0.0.1 Question 1a

Granularity refers to the level of detail in a dataset—what each row represents in terms of time, space, or entity. In this dataset, each row corresponds to **bike-sharing data per hour** in Washington, DC. Based on the granularity and the variables present in the data, what might be some of the limitations of using this data?

What are two additional data categories/variables that one could collect to address some of these limitations?

The dataset provides hourly bike-sharing data, meaning each row represents the total rentals in a given hour. While this granularity allows for temporal trend analysis, it has key limitations: 1. No Trip Origins/Destinations: The dataset lacks location data, making it difficult to analyze popular routes, station imbalances, and commuting patterns. 2. No User Demographics: Without data on age, gender, or income, we cannot assess who is using the service or whether access is equitable.

To address these gaps, the dataset could include: Trip Level Geospatial Data (start and end station locations) for spatial analysis. User Profile Data (optional demographic indicators) to understand ridership patterns across different populations.

These additions would enhance insights into who, where, and why people use bike-sharing, informing urban planning and equity-focused policies

---

0.0.2 Question 3a

Use the `sns.histplot`(documentation) function to create a plot that overlays the distribution of the daily counts of bike users.

- Use blue to represent `casual` riders, and red to represent `registered` riders.

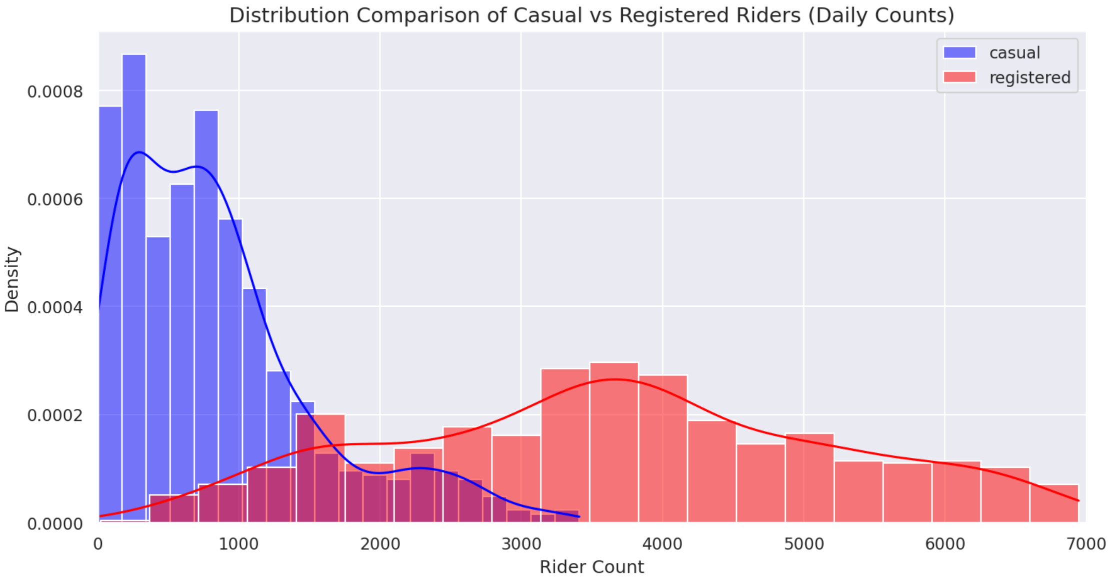
The temporal granularity of the records should be daily counts, which you should have after completing question 2c. In other words, you should be using `daily_counts` to answer this question.

Hints: - You will need to set the `stat` parameter appropriately to match the desired plot. - The `label` parameter of `sns.histplot` allows you to specify, as a string, how the plot should be labeled in the legend. Although label is not explicitly documented in Seaborn, it works because `sns.histplot` internally relies on `matplotlib`, which supports the label parameter. For example, passing in `label="My data"` would give your plot the label `"My data"` in the legend. - You will need to make two calls to `sns.histplot`.

