Task 1: Comparing Visualizations

COURSE NAME

CSDS 413 Introduction to Data Analysis

Authors:

Jacob Anderson
Wiam Skakri

September 20, 2025

Contents

Context

In data science, the choice of visualization plays a critical role in shaping how insights are derived and communicated. Different visual encodings can highlight or obscure structure in data — including spread, skew, outliers, modality, or differences between categories. In this task, you will explore how three different types of visualizations can be used to compare distributions across categories, and evaluate which is most appropriate depending on the dataset context.

For this task, you are provided with three datasets, each containing categorical grouping variables and a numerical measurement. Each dataset comes with a research scenario/question. Your task is to clean the data, visualize the distribution across categories using multiple plotting techniques, and discuss which visualization is the most appropriate in addressing the research question for each dataset.

1 Best-Selling Albums Dataset

Attributes: Year, Ranking, Artist, Album, Genre, Worldwide Sales, Tracks, Album Length

Scenario: A media analytics firm is interested in understanding whether certain genres consistently produce top-selling albums or if success is more scattered across genres.

Research Question: How does the distribution of album sales vary across music genres for albums in the previous decade (released after 2015), and are high-sales outliers concentrated in certain genres?

1.1 Part A: Data Cleaning and Preprocessing

First, filter your dataset so that only the variables critical for your analysis remain. Then clean your data so that there is consistency in variable types, capitalization, and handle any missing or invalid values.

—

The album data set had multiple issues that needed to be addressed. First, we pre-processed the data points by thresholding them based on their 'Year' attribute to include only those released after 2015 because that is the window of interest to this hypothetical firm. Then we removed the features not critical to the analysis of album sales by genre, leaving only 'Worldwide Sales (Est.)' and 'Genre'. On that note, we standardized the column names under snake case as 'album_sales' and 'genres', respectively.

In terms of cleaning the attribute values themselves, we standardized the casing of the genre values and then checked this step by printing a dictionary of genres as keys, mapped to their respective counts. Before this casing step was enforced, printing the dictionary revealed that one genre value was written as "HIp Hop" and had defined two separate keys for the one genre.

The final cleaning step expresses each of the album sale values as an integer data type without any commas so that they would not be conflated as delimiters in the CSV file.

We wrote this utility function to accomplish the task:

```
def clean_preprocess_albums_data(input_csv: str, output_csv: str) -> pd.DataFrame:
    """
    Cleans and preprocesses the albums dataset by removing irrelevant
    features and data points, handling missing/invalid values, standardizing
    capitalization, and converting data types.

    :param input_csv:: Path to the input CSV file containing the albums dataset.
    :type input_csv: str
    :param output_csv: Filename for the clean data.
    :type output_csv: str
    :returns: pd.DataFrame
    :rtype: pd.DataFrame
    """
    df = pd.read_csv(input_csv)

    # Keeps critical variables, relevant data points, removes missing value rows,
    # and renames columns more appropriately
    df = df[df['Year'] > 2015]
    df = df.iloc[:, [4, 7]]
    df.dropna(inplace=True)
    df.columns = ['album_sales', 'genre']

    # Confirms there aren't duplicate genres due to misspelling/invalid vals
    # and standardizes capitalization
    df['genre'] = df['genre'].str.lower()
    print(df['genre'].value_counts().to_dict())

    # Reformats albums sales as integers
    df['album_sales'] = df['album_sales'].str.replace(',', '').astype(int)
```

```
    os.makedirs(os.path.dirname(f'../datasets/clean/{output_csv}'), exist_ok=True)
    df.to_csv(f'../datasets/clean/{output_csv}', index=False)
    return df
```

Figure 1: Best-Selling Albums data pre-processing function.

Below is a comparison of the head of each version to illustrate the changes that were made:

Top_10_Albums_By_Year.csv

```
Year,Ranking,Artist,Album,Worldwide Sales (Est.),Tracks,Album Length,Genre
1990,1,Madonna,The Immaculate Collection,"30,000,000",17,73:32,Pop
1990,2,New Kids On The Block,Step By Step,"20,000,000",12,47:44,Pop
1990,3,Garth Brooks,No Fences,"18,770,000",10,34:34,Country
1990,4,MC Hammer,Please Hammer Don't Hurt Em,"18,000,000",13,59:04,Hip Hop
1990,5,Mariah Carey,Mariah Carey,"15,000,000",11,46:44,Pop
1990,6,Movie Soundtrack,Aashiqui,"15,000,000",12,58:13,World
1990,7,Whitney Houston,I'm Your Baby Tonight,"10,000,000",11,53:45,Pop
1990,8,Phil Collins,Serious Hits... Live!,"9,956,520",15,76:53,Rock
1990,9,Enigma,MCMXC A.D.,"8,838,000",7,40:16,Pop
1990,10,The Three Tenors,Carreras Domingo Pavarotti In Concert 1990,"8,533,000",17,67:55,Classical
...
```

album_sales_by_genre.csv

```
album_sales,genre
7657000,hip hop
6111355,hip hop
4421666,r&b
4207235,pop
4170954,pop
3661560,pop
3462374,pop
3418440,pop
3189149,pop
2727078,pop
...
```

Figure 2: Raw vs. Cleaned Best-Selling Albums dataset.

1.2 Part B: Generate Three Visualizations

Produce the following types of plots:

- **Error Bar Plot:** Show the mean and variability (e.g., standard error or 95% confidence intervals) of the numerical variable across each category.
- **Barcode Chart:** Also known as a strip plot or rug plot. Shows individual data points across categories.
- **Histogram:** Plot the distribution of the numerical variable, grouped by the categorical variable (using hue or facet).

