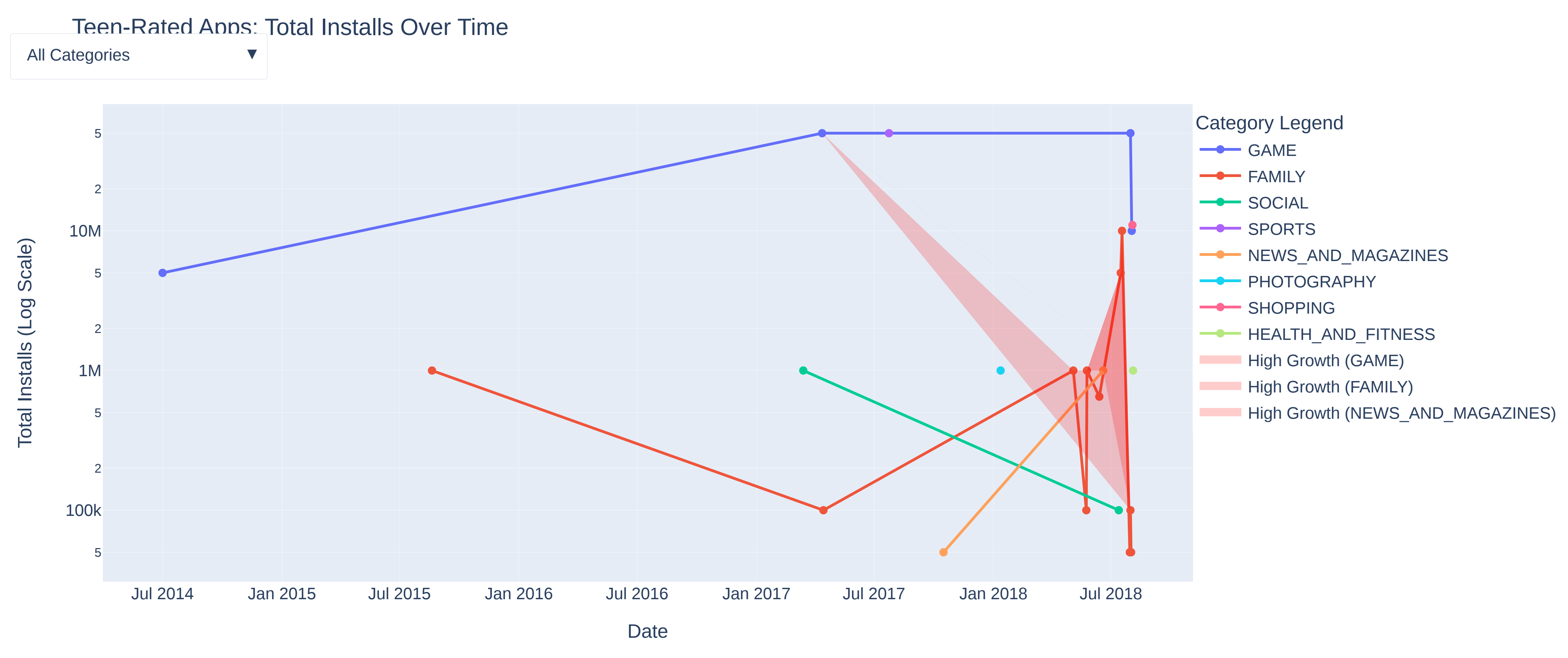
In [1]:

```
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
import datetime as dt

df = pd.read_csv("GooglePlayStore.zip", usecols=["App", "Category", "Installs", "Content Rating", "Last Updated"])

df["Installs"] = df["Installs"].str.replace("[+,]", "", regex=True).astype(float)
df["Last Updated"] = pd.to_datetime(df["Last Updated"], errors="coerce")
df = df.dropna(subset=["Last Updated"])

filtered_df = df[
    (df["Content Rating"] == "Teen") &
    (df["App"].str.startswith("E")) &
    (df["Installs"] > 10000) &
    (df["Category"].notnull()) &
    (df["Last Updated"].notnull())
]

filtered_df = filtered_df.groupby(["Last Updated", "Category", "Content Rating"], as_index=False)["Installs"].sum()
filtered_df = filtered_df.sort_values("Last Updated")

filtered_df["Prev Installs"] = filtered_df.groupby("Category")["Installs"].shift(1)
filtered_df["MoM Growth"] = ((filtered_df["Installs"] - filtered_df["Prev Installs"]) / filtered_df["Prev Installs"]) * 100

current_time = dt.datetime.now().time()
if dt.time(18, 0) <= current_time <= dt.time(21, 0):
    fig = px.line(
        filtered_df,
        x="Last Updated",
        y="Installs",
        color="Category",
        title="Teen-Rated Apps: Total Installs Over Time",
        markers=True,
        hover_data=["MoM Growth", "Content Rating"]
    )

    fig.update_yaxes(title_text="Total Installs (Log Scale)", type="log")
    fig.update_xaxes(title_text="Date")

    for category in filtered_df["Category"].unique():
        category_data = filtered_df[filtered_df["Category"] == category]
        growth_exceeds_20 = category_data[category_data["MoM Growth"] > 20]

        fig.add_trace(go.Scatter(
            x=growth_exceeds_20["Last Updated"],
            y=growth_exceeds_20["Installs"],
            fill='tonexty',
            fillcolor="rgba(255, 0, 0, 0.2)",
            mode="none",
            name=f"High Growth ({category})"
        ))

    fig.update_layout(legend=dict(x=1, y=1, title="Category Legend"))

    fig.update_layout(
        updatemenus=[{
            "buttons": [
                {"label": "All Categories", "method": "update", "args": [{"visible": True}]}
            ] + [
                {
                    "label": category,
                    "method": "update",
                    "args": [{"visible": [(c == category) for c in filtered_df["Category"].unique()]}]
                }
                for category in filtered_df["Category"].unique()
            ],
            "direction": "down",
            "showactive": True,
            "x": 0.15, "y": 1.15
        }]
    )

    fig.show()
else:
    print("The graph is only available between 6 PM IST and 9 PM IST.")
```

```
C:\Users\sh\anaconda3\Lib\site-packages\_plotly_utils\basevalidators.py:106: FutureWarning: The behavior of DatetimeProperties.to_pydatetime is deprecated, in a future version this will retur
