```
!pip install pandas plotly dash
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: plotly in /usr/local/lib/python3.12/dist-packages (5.24.1)
Collecting dash
  Downloading dash-3.2.0-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.12/dist-packages (from plotly) (8.5.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from plotly) (25.0)
Requirement already satisfied: Flask<3.2,>=1.0.4 in /usr/local/lib/python3.12/dist-packages (from dash) (3.1.2)
Requirement already satisfied: Werkzeug<3.2 in /usr/local/lib/python3.12/dist-packages (from dash) (3.1.3)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.12/dist-packages (from dash) (8.7.0)
Requirement already satisfied: typing-extensions>=4.1.1 in /usr/local/lib/python3.12/dist-packages (from dash) (4.15.0)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from dash) (2.32.4)
Collecting retrying (from dash)
  Downloading retrying-1.4.2-py3-none-any.whl.metadata (5.5 kB)
Requirement already satisfied: nest-asyncio in /usr/local/lib/python3.12/dist-packages (from dash) (1.6.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from dash) (75.2.0)
Requirement already satisfied: blinker>=1.9.0 in /usr/local/lib/python3.12/dist-packages (from Flask<3.2,>=1.0.4->dash) (1.9.0)
Requirement already satisfied: click>=8.1.3 in /usr/local/lib/python3.12/dist-packages (from Flask<3.2,>=1.0.4->dash) (8.3.0)
Requirement already satisfied: itsdangerous>=2.2.0 in /usr/local/lib/python3.12/dist-packages (from Flask<3.2,>=1.0.4->dash) (2.2.0)
Requirement already satisfied: jinja2>=3.1.2 in /usr/local/lib/python3.12/dist-packages (from Flask<3.2,>=1.0.4->dash) (3.1.6)
Requirement already satisfied: markupsafe>=2.1.1 in /usr/local/lib/python3.12/dist-packages (from Flask<3.2,>=1.0.4->dash) (3.0.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Requirement already satisfied: zipp>=3.20 in /usr/local/lib/python3.12/dist-packages (from importlib-metadata->dash) (3.23.0)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->dash) (3.4.3)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests->dash) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->dash) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests->dash) (2025.10.5)
Downloading dash-3.2.0-py3-none-any.whl (7.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.9/7.9 MB 68.9 MB/s eta 0:00:00
Downloading retrying-1.4.2-py3-none-any.whl (10 kB)
Installing collected packages: retrying, dash
Successfully installed dash-3.2.0 retrying-1.4.2
```

```
import pandas as pd
import plotly.express as px
from dash import Dash, dcc, html, Input, Output
```

```
from google.colab import files
uploaded = files.upload()
```

Choose files SampleSuperstore.csv

SampleSuperstore.csv(text/csv) - 2287806 bytes, last modified: 16/10/2025 - 100% done

```
Saving SampleSuperstore.csv to SampleSuperstore (1).csv
```

```
df = pd.read_csv("SampleSuperstore.csv", encoding='ISO-8859-1')
```

```
app = Dash(__name__)
app.title = "Sales Insights Dashboard"
```

```
app.layout = html.Div([
    html.H1("Sales Insights Dashboard - Sample Superstore", style={'textAlign': 'center'}),

    html.Div([
        html.Label("Select Category:"),
        dcc.Dropdown(
            options=[{'label': cat, 'value': cat} for cat in df['Category'].unique()],
            value=df['Category'].unique()[0],
            id='category-filter'
        )
    ], style={'width': '30%', 'display': 'inline-block'}),

    dcc.Graph(id='bar-chart'),
    dcc.Graph(id='line-chart'),
    dcc.Graph(id='scatter-plot'),
    dcc.Graph(id='map-chart')
])
```

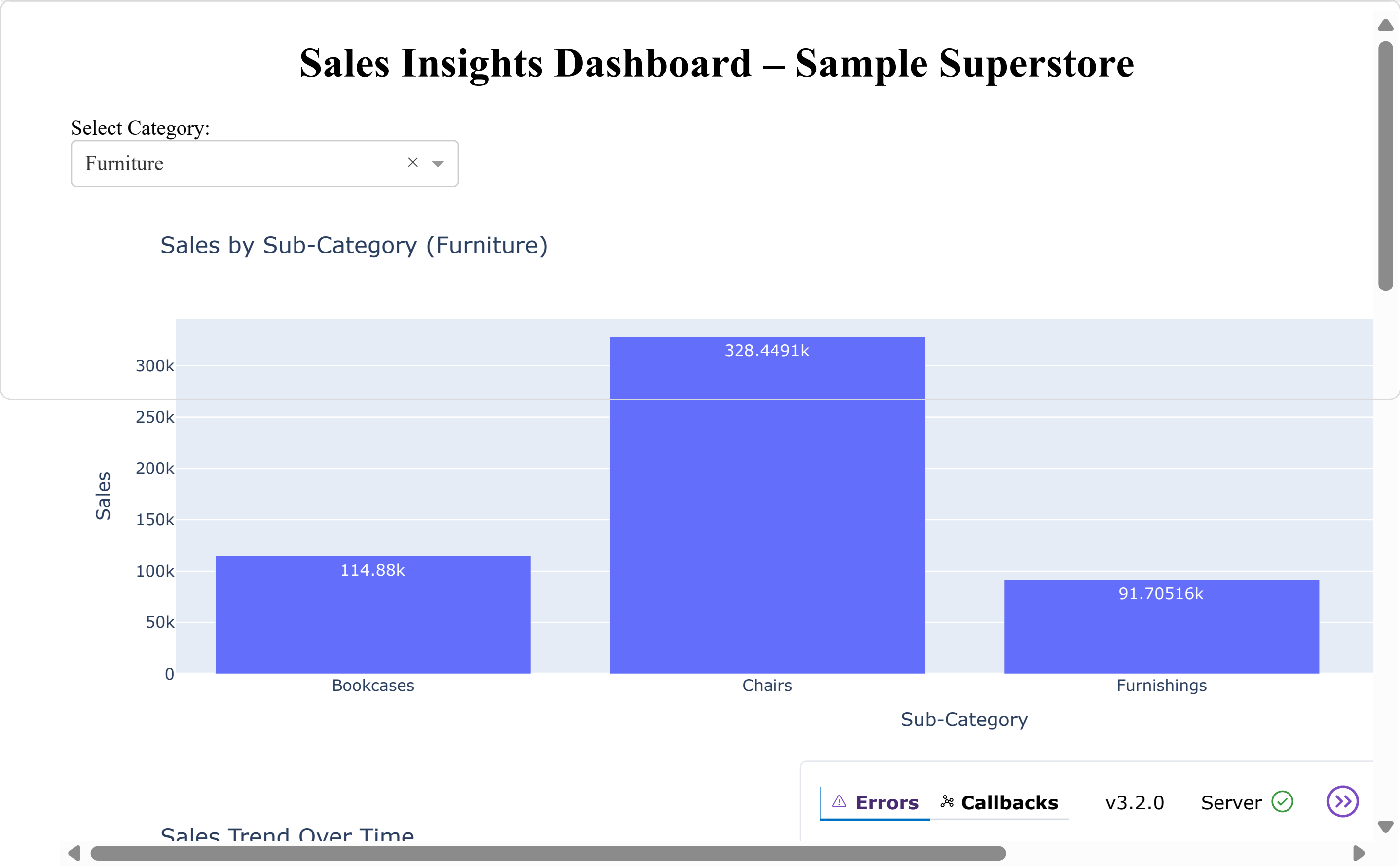
```
@app.callback(
    Output('bar-chart', 'figure'),
    Input('category-filter', 'value')
)
def update_bar_chart(selected_category):
    filtered_df = df[df['Category'] == selected_category]
    fig = px.bar(filtered_df.groupby('Sub-Category')['Sales'].sum().reset_index(),
                 x='Sub-Category', y='Sales', title=f"Sales by Sub-Category ({selected_category})",
                 text_auto=True)
    return fig
```

```
@app.callback(
    Output('line-chart', 'figure'),
    Input('category-filter', 'value')
)
def update_line_chart(selected_category):
    df['Order Date'] = pd.to_datetime(df['Order Date'])
    filtered_df = df[df['Category'] == selected_category]
    fig = px.line(filtered_df.groupby('Order Date')['Sales'].sum().reset_index(),
                  x='Order Date', y='Sales', title="Sales Trend Over Time")
    return fig
```

```
@app.callback(
    Output('scatter-plot', 'figure'),
    Input('category-filter', 'value')
)
def update_scatter_plot(selected_category):
    filtered_df = df[df['Category'] == selected_category]
    fig = px.scatter(filtered_df, x='Sales', y='Profit', color='Region',
                     title="Sales vs Profit by Region", hover_data=['Sub-Category'])
    return fig
```

```
@app.callback(
    Output('map-chart', 'figure'),
    Input('category-filter', 'value')
)
def update_map_chart(selected_category):
    filtered_df = df[df['Category'] == selected_category]
    fig = px.choropleth(filtered_df.groupby('State')['Sales'].sum().reset_index(),
                        locations='State', locationmode='USA-states', color='Sales',
                        scope='usa', title="Sales by State")
    return fig
```

```
if __name__ == '__main__':
    app.run(debug=True)
```



Sales Insights Dashboard – Sample Superstore

Select Category:

Furniture