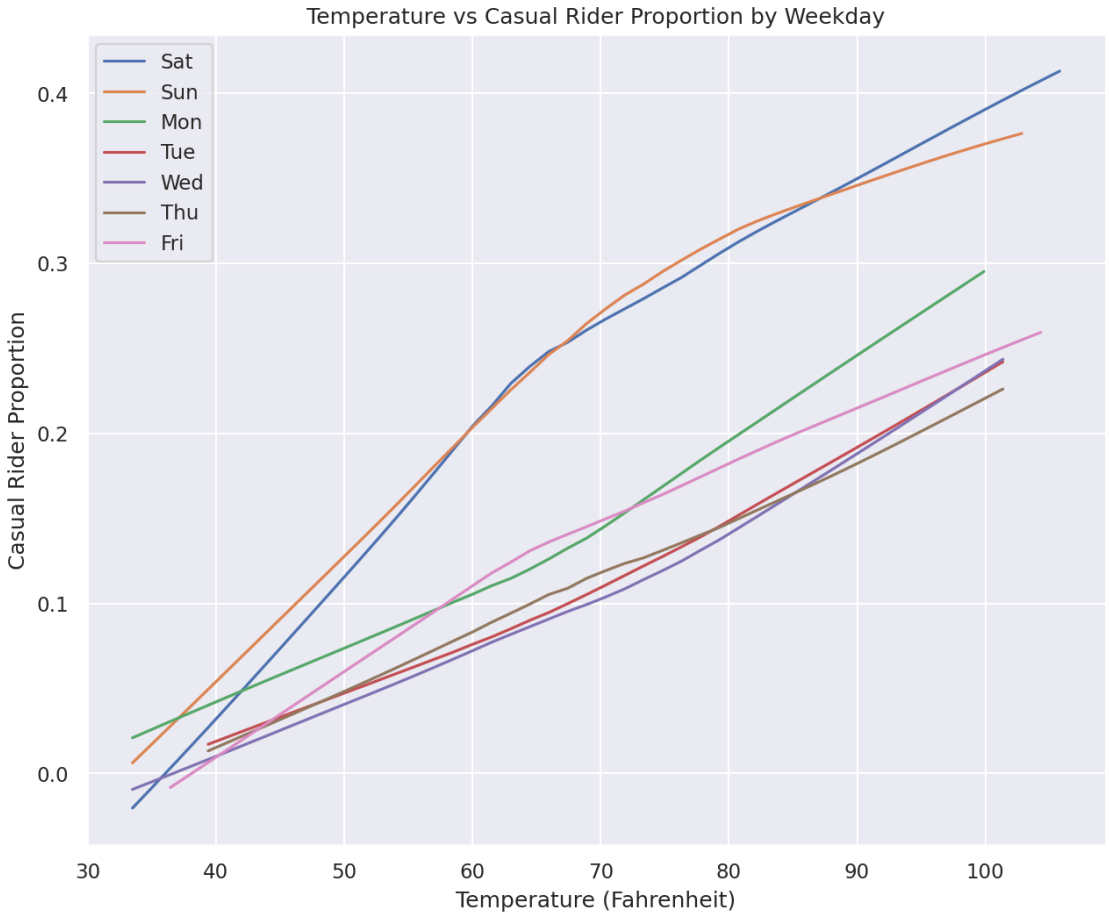
You do not need to match the colors on our sample plot as long as the colors in your plot make it easy to distinguish which day they represent.

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 [106]:
```
from statsmodels.nonparametric.smoothers_lowess import lowess

plt.figure(figsize=(10,8))
for i in bike['weekday'].unique():
    unique_day = bike[bike['weekday'] == i].copy()
    unique_day['fahrenheit'] = unique_day['temp'] * (9 / 5) * 41 + 32
    lowess_draw = lowess(unique_day['prop_casual'], unique_day['fahrenheit'], return_sorted =
    sns.lineplot(unique_day['fahrenheit'], lowess_draw, label = i)

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


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
30
40
50
60
70
80
90
100
Temperature (Fahrenheit)

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?

From the curve plot, I can see that proportion of casual riders increases as the temperature rises. I also see that the proportion of casual riders is higher on the weekends 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.

The bike data as it is can only do so much to help me assess equity. We only know whether the rider is a registered biker or a casual biker from the data. In order to assess equity, we would need more information about either the riders, regions that the bikes are being used, or both. We want to assess socio-economic classes, but we can't ask the bikers to input their social class - that would be weird and awkward, so we can use the regions that bikes are being used in order to make use of patterns and types of regions. I would also like to know what race and gender the riders are, but it would be hard to obtain those information.

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!

Yes, I would expand bike sharing to additional cities across the U.S. For example, we can see that from the lowess plot, that casual riders tend to ride bikes in hotter days. So we can use this information to show that there will be benefits in hotter areas. Also, we can use the kernel plot to infer that populated cities have more people riding bikes. There were many information that could be inferred to make decisions in additional cities in the U.S.