n a Series containing python datetime objects instead of an ndarray. To retain the old behavior, call `np.array` on the result
  v = v.dt.to_pydatetime()
C:\Users\sh\anaconda3\Lib\site-packages\_plotly_utils\basevalidators.py:106: FutureWarning: The behavior of DatetimeProperties.to_pydatetime is deprecated, in a future version this will retur
n a Series containing python datetime objects instead of an ndarray. To retain the old behavior, call `np.array` on the result
  v = v.dt.to_pydatetime()
C:\Users\sh\anaconda3\Lib\site-packages\_plotly_utils\basevalidators.py:106: FutureWarning: The behavior of DatetimeProperties.to_pydatetime is deprecated, in a future version this will retur
n a Series containing python datetime objects instead of an ndarray. To retain the old behavior, call `np.array` on the result
  v = v.dt.to_pydatetime()
C:\Users\sh\anaconda3\Lib\site-packages\_plotly_utils\basevalidators.py:106: FutureWarning: The behavior of DatetimeProperties.to_pydatetime is deprecated, in a future version this will retur
n a Series containing python datetime objects instead of an ndarray. To retain the old behavior, call `np.array` on the result
  v = v.dt.to_pydatetime()
C:\Users\sh\anaconda3\Lib\site-packages\_plotly_utils\basevalidators.py:106: FutureWarning: The behavior of DatetimeProperties.to_pydatetime is deprecated, in a future version this will retur
n a Series containing python datetime objects instead of an ndarray. To retain the old behavior, call `np.array` on the result
  v = v.dt.to_pydatetime()
C:\Users\sh\anaconda3\Lib\site-packages\_plotly_utils\basevalidators.py:106: FutureWarning: The behavior of DatetimeProperties.to_pydatetime is deprecated, in a future version this will retur
n a Series containing python datetime objects instead of an ndarray. To retain the old behavior, call `np.array` on the result
  v = v.dt.to_pydatetime()
C:\Users\sh\anaconda3\Lib\site-packages\_plotly_utils\basevalidators.py:106: FutureWarning: The behavior of DatetimeProperties.to_pydatetime is deprecated, in a future version this will retur
n a Series containing python datetime objects instead of an ndarray. To retain the old behavior, call `np.array` on the result
  v = v.dt.to_pydatetime()
C:\Users\sh\anaconda3\Lib\site-packages\_plotly_utils\basevalidators.py:106: FutureWarning: The behavior of DatetimeProperties.to_pydatetime is deprecated, in a future version this will retur
n a Series containing python datetime objects instead of an ndarray. To retain the old behavior, call `np.array` on the result
  v = v.dt.to_pydatetime()
C:\Users\sh\anaconda3\Lib\site-packages\_plotly_utils\basevalidators.py:106: FutureWarning:

The behavior of DatetimeProperties.to_pydatetime is deprecated, in a future version this will return a Series containing python datetime objects instead of an ndarray. To retain the old behav
ior, call `np.array` on the result
```



In [ ]: