# Comparative results

V-RECS vs GPT-4

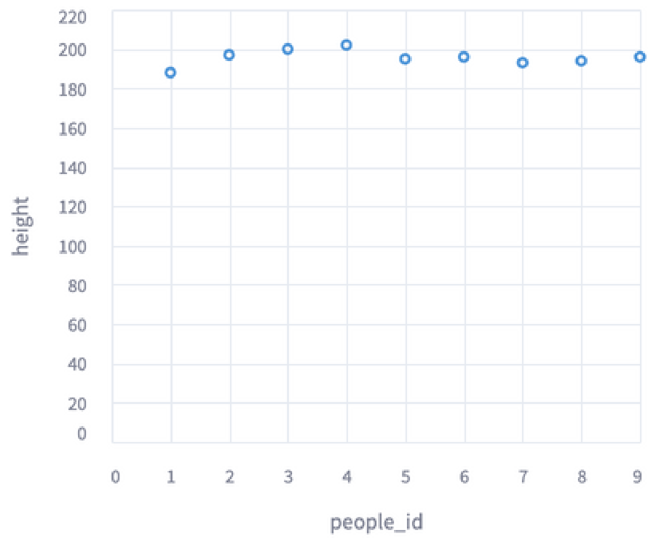
## Similar responses

**Query**

A bar chart for showing the number of the countries that have managers of age above 50 or below 46 , list in descending by the bars .

**Dataset Description**

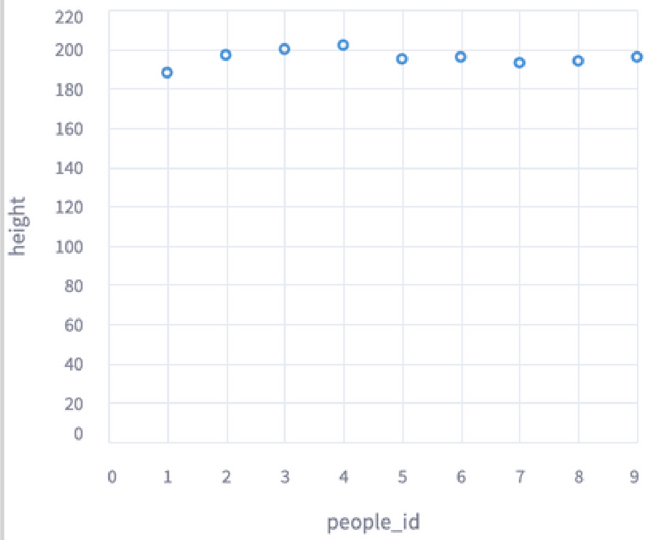
[('manager_id', 'numeric'), ('name', 'categorical'), ('country', 'categorical'), ('working_year_starts', 'categorical'), ('age', 'numeric'), ('level', 'numeric')]



The features selected, People_ID and Height, are the most relevant to the user's request. People_ID is a numeric value that uniquely identifies each individual in the dataset, and Height is a numeric value that represents the height of each individual. By plotting these two features against each other, we can visually inspect if there is any correlation between People_ID and Height. Other features like'sex', 'name', and 'date_of_birth' were not selected because they do not directly contribute to the understanding of the correlation between People_ID and Height. The chart is a point plot with 'people_id' on the x-axis and 'height' on the y-axis. There is no aggregation function applied. Other instructions to generate other data visualizations, based on the generated one, could include:

- Which person_id has the most number of people?
- Are there any person_ids with only one person?
- Is there a correlation between the number of people and the person_id?

V-RECS



The features selected are the best among all the others from the dataset based on the user instruction because the user specifically asked for a correlation between 'People_ID' and 'Height'. These are both quantitative variables, which are suitable for a scatter plot. The scatter plot visualizes the relationship between 'People_ID' and 'Height'. Each point on the plot represents a person, with their ID on the x-axis and their height on the y-axis. If there is a correlation between the two variables, it will be visible as a trend in the points. Other instructions to generate other data visualizations, based on the generated one, could be:

- A scatter chart showing the correlation between 'People_ID' and 'Weight'.
- A bar chart showing the distribution of 'Sex' in the dataset.
- A line chart showing the change in average 'Height' over time, if 'Date_of_Birth' is converted to a suitable format.

GPT-4

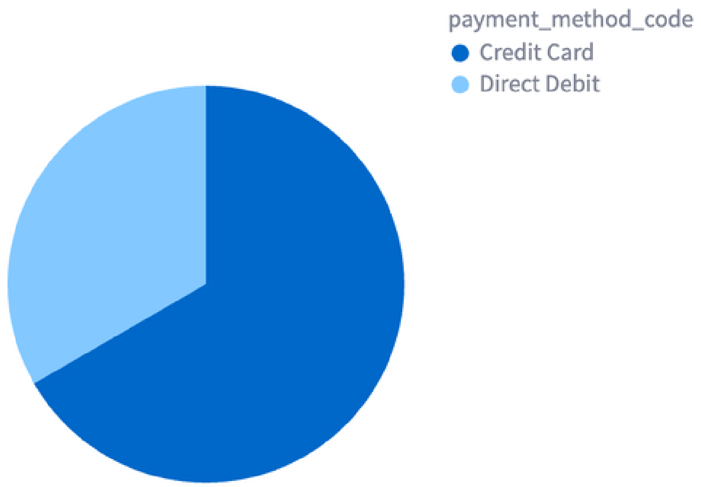
## Similar responses

**Query**

A pie chart about how many customers use each payment method ?

**Dataset Description**

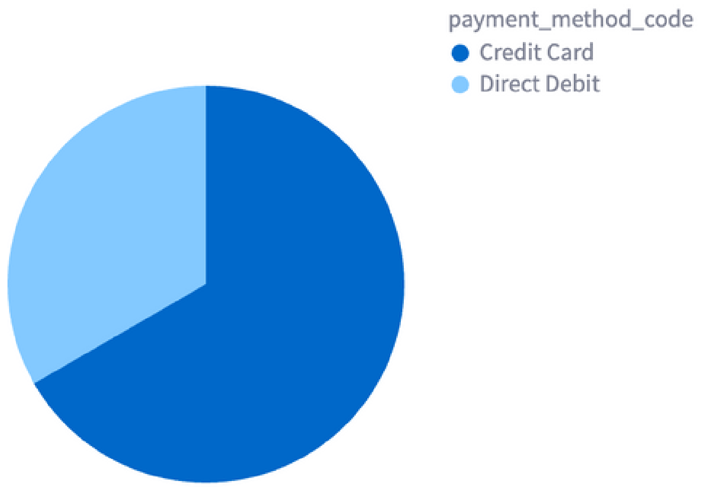
[('customer_id', 'numeric'), ('payment_method_code', 'categorical'), ('customer_code', 'categorical'), ('customer_name', 'categorical'), ('customer_address', 'categorical'), ('customer_phone', 'categorical'), ('customer_email', 'categorical')]



The 'payment_method_code' column is chosen because it directly corresponds to the user's interest in the payment methods used by customers. The 'count' aggregate function is used to quantify the number of customers using each method. The 'group' transform is applied to group the data by payment method, which allows us to calculate the count for each method. Other columns like 'customer_id', 'customer_code', 'customer_name', 'customer_address', 'customer_phone', and 'customer_email' are not selected as they do not directly contribute to the user's request. The visualization is an arc chart. The x-axis represents the 'payment_method_code', and the y-axis represents the count of 'payment_method_code'. The data is grouped by 'payment_method_code'. Other instructions to generate other data visualizations, based on the generated one, could include:

- Are there any payment methods that are significantly more popular than others?
- Are there any payment methods that are rarely used by customers?
- Is there a correlation between the popularity of a payment method and its convenience or security features?

**V-RECS**



The features selected are the best among all the others from the dataset based on the user instruction because 'customer_id' allows us to count the number of customers and 'payment_method_code' allows us to categorize the data by payment method. The visualization is a pie chart that shows the distribution of customers across different payment methods. Each slice of the pie represents a different payment method, and the size of the slice corresponds to the number of customers using that method. Other instructions to generate other data visualizations, based on the generated one, could be:

- A bar chart showing the number of customers using each payment method.
- A pie chart showing the distribution of customers across different customer codes.
- A bar chart showing the number of customers from each customer address.

**GPT-4**

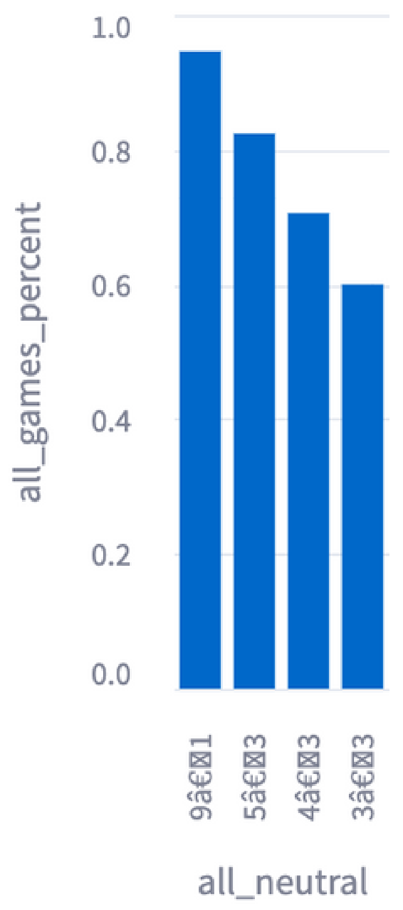
**Inverted Axes**

**Query**

Return a bar chart about the distribution of All Neutral and All_Games_Percent and sort by the y-axis in descending

**Dataset Description**

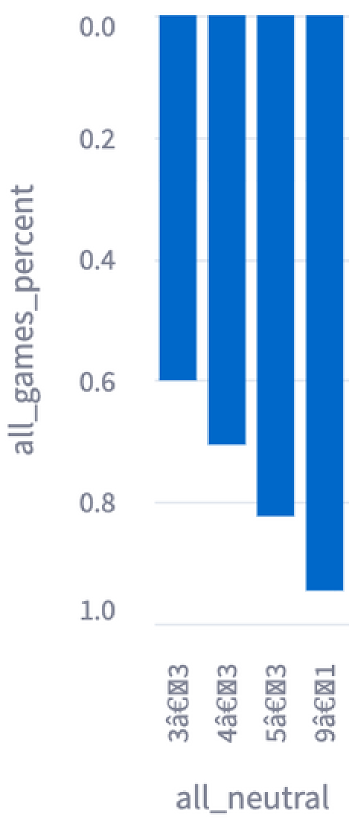
[('team_id', 'numeric'), ('school_id', 'numeric'), ('team_name', 'categorical'), ('acc_regular_season', 'categorical'), ('acc_percent', 'categorical'), ('acc_home', 'categorical'), ('acc_road', 'categorical'), ('all_games', 'categorical'), ('all_games_percent', 'numeric'), ('all_home', 'categorical'), ('all_road', 'categorical'), ('all_neutral', 'categorical')]



The user is interested in understanding the distribution of the 'all_neutral' and 'all_games_percent' variables. The 'all_neutral' variable represents the number of games played in neutral locations, while the 'all_games_percent' variable represents the percentage of all games played. These two variables can provide insights into the performance of the team in neutral locations and how it compares to all games played. The data is sorted in descending order on the y-axis to easily identify the highest values. The visualization is a bar chart where the x-axis represents 'all_neutral' and the y-axis represents 'all_games_percent'. There is no aggregation function applied. The bars are sorted in descending order based on the y-axis values. Other instructions to generate other data visualizations, based on the generated one, could include:

- What insights can be drawn from the percentage of neutral games played in the basketball match?
- Are there any noticeable trends in the percentage of neutral games played over time in the basketball match?

**V-RECS**



The features selected are the best among all the others from the dataset based on the user instruction because 'all_neutral' and 'all_games_percent' are the only ones that match the user's request. The user wants to see the distribution of 'all_neutral' and 'all_games_percent', and these are the only columns that can provide this information. The visualization is a bar chart that shows the distribution of 'all_neutral' and 'all_games_percent'. The x-axis represents the 'all_neutral' category, and the y-axis represents the 'all_games_percent'. The bars are sorted in descending order based on the y-axis, which means the bar with the highest 'all_games_percent' is at the top. Other instructions to generate other data visualizations, based on the generated one, could be:

- Return a line chart about the distribution of 'all_neutral' and 'all_games_percent', and sort by the y-axis in ascending order.
- Return a scatter plot about the distribution of 'all_neutral' and 'all_games_percent', and color the points by 'team_name'.
- Return a bar chart about the distribution of 'all_home' and 'all_games_percent', and sort by the y-axis in descending order.

**GPT-4**

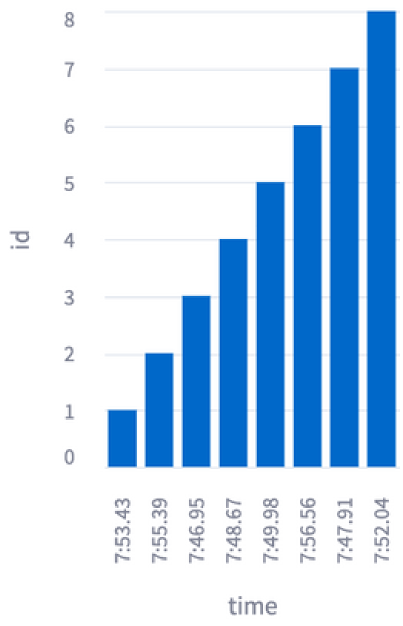
**Query**

Return a bar chart about the distribution of Time and ID , and sort y-axis in ascending order please .

**GPT Hallucination**

**Dataset Description**

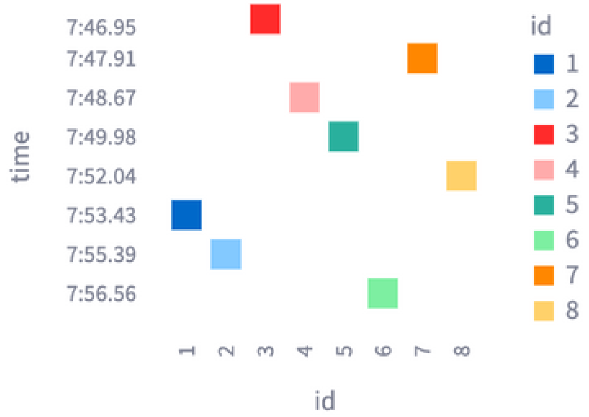
[('id', 'numeric'), ('name', 'categorical'), ('nationality', 'categorical'), ('meter_100', 'numeric'), ('meter_200', 'categorical'), ('meter_300', 'categorical'), ('meter_400', 'categorical'), ('meter_500', 'categorical'), ('meter_600', 'categorical'), ('meter_700', 'categorical'), ('time', 'categorical')]



The user is interested in understanding the distribution of Time and ID. The Time column is selected as it represents the time period, which is a key factor in understanding the distribution. The ID column is selected as it uniquely identifies each data point, allowing for a clear visualization of the distribution. The data is sorted in ascending order on the y-axis to make it easier to identify patterns and trends. The visualization is a bar chart with 'Time' on the x-axis and 'ID' on the y-axis. There is no aggregation function applied. The bars are sorted in ascending order based on the 'ID'. Other instructions to generate other data visualizations, based on the generated one, could include:

- Which swimmers consistently perform well or poorly?
- Are there any swimmers who are significantly faster or slower than the others?

V-RECS



The features selected are the best among all the others from the dataset based on the user instruction because the user specifically asked for a bar chart about the distribution of 'Time' and 'ID'. The other columns in the dataset are not relevant to this specific request. The visualization is a bar chart that shows the distribution of 'Time' and 'ID'. Each bar represents an 'ID' and the length of the bar corresponds to the 'Time'. The bars are sorted in ascending order based on 'Time'. Other instructions to generate other data visualizations, based on the generated one, could be:

- Return a line chart about the distribution of 'Time' and 'ID'.
- Return a scatter plot about the distribution of 'Time' and 'ID'.
- Return a bar chart about the distribution of 'Time' and 'ID', and sort y-axis in descending order.
- Return a bar chart about the distribution of 'Time' and 'ID', and color the bars based on 'ID'.

GPT-4

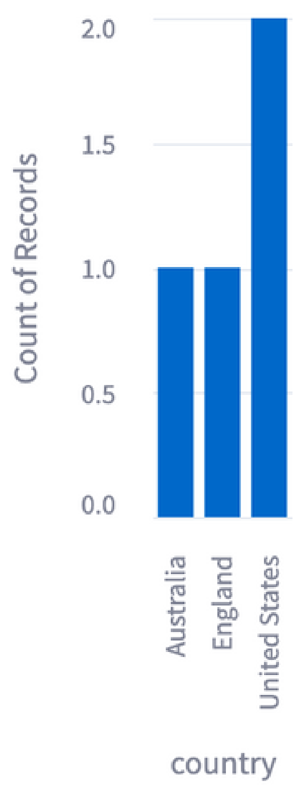
## Wrong ordering

**Query**

A bar chart for showing the number of the countries that have managers of age above 50 or below 46 , list in descending by the bars .

