

# PyGWalker

PyGWalker (Python binding of Graphic Walker) can simplify your Jupyter Notebook data analysis and data visualization workflow. It allows data scientists to analyze data and visualize patterns with simple drag-and-drop operations.

- Install PyGWalker  
pip install pygwalker

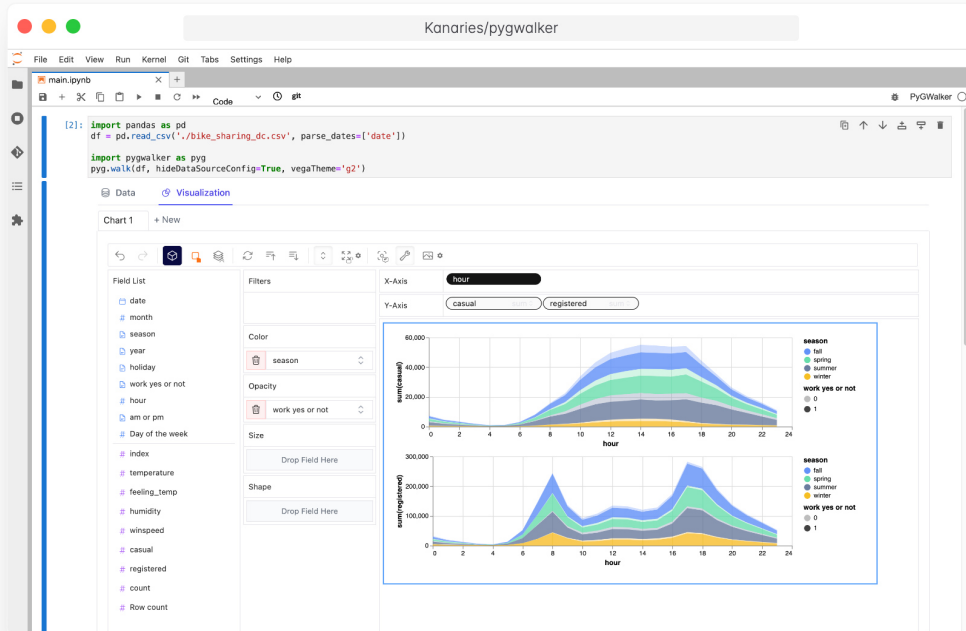
- Importing libraries  
import pandas as pd  
import pygwalker as pyg

- Loading data  
df = pd.read\_csv(r"Superstore\_USA.csv")

- Now take DataFrame into PyGWalker  
pyg.walk(df, vegaTheme = "vega")  
★ Or you can store the dataframe into another variable as per below example  
new\_df = pyg.walk(df, vegaTheme = "vega")

# PyGWalker

Turn your pandas dataframe into a Tableau-style User Interface for visual analysis



## <<< PyGWalker: A Python Library for Exploratory Data Analysis with Visualization >>>

PyGWalker can simplify your Jupyter Notebook data analysis and data visualization workflow, by turning your pandas dataframe into a Tableau-style User Interface for visual exploration.

PyGWalker (pronounced like "Pig Walker", just for fun) is named as an abbreviation of "Python binding of Graphic Walker". It integrates Jupyter Notebook (or other jupyter-based notebooks) with Graphic Walker, a different type of open-source alternative to Tableau. It allows data scientists to analyze data and visualize patterns with simple drag-and-drop operations.

## Install PyGWalker

# <<< PyGWalker: A Python Library for Exploratory Data Analysis with Visualization >>>

PyGWalker can simplify your Jupyter Notebook data analysis and data visualization workflow, by turning your pandas dataframe into a Tableau-style User Interface for visual exploration.

PyGWalker (pronounced like "Pig Walker", just for fun) is named as an abbreviation of "Python binding of Graphic Walker". It integrates Jupyter Notebook (or other jupyter-based notebooks) with Graphic Walker, a different type of open-source alternative to Tableau. It allows data scientists to analyze data and visualize patterns with simple drag-and-drop operations.

## Install PyGWalker

```
In [ ]: 1 pip install pygwalker
```

## Importing libraries

```
In [23]: 1 import pandas as pd
          2 import pygwalker as pyg
```

## Checking directories

In [24]: 1 `dir(pyg)`

Out[24]: ['GWalker',  
          '\_\_builtins\_\_',  
          '\_\_cached\_\_',  
          '\_\_doc\_\_',  
          '\_\_file\_\_',  
          '\_\_loader\_\_',  
          '\_\_name\_\_',  
          '\_\_package\_\_',  
          '\_\_path\_\_',  
          '\_\_spec\_\_',  
          'base',  
          'gwalker',  
          'to\_html',  
          'utils',  
          'walk']

## Loading data

In [25]:

1

df = pd.read\_csv(r'Superstore\_USA.csv')

2

df.head()

Out[25]:

|   | Row ID | Order Priority | Discount | Unit Price | Shipping Cost | Customer ID | Customer Name   | Ship Mode   | Customer Segment | Product Category | ... | Region  | State or Province | City      | Postal Code | Order Date | Ship Date  |
|---|--------|----------------|----------|------------|---------------|-------------|-----------------|-------------|------------------|------------------|-----|---------|-------------------|-----------|-------------|------------|------------|
| 0 | 18606  | Not Specified  | 0.01     | 2.88       | 0.50          | 2           | Janice Fletcher | Regular Air | Corporate        | Office Supplies  | ... | Central | Illinois          | Addison   | 60101       | 28-05-2012 | 30-05-2012 |
| 1 | 20847  | High           | 0.01     | 2.84       | 0.93          | 3           | Bonnie Potter   | Express Air | Corporate        | Office Supplies  | ... | West    | Washington        | Anacortes | 98221       | 07-07-2010 | 08-07-2010 |
| 2 | 23086  | Not Specified  | 0.03     | 6.68       | 6.15          | 3           | Bonnie Potter   | Express Air | Corporate        | Office Supplies  | ... | West    | Washington        | Anacortes | 98221       | 27-07-2011 | 28-07-2011 |
| 3 | 23087  | Not Specified  | 0.01     | 5.68       | 3.60          | 3           | Bonnie Potter   | Regular Air | Corporate        | Office Supplies  | ... | West    | Washington        | Anacortes | 98221       | 27-07-2011 | 28-07-2011 |
| 4 | 23088  | Not Specified  | 0.00     | 205.99     | 2.50          | 3           | Bonnie Potter   | Express Air | Corporate        | Technology       | ... | West    | Washington        | Anacortes | 98221       | 27-07-2011 | 27-07-2011 |

5 rows × 24 columns

◀

▶

Now take DataFrame into PyGWalker

```
In [26]: 1 pyg.walk(df, vegaTheme = 'vega') #You can view the dataframe in a table and configure the analytic types and semantic types  
2 # Or you can store dataframe into another variable as per below  
3 # new_df = pyg.walk(df, vegaTheme = 'vega')
```

Chart 1 + New



Field List

- Order Priority
- Customer Name
- Ship Mode
- Customer Segment
- Product Category
- Product Sub-Category
- Product Container
- Product Name
- Region
- State or Province
- City
- Order Date
- Ship Date

- # index
- # Row ID
- # Discount
- # Unit Price
- # Shipping Cost
- # Customer ID
- # Product Base Margin
- # Postal Code
- # Profit
- # Quantity ordered new
- # Sales
- # Order ID
- # Row count

Filters

Color

Customer Segment

Opacity

Drop Field Here

Size

Drop Field Here

Shape

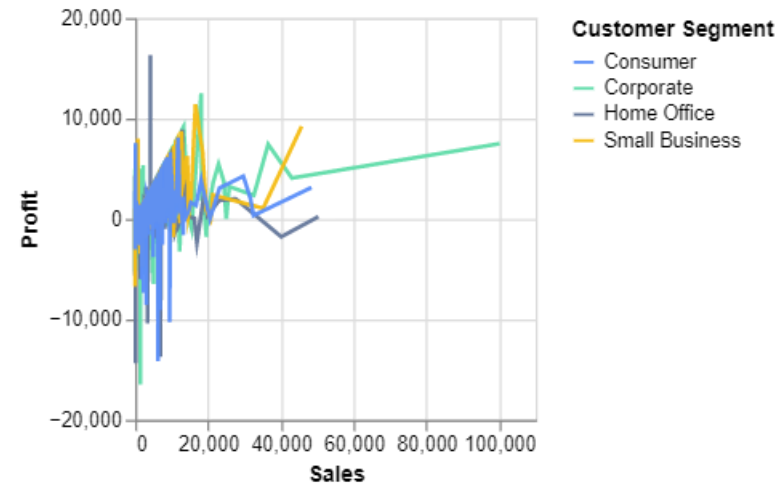
Drop Field Here

X-Axis

Sales

Y-Axis

Profit



**Now visualize as per requirements**



```
In [27]: 1 pyg.walk(df, vegaTheme = 'vega')
```



## Field List

- Order Priority
- Customer Name
- Ship Mode
- Customer Segment
- Product Category
- Product Sub-Category
- Product Container
- Product Name
- Region
- State or Province
- City
- Order Date
- Ship Date