Error Bar Plot

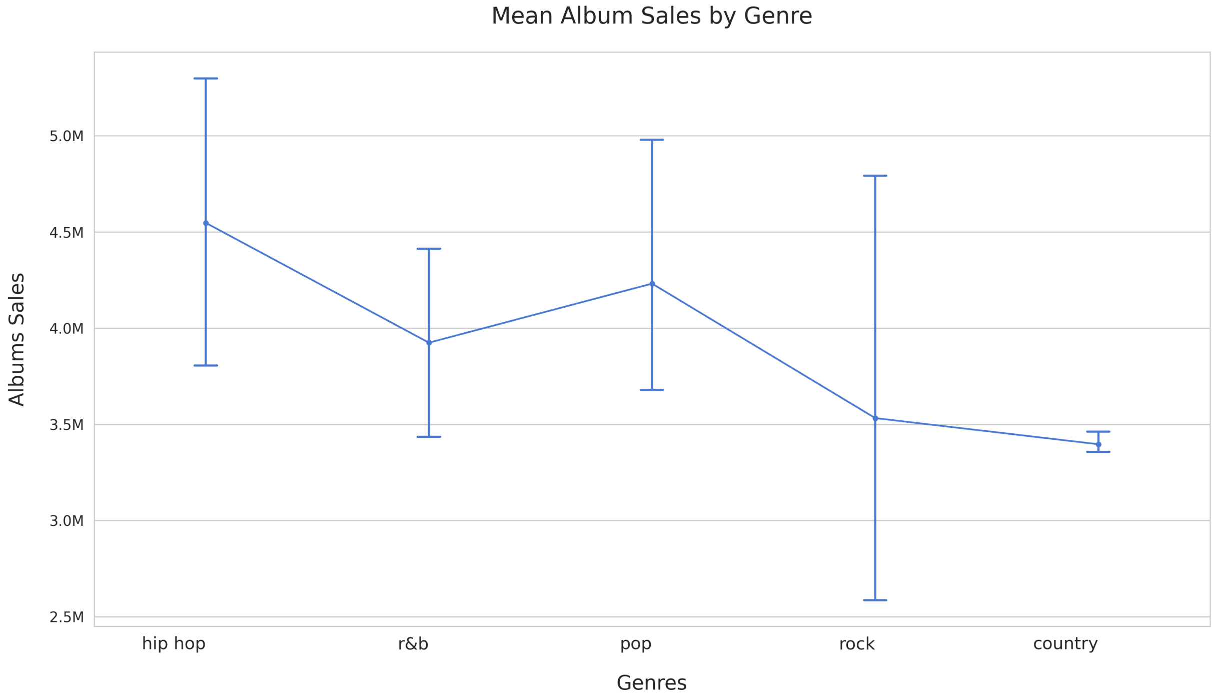

Figure 3: Mean Album Sales by Genre with 95% confidence intervals.

Barcode Chart

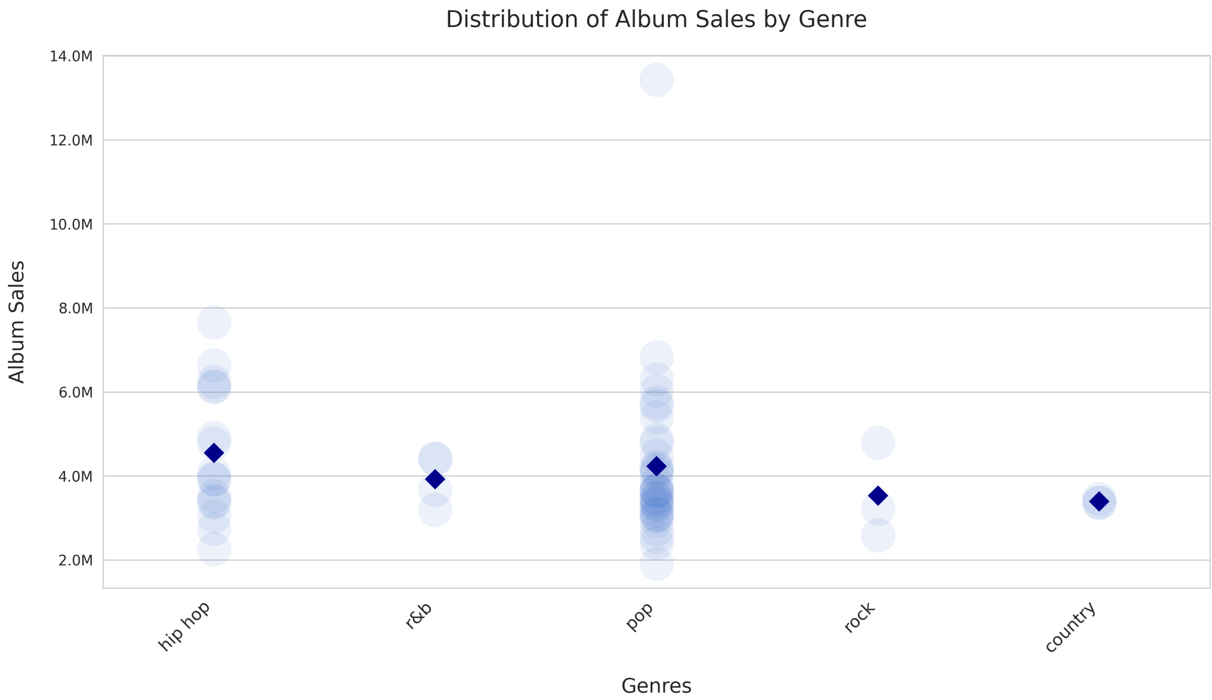

Figure 4: Average Distribution of Album Sales by Genre.

Histogram

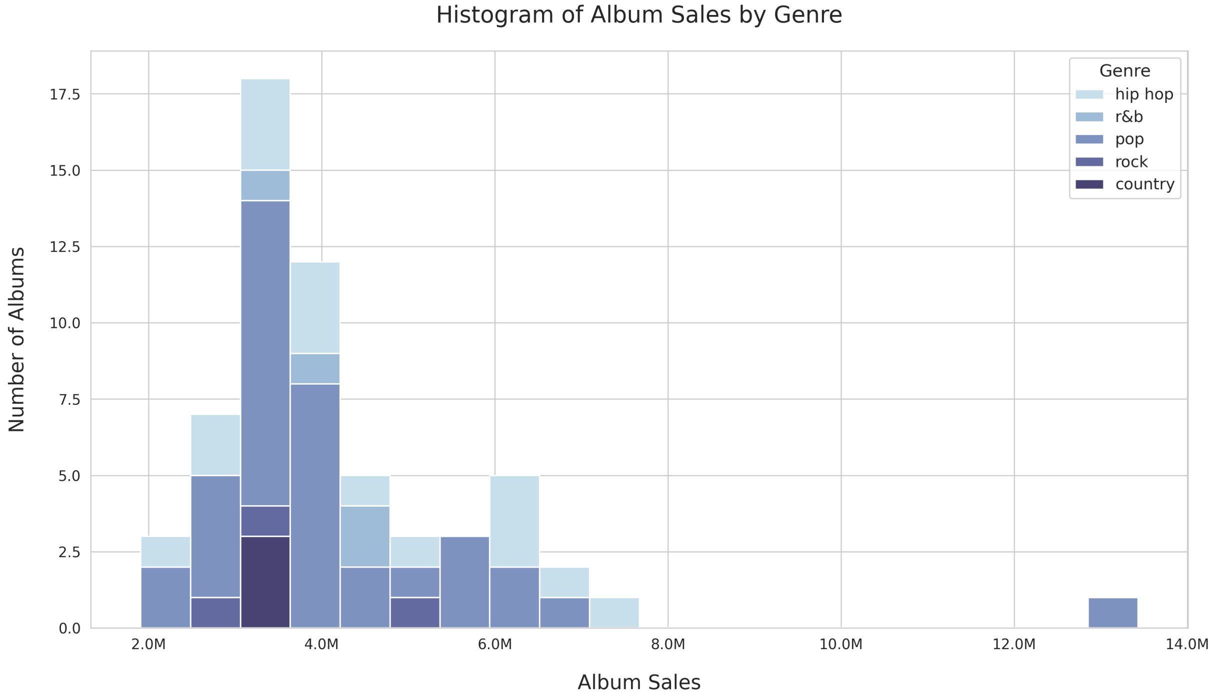

Figure 5: Histogram of Album Sales by Genre.

1.3 Part C: Evaluate and Justify Visualization

For the dataset:

- Discuss the advantages and disadvantages of each visualization type.
- Decide which visualization is best for the research question.
- Support your answer with evidence from the plots and reasoning based on dataset size, shape, or structure.

Advantages and Disadvantages

The error bar plot well-represents the spread of the data out to our specific confidence level, giving us a notion of uncertainty in this understanding of the distributions, and shows clearly the average number of album sales for each genre. Where the error bar plot loses out, however, is in their ability to represent outliers/non-general cases out to our uncertainty threshold. This plot gives us no knowledge of what the distribution looks like outside the error bars, and within the error bars, we do not get to see the shape that the distribution's mass forms directly, it only signals this shape by the information we can gather from the mean points and the error bar lengths. We can also see in later visualizations that the mass distribution for each genre does look a bit different and so there is some necessary context getting lost for this data. Additionally, while the notion of uncertainty in understanding variance is informative, it is not as useful for the genres where data is very sparse, and so in the cases of genres like Rock or Country, the understanding of their variance in this plot could very well be over- or under- exaggerated.

The strip plot does a great job at contending with the spread of the data; as we saw in class, the density of the points is a very natural indicator of how the mass of the distribution is laid out across its domain, and we also get on top of that direct knowledge of what outliers exist for each genre and what that behavior looks like comparatively for each genre. We can see from the figure one relatively extreme outlier in the Pop genre that was not obvious before with the error bar plot; an Ed Sheeran album called "Divide" from 2017, achieving just north of 13.4 million sales. This plot also tends to include a mean indicator, which we have included as a distinct, opaque symbol overtop the data points, and this is very helpful in understanding the difference between the extreme behavior and the typical behavior for each genre. The sole issue we have with this visualization is that it is still one step/"layer" away from getting the full benefit of viewing a distribution plotted for each genre directly. Nevertheless, for this particular context, we are examining a fairly small number of examples (only releases after 2015), so any issues with too much overlap and a lack of clarity in how the mass is distributed is not something that we will struggle with much here.

The histogram is primarily helpful in the exact area where the strip plot can fail, where perhaps the mass of the distribution is hard to interpret due to this overlapping quality of its visuals and perhaps we could be looking at a larger set of samples in some cases, and so here we get the benefit of most directly viewing how the mass is distributed over the different bins of album sales for each genre. Regarding each genre though, one concern we have is that, regardless of every color palette we tried, reading the behavior of each distribution comparatively i.e. between genres is obfuscated by the fact that they are all plotted on the same figure and the shapes of these distributions are not as smooth because of how few samples we are analyzing. Where we benefit much more with histograms is in examining the extreme values and in understanding the general tendencies of the data. Again, we would argue that strip plots represent these behaviors better for this data, but nonetheless histograms do well to provide us with this context, certainly well enough to make informed claims about the samples. The sole struggle here is just with comparing the genres as effectively.

Which visualization is best for the research question

How does the distribution of album sales vary across music genres for albums in the previous decade (released after 2015), and are high-sales outliers concentrated in certain genres? The strip plot is best for answering this research question.

Evidence-based Support for this answer

First, let's compare the information we collect from the error bar plot versus the strip plot. Both give us very clearly the mean values for each genre, so when contending with the question of album sales varying across genres, we can equally speak to the general behavior of each genre. Both tell us that Hip-Hop and Pop sells the most albums on average, having a noticeably higher mean, followed by R&B and Rock, and trailed by

Country with a noticeably lower average sale performance. We will concede though that the error bar plot does do marginally better to comparatively visualize the mean values across genres due to the fact that it does not have to scale to extreme outliers in the data, allowing it to be more expressive in this respect. Nonetheless, one can still look at the strip plot and immediately make the comparisons we just did, it is just not as obvious.

Regrading the variance of the data, the strip plot gives us information more relevant to our research question, as we are concerned with high-sales outliers in our data. It is obvious from the shape of our data for each genre that the error bar plot completely misses the primary point to take away from the data in this respect; that being the high-sale outlier in the Pop genre. The strip plot makes it immediately clear and is the obvious choice as far as understanding the whole second part of our research question. Additionally, the error bars, while informative in the case of genres with a lot of exposure in the top 10 over the past decade, struggle to provide as much leverage to make claims about the general behavior of sparser genres. Again, as the strip plot provide a much more direct visualization of how the data is distributed, it is the obvious choice over error bar plot for answering the research question.

As far as choosing between the strip plot and the histogram, we confirmed in class and previously from the discussion of advantages and disadvantages that the strip plot mainly loses value in the cases that there is so much data, or perhaps the data is so concentrated toward the distribution's center, that it could be hard to understand the variance for each genre. That is not the case here and the strip plot actually does quite a good job at representing the relative spread in the data across genres; in this case, it's like staring at the distributions from a bird's-eye view, uninhibited by any overlapping masses across genres and expressive enough to readily show qualities like the extra mass of the left side of the pop genre distribution of album sales and the relative difference in variance between rock, pop, and hip hop.

2 Anime Dataset

Attributes: Rank, Name, Japanese_name, Type, Episodes, Studio, Release_season, Tags, Rating, Release_year, End_year, Description, Content_Warning, Related_Mange, Related_anime, Voice_actors, staff

Scenario: A streaming service is considering expanding its short anime series catalog (< 25 episodes) and wants to understand how viewer ratings differ between anime TV series and movies released after 2015. The goal is to determine which format generally receives better audience reception to inform licensing and promotion strategies.

Research Question: How do audience ratings compare between anime TV series and movies released after 2015, and which format generally receives higher ratings?

2.1 Part A: Data Cleaning and Preprocessing

First, filter your dataset so that only the variables critical for your analysis remain. Then clean your data so that there is consistency in variable types, capitalization, and handle any missing or invalid values.

—

We first filtered by Year to keep everything released after 2015, then removed all of the columns irrelevant to the research question, leaving 'Type' and 'Rating'. After doing a drop of all rows with missing values and renaming the columns so they are consistent to how the albums data was set up, there was much whitespace left in the attribute values of the type column, so that was stripped away.

At that point we could force all of the type values to lowercase and read out a dictionary to ensure that there were not any misspelling concerns. Below is the corresponding utility function:

```
def clean_preprocess_anime_data(input_csv: str, output_csv: str) -> pd.DataFrame:
    """
    Cleans and preprocesses the anime dataset by filtering out the irrelevant features
    and datapoints and reformatting the columns names and attribute values

    :param input_csv:: Path to the input CSV file containing the anime dataset.
    :type input_csv: str
    :param output_csv: Filename for the clean data.
    :type output_csv: str
    :returns: pd.DataFrame
    :rtype: pd.DataFrame
    """
    df = pd.read_csv(input_csv)

    # Keeps relevant variables, relevant data poitns, removes the
    # missing value rows, renames the columns, and strips the whitespace
    # out for the type col
    df = df[df['Release_year'] > 2015]
    df = df.loc[:, ['Type', 'Rating']]
    df.dropna(inplace=True)
    df.columns = ['type', 'rating']
    df['type'] = df['type'].str.strip()

    # Standardizes to lower case and filters out irrelevant types
    df['type'] = df['type'].str.lower()
    df = df[df['type'].isin(['tv', 'movie'])]
    print(df['type'].value_counts().to_dict())

    os.makedirs(f'../datasets/clean/', exist_ok=True)
    df.to_csv(f'../datasets/clean/{output_csv}', index=False)
    return df
```

Figure 6: Anime data pre-processing function.

Below is a comparison for this dataset to illustrate the changes:

Anime.csv

```
Rank,Name,Japanese_name,Type,Episodes,Studio,Release_season,Tags,Rating,Release_year,End_year,Descripti
1,Demon Slayer: Kimetsu no Yaiba - Entertainment District Arc, Kimetsu no Yaiba: Yuukaku-hen,TV   ,,ufo
Original Creator, Haruo Sotozaki
Director, Akira Matsushima
Character Design, Aimer
Song Performance","Koyoharu Gotouge : Original Creator, Haruo Sotozaki : Director, Akira Matsushima : C
2,Fruits Basket the Final Season, Fruits Basket the Final,TV   ,13.0,TMS Entertainment,Spring,"Drama, F
Original Creator, Yoshihide Ibata
Director & Episode Director & Storyboard, Taku Kishimoto
Screenplay & Series Composition, Masaru Yokoyama
Music, Masaru Shindou
Character Design & Chief Animation Director, Baek-Ryun Chae
Photography Director, Youko Koyama
Art Director, Mika Sugawara
Color Design","Natsuki Takaya : Original Creator, Yoshihide Ibata : Director & Episode Director & Story
3,Mo Dao Zu Shi 3, The Founder of Diabolism 3,Web  ,12.0,B.C MAY PICTURES,,"Fantasy, Ancient China, Chi
Original Creator, Xiong Ke
Chief Director, Ma Chendi
Chief Director, Sun Yujing
Music, Weng Teng
Music, Feng Shuo
Music, Shen Lin
Character Design & Chief Animation Director, Liang Sha
Screenplay","Mo Xiang Tong Xiu : Original Creator, Xiong Ke : Chief Director, Ma Chendi : Chief Directo
4,Fullmetal Alchemist: Brotherhood, Hagane no Renkinjutsushi: Full Metal Alchemist,TV   ,64.0,Bones,Spr
Original Creator, Yasuhiro Irie
Director, Akira Senju
Music, Hiroki Kanno
2Nd Key Animator & Animation Director & Assistant Animation Director & Character Design & Key Animator,
Producer, Ryou Ooyama
Producer, Nobuyuki Kurashige
Producer, Noritomo Yonai
Producer","Hiromu Arakawa : Original Creator, Yasuhiro Irie : Director, Akira Senju : Music, Hiroki Kan
5,Attack on Titan 3rd Season: Part II, Shingeki no Kyojin Season 3: Part II,TV   ,10.0,WIT Studio,Sprin
Original Creator, Tetsurou Araki
Chief Director, Masashi Koizuka
Director, Tetsuya Wakano
Assistant Director, Yasuko Kobayashi
Series Composition, Hiroyuki Sawano
Music, Kyouji Asano
Character Design, Kazuhiro Yamada
Photography Director","Hajime Isayama : Original Creator, Tetsurou Araki : Chief Director, Masashi Koiz
...
```

anime.csv

```
type,rating
tv,4.6
tv,4.6
tv,4.57
tv,4.56
tv,4.56
...
```

Figure 7: Raw vs. Cleaned Anime dataset.

2.2 Part B: Generate Three Visualizations

Produce the following types of plots:

- **Error Bar Plot:** Show the mean and variability (e.g., standard error or 95% confidence intervals) of the numerical variable across each category.
- **Barcode Chart:** Also known as a strip plot or rug plot. Shows individual data points across categories.
- **Histogram:** Plot the distribution of the numerical variable, grouped by the categorical variable (using hue or facet).

Error Bar Plot

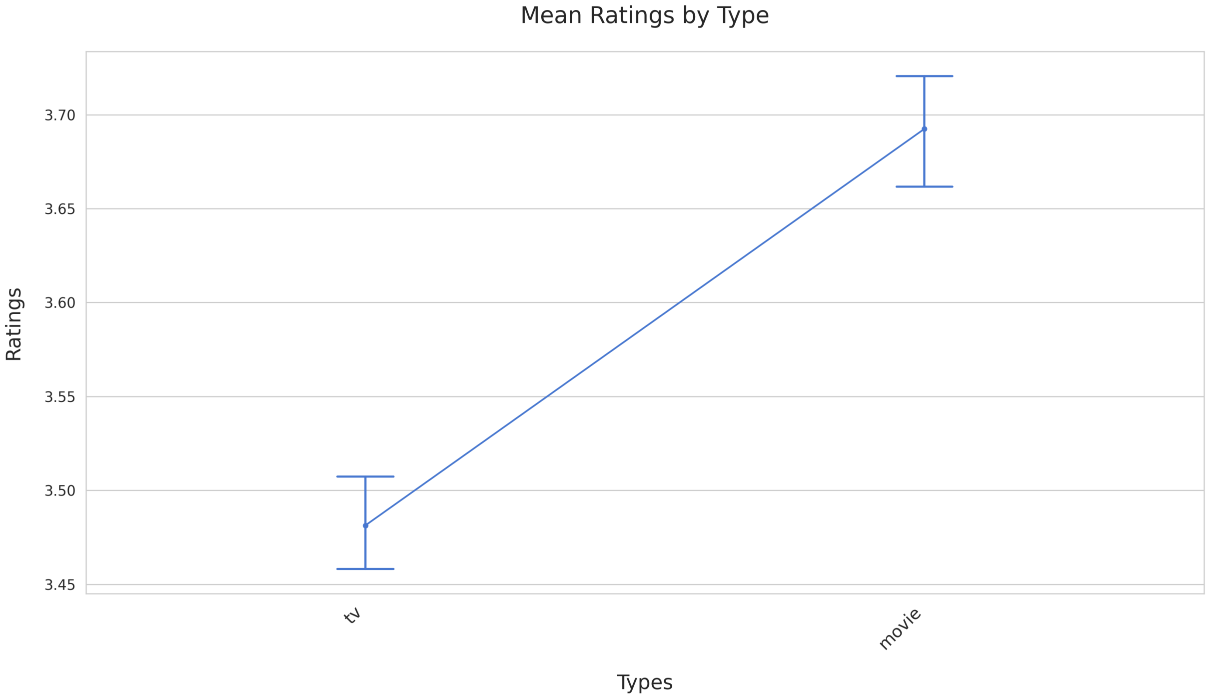

Figure 8: Mean Ratings by Type with 95% confidence intervals.