**Dataset Description**

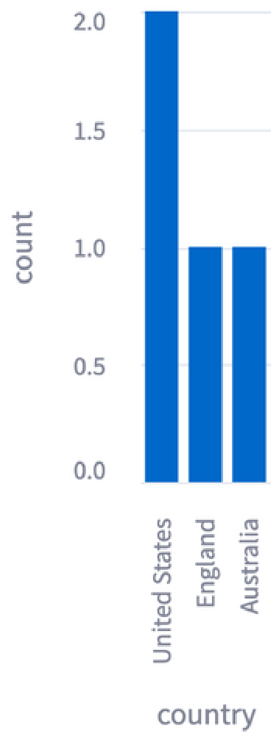
[('manager_id', 'numeric'), ('name', 'categorical'), ('country', 'categorical'), ('working_year_starts', 'categorical'), ('age', 'numeric'), ('level', 'numeric')]



The 'country' and 'age' columns are selected based on the user's request to see the distribution of countries based on the age of the managers. The 'count' aggregate function is used to count the number of countries that meet the age criteria. The data is then grouped by country and sorted in descending order to provide a clear view of the countries with the most managers. This is a bar chart where the x-axis represents different countries and the y-axis represents the count of each country. The data is filtered to include only those countries where the age is greater than 50 or less than 46. The bars are grouped by country and sorted in descending order. Other instructions to generate other data visualizations, based on the generated one, could include:

- What insights can be drawn from the distribution of countries with age greater than 50 or less than 46?
- Are there any countries that stand out in terms of the number of age greater than 50 or less than 46?

**V-RECS**



The features selected are the best among all the others from the dataset based on the user instruction because 'country' and 'age' are the only relevant columns to the user's request. The 'country' column is used to group the data, and the 'age' column is used to filter the data. The visualization is a bar chart that shows the number of managers in each country who are either above 50 or below 46 years old. The countries are listed in descending order by the number of such managers. Other instructions to generate other data visualizations, based on the generated one, could be:

- A bar chart showing the number of managers in each country that have a working year start above or below a certain year.
- A bar chart showing the number of managers in each country that are at a certain level.
- A bar chart showing the number of managers in each country that have a certain name.

**GPT-4**

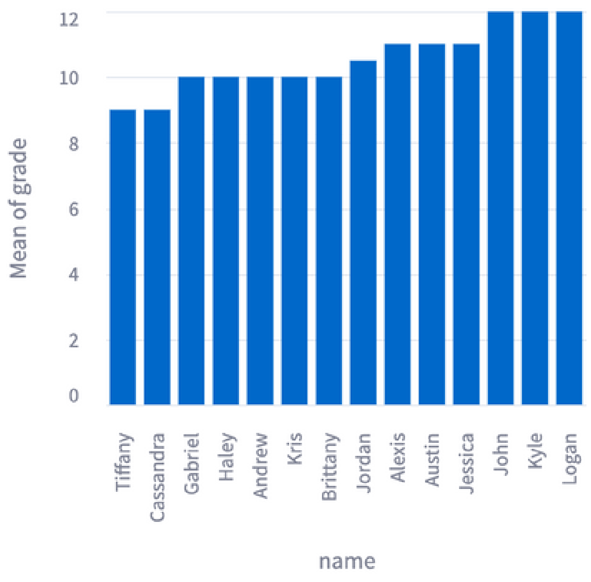
**Wrong ordering**

**Query**

Draw a bar chart for what are the names and the average grades for each high schooler ? , and list by the y axis from low to high .

**Dataset Description**

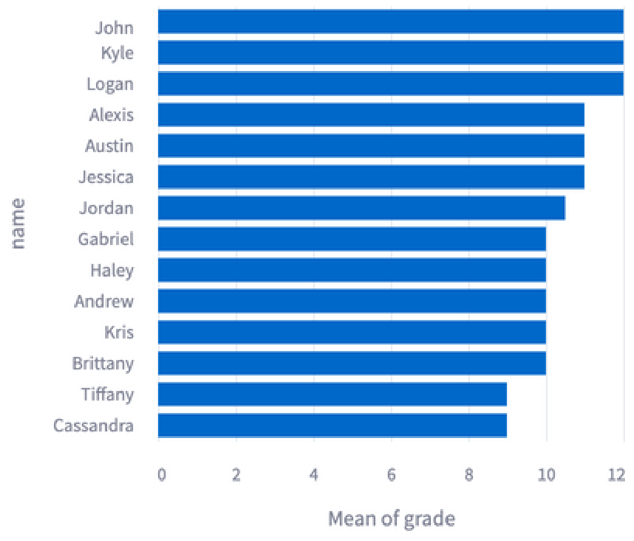
[('id', 'numeric'), ('name', 'categorical'), ('grade', 'numeric')]



The 'name' and 'grade' columns are selected because they directly answer the user's question. The 'name' column provides the names of the high schoolers, and the 'grade' column provides their grades. The aggregate function'mean' is used to calculate the average grade for each high schooler. The data is then sorted in ascending order to provide a clear view of the average grades from low to high. The visualization is a bar chart where the x-axis represents the 'name' and the y-axis represents the mean 'grade'. The bars are grouped by 'name' and sorted in ascending order based on 'grade'. Other instructions to generate other data visualizations, based on the generated one, could include:

- Who has the highest average grade among all highschoolers?
- Are there any highschoolers who have significantly lower average grades than others?