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.

For all visualizations in Data 100, our grading team will evaluate your plot based on its similarity to the provided example. While your plot does not need to be *identical* to the example shown, we do expect it to capture its main features, such as the **general shape of the distribution**, the **axis labels**, the **legend**, and the **title**. It is okay if your plot contains small stylistic differences, such as differences in color, line weight, font, or size/scale.

```
In [47]: # Aggregate the data to daily counts
         daily_counts = bike.groupby("dteday").agg({
             "casual": "sum",
             "registered": "sum"
         }).reset_index()

         # Set figure size
         plt.figure(figsize=(12, 6))

         # Create histograms for casual and registered riders (daily basis) with density scaling
         sns.histplot(
             data=daily_counts,
             x="casual",
             bins=20,
             kde=True,
             color="blue",
    label="casual",
    alpha=0.5,
    stat="density"  # Normalize to density instead of frequency
)

sns.histplot(
    data=daily_counts,
    x="registered",
    bins=20,
    kde=True,
    color="red",
    label="registered",
    alpha=0.5,
    stat="density"  # Normalize to density instead of frequency
)

# Adjust x-axis range
plt.xlim([0, 7000])

# Title and labels
plt.title("Distribution Comparison of Casual vs Registered Riders (Daily Counts)", fontsize=14
plt.xlabel("Rider Count", fontsize=12)
plt.ylabel("Density", fontsize=12)  # Correct y-axis label

# Move legend inside the plot
plt.legend(loc="upper right")

# Show the plot
plt.show()
```



0.0.3 Question 3b

In the cell below, describe 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 casual riders' density curve is unimodal, with a peak between 200 and 1000 daily riders. The distribution is right-skewed, meaning that most days have relatively low casual ridership, but there are some days with significantly higher usage. The long right tail extends beyond 3000 riders, indicating that on rare occasions, casual ridership spikes significantly. The spread is narrower compared to registered riders, showing that casual bike usage is more sporadic and event-driven.

The registered riders' density curve is also unimodal, with a peak around 3500–4000 daily riders. Unlike casual riders, the registered riders' distribution is more symmetric, meaning that daily counts tend to cluster around the peak without extreme fluctuations. The right tail extends toward 7000 riders, but the density declines gradually, suggesting that high-demand days occur with relative frequency. The wider spread of registered riders' counts suggests that they have more stable and predictable usage patterns compared to casual riders.

In terms of gaps and outliers, there are no significant gaps in either distribution, meaning both groups consistently rent bikes throughout the dataset. However, casual riders show greater fluctuations, with more extreme low and high counts, indicating that their ridership is more influenced by external factors such as weather, holidays, or special events. Registered riders, on the other hand, maintain a steadier daily rental pattern with fewer outliers.._

0.0.4 Question 3c

The density plots do not show us how the counts for `registered` and `casual` riders vary together.

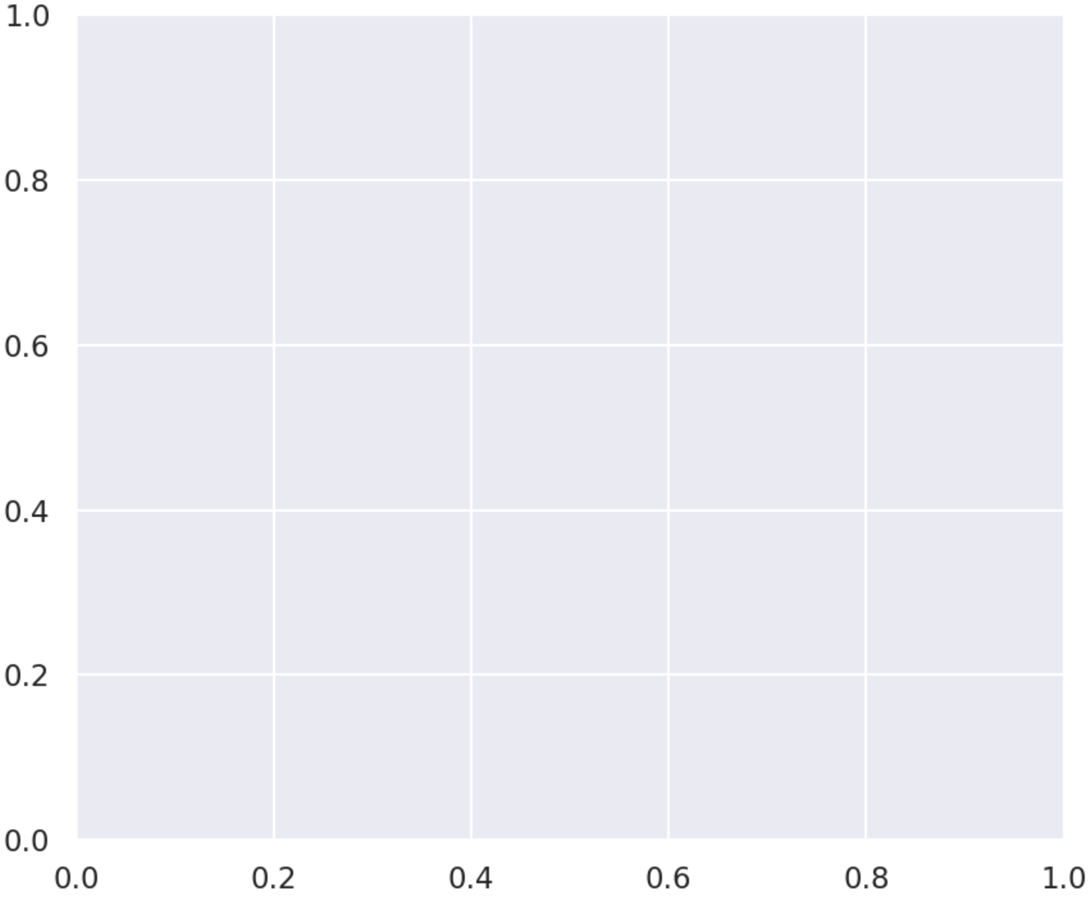
Use `sns.lmplot` (documentation) to create a scatter plot to investigate the relationship between casual and registered counts.

- Use the `bike` `DataFrame` to plot hourly counts instead of daily counts.
- Color the points in the scatter plot 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.

Hints: * Check out this helpful tutorial on `lmplot`. * There are many points in the scatter plot, so make them small to help reduce overplotting. Check out the `scatter_kws` parameter of `lmplot`. * You can set the `height` parameter if you want to adjust the size of the `lmplot`. * Add a descriptive title and axis labels for your plot. * It is okay if the scales of your `x` and `y` axis (i.e., the numbers labeled on the two axes) are different from those used in the provided example.

```
In [48]: sns.set(font_scale=1) # This line automatically makes the font size a bit bigger on the plot.
         # Set the Seaborn theme for a clean style
         sns.set_theme(style="darkgrid")

         # Convert workingday to categorical labels
         bike["workingday"] = bike["workingday"].map({0: "no", 1: "yes"})

         # Create the scatter plot with regression lines
         sns.lmplot(
             data=bike,  # Use the hourly dataset
             x="casual",  # X-axis: casual riders
             y="registered",  # Y-axis: registered riders
             hue="workingday",  # Color by working or non-working day
             palette={"yes": "chocolate", "no": "royalblue"},  # Match colors
             height=5,  # Adjust figure height
             aspect=1.2,  # Adjust aspect ratio
             scatter_kws={"s": 10, "alpha": 0.6},  # Adjust point size & transparency
             line_kws={"linewidth": 2},  # Make regression lines thicker
         )

         # Add title and labels
         plt.title("Casual vs Registered Riders on Working and Non-Working Days", fontsize=14)
         plt.xlabel("Casual", fontsize=12)
         plt.ylabel("Registered", fontsize=12)

         # Show the plot
         plt.show()
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[48], line 9
      6 bike["workingday"] = bike["workingday"].map({0: "no", 1: "yes"})
      8 # Create the scatter plot with regression lines
----> 9 sns.lmplot(
     10     data=bike,  # Use the hourly dataset
     11     x="casual",  # X-axis: casual riders
     12     y="registered",  # Y-axis: registered riders
     13     hue="workingday",  # Color by working or non-working day
     14     palette={"yes": "chocolate", "no": "royalblue"},  # Match colors
     15     height=5,  # Adjust figure height
     16     aspect=1.2,  # Adjust aspect ratio
     17     scatter_kws={"s": 10, "alpha": 0.6},  # Adjust point size & transparency
     18     line_kws={"linewidth": 2},  # Make regression lines thicker
     19 )
     21 # Add title and labels
     22 plt.title("Casual vs Registered Riders on Working and Non-Working Days", fontsize=14)

File /srv/conda/envs/notebook/lib/python3.11/site-packages/seaborn/regression.py:640, in lmplot(data,
    637     ax.update_datalim(xys, updatey=False)
    638     ax.autoscale_view(scaley=False)
--> 640 facets.map_dataframe(update_datalim, x=x, y=y)
    642 # Draw the regression plot on each facet
    643 regplot_kws = dict(
    644     x_estimator=x_estimator, x_bins=x_bins, x_ci=x_ci,
    645     scatter=scatter, fit_reg=fit_reg, ci=ci, n_boot=n_boot, units=units,
   (…)
    649     scatter_kws=scatter_kws, line_kws=line_kws,
    650 )

File /srv/conda/envs/notebook/lib/python3.11/site-packages/seaborn/axisgrid.py:809, in FacetGrid.map_d
    806 ax = self.facet_axis(row_i, col_j, modify_state)
    808 # Decide what color to plot with
--> 809 kwargs["color"] = self._facet_color(hue_k, kw_color)
    811 # Insert the other hue aesthetics if appropriate
    812 for kw, val_list in self.hue_kws.items():

File /srv/conda/envs/notebook/lib/python3.11/site-packages/seaborn/axisgrid.py:838, in FacetGrid._face
    836 def _facet_color(self, hue_index, kw_color):
--> 838     color = self._colors[hue_index]
    839     if kw_color is not None:
    840         return kw_color

IndexError: list index out of range
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

0.0.5 Question 3d

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 a distinct difference in the relationship between casual and registered riders based on whether the day is a working day. On working days, there is a strong positive correlation between the two, with registered riders consistently outnumbering casual riders. The trendline for working days is steeper, indicating that as the number of casual riders increases, the number of registered riders increases even more sharply. This suggests that registered users are likely commuting to work, while casual riders make up a smaller portion of total bike usage.

On non-working days, the trendline is less steep, and there is greater variation in casual rider counts. This suggests that bike usage patterns on weekends are different from weekdays, with more casual riders and a relatively smaller increase in registered riders.

Overplotting affects the ability to clearly see individual data points, especially in areas where many points overlap, such as the lower left of the plot where rider counts are low. This makes it difficult to identify finer details or potential outliers. While transparency and smaller point sizes help, using alternative visualizations like hexbin plots or density plots could provide a clearer representation of the data distribution._

0.0.6 Question 4a

Generate a bivariate kernel density plot with workday and non-workday separated using the `daily_counts` `DataFrame`. It should look like the first plot displayed above.

Hint: You only need to call `sns.kdeplot` once. Take a look at the `hue` parameter and adjust other inputs as needed.

After you get your plot working, experiment by setting `fill=True` in `kdeplot` to see the difference between the shaded and unshaded versions.

- But, **please submit your work with `fill=False`.**

```
In [ ]: # Set the figure size for the plot
        plt.figure(figsize=(12,8))

