

Consider the below JSON array of objects describing low-level analytic tasks (fundamental operations that users perform when interacting with data visualizations) as a list of their "Name", "Description" and "Pro Forma Abstract" (a concise summary outlining the main elements for a given natural language query), with "Examples" (example natural language queries about different datasets), "Attribute Data Types and Visual Encodings" (the data type of the column titles in the provided dataset along with the preferred visual encodings in the recommended visualization), "Attributes and Visual Encoding Descriptions", and "Recommended Visualizations":

```
[
{
  "Name": "Correlation",
  "Description": "Given a set of data cases and two attributes, determine useful relationships between the values of those attributes.",
  "Pro Forma Abstract": "What is the correlation between attributes X and Y over a given set S of data cases?",
  "Examples": ["Is there a correlation between carbohydrates and fat?", "Do different genders have a preferred payment method?", "What is the relationship between budget and gross?"],
  "Attribute Data Types and Visual Encodings":
  [{
    "X axis": "Quantitative",
    "Y axis": "Quantitative",
    "Other Encoding": "<choose from : [\"Nominal\", \"Ordinal\", \"Quantitative\", \"Temporal\"]>",
    "Sort": null,
    "Filter": null,
  }],
  "Attributes and Visual Encodings Description": "The \"X axis\" key indicates that the first attribute is used for the horizontal or x-axis of the visualization. Similarly the \"Y axis\" key is used for the vertical or y-axis of the visualization. The attributes in the \"Other Encoding\" key are optional and are to be used in encodings in color, shape, size, and opacity. The \"Sort\" and \"Filter\" keys should not be used in this task.",
  "Recommended Visualization":["Scatterplot"]
},
{
  "Name": "Derived Value",
  "Description": "Given a set of data cases, compute an aggregate numeric representation of those data cases.",
  "Pro Forma Abstract": "What is the value of aggregation function F over a given set S of data cases?",
  "Examples": ["What is the average calorie content of Post cereals?", "What is the gross income of all stores combined?", "How many manufacturers of cars are there?"],
  "Attribute Data Types and Visual Encodings":
  [{
    "X axis": "Nominal" or "Ordinal",
    "Y axis": "Quantitative",
    "Other Encoding": "<choose from : [\"Nominal\", \"Ordinal\", \"Quantitative\", \"Temporal\"]>",
    "Sort": null,
    "Filter": null,
  }],
  "Attributes and Visual Encodings Description": "The \"X axis\" indicates that the first attribute is used for the horizontal or x-axis of the visualization. Similarly the \"Y axis\" is used for the vertical or y-axis of the visualization. The attributes in the \"Other Encoding\" key are optional and are to be used in encodings in color, shape, size, and opacity. The \"Sort\" and \"Filter\" keys should not be used in this task.",
  "Recommended Visualization" : ["Bar Chart"]
},
{
  "Name": "Filter",
  "Description": "Given some concrete conditions on attribute values, find data cases satisfying those conditions.",
  "Pro Forma Abstract": "Which data cases satisfy conditions {A, B, C, ...}?",
```

"Examples": ["What Kellogg's cereals have high fiber?", "What comedies have won awards?", "Which funds underperformed the SP-500?", "Show the visualization only for films that have grossed more than 100 million dollars"],

"Attribute Data Types and Visual Encodings":

```
{
  "X axis": null
  "Y axis": null
  "Other Encoding": null,
  "Sort": null,
  "Filter": True,
}
```

"Attributes and Visual Encodings Description": "The \"X axis\", \"Y axis\", \"Other Encoding\", and \"Sort\" keys should not be used for this task. However, the \"Filter\" key should be used. As the \"Filter\" key is set to True, this indicates that the ensuing visualization must satisfy the conditions requested by the input natural language query. As such, the ensuing visualization must only display data points that satisfy these conditions. The attribute requested for the filter task can be any datatype (Quantitative, Nominal, Ordinal, or Temporal). If there is a \"Filter\" task detected in the input natural language query, please add the \"transform\" property in Vega-Lite to the Vega-Lite specification. This action will apply the filter specified in the natural language query. (Link to Vega-Lite \"transform\" property: <https://vega.github.io/vega-lite/docs/filter.html>)",

"Visualization Recommendation":["Line Chart", "Scatter Plot", "Strip Plot", "Histogram", "Bar Chart", "Heatmap"]

},

{

"Name": "Trend",

"Description": "Trend is the direction of the data over time, which may be increasing, decreasing, or flat",

"Pro Forma Abstract": "What is the direction of values for attribute(a) in the span of Time(t)?",

"Examples": ["Is there a trend of increasing film length over the years?", "How have production budgets changed over the years?"],

"Attribute Data Types and Visual Encodings":

```
{
  "X axis": "Temporal",
  "Y axis": "Quantitative",
  "Other Encoding": "<choose from : [\"Quantitative\", \"Nominal\", \"Ordinal\"]>",
  "Sort": null,
  "Filter": null,
}
```

"Attributes and Visual Encodings Description": "The \"X axis\" indicates that the first attribute is used for the horizontal or x-axis of the visualization. Similarly the \"Y axis\" is used for the vertical or y-axis of the visualization. Attributes in the curly brackets {} are optional and are to be used in the following encodings: color, shape, size, and opacity. The \"Sort\" and \"Filter\" keys are not to be used for this task.",

"Visualization Recommendation":["Line Chart"]

},

{

"Name": "Distribution",

"Description": "Given a set of data cases and a quantitative attribute of interest, characterize the distribution of that attribute's values over the set",

"Pro Forma Abstract": "What is the distribution of values of attribute A in a set S of data cases?",

"Examples": ["What is the age distribution of shoppers?", "What is the distribution of carbohydrates in cereals?", "What is the count of movies for each genre?"],

"Attribute Data Types and Visual Encodings":

```
{
  "X axis": "Quantitative" or "Nominal" or "Ordinal",
  "Y axis": "Quantitative",
  "Other Encoding": "<choose from : [\"Nominal\", \"Ordinal\", \"Quantitative\", \"Temporal\"]>",
  "Sort": null,
}
```

```

    "Filter": null,
  }],
  "Attributes Description": "The \"X axis\" indicates that the first attribute is used for the horizontal or x-axis of the visualization. Similarly the \"Y axis\" is used for the vertical or y-axis of the visualization. Attributes in the curly brackets {} are optional and are to be used in encodings in color, shape, size, and opacity.",
  "Visualization Recommendation":["Histogram"]
},
{
  "Name": "Sort",
  "Description": "Given a set of data cases, rank them according to some ordinal metric.",
  "Pro Forma Abstract": "What is the sorted order of a set S of data cases according to their value of attribute A?",
  "Examples": ["Order the cars by weight", "Rank the cereals by calories."],
  "Attribute Data Types and Visual Encodings":
  [{
    "X axis": null
    "Y axis": null
    "Other Encoding": null,
    "Sort": True
    "Filter": null
  }],
  "Attributes and Visual Encodings Description": "The \"X axis\", \"Y axis\", \"Other Encoding\", and \"Filter\" keys should not be used for this task. However, the \"Sort\" key should be used. As the \"Sort\" key is set to True, this indicates that the ensuing visualization must satisfy the order/ranking requested by the input natural language query. The attribute requested for the sort task must be of the Quantitative datatype.",
  "Visualization Recommendation":["Bar Chart"]
},
{
  "Name": "Find Extremum",
  "Description": "Find data cases possessing an extreme value of an attribute over its range within the data set.",
  "Pro Forma Abstract": "What are the top/bottom N data cases with respect to attribute A?",
  "Examples": ["What is the car with the highest MPG?", "What director/film has won the most awards?", "What Robin Williams film has the most recent release date?"],
  "Attribute Data Types and Visual Encodings":
  [{
    "X axis": null
    "Y axis": null
    "Other Encoding": null,
    "Sort": True
    "Filter": null
  }],
  "Attributes and Visual Encodings Description": "The \"X axis\", \"Y axis\", \"Other Encoding\", and \"Filter\" keys should not be used for this task. However, the \"Sort\" key should be used. As the \"Sort\" key is set to True, this indicates that the ensuing visualization must satisfy the order/ranking requested by the input natural language query. The attribute requested for the sort task must be of the Quantitative datatype.",
  "Visualization Recommendation":["Bar Chart"]
}
]`

```

Also consider the below JSON array of objects describing low-level operations for follow-up queries as a list of their "Name", "Instruction", and "Examples":

```

[
{
  "Name": "Add Attribute",

```

```

"Instruction": "Given a previous analytic specification, add the attribute that is detected in the input natural language query to the previous specification.",
"Examples": ["Add genre", "Group by genre", "Put genre as well"]
},
{
  "Name": "Remove Attribute",
  "Instruction": "Given a previous analytic specification, remove the attribute that is detected in the input natural language query to the previous specification.",
  "Examples": ["Remove genre", "Show only production budget"]
},
{
  "Name": "Replace Attribute",
  "Instruction": "Given a previous analytic specification, replace the attribute that is detected in the input natural language query and is also in the previous analytic specification with the attribute that is detected in the natural language query, but is not in the previous specification.",
  "Examples": ["Replace gross with imdb rating", "Show gross instead of imdb rating"]
},
{
  "Name": "Add Task",
  "Instruction": "Given a previous analytic specification, add the task that is detected in the input natural language query to the previous specification. The visualization specification should reflect this new task that is added from the natural language query. Also please keep the previous tasks from the previous specification in the new modified analytic specification.",
  "Examples": ["Sort by worldwide gross in the descending order", "Show only horror movies", "What are their average production budgets?"]
},
{
  "Name": "Remove Task",
  "Instruction": "Given a previous analytic specification, remove the task that is detected in the input natural language query to the previous specification.",
  "Examples": ["Undo the sorting", "Undo the average"]
},
{
  "Name": "Replace Task",
  "Instruction": "Given a previous analytic specification, replace the attribute that is detected in the input natural language query and is also in the previous analytic specification with the attribute that is detected in the natural language query, but is not in the previous specification. Also please keep the other tasks from the previous specification that will not be replaced in the new modified analytic specification.",
  "Examples": ["Replace average with sum", "Show me total instead of average", "What about the lowest instead of highest?"]
},
{
  "Name": "Replace Vis",
  "Instruction": "Given a previous analytic specification, replace the visualization type that is detected in the input natural language query and is also in the previous analytic specification with the visualization type that is detected in the natural language query, but is not in the previous specification.",
  "Examples": ["Replace the vis with a bar chart", "As a line chart now", "Show a tick plot instead"]
}
]

```

If a field called ****PREVIOUS ANALYTIC SPECIFICATION**** appears below the natural language query, classify the below natural language query into the respective follow-up operations they map to. Utilize the previous analytic specification (including the attributeMap, taskMap, and visList) and modify this specification to reflect the changes specified and requested in the natural language query. Return the visualization type in the form of a Vega-Lite specification where it reads data from url: < " INSERT DATASET URL HERE " >.

However, if there is no field called PREVIOUS ANALYTIC SPECIFICATION that is below the natural language query, then using the above definitions, classify the below natural language queries into the respective analytic tasks they map to. There can be one or more analytic tasks detected in the input natural language query. Return the visualization type in the form of a Vega-Lite specification where it reads data from the url above. PLEASE ensure that the schema used in your Vega-Lite specification is <"INSERT VEGA-LITE SCHEMA URL HERE">.

Here's a subset of the original dataset with actual columns and rows for reference.

< "INSERT DATASET COLUMNS AND FIRST 10 ROWS HERE" >

If there is no PREVIOUS ANALYTIC SPECIFICATION that appears below the query, detect any attributes, tasks, and visualizations in the dataset that the provided query references, and place the detected dataset columns in the attributeMap, taskMap and visList property of the JSON below. Each Query can have more than one task and visualization type they can map to. Each property in the "attributeMap" JSON should be populated with the extracted dataset column (e.g. "Worldwide Gross"). There can be multiple attributes, tasks, and visualizations that are detected, but make sure that each attribute, task, and visualization in the attributeMap, taskMap, and visList is unique. Put each attribute into the attributeMap, task into taskMap, and Visualization into visList JSON. There can also be multiple possible visualization specifications as well. For each possible visualization specification, you can include a dictionary that has the required contents to the "visList" list.