**V-RECS**



The 'name' and 'grade' columns are the best among all the others from the dataset based on the user instruction because the user wants to know the average grades for each high schooler. The 'name' column represents each high schooler and the 'grade' column represents their grades. The visualization is a bar chart where each bar represents a high schooler. The length of the bar corresponds to the average grade of the high schooler. The bars are sorted in ascending order of the average grade. Other instructions to generate other data visualizations, based on the generated one, could be:

- Draw a line chart to show the trend of the average grades over time.
- Draw a scatter plot to show the relationship between the grades and another variable, such as the number of hours studied per week.
- Draw a histogram to show the distribution of the grades.

**GPT-4**

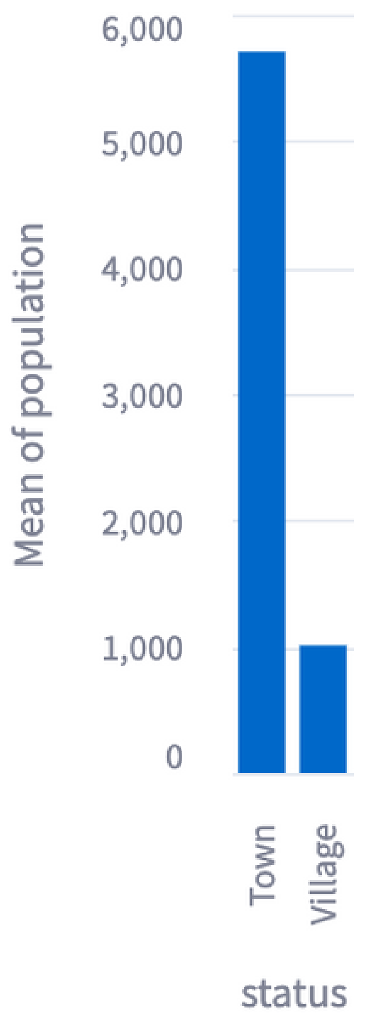
## GPT better then VRECS

**Query**

What are the statuses and average populations of each city Plot them as bar chart , and I want to order names from high to low order .

**Dataset Description**

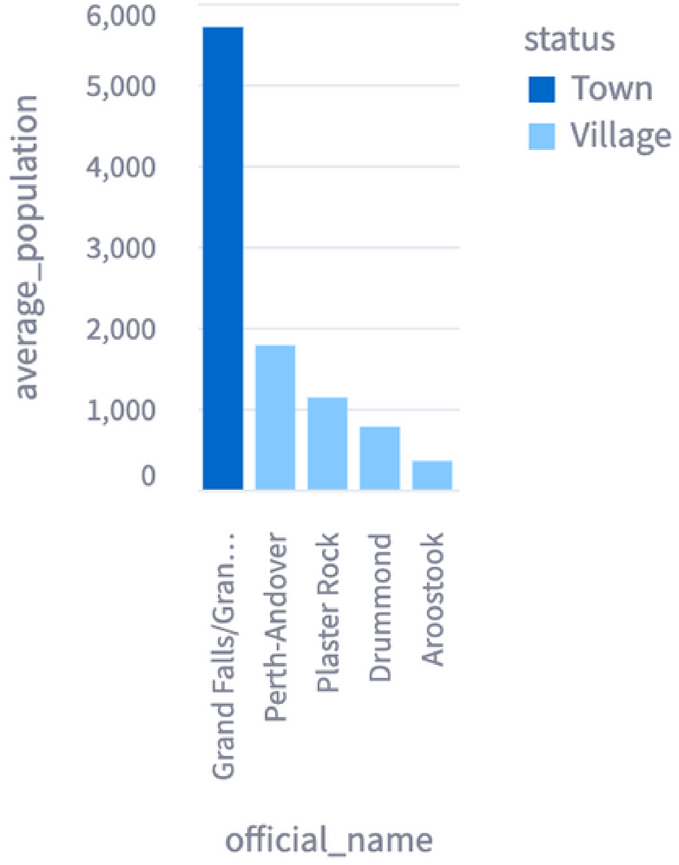
[('city_id', 'numeric'), ('official_name', 'categorical'), ('status', 'categorical'), ('area_km_2', 'numeric'), ('population', 'numeric'), ('census_ranking', 'categorical')]



The'status' and 'population' columns are selected because they directly answer the user's question. The'status' column provides the different statuses of the cities, and the 'population' column provides the population of each city. By calculating the mean population for each city and sorting the cities in descending order, we can provide the user with the information they are looking for. Other columns like 'city_id', 'official_name', 'area_km_2', and 'census_ranking' are not selected as they do not contribute to answering the user's question. This is a bar chart where the x-axis represents the 'Status' and the y-axis represents the mean 'Population'. The bars are grouped by 'Status' and sorted in descending order. Other instructions to generate other data visualizations, based on the generated one, could include:

- What insights can be drawn from the average population of each city?
- Are there any cities that stand out in terms of average population?
- How does the average population vary across different cities?