        # Create a KDE plot comparing registered and casual riders, separated by workingday
        sns.kdeplot(
            data=daily_counts,
            x="casual",
            y="registered",
            hue="workingday",
            fill=False,  # Ensures the plot remains unshaded as required in the question
            common_norm=False,
            levels=15,  # Adjust contour levels for better visualization
            alpha=0.7,
            linewidths=2
        )

        # Titles and labels
        plt.title("KDE Plot Comparison of Registered vs Casual Riders", fontsize=14)
        plt.xlabel("Casual Riders", fontsize=12)
        plt.ylabel("Registered Riders", fontsize=12)

        # Show plot
        plt.show()
```

0.0.7 Question 4b

With some modification to your Question 4a code (this modification is not in scope), we can generate the plot above.

In your own words, describe what the lines and the color shades of the lines signify about the data. What does each line and color represent?

Hint: You may find it helpful to compare it to a contour or topographical map as shown here.

_The bivariate KDE plot visualizes the density of registered and casual riders, with different colors representing workdays and non-workdays. Similar to contour lines on a topographic map, the lines in the plot indicate areas of varying density—regions with closer contour lines signify steeper changes in density, while wider-spaced lines indicate more gradual variations.

Darker shades represent higher density areas, meaning that most registered and casual riders fall within these regions. The plot shows that on workdays (red contours), registered riders tend to be more concentrated in higher numbers, while on non-workdays (blue contours), there is a broader spread of casual riders, with some peak densities occurring at lower registered rider counts.

This visualization highlights the behavioral differences between casual and registered riders, where registered users are more consistently high in number during workdays, while casual riders exhibit more variation, especially on non-workdays._

0.0.8 Question 4c

What additional details about the riders can you identify from this contour plot that were difficult to determine from the scatter plot?

The contour plot provides a clearer representation of density patterns that were difficult to discern in the scatter plot due to overplotting. It highlights the most common combinations of registered and casual riders by showing areas of high concentration through darker shades and closely spaced contour lines. Unlike the scatter plot, which only shows individual data points, the contour plot reveals that registered riders tend to cluster at higher counts on workdays, while casual riders are more dispersed on non-workdays. Additionally, it helps identify the density distribution across different rider groups, making it easier to see trends and patterns that were previously obscured by overlapping points in the scatter plot.

0.0.9 Question 5b

Let’s examine the behavior of riders by plotting the **average number of riders** for each **time category** (using the `time_category` column), separated by rider type.

Your plot should look like the plot below. It’s fine if your plot’s colors don’t match ours exactly.

Hint:
To label the x-axis correctly, use `plt.xticks()` to manually set tick positions and labels. You may need to rotate the labels for readability. Refer to the documentation for more details.

```
In [ ]: # Group by time category and calculate means
        time_category_means = (
            bike.groupby("time_category")[["casual", "registered"]].mean()
        )

        plt.figure(figsize=(10, 7))
        sns.lineplot(
            data=time_category_means.reset_index(),
            x="time_category",
            y="casual",
            label="Casual Riders",
        )
        sns.lineplot(
            data=time_category_means.reset_index(),
            x="time_category",
            y="registered",
            label="Registered Riders",
        )

        plt.xlabel("Time of the Day (Categories)")
        plt.ylabel("Average Count")
        plt.title("Average Count of Casual vs. Registered Riders by Time Categories")
        plt.xticks(
            ticks=range(len(time_category_means)),  # Order categories
            labels=["Midnight", "Morning", "Lunch Time", "Afternoon", "Evening", "Night"],
            rotation=45 # Rotate x-axis labels for readability
        )
        plt.legend()
        plt.tight_layout()
```

0.0.10 Question 5c

Next, analyze how the average count of casual and registered riders varies by month (`mnth`).

Compute the average number of casual and registered riders for each month in the dataset and create a line plot showing the trends.

Your plot should look like the plot below. It's fine if your plot's colors don't match ours exactly.