Furthermore, note that the input natural language (NL) query can also use ambiguous language with partial references to data attributes. In these cases, the attributeMap also includes an "isAmbiguous" field. The field can either be True or False. Set the field to True if the queryPhrase could refer to other attributes in the dataset. Otherwise set the field to False. If there are ambiguous attributes detected, generate Vega-Lite specifications for each ambiguous attribute that encode the ambiguous attribute in the visualization. Furthermore, if there are no tasks detected in the NL query, infer the task that is best suited with the detected attributes' datatypes. Generate a visualization specification using this inferred task and detected attributes.

Here is the JSON object that the response should be returned as if and only if there is no PREVIOUS ANALYTIC SPECIFICATION attached to the natural language query:

```
{
  "query": <Add NL query that is being parsed here>,
  "query_raw": <Add NL query that is being parsed here>,
  "dataset": <Add dataset URL here>,
  "visList": [
    {
      "attributes": [List of dataset attributes detected],
      "queryPhrase": [<Keywords found in query that were used to detect the taskMap and the recommended visualization>],
      "visType": "None",
      "tasks": [Add the list of tasks detected here. For example, list "Correlation","Derived Value","Filter","Distribution" etc. ],
      "inferenceType": <Can be one of two values: "Explicit" or "Implicit". Set the value to "Explicit" if the visualization's "queryPhrase" explicitly references a visualization type. Otherwise set the value to "Implicit".>
      "vISpec": <Add the Vega-Lite specification of the visualization recommended here.>
    }
  ],
  "attributeMap": {
    <Dataset column that was detected (should have same value as "name")>: {
      "name": <Dataset column that was detected>,
      "queryPhrase": [<Keywords found in query that were used to detect the dataset attribute>]
      "encode": <Boolean value depending on if the attribute appears on either of the axes or color in the Vega-Lite specification. The boolean value should be output as true or false in all lowercase letters.>
    }
  }
}
```

```

},
"metric": <[Can be one of two values: "attribute_exact_match" or "attribute_similarity_match". Set the
value to "attribute_exact_match" if the attribute was found directly in the query. Set the value to
"attribute_similarity_match" if the query uses a synonym for the attribute.]>,
"inferenceType": <Can be one of two values: "Explicit" or "Implicit". Set the value to "Explicit" if the
attribute's "queryPhrase" references an attribute name. Set the value to "Implicit" if the queryPhrase
directly references values found in the attribute's values.>
"isAmbiguous": <Can be either True or False. Set the field to True if the queryPhrase could refer other
attributes in the dataset. Otherwise set the field to False.>
"ambiguity": [<Populate this list with all the different attributes in the dataset that the queryPhrase can
refer to if isAmbiguous is set to True. Otherwise keep this list empty.
],
"taskMap": {
  <Task that was detected>: [
    {
      "task": <Task that was detected>,
      "queryPhrase": [<Keywords found in query that were used to detect the task>],
      "values": [<If the "Filter" task was detected, put the filter value in here>],
      "attributes": [<Populate with the attributes that the task is mapped to>],
      "operator": "<Can be one of "IN", "GT", "EQ", "AVG", "SUM". "GT" is greater than. "EQ" is equals. "GT"
and "EQ" are used for quantitative filters. "IN" is used for nominal filters. "AVG" and "SUM" are used for
derived value tasks. "SUM" is for summation and "AVG" is for average. Keep the string empty
otherwise.>"
      "inferenceType": <Can be one of two values: "Explicit" or "Implicit". Set the value to "Explicit" if the
"queryPhrase" directly requests for a task to be applied. Set the value to "Implicit" if the task is derived
implicitly from the "queryPhrase".>
    }
  ]
}
"visType": <Put the visualization type that was explicitly specified in the query. Put the string "None" if vis
type was not specified in the query.>,
"visQueryPhrase": <Keywords found in query that were used to detect the visType. Put the string "None" if
vis type was not specified in the query.>
}

```

If there is a PREVIOUS ANALYTIC SPECIFICATION provided, then utilize it (including the attributeMap, taskMap, and visList) and modify this specification to reflect the changes that are specified and requested in the natural language query. Then return this modified JSON object as output. The Vega-Lite specification should also change and a new visualization should be generated as a result of the modification. Also, in the modified JSON object, include a new property called "followupType". This "followupType" should include the classified follow-up operation (add attribute, replace task, etc.) that was detected in the natural language query.

Please do not add any extra prose to your response. I only want to see the JSON output.