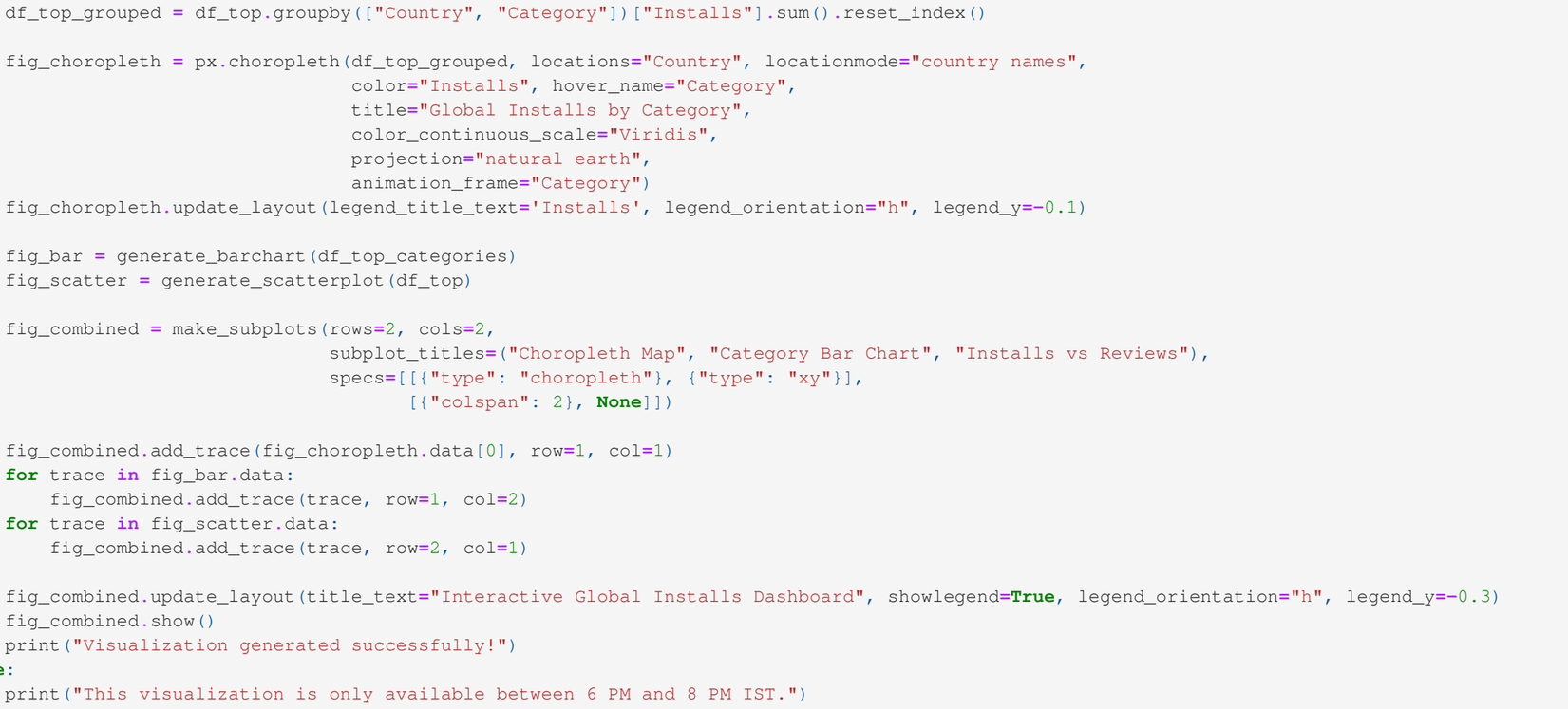
In [1]:
```
import pandas as pd
import plotly.express as px
import datetime
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import random

df = pd.read_csv("GooglePlayStore.zip")

def preprocess_installs(installs):
    return int(installs.replace(",", "").replace("+", "")) if isinstance(installs, str) and installs.replace(",", "").replace("+", "").isdigit() else 0

df["Installs"] = df["Installs"].apply(preprocess_installs)
df_filtered = df[~df["Category"].str.startswith(("A", "C", "G", "S"))]
df_top_categories = df_filtered.groupby("Category")["Installs"].sum().nlargest(5).reset_index()
df_top = df_filtered[df_filtered["Category"].isin(df_top_categories["Category"])]

world_countries = ["United States", "India", "Germany", "Brazil", "Canada", "France", "Australia", "China", "United Kingdom", "Japan"]
df_top["Country"] = [random.choice(world_countries) for _ in range(len(df_top))]

def generate_barchart(df):
    fig_bar = px.bar(df, x="Category", y="Installs", color="Category",
                     title="Top 5 Categories by Installs",
                     text="Installs", color_discrete_sequence=px.colors.qualitative.Set2)
    fig_bar.update_traces(texttemplate='%{text}', textposition='outside')
    fig_bar.update_layout(legend_title_text='Category', legend_orientation="h", legend_y=1.15)
    return fig_bar

def generate_scatterplot(df):
    fig_scatter = px.scatter(df, x="Reviews", y="Installs", color="Category",
                             title="Installs vs. Reviews", size="Installs", hover_name="App")
    fig_scatter.update_layout(legend_title_text='Category', legend_orientation="h", legend_y=-0.2)
    return fig_scatter

current_time = datetime.datetime.now().time()
allowed_start = datetime.time(18, 0, 0)
allowed_end = datetime.time(20, 0, 0)

if allowed_start <= current_time <= allowed_end:
    print("Generating visualization... Please wait.")
    df_top_grouped = df_top.groupby(["Country", "Category"])["Installs"].sum().reset_index()

    fig_choropleth = px.choropleth(df_top_grouped, locations="Country", locationmode="country names",
                                   color="Installs", hover_name="Category",
                                   title="Global Installs by Category",
                                   color_continuous_scale="Viridis",
                                   projection="natural earth",
                                   animation_frame="Category")
    fig_choropleth.update_layout(legend_title_text='Installs', legend_orientation="h", legend_y=-0.1)

    fig_bar = generate_barchart(df_top_categories)
    fig_scatter = generate_scatterplot(df_top)

    fig_combined = make_subplots(rows=2, cols=2,
                                 subplot_titles=("Choropleth Map", "Category Bar Chart", "Installs vs Reviews"),
                                 specs=[[{"type": "choropleth"}, {"type": "xy"}],
                                        [{"colspan": 2}, None]])

    fig_combined.add_trace(fig_choropleth.data[0], row=1, col=1)
    for trace in fig_bar.data:
        fig_combined.add_trace(trace, row=1, col=2)
    for trace in fig_scatter.data:
        fig_combined.add_trace(trace, row=2, col=1)

    fig_combined.update_layout(title_text="Interactive Global Installs Dashboard", showlegend=True, legend_orientation="h", legend_y=-0.3)
    fig_combined.show()
    print("Visualization generated successfully!")
else:
    print("This visualization is only available between 6 PM and 8 PM IST.")
```

```
C:\Users\sh\AppData\Local\Temp\ipykernel_7720\923555146.py:19: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_top["Country"] = [random.choice(world_countries) for _ in range(len(df_top))]
Generating visualization... Please wait.
```



```
Visualization generated successfully!
```

In [ ]: