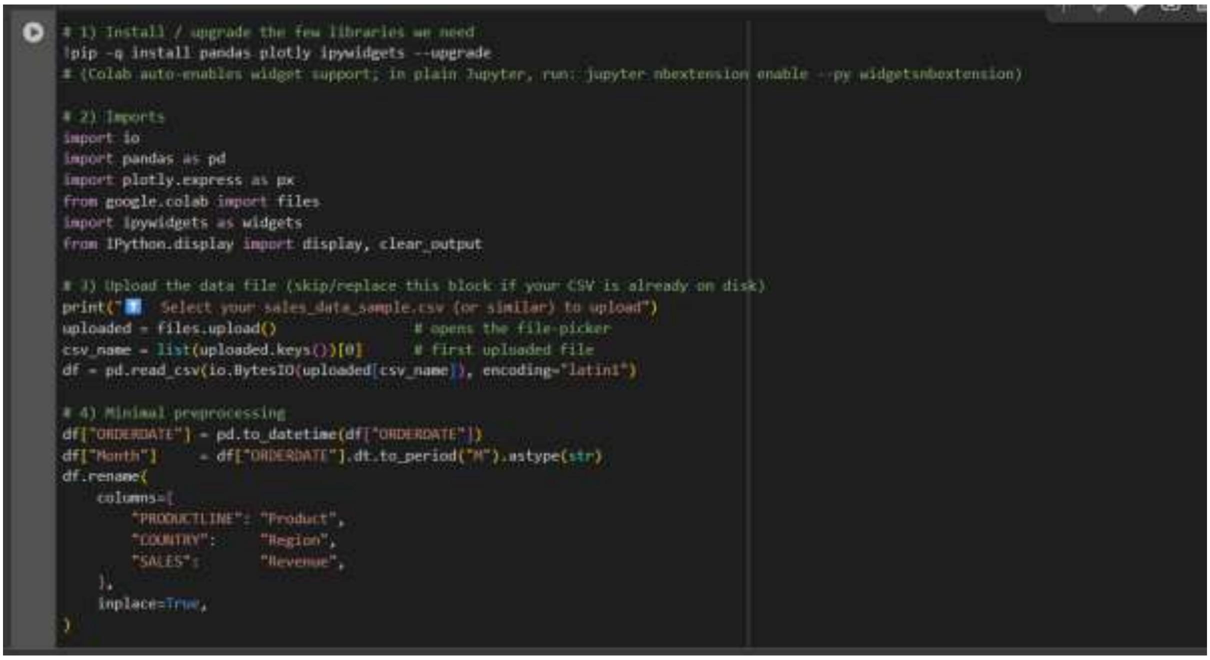
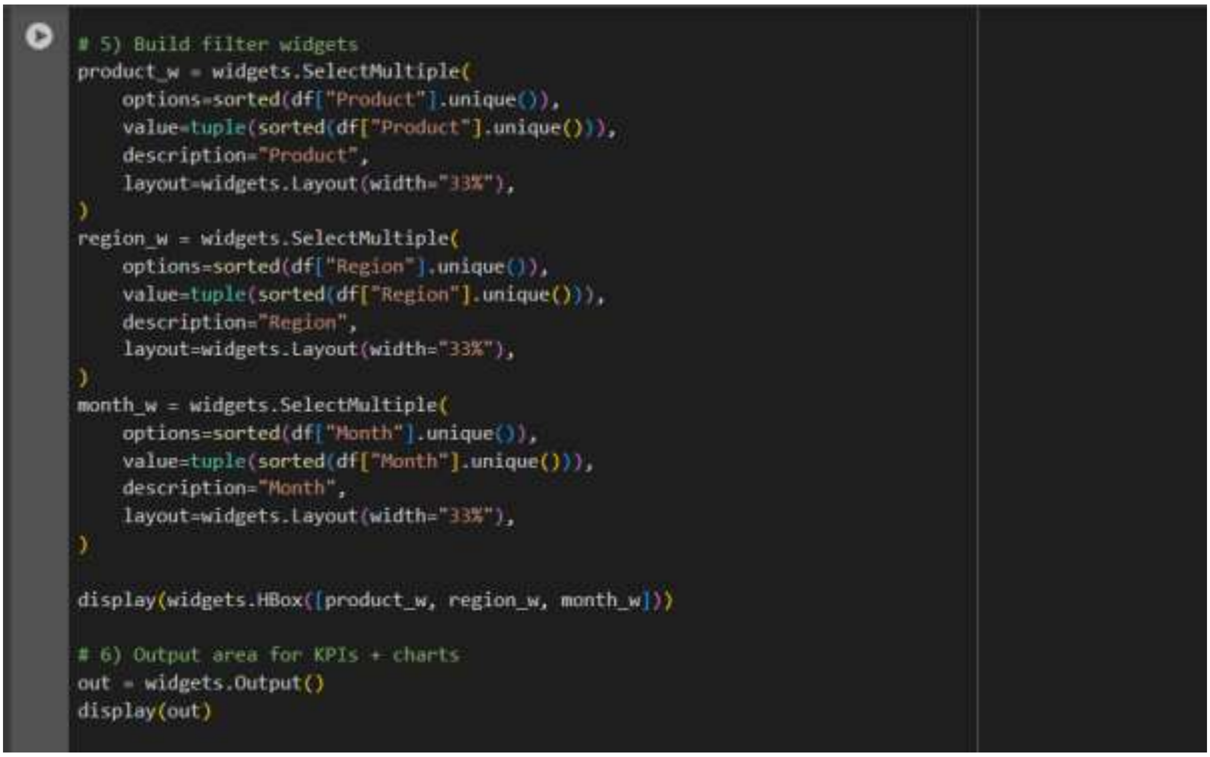
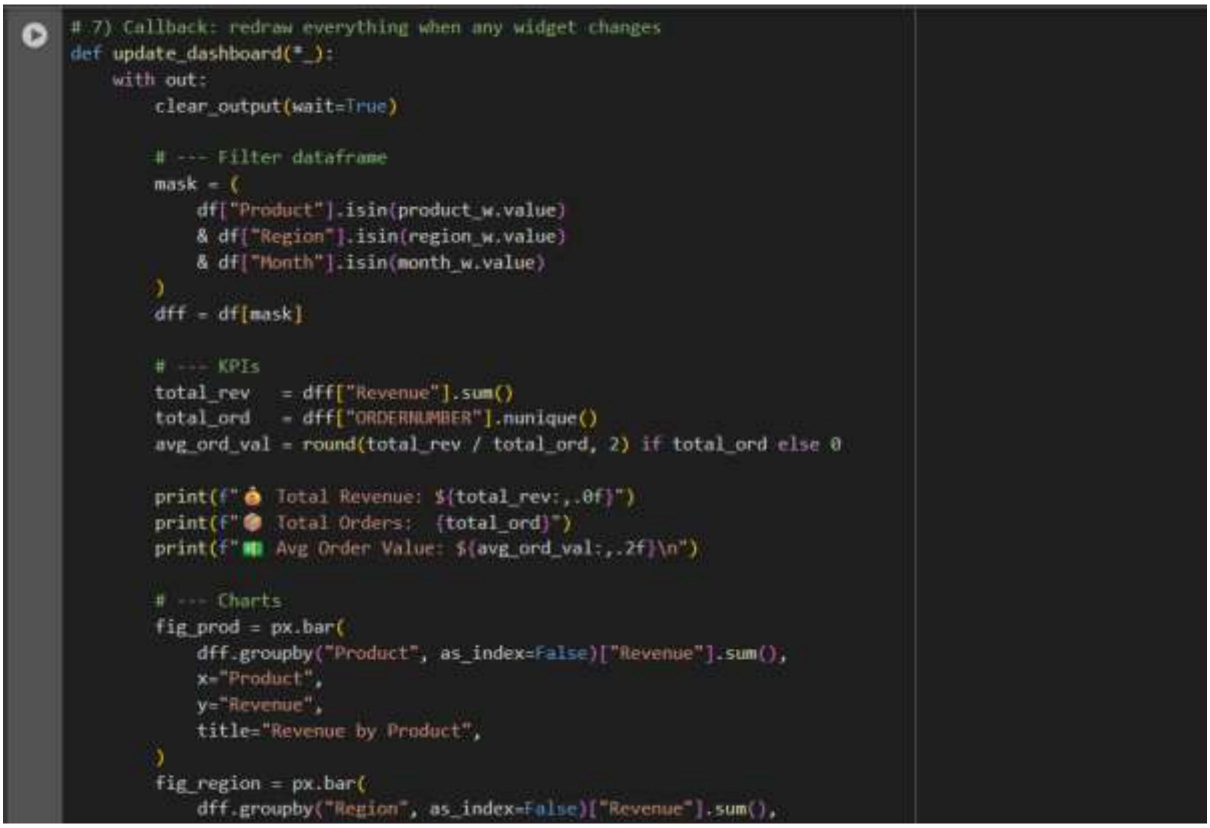
Code:



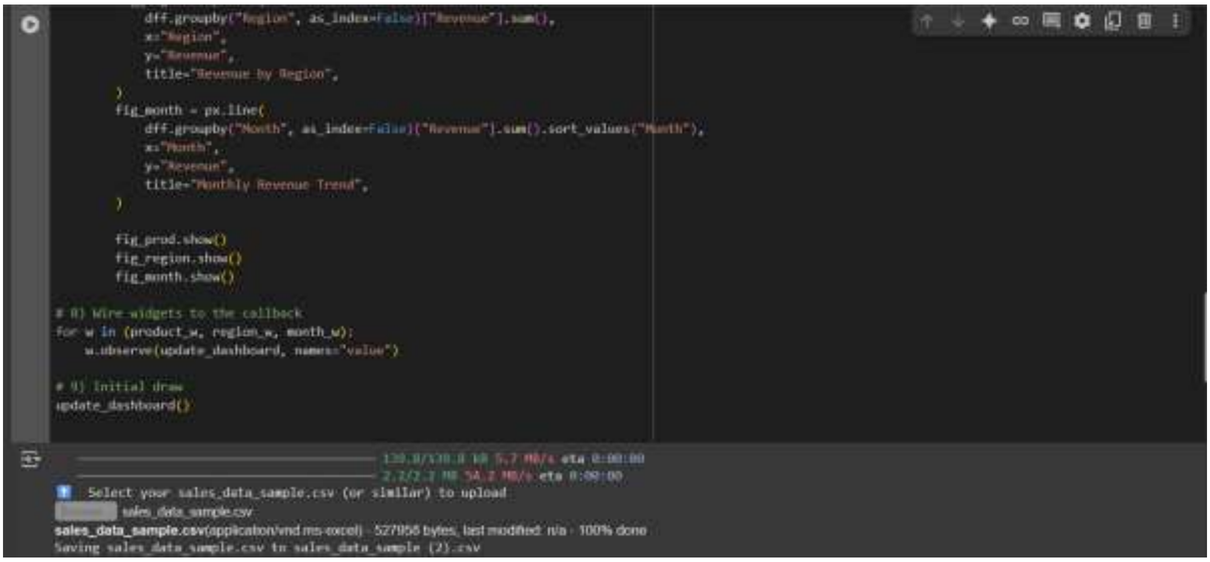
```
# 1) Install / upgrade the few libraries we need
!pip -q install pandas plotly ipywidgets --upgrade
# (Colab auto-enables widget support; in plain Jupyter, run: jupyter nbextension enable --py widgetsnbextension)

# 2) Imports
import io
import pandas as pd
import plotly.express as px
from google.colab import files
import ipywidgets as widgets
from IPython.display import display, clear_output

# 3) Upload the data file (skip/replace this block if your CSV is already on disk)
print("⬆️  Select your sales_data_sample.csv (or similar) to upload")
uploaded = files.upload()                    # opens the file-picker
csv_name = list(uploaded.keys())[0]          # first uploaded file
df = pd.read_csv(io.BytesIO(uploaded[csv_name]), encoding="latin1")