Barcode Chart

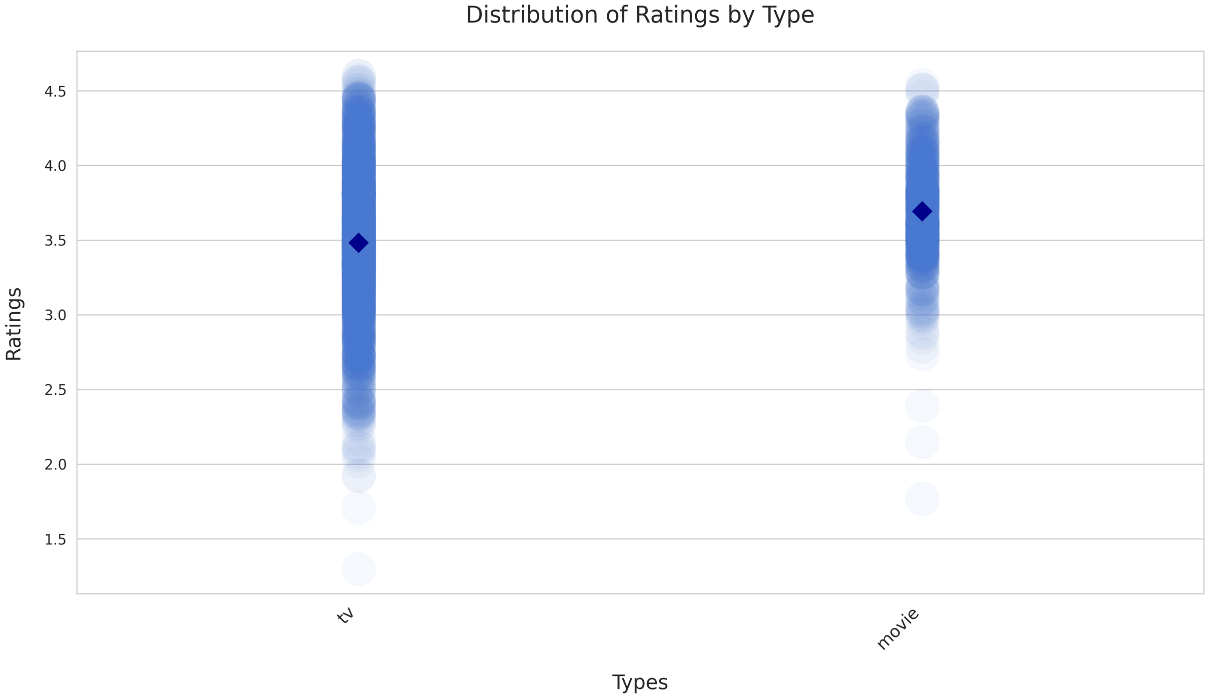

Figure 9: Average Distribution of Ratings by Type.

Histogram

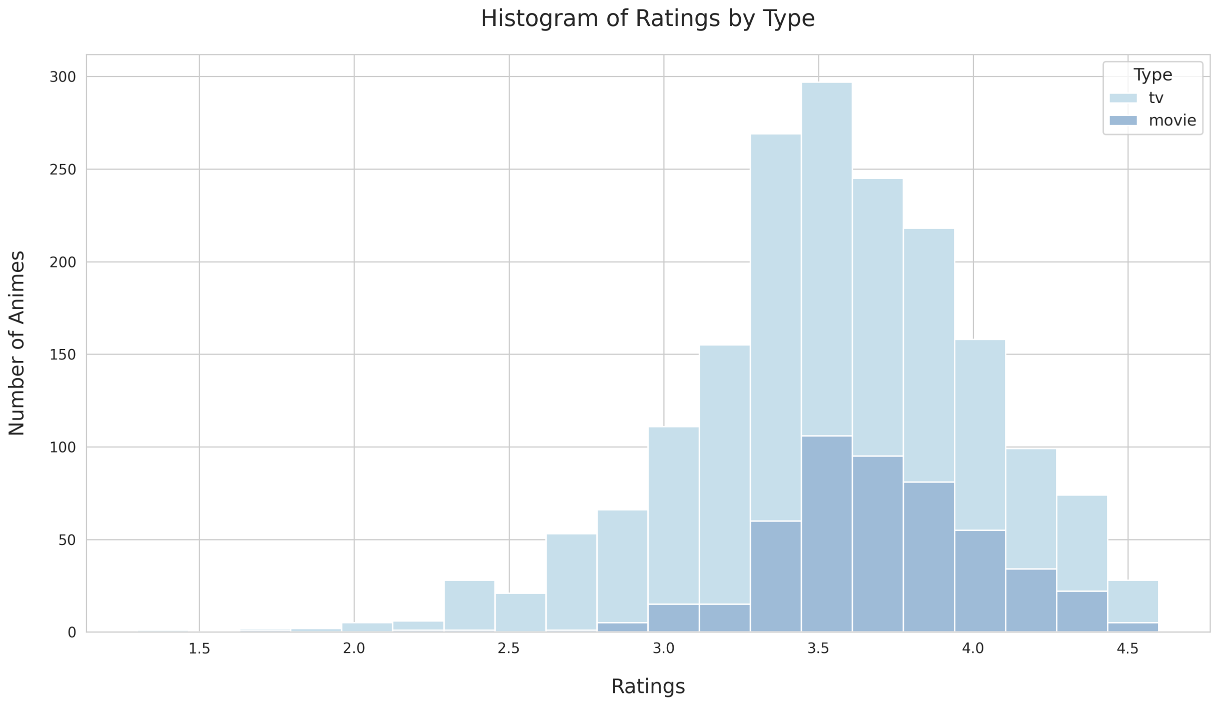

Figure 10: Histogram of Ratings by Type.

2.3 Part C: Evaluate and Justify Visualization

For the dataset:

- Discuss the advantages and disadvantages of each visualization type.
- Decide which visualization is best for the research question.
- Support your answer with evidence from the plots and reasoning based on dataset size, shape, or structure.