- # index
- # Row ID
- # Discount
- # Unit Price
- # Shipping Cost
- # Customer ID
- # Product Base Margin
- # Postal Code
- # Profit
- # Quantity ordered new
- # Sales
- # Order ID
- # Row count

## Filters

Drop Field Here

Opacity

Shipping Cost

sum

## Size

Drop Field Here

## Shape

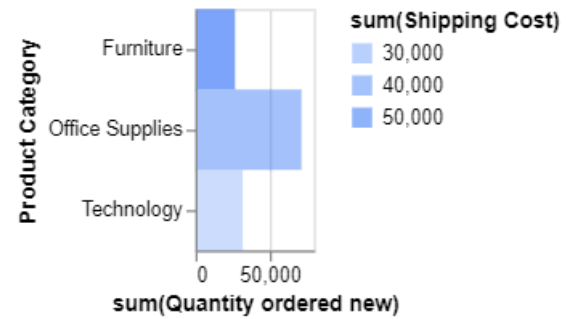
Drop Field Here

X-Axis

Quantity ordered new sum

Y-Axis

## Product Category





```
In [28]: 1 pyg.walk(df, vegaTheme = 'vega')
```



- Field List
- Order Priority

Customer Name

Ship Mode

Customer Segment

Product Category

Product Sub-Category

Product Container

Product Name

Region

State or Province

City

Order Date

Ship Date
- # index

# Row ID

# Discount

# Unit Price

# Shipping Cost

# Customer ID

# Product Base Margin

# Postal Code

# Profit

# Quantity ordered new

# Sales

# Order ID

# Row count

Filters

Color

Product Category

Opacity

Drop Field Here

Size

Drop Field Here

Shape

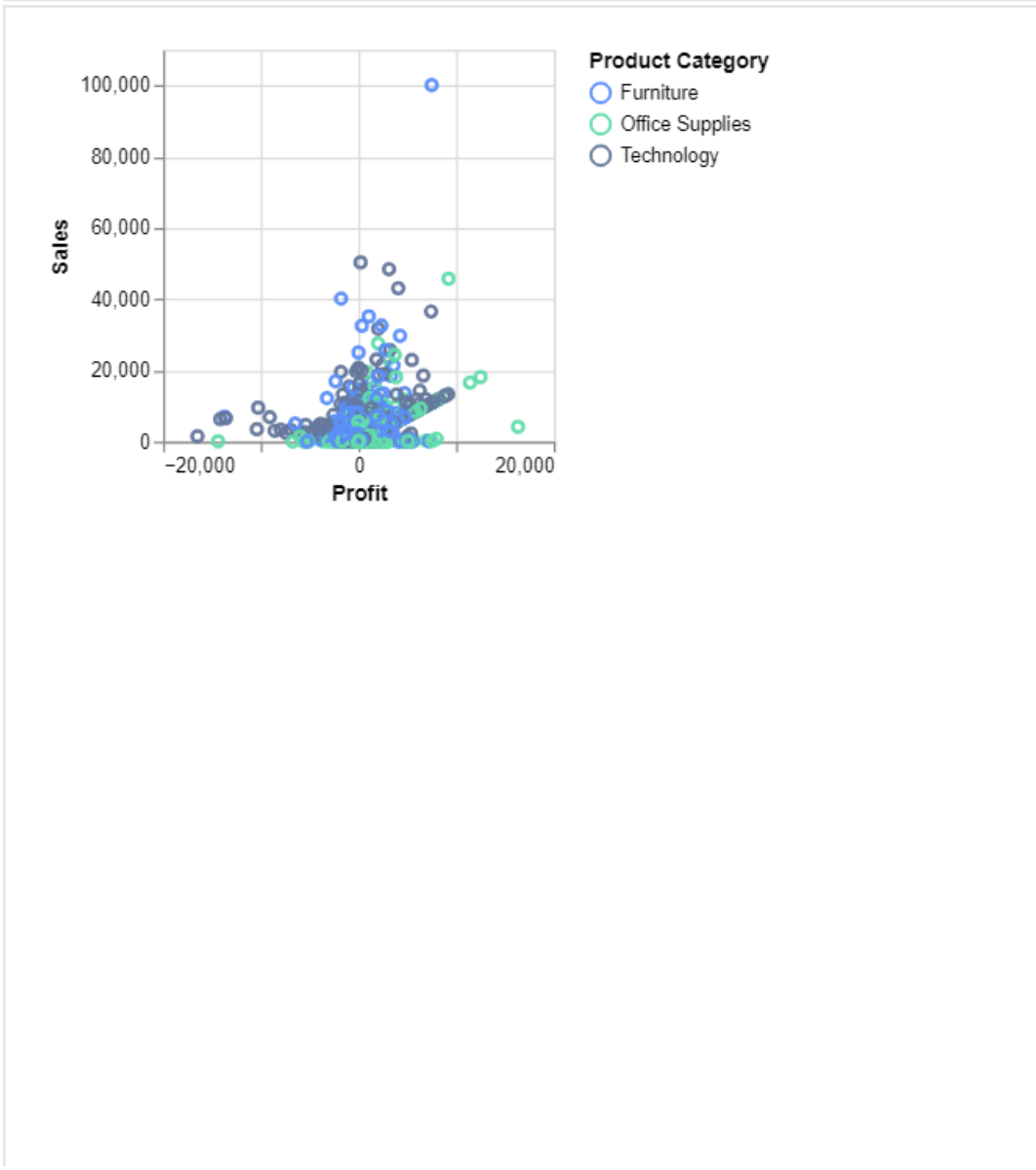
Drop Field Here

X-Axis

Profit

Y-Axis

Sales

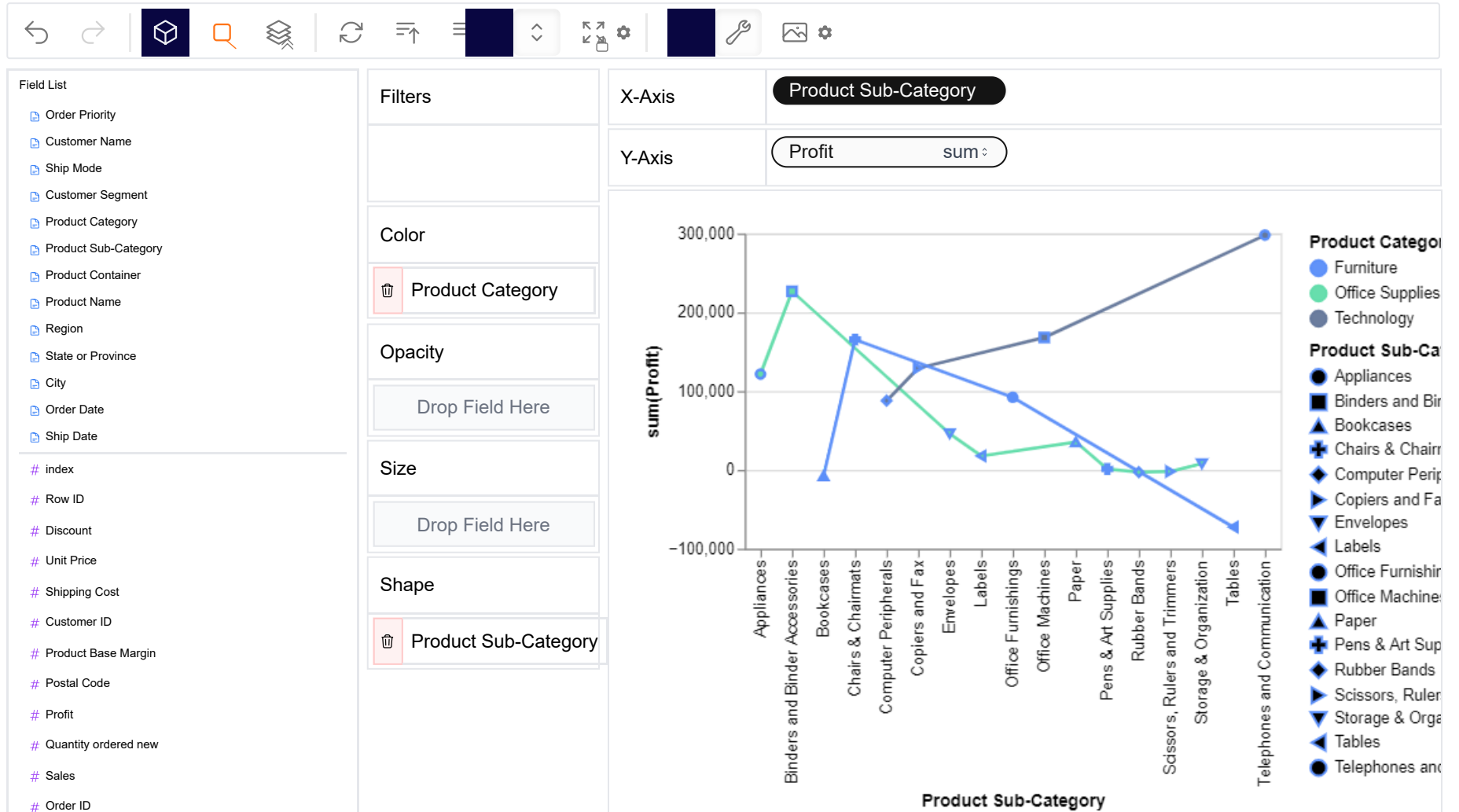




```
In [29]: 1 pyg.walk(df, vegaTheme = 'vega')
```

### Chart 1

+ New





**Upload another datasets**

In [30]:

```
1 df_df = pd.read_csv(r'vix-daily_csv.csv')
2 pyg.walk(df_df, vegaTheme = 'vega')
```

Chart 1

+ New



Field List

Date

# index  
# VIXOpen  
# VIXHigh  
# VIXLow  
# VIXClose  
# Row count

Filters

Color

Drop Field Here

Opacity

Drop Field Here

Size

Drop Field Here

Shape

Drop Field Here

X-Axis

index

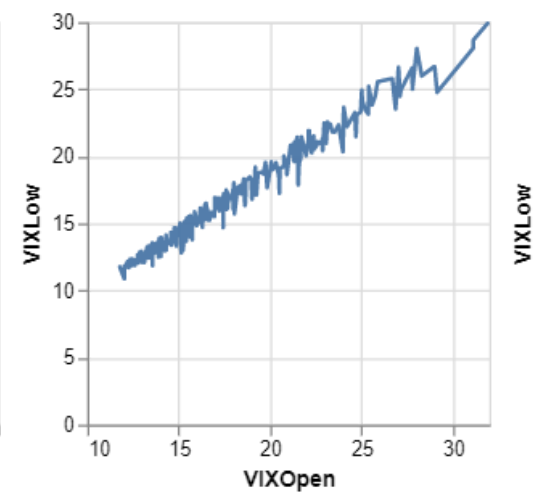
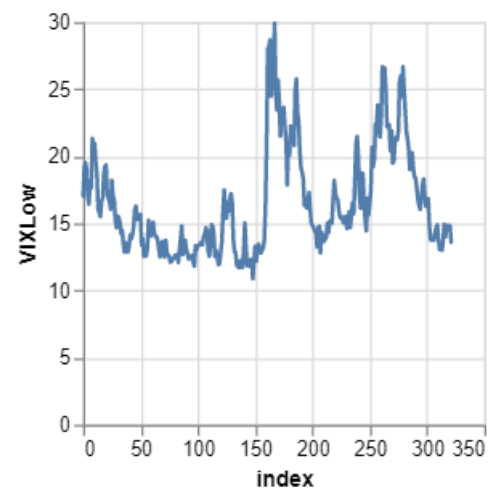
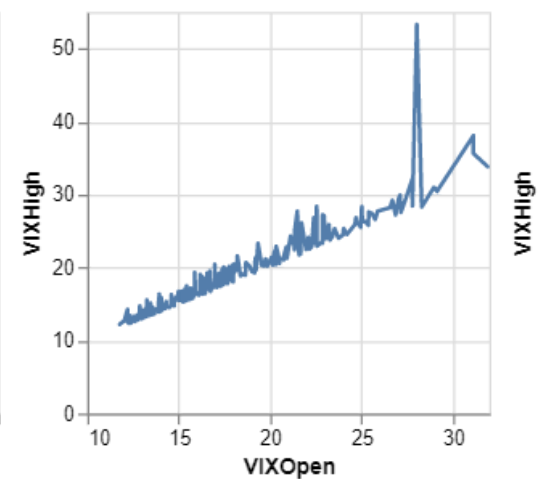
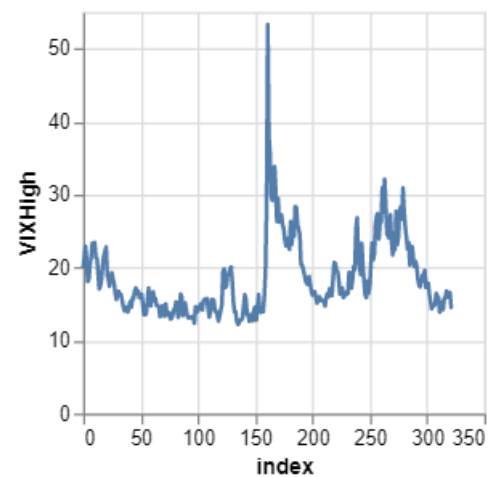
VIXClose