# 4) Minimal preprocessing
df["ORDERDATE"] = pd.to_datetime(df["ORDERDATE"])
df["Month"]     = df["ORDERDATE"].dt.to_period("M").astype(str)
df.rename(
    columns={
        "PRODUCTLINE": "Product",
        "COUNTRY":     "Region",
        "SALES":       "Revenue",
    },
    inplace=True,
)
```



```
# 5) Build filter widgets
product_w = widgets.SelectMultiple(
    options=sorted(df["Product"].unique()),
    value=tuple(sorted(df["Product"].unique())),
    description="Product",
    layout=widgets.Layout(width="33%"),
)
region_w = widgets.SelectMultiple(
    options=sorted(df["Region"].unique()),
    value=tuple(sorted(df["Region"].unique())),
    description="Region",
    layout=widgets.Layout(width="33%"),
)
month_w = widgets.SelectMultiple(
    options=sorted(df["Month"].unique()),
    value=tuple(sorted(df["Month"].unique())),
    description="Month",
    layout=widgets.Layout(width="33%"),
)

display(widgets.HBox([product_w, region_w, month_w]))

# 6) Output area for KPIs + charts
out = widgets.Output()
display(out)
```

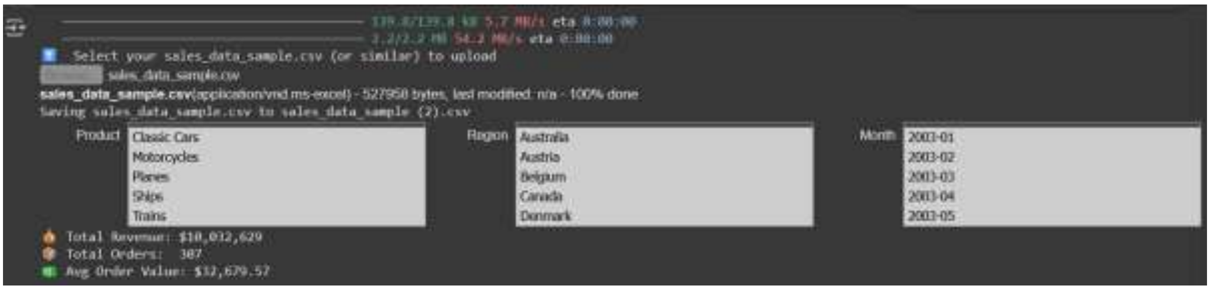


```python
# 7) Callback: redraw everything when any widget changes
def update_dashboard(*_):
    with out:
        clear_output(wait=True)

        # --- Filter dataframe
        mask = (
            df["Product"].isin(product_w.value)
            & df["Region"].isin(region_w.value)
            & df["Month"].isin(month_w.value)
        )
        dff = df[mask]

        # --- KPIs
        total_rev   = dff["Revenue"].sum()
        total_ord   = dff["ORDERNUMBER"].nunique()
        avg_ord_val = round(total_rev / total_ord, 2) if total_ord else 0

        print(f"💰 Total Revenue: ${total_rev:,.0f}")
        print(f"📦 Total Orders:  {total_ord}")
        print(f"💵 Avg Order Value: ${avg_ord_val:,.2f}\n")

        # --- Charts
        fig_prod = px.bar(
            dff.groupby("Product", as_index=False)["Revenue"].sum(),
            x="Product",
            y="Revenue",
            title="Revenue by Product",
        )
        fig_region = px.bar(
            dff.groupby("Region", as_index=False)["Revenue"].sum(),
```



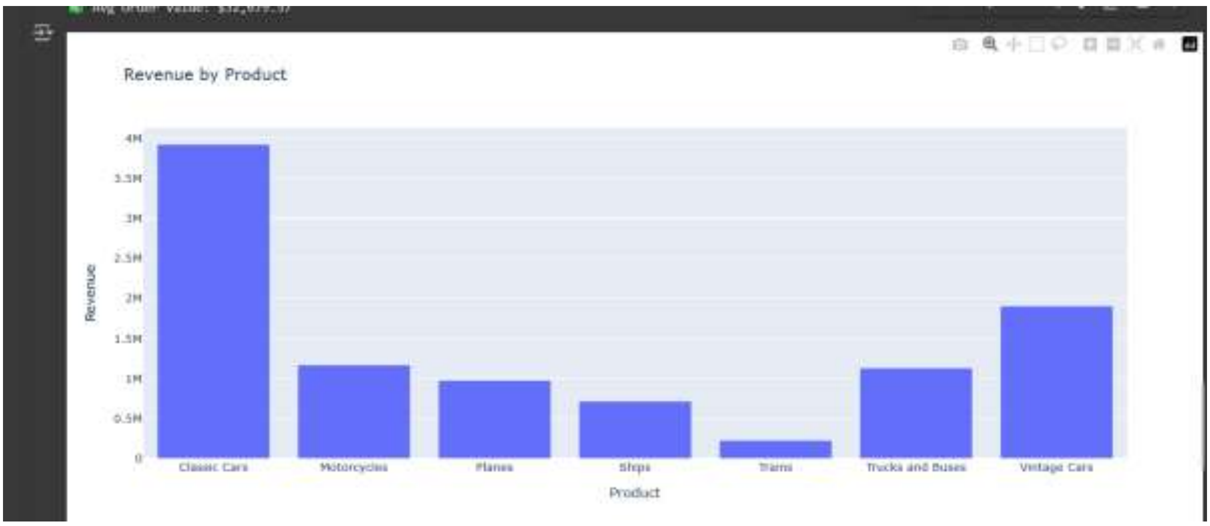
```python
            dff.groupby("Region", as_index=False)["Revenue"].sum(),
            x="Region",
            y="Revenue",
            title="Revenue by Region",
        )
        fig_month = px.line(
            dff.groupby("Month", as_index=False)["Revenue"].sum().sort_values("Month"),
            x="Month",
            y="Revenue",
            title="Monthly Revenue Trend",
        )

        fig_prod.show()
        fig_region.show()
        fig_month.show()

# 8) Wire widgets to the callback
for w in (product_w, region_w, month_w):
    w.observe(update_dashboard, names="value")

# 9) Initial draw
update_dashboard()
```

```
   ──────────────────────────────── 139.8/139.8 kB 5.7 MB/s eta 0:00:00
   ──────────────────────────────── 2.2/2.2 MB 54.2 MB/s eta 0:00:00
⬆️ Select your sales_data_sample.csv (or similar) to upload
sales_data_sample.csv
sales_data_sample.csv(application/vnd.ms-excel) - 527958 bytes, last modified: n/a - 100% done
Saving sales_data_sample.csv to sales_data_sample (2).csv
```



```
   ──────────────────────────────── 139.8/139.8 kB 5.7 MB/s eta 0:00:00
   ──────────────────────────────── 2.2/2.2 MB 54.2 MB/s eta 0:00:00
⬆️ Select your sales_data_sample.csv (or similar) to upload
sales_data_sample.csv
sales_data_sample.csv(application/vnd.ms-excel) - 527958 bytes, last modified: n/a - 100% done
Saving sales_data_sample.csv to sales_data_sample (2).csv
```

```
💰 Total Revenue: $10,032,629
📦 Total Orders:  307
💵 Avg Order Value: $32,679.57
```



```
Avg Order Value: $32,079.37
```