3 Algorithm Performance Dataset

Attributes: Algorithm, Epoch, Accuracy, Trial Number

Scenario: You are testing two reinforcement learning (RL) algorithms on a sequential decision task. To avoid overfitting and simulate real-world noise, you shuffle the dataset for each trial and run 10 independent trials per algorithm. For each trial, you track the accuracy across 10 training epochs (one pass through a dataset). Due to how you shuffle your data and algorithmic stochasticity, accuracy results vary across trials.

Research Question: Which algorithm performs more accurately on average across epochs, and how does the use of a visualization help you assess reliability and variation of each algorithm?

3.1 Part A: Data Cleaning and Preprocessing

First, filter your dataset so that only the variables critical for your analysis remain. Then clean your data so that there is consistency in variable types, capitalization, and handle any missing or invalid values.

—–

The scenario specified that we are interested in epochs 1-10 over 10 trials, so we first filtered out any epochs and/or runs beyond that, for which there was one epoch 11 in the raw data. After also dropping any rows with missing accuracies, for which there were a few, we then finished by standardizing the format of the Algorithms columns and renaming the columns, to be consistent with the other two clean datasets.

The Algorithms column had inconcsistent casing of the word 'Algorithm', which is also irrelevant entirely to grabbing the information of which algorithm we are observing, so the casing was standardized and the term was removed from all of the values in this row. There was also whitespace in some cases, so each record was stripped, and we were left with clean 'a'/'b' labels to work with.

Below is the corresponding utility function:

```
def clean_preprocess_algorithms_data(input_csv: str, output_csv: str) -> pd.DataFrame:
    """
    Cleans and preprocesses the algorithms dataset by constraining the trials and epochs
    to the first 10 and then cleans the inconsistent entry of algorithm labels.

    :param input_csv:: Path to the input CSV file containing the algorithm trials dataset.
    :type input_csv: str
    :param output_csv: Filename for the clean data.
    :type output_csv: str
    :returns: pd.DataFrame
    :rtype: pd.DataFrame
    """
    df = pd.read_csv(input_csv)

    df.dropna(inplace=True)
    df = df.loc[:, ['Epoch', 'Algorithm', 'Run', 'Accuracy']]

    # Contrains to trial and epoch values within 1-10
    df = df[(df['Epoch'] >= 1) & (df['Epoch'] <= 10)]
    df = df[(df['Run'] >= 1) & (df['Run'] <= 10)]

    # Standardizes algorithms att value format
    df['Algorithm'] = df['Algorithm'].str.strip()
    df['Algorithm'] = df['Algorithm'].str.lower()
    df['Algorithm'] = df['Algorithm'].str.replace('algorithm ', '')

    df.columns = ['epoch', 'algorithm', 'run', 'accuracy']

    os.makedirs(f'../datasets/clean/', exist_ok=True)
    df.to_csv(f'../datasets/clean/{output_csv}', index=False)
    return df
```

Figure 11: Algorithm Performance data pre-processing function.

Below is a comparison for this dataset to illustrate the changes:

algorithm_trials.csv

```
,Epoch,Algorithm,Run,Accuracy
0,1, algorithm a ,1,0.0464
1,2, algorithm a ,1,0.0069
2,3, algorithm a ,1,0.0992
3,4, algorithm a ,1,0.241
4,5, algorithm a ,1,
5,6, algorithm a ,1,0.4813
6,7, algorithm a ,1,0.8574
7,8, algorithm a ,1,0.9422
8,9, algorithm a ,1,0.915
9,10, algorithm a ,1,1.0
...
```

algo_accuracy_by_epoch.csv

```
epoch,algorithm,run,accuracy
1,a,1,0.0464
2,a,1,0.0069
3,a,1,0.0992
4,a,1,0.241
6,a,1,0.4813
7,a,1,0.8574
8,a,1,0.9422
9,a,1,0.915
10,a,1,1.0
1,a,2,0.0
...
```

Figure 12: Raw vs. Cleaned Algorithm Performance dataset.

3.2 Part B: Generate Three Visualizations

Produce the following types of plots:

- **Error Bar Plot:** Show the mean and variability (e.g., standard error or 95% confidence intervals) of the numerical variable across each category.
- **Barcode Chart:** Also known as a strip plot or rug plot. Shows individual data points across categories.
- **Histogram:** Plot the distribution of the numerical variable, grouped by the categorical variable (using hue or facet).

Error Bar Plot

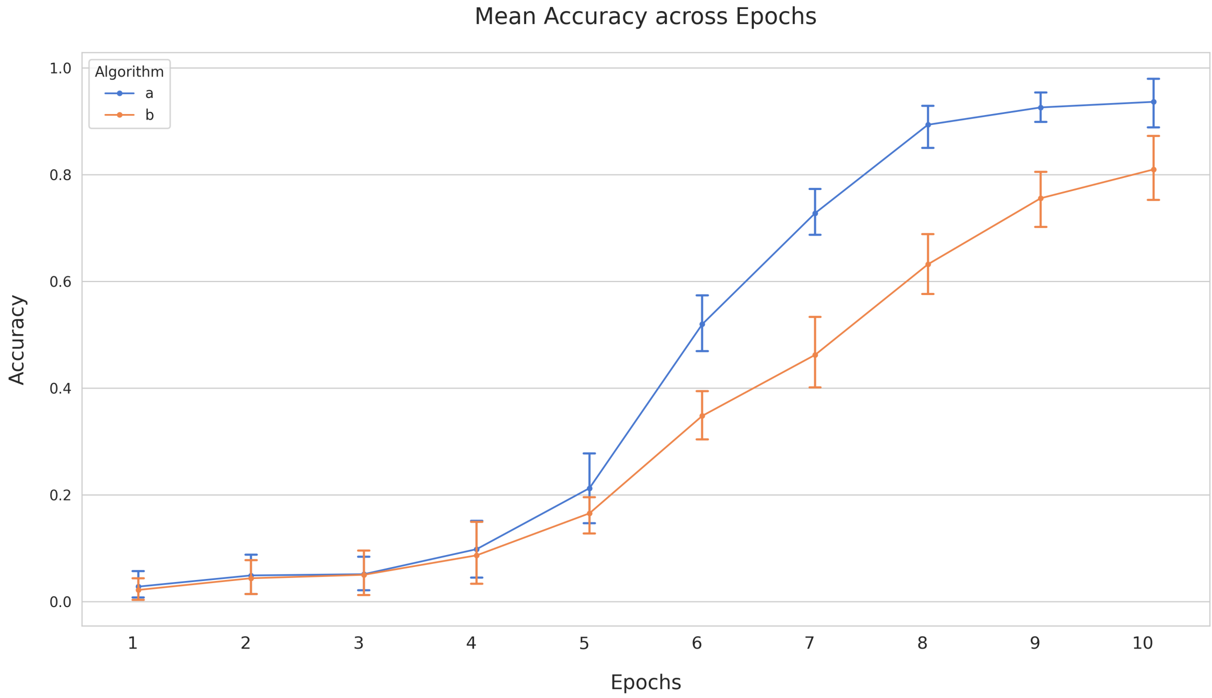

Figure 13: Mean Accuracy across Epochs by Algorithm with 95% confidence intervals.

