Lab 3: Data Visualization

- Student ID：1851049
- Name：陈中悦

Data Analysis

Objectives

The `google-play-store-apps` dataset is chosen as the data visualization object for this assignment.

This dataset records information about various categories of Apps in the Google App Store, with a total of 10840 data, including the `App name` , `Category` , `Rating` , `Reviews` , `Size` , `Installs` , `Type` , `Price` , `Content Rating` , `Genres` , `Last Updated` , `Current Ver` , `Android Ver` .

This assignment takes `Category` and `Type` as the main classification basis, based on which a pie chart shows the percentage of the number of Apps of each content rating; at the same time, the names of the top ten installed Apps are listed.

`Content Rating` is used as the secondary basis for classification. The user selects a rating, based on which the relationship between `Reviews` and `Installs` is shown using a **scatter chart**; the relationship between `Installs` and `Size` is shown using a **folding line chart**; and the relationship of `Installs-Rating` and `reviews-rating` is shown using two **bar charts**, respectively.

Characteristics

The characteristics of some of the attributes in this dataset used in the assignment are as follows:

Category

This attribute indicates the App category and has a total of 33 values as follows.

ART_AND_DESIGN
AUTO_AND_VEHICLES
BEAUTY
BOOKS_AND_REFERENCE
BUSINESS
COMICS
COMMUNICATION
DATING
EDUCATION
ENTERTAINMENT
EVENTS
FINANCE
FOOD_AND_DRINK
HEALTH_AND_FITNESS
HOUSE_AND_HOME
LIBRARIES_AND_DEMO
LIFESTYLE
GAME
FAMILY
MEDICAL
SOCIAL
SHOPPING
PHOTOGRAPHY
SPORTS
TRAVEL_AND_LOCAL
TOOLS
PERSONALIZATION
PRODUCTIVITY
PARENTING
WEATHER
VIDEO_PLAYERS

ART_AND_DESIGN
NEWS_AND_MAGAZINES
MAPS_AND_NAVIGATION

Rating

This attribute indicates the rating of each App, where:

- `NaN` Indicates that the App is not rated and that such Apps are not involved in data visualization involving rating.
- `Rating` takes values in the range `1.0~5.0` . The visualization process divides the rating into six intervals: `[0, 1)` , `[1,2)` , `[2,3)` , `[3,4)` , `[4,5)` , `[5, inf)`

Size

This attribute indicates the storage capacity required for each App, where:

- `Varies with device` means that the size of some Apps vary from different devices.
- The rest of the Apps take values in the range `8.5k~1020.0k` and `1.0M~100.0M` . The visualization process divides `Size` into the following 8 intervals: `0~300K` , `300K~600K'` , `600K~900K` , `900K~25M` , `25M~50M` , `50M~75M'` , `75M~100M` ', `Varies with device` '

Insatlls

This property indicates the number of downloads for each App and takes values ranging from `0~1,000,000,000+` , and no further details will be given here.

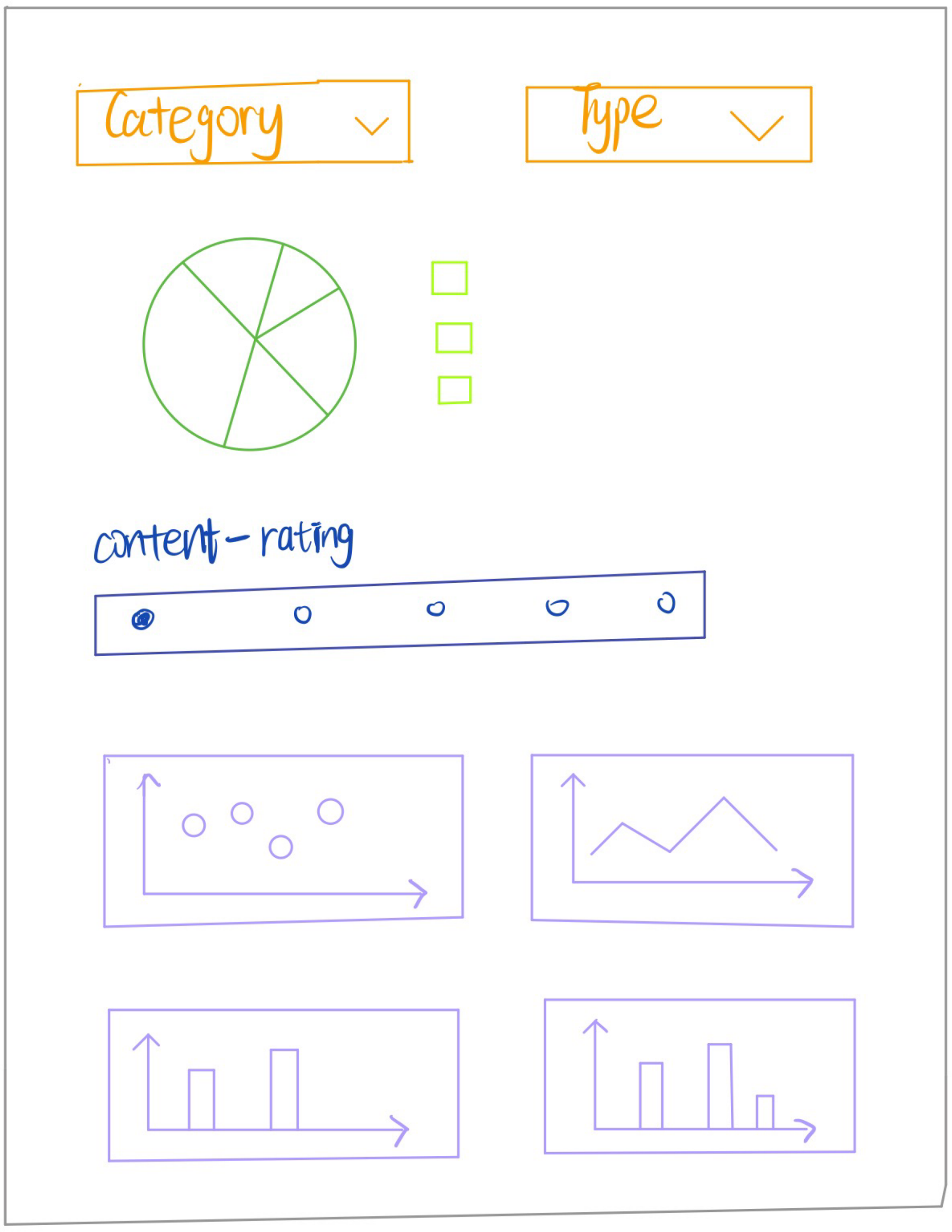
Content Rating

This attribute indicates the hierarchy of the content of each App and has the following six values:

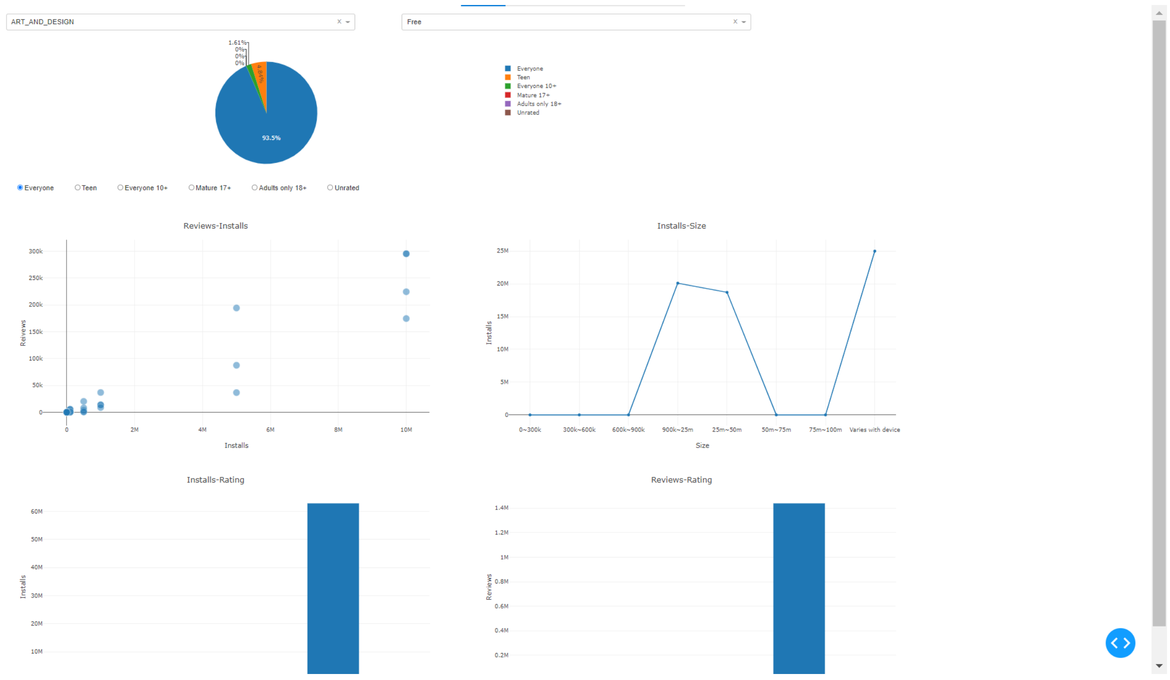
Everyone
Teen
Everyone 10+
Mature 17+
Adults only 18+
Unrated

Layout of Designed Dashboard

Since `Category` , `Type` and `Content Rating` are the main classification bases in this data visualization process, the visualized information needs to be pre-screened based on the above classification bases, so the layout shown below is used.

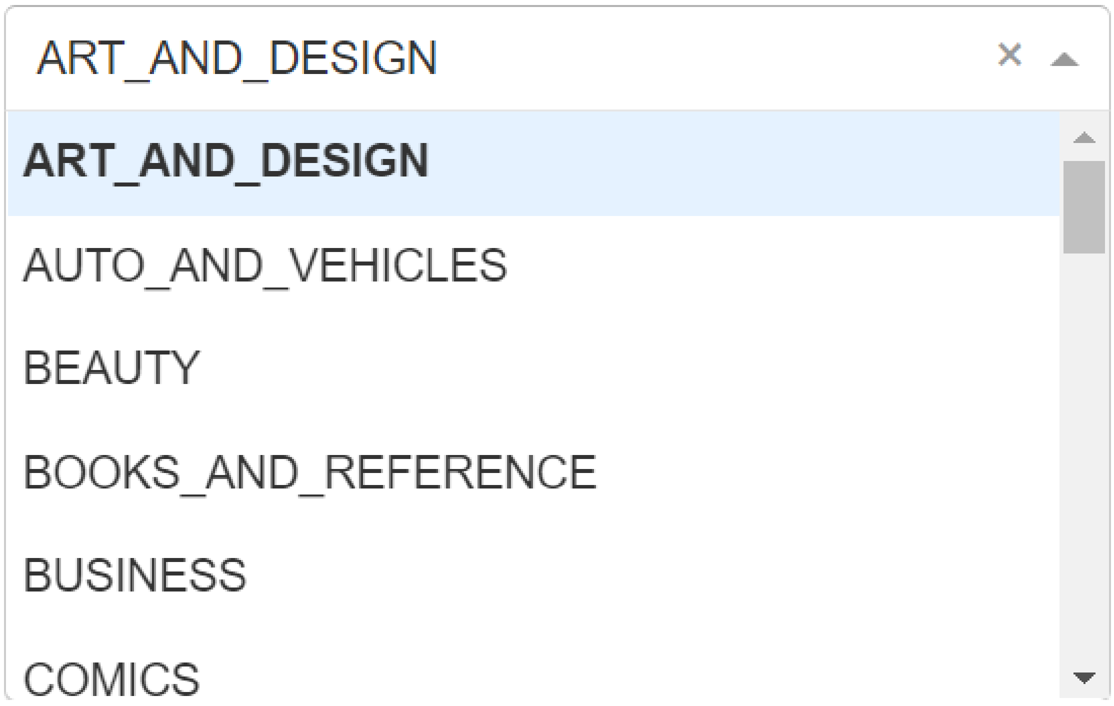


A general overview of the interface is as follows;



1. **Category Dropdown**

   For users to select the App category, the default value is `ART_AND_DESIGN`



ART_AND_DESIGN

ART_AND_DESIGN

AUTO_AND_VEHICLES

BEAUTY

BOOKS_AND_REFERENCE

BUSINESS

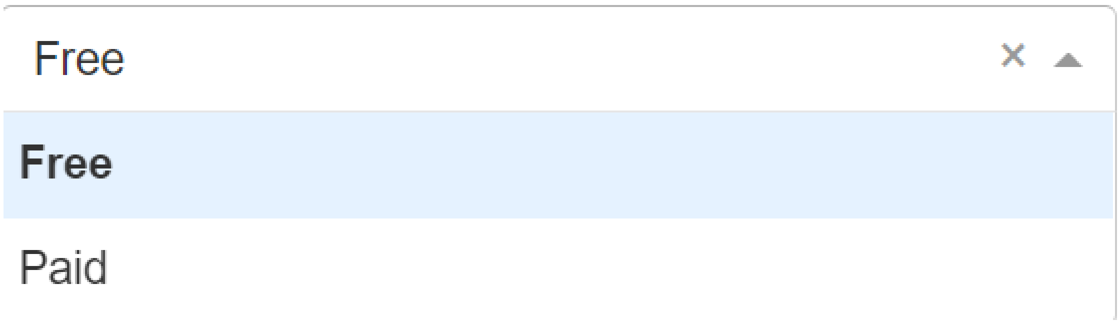
COMICS

Core Code:

```python
# category
    html.Div([
        dcc.Dropdown(
            id="Category",
            options=[{'label': i, 'value': i} for i in categories],
            value='ART_AND_DESIGN'
        )],
        style={'width': '30%', 'display': 'inline-block','padding':'10px 50px'}),
```

2. **Type Dropdown**

   For users to choose the cost type of the App, the default value is `Free`

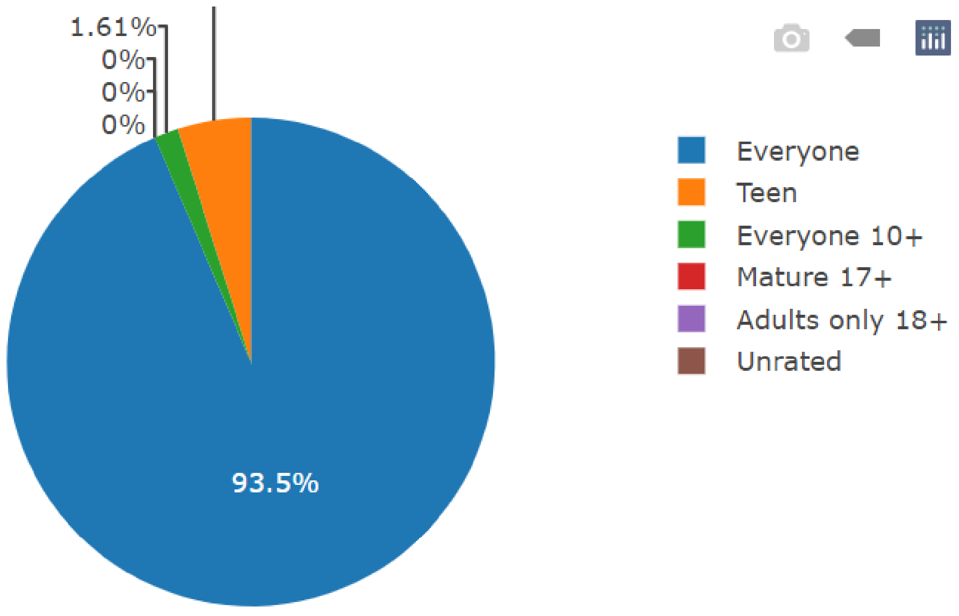


Core Code:

```
# type
    html.Div([
        dcc.Dropdown(
            id="Type",
            options=[{'label': i, 'value': i} for i in types],
            value='Free'
        )],
        style={'width': '30%', 'display': 'inline-block', 'padding': '10px
50px'}),
```

3. **Content Rating-Nums Pie Chart**

   Show the number of Apps of each Content Rating under current Category and Type.

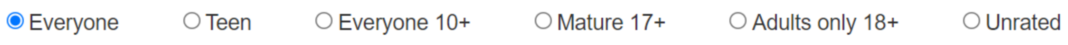


Core Code:

```
def update_pie(category, type):
    dff=df[(df['Category']==category)&(df['Type']==type)]

    content_rating_list=list(content_rates)
    content_rating_values=[]
    for i in range(len(content_rating_list)):
        content_rating_values.append(0)
    for index,row in dff.iterrows():
        content_rating_values[content_rating_list.index(row[8])]+=1
    print('content_rating_values',content_rating_values)


    return {
        'data':[go.Pie(
        labels=content_rating_list,
        values=content_rating_values
    )],
        'layout':
            go.Layout(
                margin={
                    'l': 130,
                    'b': 30,
                    't': 50,
                    'r': 0
                },
                height=300,
                hovermode='closest'
            )
    }
```

4. **Content Rating Radio Items**

   For users to select the App's `Content Rating`, the default value is `Everyone`.

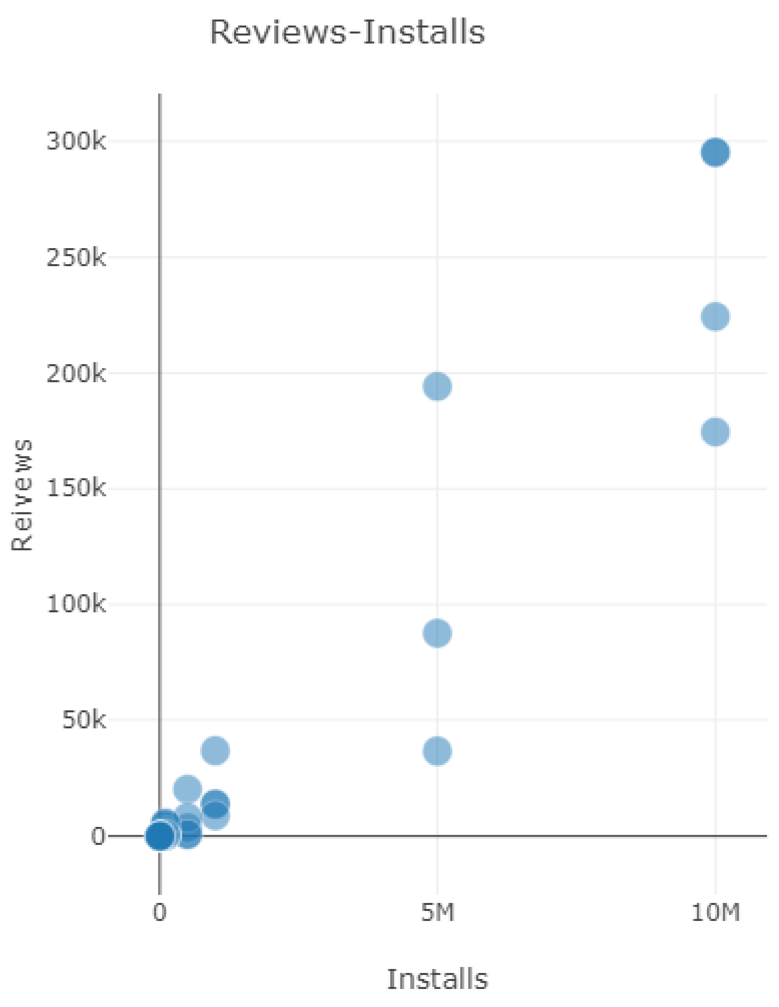

[✓ Everyone] Teen Everyone 10+ Mature 17+ Adults only 18+ Unrated

Core Code:

```
# content-rating radio-items
    html.Div([
        dcc.RadioItems(
            id='content-rating-radio',
            options=[
                {'label':i,'value':i} for i in content_rates
            ],
            value='Everyone',
            labelStyle={'padding':'0 20px','display': 'inline-block'}
        )
    ],
    style={'padding':'10px 50px','margin':'0 auto'}),
```

5. **Reviews-Installs Scatters Chart**

   Show the scattered relationship between Reviews and Installs under current Category, Type and Content Rating.

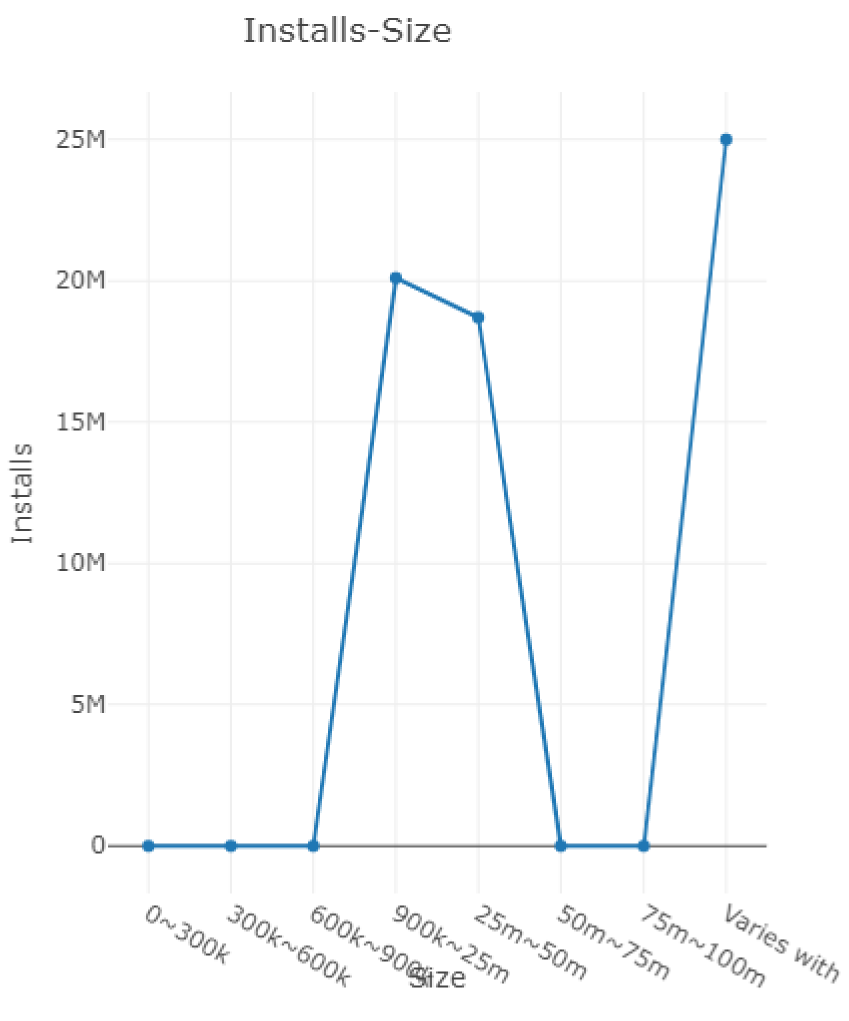


Core Code:

```python
# reviews-installs散点图
@app.callback(
    dash.dependencies.Output('reviews-installs', 'figure'),
    [
        dash.dependencies.Input('Category', 'value'),
        dash.dependencies.Input('Type', 'value'),
        dash.dependencies.Input('content-rating-radio','value')
    ]
)
def update_RI_scatters(category, type, radio):
    print(category, type, radio)
    dff = df[(df['Category'] == category) & (df['Content Rating'] == radio)&
(df['Type']==type)]


    reviews,installs,names=[],[],[]
    for index, row in dff.iterrows():
        reviews.append(row['Reviews'])
        installs.append(row['Installs'])
        names.append(row['App'])

    return create_scatter(installs, 'Installs',reviews,'Reivews',names,'Reviews-
Installs')
```

6. **Installs-Size Folding Line Chart**

   Show the current Category, Type and Content Rating of the installs and Size of the line relationship.

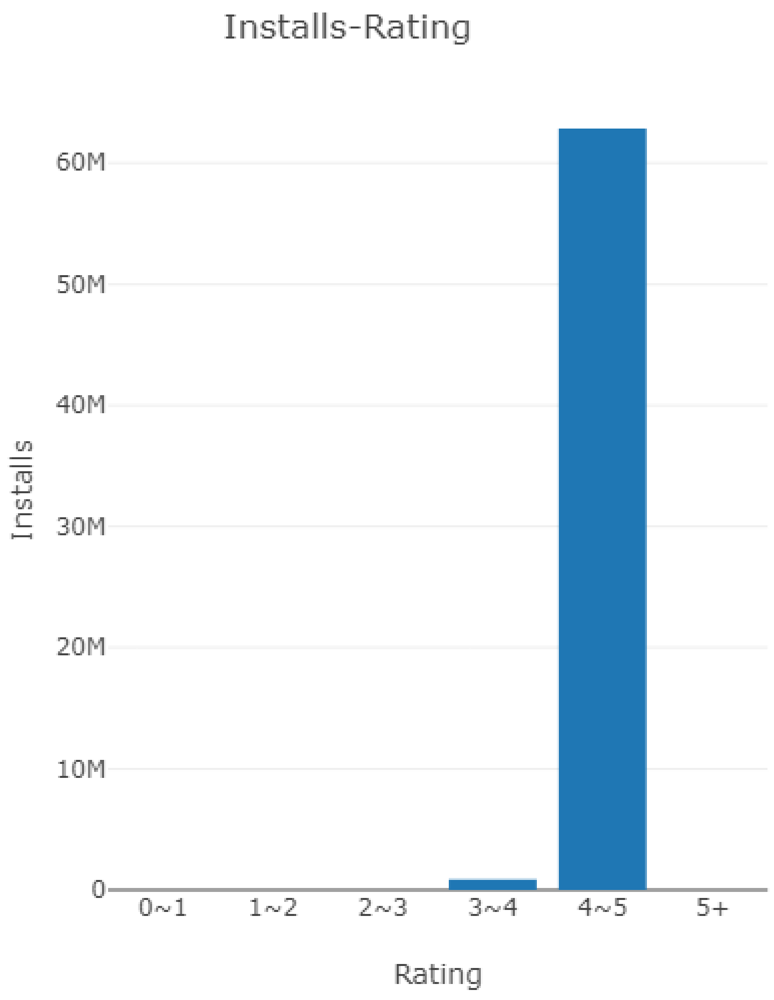


Core Code:

```
# installs-size 折线图
@app.callback(
    dash.dependencies.Output('installs-size', 'figure'),
    [
        dash.dependencies.Input('Category', 'value'),
        dash.dependencies.Input('Type', 'value'),
        dash.dependencies.Input('content-rating-radio','value')
    ]
)
def update_IS_series(category,type,radio):
    dff = df[(df['Category'] == category) & (df['Content Rating'] == radio)&
(df['Type']==type)]


    installs,sizes=[0,0,0,0,0,0,0,0],[]
    sizes = [
        '0~300k', '300k~600k', '600k~900k', '900k~25m', '25m~50m',
        '50m~75m', '75m~100m', 'Varies with device'
    ] # 单位m
    for index,row in dff.iterrows():
        print(row['Size'])
        if row['Size']=='Varies with device':
            installs[-1]+=row['Installs']
        elif row['Size'][-1]=='k':
            i=int(float(row['Size'][:-1])//300)
            print('软件大小：',row['Size'],'下标：',i)
            installs[i]+=row['Installs']
        else:
            i=int(float(row['Size'][:-1])//25)+3
            installs[i]+=row['Installs']
    print(installs)

    return {
        'data': [{
            "x": sizes,
            "y": installs,
            "mode": "lines+markers",
            'type': 'Scatter',
            'name': 'Line'
        }],
        'layout':
            go.Layout(
                xaxis={
                    'title': 'Size',
                },
                yaxis={
                    'title': 'Installs',
                },
                title='Installs-Size',

                margin={
                    'l': 130,
                    'b': 50,
                    't': 50,
                    'r': 40
                },
                height=500,
                hovermode='closest')
    }
```

7. **Installs-Rating Bar Chart**

   Show the bar relationship of Installs and Rating under current Category, Type and Content Rating.