```
In [ ]: # Group by month and calculate mean rider counts
        avg_riders_by_month = bike.groupby("mnth")[["casual", "registered"]].mean()

        plt.figure(figsize=(10, 7))

        # Plot casual riders
        sns.lineplot(
            data=avg_riders_by_month.reset_index(),
            x="mnth",
            y="casual",
            label="Casual Riders"
        )

        # Plot registered riders
        sns.lineplot(
            data=avg_riders_by_month.reset_index(),
            x="mnth",
            y="registered",
            label="Registered Riders"
        )

        # Formatting
        plt.xlabel("Month")
        plt.ylabel("Average Rider Count")
        plt.title("Average Number of Casual vs. Registered Riders by Month")
        plt.xticks(
            ticks=range(1, 13),  # Months range from 1 to 12
            labels=[
                "Jan", "Feb", "Mar", "Apr", "May", "Jun",
                "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"
            ],
            rotation=45  # Rotate x-axis labels for readability
        )
        plt.legend()
        plt.tight_layout()
```

0.0.11 Question 5d

What can you observe from the plots generated in **5b** and **5c**?

Discuss your observations for both types of riders, and hypothesize about the meaning of the peaks and troughs of both riders' distributions.

Observations from 5b (Time Categories) • Registered riders have a pronounced peak in the morning (5 AM - 11 AM) and evening (5 PM - 9 PM), with a sharp drop in the afternoon. This suggests that registered riders primarily use the service for commuting, likely traveling to and from work or school. • Casual riders, on the other hand, exhibit a more stable distribution throughout the day, with slightly higher activity during lunch time and evening. This indicates that casual riders might be using bikes more for leisure or non-commute-related trips.

Interpretation of Peaks and Troughs • The peaks in registered riders' activity during the morning and evening indicate a strong correlation with work-related travel. • The peaks for casual riders during lunch time and evenings suggest recreational or flexible-purpose biking.