**V-RECS**



The features selected are the best among all the others from the dataset based on the user instruction because 'official_name' represents the city, 'status' provides the status of each city, and 'population' is used to calculate the average population of each city, which are all the information required by the user. The visualization is a bar chart where each bar represents a city. The length of the bar corresponds to the average population of the city, and the color of the bar indicates the status of the city. The cities are ordered from high to low based on their average population. Other instructions to generate other data visualizations, based on the generated one, could be:

- Plot the total area of each city as a bar chart, ordered from high to low.
- Plot the average population of each city status as a bar chart, ordered from high to low.
- Plot the number of cities in each status as a pie chart.

**GPT-4**

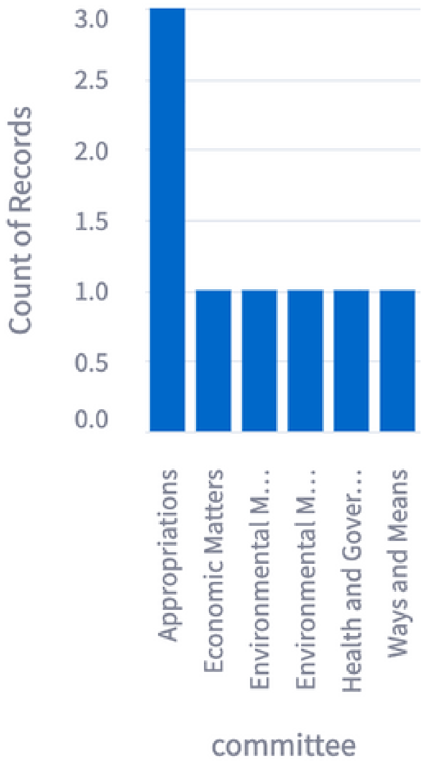
**VRECS better then GPT**

**Query**

Show the number of delegates in each committee with a bar chart , and display in descending by the y axis .

**Dataset Description**

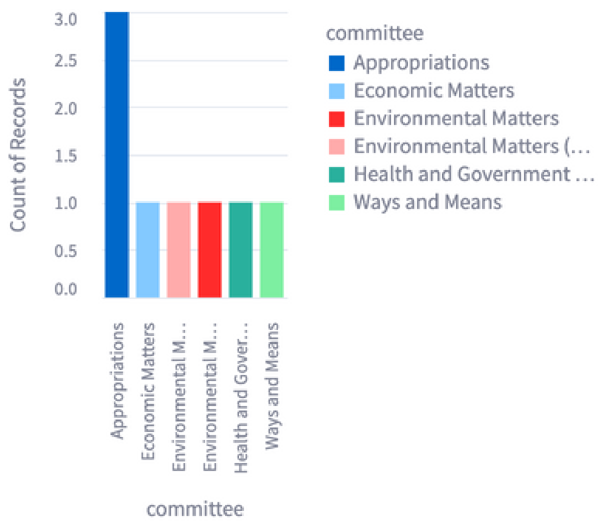
[('election_id', 'numeric'), ('counties_represented', 'categorical'), ('district', 'numeric'), ('delegate', 'categorical'), ('party', 'numeric'), ('first_elected', 'numeric'), ('committee', 'categorical')]



The 'committee' column is chosen because it directly corresponds to the user's request of showing the number of delegates in each committee. The 'count' aggregate function is used to calculate the number of delegates in each committee. The 'group' transform is used to group the data by committee, and the'sort' transform is used to sort the data in descending order by the count of delegates, as per the user's request. The other columns in the dataset, such as 'election_id', 'counties_represented', 'district', 'delegate', 'party', 'first_elected', are not relevant to the user's request and are therefore not selected. The visualization is a bar chart where each bar represents a different committee. The height of each bar corresponds to the count of each committee. The bars are sorted in descending order based on the count. Other instructions to generate other data visualizations, based on the generated one, could include:

- What insights can be drawn from the distribution of data points across different committees?
- Are there any committees that stand out in terms of the number of data points?
- Can any trends or patterns be identified in the distribution of data points across committees?

**V-RECS**



The 'delegate' and 'committee' columns are the best among all the others from the dataset based on the user instruction because they directly correspond to the user's request to show the number of delegates in each committee. The visualization is a bar chart where each bar represents a committee. The length of the bar corresponds to the number of delegates in that committee. The committees are sorted in descending order by the number of delegates. Other instructions to generate other data visualizations, based on the generated one, could be:

- Show the number of committees each delegate is part of with a bar chart, and display in descending order by the y axis.
- Show the distribution of the number of delegates in each committee with a histogram.
- Show the number of delegates in each committee with a pie chart.

**GPT-4**

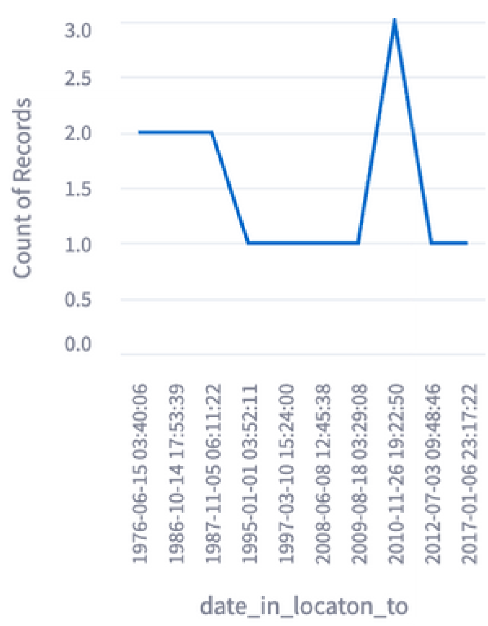
Last example

**Query**

Give me line charts of worldwide gross how many date in locaton to over year date in locaton to by major genres Location_Code , display from high to low by the x axis .

**Dataset Description**

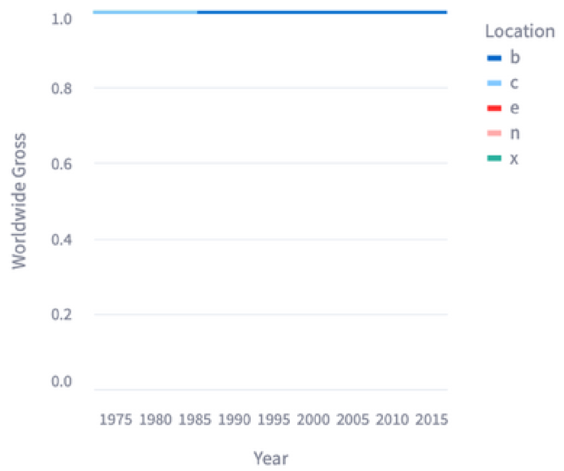
[('document_id', 'numeric'), ('location_code', 'categorical'), ('date_in_location_from', 'temporal'), ('date_in_locaton_to', 'temporal')]



The 'date_in_locaton_to' column is chosen because it provides the temporal aspect of the data, which is crucial for understanding trends over time. The 'count' aggregate function is used to quantify the number of dates in each location. The 'group' transform is applied to group the data by location, and the'sort' transform is used to sort the data in descending order, making it easier to identify locations with the highest number of dates. The visualization represents a line chart where the x-axis is the date_in_location_to and the y-axis is the count of date_in_location_to. The data is grouped by location_code and sorted in descending order by the date_in_location_to. Other instructions to generate other data visualizations, based on the generated one, could include:

- What trends can be observed in the number of documents over time?
- Are there any locations that stand out due to their high or low number of documents?
- How does the number of documents vary across different locations?

V-RECS



The features selected are the best among all the others from the dataset based on the user instruction because they directly correspond to the user's request. The user asked for a line chart of worldwide gross by date in location, sorted from high to low on the x-axis. The 'date_in_location_from' and 'date_in_locaton_to' columns provide the date information, and the 'location_code' column provides the location information. The visualization shows a line chart where the x-axis represents the years, and the y-axis represents the worldwide gross. Each line represents a different location, and the height of the line at any given point represents the worldwide gross for that location in that year. Other instructions to generate other data visualizations, based on the generated one, could include:

- Give me a bar chart of worldwide gross by location, sorted from high to low.
- Give me a scatter plot of worldwide gross by date in location, with each point colored by location.
- Give me a line chart of worldwide gross by year, with each line representing a different location.

GPT-4