VIXOpen

Y-Axis

VIXHigh

VIXLow





In [31]:

```
1 df1 = pd.read_csv(r'FitBit.csv')  
2 pyg.walk(df1, vegaTheme = 'vega')
```



- Field List
- ActivityDate
- 
- # index
  - # Id
  - # TotalSteps
  - # TotalDistance
  - # TrackerDistance
  - # LoggedActivitiesDistance
  - # VeryActiveDistance
  - # ModeratelyActiveDistance
  - # LightActiveDistance
  - # SedentaryActiveDistance
  - # VeryActiveMinutes
  - # FairlyActiveMinutes
  - # LightlyActiveMinutes
  - # SedentaryMinutes
  - # Calories
  - # Row count

Filters

Color

VeryActiveMinutes

Opacity

Drop Field Here

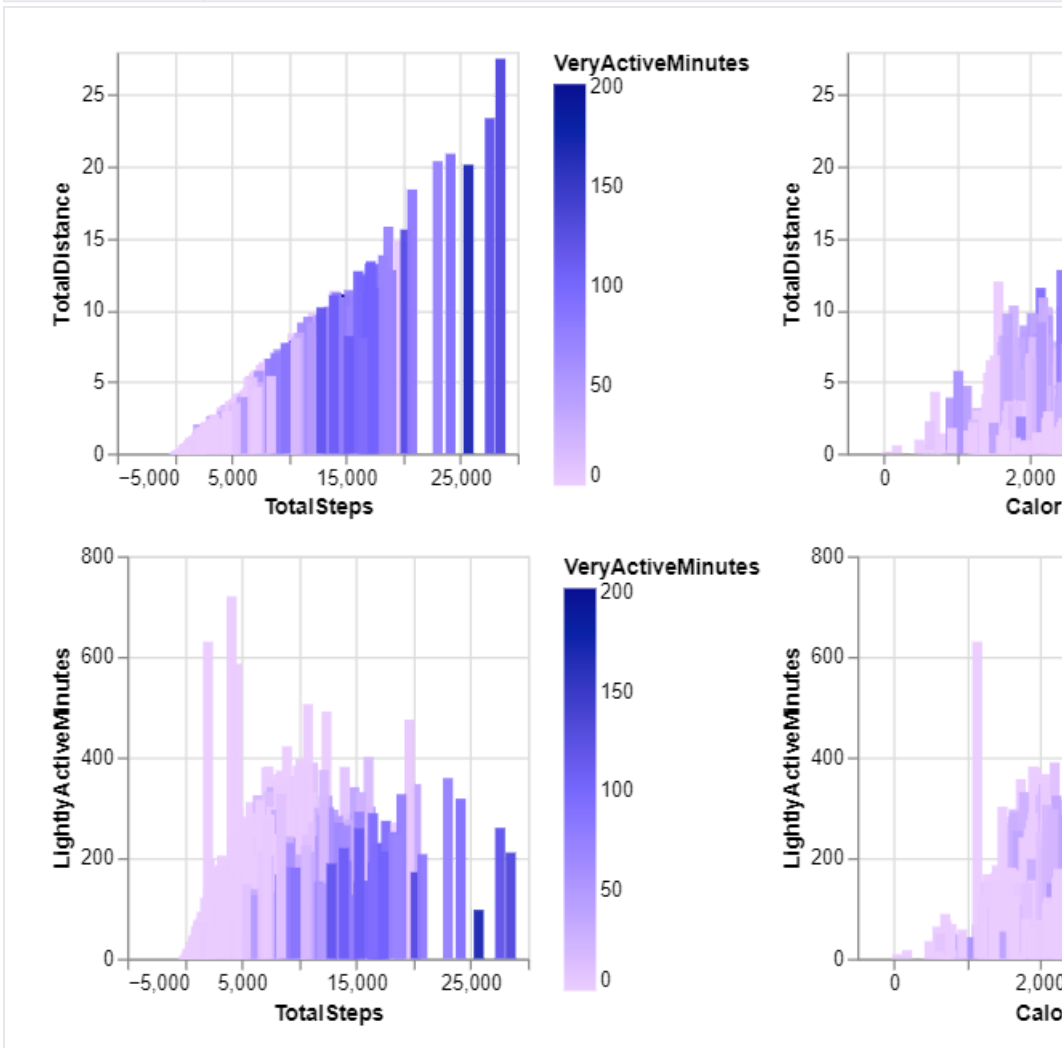
Size

Drop Field Here

Shape

Drop Field Here

|        |               |                      |
|--------|---------------|----------------------|
| X-Axis | TotalSteps    | Calories             |