Observations from 5c (Monthly Trends) • Registered riders show a strong seasonal trend, peaking during the summer months (June - September) and declining in the winter months (November - February). This may be influenced by weather conditions, with more people choosing biking as a reliable commuting option in warm months. • Casual riders follow a similar trend but with more pronounced growth in summer, suggesting an increase in leisure cycling when the weather is pleasant. • Both types of riders show a decline during the winter months, likely due to colder temperatures and adverse weather conditions.

0.0.12 Question 6b

Draw 7 smoothed curves on a single plot, one for each day of the week.

- The x-axis should be the temperature (as given in the 'temp' column).
- The y-axis should 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.

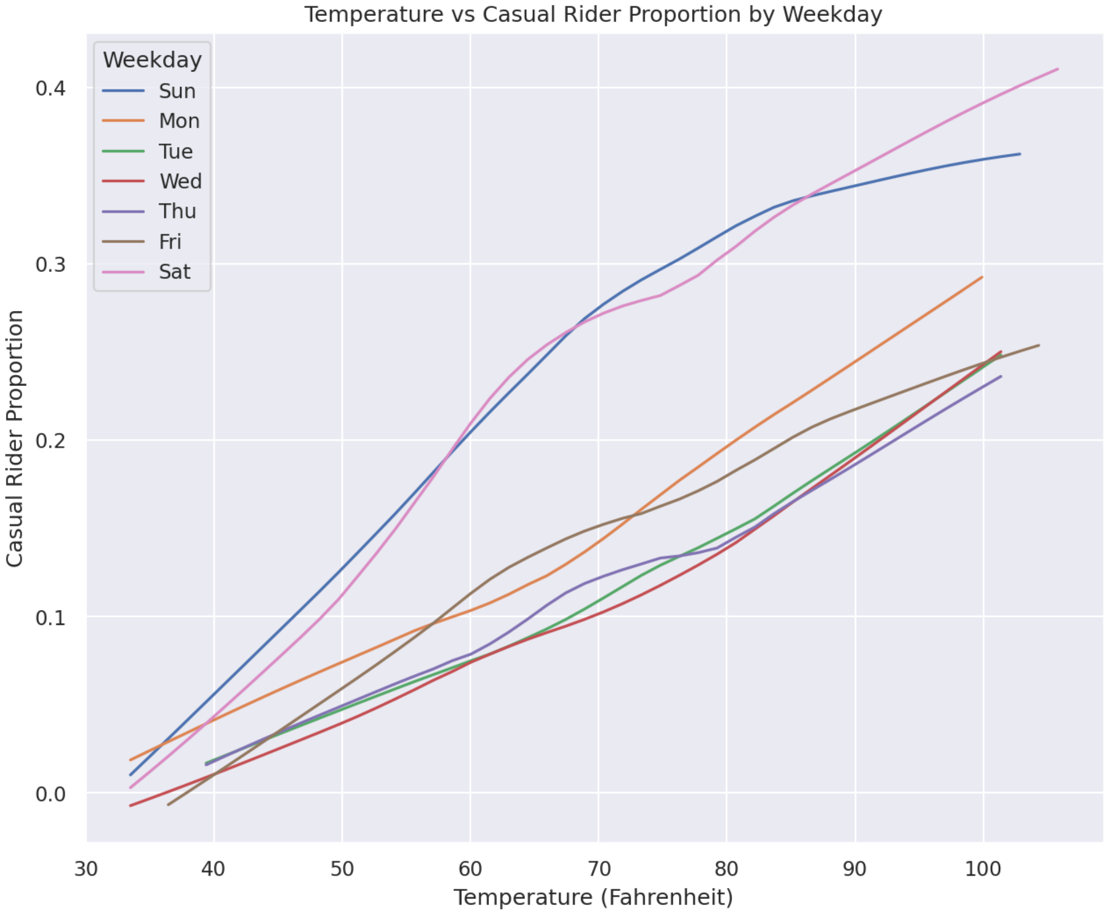
You do not need to match the colors on our sample plot as long as the colors in your plot make it easy to distinguish which day they represent.

Hints: * Start by plotting only one day of the week to make sure you can do that first. Then, consider using a `for` loop to repeat this plotting operation for all days of the week.

- The `lowess` function expects the `y` coordinate first, then the `x` coordinate. You should also set the `return_sorted` field to `False`.
- **You will need to rescale the normalized temperatures stored in this dataset to Fahrenheit values.** Look at the section of this notebook titled 'Loading Bike Sharing Data' for a description of the (normalized) temperature field to know how to convert back to Celsius first. After doing so, convert it to Fahrenheit. By default, the temperature field ranges from 0.0 to 1.0. In case you need it, $\text{Fahrenheit} = \text{Celsius} \times \frac{9}{5} + 32$. If you prefer plotting temperatures in Celsius, that's fine as well! Just remember to convert accordingly so the graph is still interpretable. In addition, for smoother curves, use `sns.lineplot` instead of Matplotlib's default plotting functions.
  This helps avoid "noisy" jagged lines that might appear with `plt.plot` or `plt.scatter`.