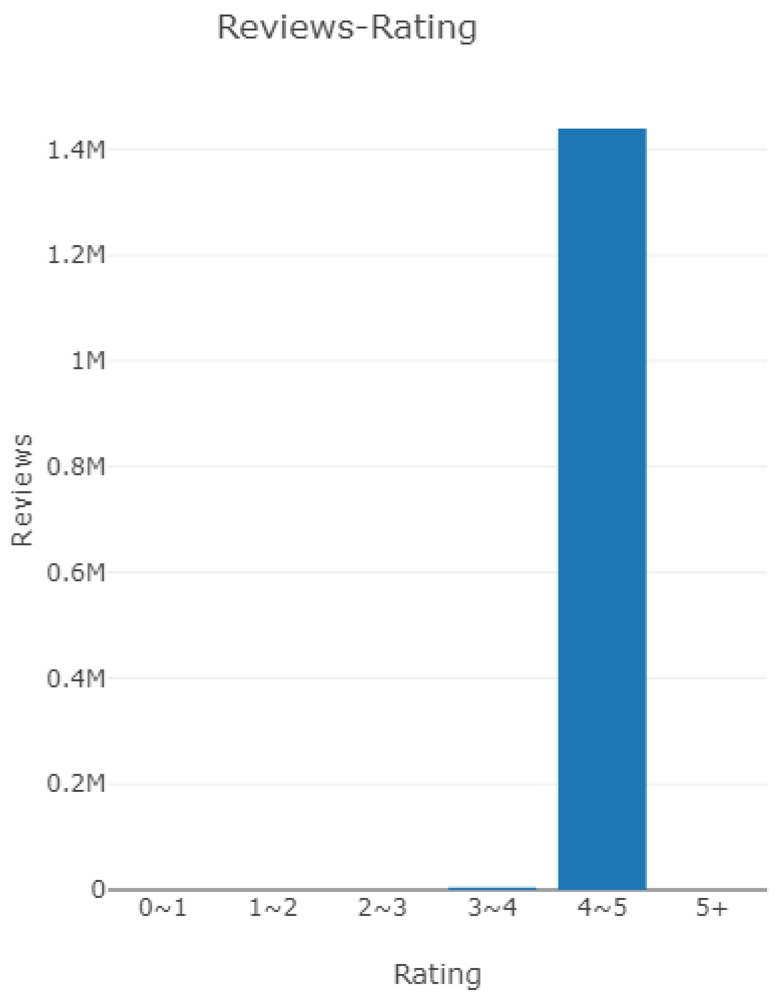


Core Code:

```
# installs-rating柱状图
@app.callback(
    dash.dependencies.Output('installs-rating', 'figure'),
    [
        dash.dependencies.Input('Category', 'value'),
        dash.dependencies.Input('Type', 'value'),
        dash.dependencies.Input('content-rating-radio','value')
    ]
)
def update_RI_scatters(category, type, radio):
    print(category, type, radio)
    dff = df[(df['Category'] == category) & (df['Content Rating'] == radio)&
(df['Type']==type)]



    installs=[0,0,0,0,0,0]
    ratings=['0~1','1~2','2~3','3~4','4~5','5+']
    for index, row in dff.iterrows():
        if not np.isnan(row['Rating']):
            installs[int(float(row['Rating']))]+=row['Installs']
    print(installs)

    return {
        'data': [
            go.Bar(
                x=ratings,
                y=installs,
            )
        ],
        'layout':
            go.Layout(
                xaxis={
                    'title': 'Rating',
                },
                yaxis={
                    'title': 'Installs',
                },
                title='Installs-Rating',
                margin={
                    'l': 130,
                    'b': 50,
                    't': 50,
                    'r': 40
                },
                height=500,
                hovermode='closest')
    }
```

8. **Reviews-Rating Bar Chart**

   Show the bar relationship between Reviews and Rating under current Category, Type and Content Rating.



Core Code:

```
# reviews-rating柱状图
@app.callback(
    dash.dependencies.Output('reviews-rating', 'figure'),
    [
        dash.dependencies.Input('Category', 'value'),
        dash.dependencies.Input('Type', 'value'),
        dash.dependencies.Input('content-rating-radio','value')
    ]
)
def update_RI_scatters(category, type, radio):
    print(category, type, radio)
    dff = df[(df['Category'] == category) & (df['Content Rating'] == radio)&
(df['Type']==type)]

    reviews = [0, 0, 0, 0, 0,0]
    ratings = ['0~1', '1~2', '2~3', '3~4', '4~5','5+']
    for index, row in dff.iterrows():
        if not np.isnan(row['Rating']):
            reviews[int(float(row['Rating']))]+=row['Reviews']
    print(reviews)

    return {
        'data': [
            go.Bar(
                x=ratings,
                y=reviews,
                )
            ],
            'layout':
                go.Layout(
                    xaxis={
                        'title': 'Rating',
                    },
                    yaxis={
                        'title': 'Reviews',
                    },
                    title='Reviews-Rating',
                    margin={
                        'l': 130,
                        'b': 50,
                        't': 50,
                        'r': 40
                    },
                    height=500,
                    hovermode='closest')
        }
```

Patterns Revealed in the Figures

- **Content Rating-Nums Pie Chart**

  As can be seen from this chart:

  - The majority of Apps are open for everyone.
  - Only a few Apps are unrated, indicating Google's stricter constraints on Apps.
- **Reviews-Installs Scatters Chart**

  As can be seen from this chart:

  - Most Apps have less than 10M installs, while some have more than 50M installs.
  - Most Apps have less than 50K reviews, while some Apps have more than 50M reviews.
  - Reviews and Installs show an overall positive correlation, i.e. Apps with a higher number of installls also harvest relatively more reviews.
  - For Apps with a high number of installs, the number of reviews and installs do not show an obvious linear relationship, so it is inferred that for the more popular Apps, there are some users who only review but do not install.
- **Installs-Size Series Chart**

  As can be seen from this chart:

  - The number of Apps who take values in the range `900k~50M` and `Varies with device` is higher, indicating that users tend to install moderately sized Apps.
  - Apps with too small or too large size have low number of installations.
- **Installs-Rating Bar Chart & Reviews-Rating Bar Chart**

  As can be seen from these two charts:

  - Apps with ratings between 4 and 5 get more installs and reviews, indicating that users prefer Apps with high ratings, which is consistent with common sense.

- Apps with lower ratings have a lower number of installs and reviews, indicating that low-rated Apps are less popular with users.
- The number of installs and reviews for Apps with rating 5 is also low, which means that the number of Apps with rating 5 is low, so the overall number of installs and reviews is low.nvcnm