Barcode Chart

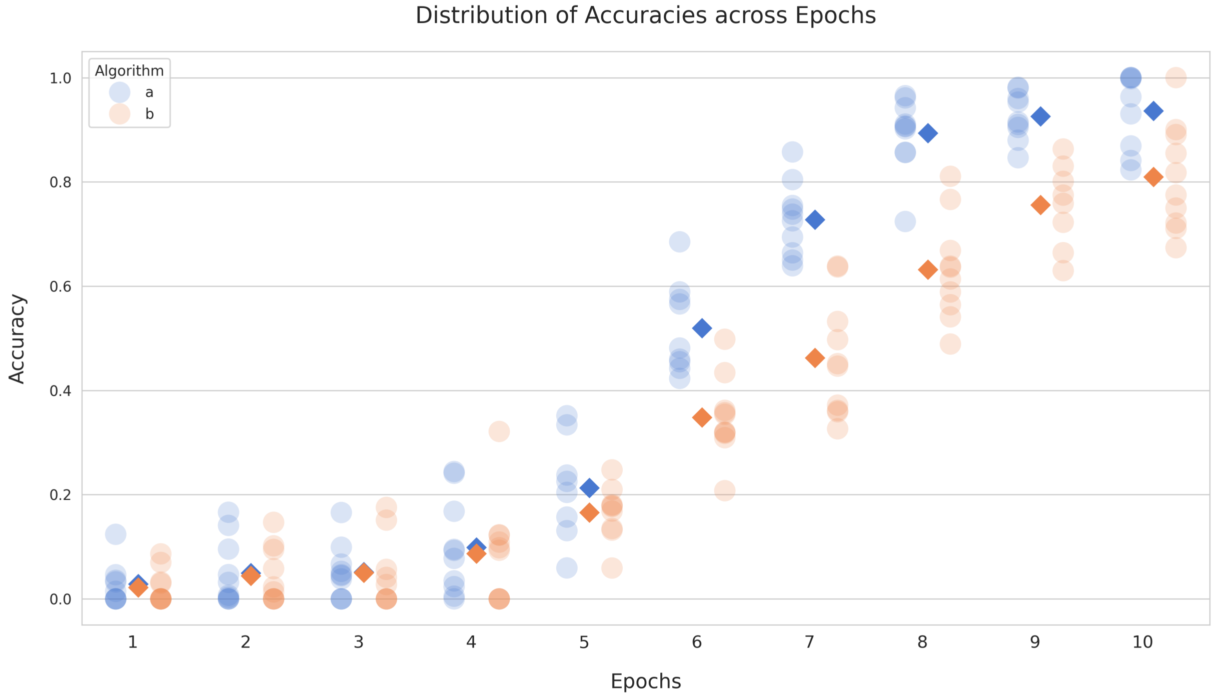

Figure 14: Average Distribution of Accuracy across Epochs by Algorithm.

Histogram

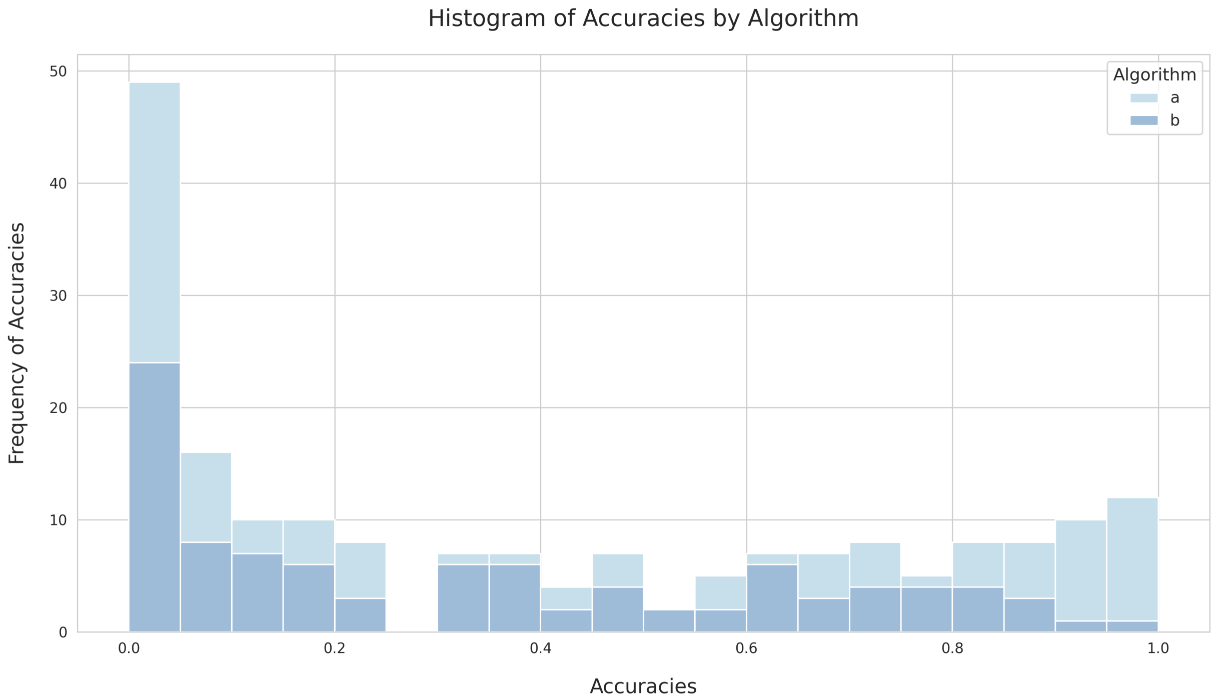

Figure 15: Histogram of Accuracies by Algorithm.

3.3 Part C: Evaluate and Justify Visualization

For the dataset:

- Discuss the advantages and disadvantages of each visualization type.
- Decide which visualization is best for the research question.
- Support your answer with evidence from the plots and reasoning based on dataset size, shape, or structure.

Task 2: Fitting and Comparing Distributions

COURSE NAME

CSDS 413 Introduction to Data Analysis

Authors:

Wiam Skakri
Jacob Anderson

September 18, 2025

Contents

Context

In this task, you will explore how different types of real-world datasets may follow different distributions. You will need to develop a set of hypotheses and perform experiments to validate your own hypotheses.