```
In [62]: from statsmodels.nonparametric.smoothers_lowess import lowess

         # Convert normalized temperature to Fahrenheit
         bike["temp_fahrenheit"] = bike["temp"] * 73.8 + 32

         # Set up the plot
         plt.figure(figsize=(10, 8))

         # Define the weekdays in order (assuming 0 = Sunday, 6 = Saturday)
```

```
weekdays = ["Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"]

# Loop through each weekday and plot the LOWESS-smoothed curve
for i, day in enumerate(weekdays):
    subset = bike[bike["weekday"] == day]  # Select only the data for the given day

    if not subset.empty:  # Ensure we have data for this day
        smoothed = lowess(subset["prop_casual"], subset["temp_fahrenheit"], frac=0.5, return_se
        sns.lineplot(x=subset["temp_fahrenheit"], y=smoothed, label=day)

# Formatting the plot
plt.xlabel("Temperature (Fahrenheit)")
plt.ylabel("Casual Rider Proportion")
plt.title("Temperature vs Casual Rider Proportion by Weekday")
plt.legend(title="Weekday")
plt.show()
```



0.0.13 Question 6c

Examine the plot above and describe how casual ridership changes with temperature. Determine if the **plot alone** provides evidence of a **causal** relationship between temperature and casual ridership, and explain your reasoning.

Finally, based on **your own intuition**, state whether you think there is a underlying causal relationship. Justify your answer.

The plot indicates a positive association between temperature and the proportion of casual riders. As temperature increases, casual ridership also rises, suggesting that warmer weather makes biking a more attractive option. This trend is especially pronounced on weekends, when casual riders are more active compared to weekdays. The pattern aligns with expectations—people are generally more inclined to engage in recreational activities, such as biking, when the weather is pleasant.

That being said, the plot alone does not establish a causal relationship between temperature and casual ridership. While the upward trend suggests a strong correlation, other confounding factors could be at play. For example, seasonality might influence both temperature and ridership simultaneously—summer months tend to be warmer and also coincide with vacation periods, which could independently drive casual ridership. Additionally, longer daylight hours in warmer months might encourage biking, and external factors such as events, holidays, or public transportation availability could also contribute to fluctuations in ridership.

Intuitively, it seems reasonable to argue that temperature has a causal effect on casual ridership—colder weather likely deters casual riders due to discomfort, while warmer weather makes biking more appealing. However, to make a stronger causal claim, additional analysis would be needed, such as controlling for seasonality, conducting a natural experiment, or using methods like difference-in-differences or instrumental variables to isolate the effect of temperature from other influencing factors.

0.0.14 Question 7a

Imagine you are working for a bike-sharing company that collaborates with city planners, transportation agencies, and policymakers 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 informs 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 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.

Note: There is no single "right" answer to this question – we are looking for thoughtful reflection and commentary on whether or not this dataset, in its current form, encodes information about equity.

Type your answer here, replacing this text.

0.0.15 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 in the US.

Based on your plots in this assignment, would you recommend expanding bike sharing to additional cities in the US? If so, what cities (or types of cities) would you suggest?

Please list at least two reasons why, and mention which plot(s) you drew your 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!

Based on the analysis of bike-sharing ridership patterns in this assignment, I would recommend expanding bike-sharing to additional cities in the US. However, this expansion should be targeted toward cities with characteristics that align with the observed usage trends. Below, I outline two key insights from the plots that support this recommendation and suggest what types of cities would be the best fit.

```
In [ ]: 1. Temperature and Ridership Trends (Question 6b & 6c)
        • Insight: The plots analyzing casual ridership and temperature suggest that ridership increase
        • Implication: This indicates that bike-sharing is more likely to succeed in cities with modera
        City Recommendation:
        • Cities with mild winters and warm summers (e.g., Austin, TX; San Diego, CA; Miami, FL).
        • Cities with strong seasonal demand but manageable winter conditions (e.g., Washington, DC; Den
```

```
In [ ]: 2. Time-of-Day Ridership Patterns (Question 5b & 5c)
        • Insight: The peak ridership times correspond to typical commuting hours, particularly among r
        • Implication: Expanding bike-sharing would be most effective in cities with high commuter dens
        City Recommendation:
        • New York City, NY; Chicago, IL; Seattle, WA – Cities with well-developed but often congested p
        • College towns like Ann Arbor, MI, and Boulder, CO, where students rely on bikes for short comm
```

```
In [ ]: Additional Considerations
        • Equity & Accessibility: The current dataset does not capture demographic information, but futu
        • Infrastructure Readiness: Cities with existing bike infrastructure (protected lanes, bike rac
```