| Y-Axis | TotalDistance | LightlyActiveMinutes |





```
In [32]: 1 pyg.walk(df1, vegaTheme = 'vega')
```



Chart 1

+ New



- Field List
- ActivityDate
- 
- # index
- # Id
- # TotalSteps
- # TotalDistance
- # TrackerDistance
- # LoggedActivitiesDistance
- # VeryActiveDistance
- # ModeratelyActiveDistance
- # LightActiveDistance
- # SedentaryActiveDistance
- # VeryActiveMinutes
- # FairlyActiveMinutes
- # LightlyActiveMinutes
- # SedentaryMinutes
- # Calories
- # Row count

Filters

Color

Drop Field Here

Opacity

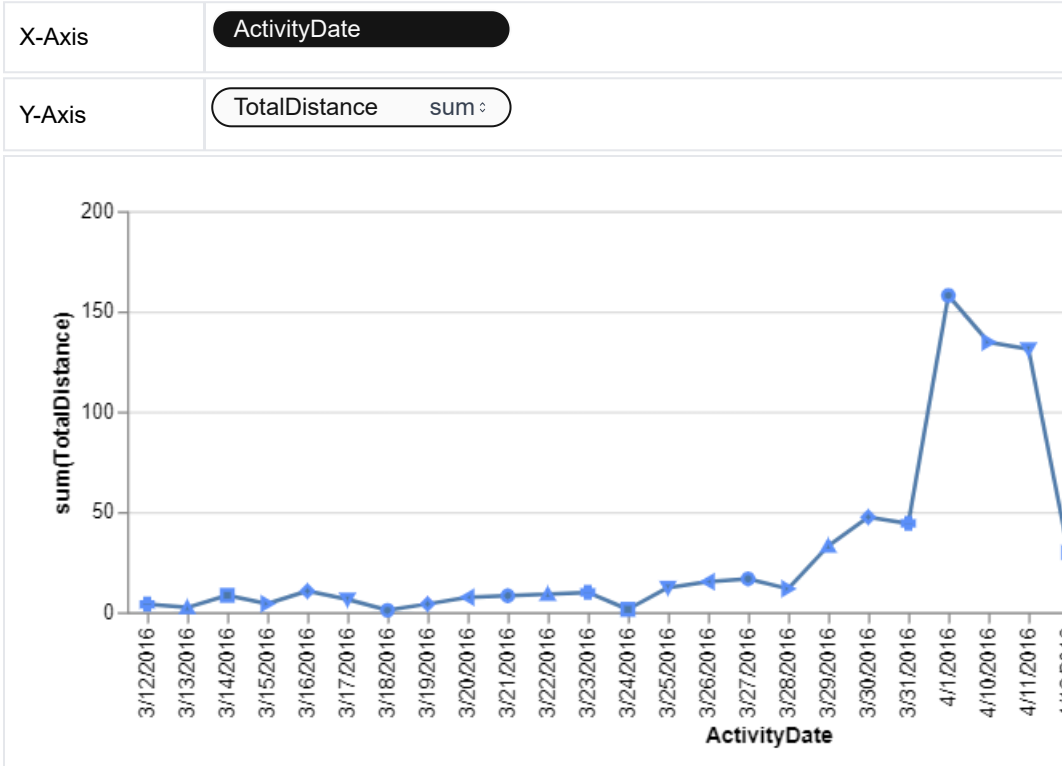
Drop Field Here

Size

Drop Field Here

Shape

TotalSteps sum ◌



**Upload another datasets**

In [33]:


```
1 import xlrd
2 df2 = pd.read_excel(r'Data_Train.xlsx')
3 pyg.walk(df2, vegaTheme = 'vega')
```

+ New




## Field List

 Airline

 Date\_of\_Journey

 Source

 Destination

 Route

Dep\_Time

 Arrival\_Time

 **Duration**

 Total\_Stops

 Additional\_Info

# index

# Price

# Row count

## Filters

Color

 Total\_Stops

Opacity

Drop Field Here

Size

Drop Field Here

Shape

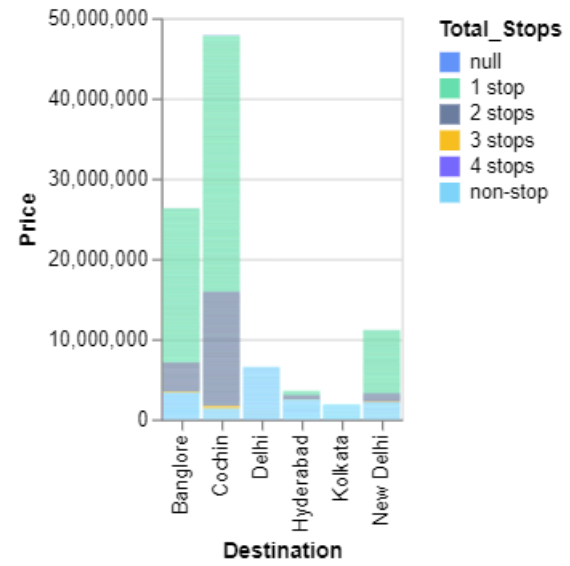
Drop Field Here

X-Axis

Destination

Y-Axis

Price





In [34]:

```
1  
2 df2 = pd.read_excel(r'Data_Train.xlsx')  
3 pyg.walk(df2, vegaTheme = 'vega')
```

Chart 1

+ New



## Field List

Airline  
Date\_of\_Journey  
Source  
Destination  
Route  
Dep\_Time  
Arrival\_Time  
Duration  
Total\_Stops  
Additional\_Info

# index  
# Price  
# Row count

## Filters

## Color

Destination

## Opacity

Drop Field Here

## Size

Drop Field Here

## Shape

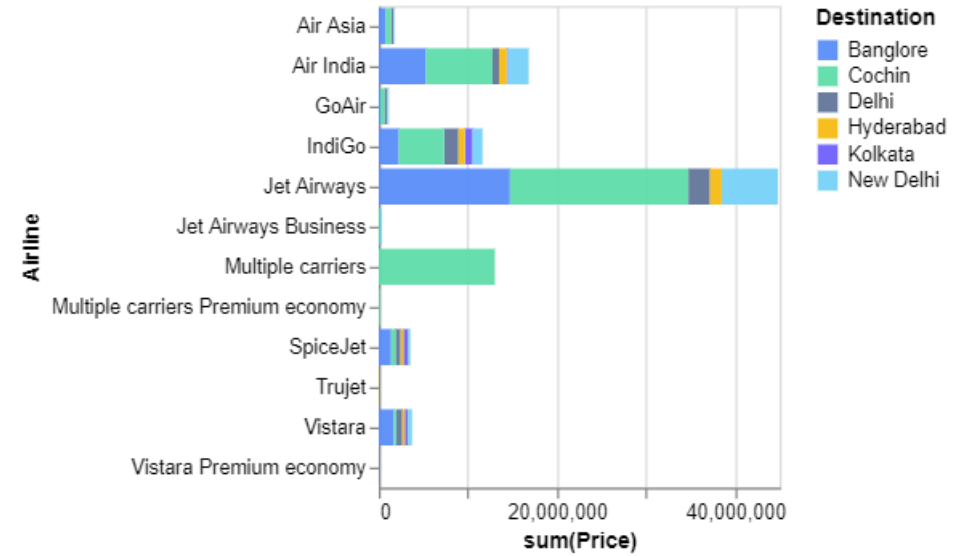
Drop Field Here

X-Axis

Price sum

Y-Axis

Airline



**Upload another datasets**



In [35]:

```
1 df3 = pd.read_csv(r'AgentLoggingReport.csv')  
2 pyg.walk(df3, vegaTheme = 'vega')
```
















- Field List
-  Agent

 Date

 Login Time

 Logout Time

 Duration

- # index
- # SL No
- # Row count

Filters

Color

Drop Field Here

Opacity

Drop Field Here

Size

Drop Field Here

Shape

Drop Field Here

X-Axis

SL No

Y-Axis

Date