1 Normal Distribution Dataset

1.1 Part A: Developing Hypotheses

Identify and collect a real-world dataset that you hypothesize follows a Normal distribution. Please be clear about the reasoning behind your hypothesis and be specific about the source of the dataset.

https://www.kaggle.com/datasets/uciml/iris

1.2 Part B: Fitting Distributions

For this exercise, we will call each of the four different theoretical distributions (normal, uniform, power law, exponential) a “model". Fit the dataset (i.e., estimate the model parameters) against each model (not just the one you hypothesized) using maximum likelihood estimation (or using any technique you think is appropriate; make sure to comment on the validity of your approach). This should result in a total of **4 parameter sets**. Report the estimated parameters in the following tabular format:

		Model			
Dataset	***# Observations***	**Normal**	**Uniform**	**Power law**	**Exponential**
Dataset 1	n_1	μ_1, σ_1	a_1, b_1	$\alpha_1, x_{\min_1}$	λ_1

Be sure to show the code you used to arrive at your final estimates clearly.

1.3 Part C: Comparing Real and Synthetic Data

For each fitted distribution (there will be 4 of them for this dataset, each corresponding to a different model), generate a synthetic sample of data points equal to the sample size of the real dataset using the respective model parameters you inferred from the real dataset.

Compare the real vs. synthetic data distributions using methods you think are the most appropriate, including visualizations. So, for this dataset, we compare the original dataset to four synthetic datasets, all with equal number of observations, but each synthetic dataset is generated using a different model.

For this dataset, identify the synthetic dataset (which corresponds to a model) that is most similar to the original data in terms of its distribution.

Now revisit your initial hypothesis. For this dataset: Did the dataset behave as expected, or was another model (assumed distribution) a better fit to the dataset? Reflect on why the observed results may differ from your expectations.

2 Uniform Distribution Dataset

2.1 Part A: Developing Hypotheses

Identify and collect a real-world dataset that you hypothesize follows a Uniform distribution. Please be clear about the reasoning behind your hypothesis and be specific about the source of the dataset.

2.2 Part B: Fitting Distributions

For this exercise, we will call each of the four different theoretical distributions (normal, uniform, power law, exponential) a “model". Fit the dataset (i.e., estimate the model parameters) against each model (not just the one you hypothesized) using maximum likelihood estimation (or using any technique you think is appropriate; make sure to comment on the validity of your approach). This should result in a total of **4 parameter sets**. Report the estimated parameters in the following tabular format:

		Model			
Dataset	***# Observations***	**Normal**	**Uniform**	**Power law**	**Exponential**
Dataset 2	n_2	μ_2, σ_2	a_2, b_2	$\alpha_2, x_{\min_2}$	λ_2

Be sure to show the code you used to arrive at your final estimates clearly.

2.3 Part C: Comparing Real and Synthetic Data

For each fitted distribution (there will be 4 of them for this dataset, each corresponding to a different model), generate a synthetic sample of data points equal to the sample size of the real dataset using the respective model parameters you inferred from the real dataset.

Compare the real vs. synthetic data distributions using methods you think are the most appropriate, including visualizations. So, for this dataset, we compare the original dataset to four synthetic datasets, all with equal number of observations, but each synthetic dataset is generated using a different model.

For this dataset, identify the synthetic dataset (which corresponds to a model) that is most similar to the original data in terms of its distribution.

Now revisit your initial hypothesis. For this dataset: Did the dataset behave as expected, or was another model (assumed distribution) a better fit to the dataset? Reflect on why the observed results may differ from your expectations.

3 Power Law Distribution Dataset

3.1 Part A: Developing Hypotheses

Identify and collect a real-world dataset that you hypothesize follows a Power Law distribution. Please be clear about the reasoning behind your hypothesis and be specific about the source of the dataset.

3.2 Part B: Fitting Distributions

For this exercise, we will call each of the four different theoretical distributions (normal, uniform, power law, exponential) a “model". Fit the dataset (i.e., estimate the model parameters) against each model (not just the one you hypothesized) using maximum likelihood estimation (or using any technique you think is appropriate; make sure to comment on the validity of your approach). This should result in a total of **4 parameter sets**. Report the estimated parameters in the following tabular format:

		Model			
Dataset	***# Observations***	**Normal**	**Uniform**	**Power law**	**Exponential**
Dataset 3	n_3	μ_3, σ_3	a_3, b_3	$\alpha_3, x_{\min_3}$	λ_3

Be sure to show the code you used to arrive at your final estimates clearly.

3.3 Part C: Comparing Real and Synthetic Data

For each fitted distribution (there will be 4 of them for this dataset, each corresponding to a different model), generate a synthetic sample of data points equal to the sample size of the real dataset using the respective model parameters you inferred from the real dataset.

Compare the real vs. synthetic data distributions using methods you think are the most appropriate, including visualizations. So, for this dataset, we compare the original dataset to four synthetic datasets, all with equal number of observations, but each synthetic dataset is generated using a different model.

For this dataset, identify the synthetic dataset (which corresponds to a model) that is most similar to the original data in terms of its distribution.

Now revisit your initial hypothesis. For this dataset: Did the dataset behave as expected, or was another model (assumed distribution) a better fit to the dataset? Reflect on why the observed results may differ from your expectations.

4 Exponential Distribution Dataset

4.1 Part A: Developing Hypotheses

Identify and collect a real-world dataset that you hypothesize follows an Exponential distribution. Please be clear about the reasoning behind your hypothesis and be specific about the source of the dataset.

4.2 Part B: Fitting Distributions

For this exercise, we will call each of the four different theoretical distributions (normal, uniform, power law, exponential) a "model". Fit the dataset (i.e., estimate the model parameters) against each model (not just the one you hypothesized) using maximum likelihood estimation (or using any technique you think is appropriate; make sure to comment on the validity of your approach). This should result in a total of **4 parameter sets**. Report the estimated parameters in the following tabular format:

		Model			
Dataset	***# Observations***	**Normal**	**Uniform**	**Power law**	**Exponential**
Dataset 4	n_4	μ_4, σ_4	a_4, b_4	$\alpha_4, x_{\min_4}$	λ_4

Be sure to show the code you used to arrive at your final estimates clearly.

4.3 Part C: Comparing Real and Synthetic Data

For each fitted distribution (there will be 4 of them for this dataset, each corresponding to a different model), generate a synthetic sample of data points equal to the sample size of the real dataset using the respective model parameters you inferred from the real dataset.

Compare the real vs. synthetic data distributions using methods you think are the most appropriate, including visualizations. So, for this dataset, we compare the original dataset to four synthetic datasets, all with equal number of observations, but each synthetic dataset is generated using a different model.

For this dataset, identify the synthetic dataset (which corresponds to a model) that is most similar to the original data in terms of its distribution.

Now revisit your initial hypothesis. For this dataset: Did the dataset behave as expected, or was another model (assumed distribution) a better fit to the dataset? Reflect on why the observed results may differ from your expectations.