HCI Assignment 4

Student Number:1952293, Name: Shuo Tang

Describe a data analysis task

Preprocess

After viewing the dataset, I decided to expand the analysis of the dataset around the category. I opened the dataset with pandas. After loading the data, I used the following code to view the analysis of different categories:

```
print(df['Category'].unique())
# output
#['ART_AND_DESIGN' 'AUTO_AND_VEHICLES' 'BEAUTY' 'BOOKS_AND_REFERENCE'
#'BUSINESS' 'COMICS' '3.9' 'COMMUNICATION' 'DATING' '4.3' 'EDUCATION'
#'ENTERTAINMENT' '4.6' 'EVENTS' 'FINANCE' 'FOOD_AND_DRINK'
#'HEALTH_AND_FITNESS' '4.5' 'HOUSE_AND_HOME' 'LIBRARIES_AND_DEMO'
#'LIFESTYLE' '4.1' 'GAME' '4.7' '4' '4.2' 'FAMILY' 'MEDICAL' 'SOCIAL'
#'SHOPPING' 'PHOTOGRAPHY' '000+"' 'SPORTS' 'TRAVEL_AND_LOCAL' '3.7'
#'TOOLS' 'PERSONALIZATION' 'PRODUCTIVITY' 'PARENTING' 'WEATHER'
#'VIDEO_PLAYERS' 'NEWS_AND_MAGAZINES' 'MAPS_AND_NAVIGATION' 'NaN' '3.8'
#'3.1' '3.6' '4.4' '3' '1.9' '4.8' '2.7']
```

It can be seen that there are ambiguous categories such as 3.1, 3.6, 4.4 and NaN. Use the following code to delete the categories and Nan starting with numbers through regular matching:

```
process_df = process_df[~df['Category'].str.contains(pat=r'^[0-9]|NaN', regex=True)]
```

Category and AndroidVer

In this section, I want to check the number of apps that are suitable for different Android versions under different categories. Therefore, we need to find out that AndroidVer is null based on the above. At the same time, if there are too many categories and the data presentation is not good-looking, we have selected the top ten apps with the following code:

```
# Remove Andorid Ver NaN
process_df = df[~df['Android Ver'].str.contains(pat=r'NaN', regex=True)]
# Extract and divide versions and categories, and count
ver_category_df = process_df.groupby(['Android Ver', 'Category'])['App'].nunique().reset_in
# Take out the top 10 categories in the total
categories = ver_category_df["Category"].value_counts().head(10).index
# Extract the number of apps contained in these categories and versions
ten_df = ver_category_df[ver_category_df['Category'].str.contains(pat=r"|".join(categories)
```

Category and Ratings, Installs, Reviews

In this section, we will show the data about the ratings, installs and reviews of different types of apps. Similarly, we need to eliminate those with empty scores in the data. We select the top 4 apps in the top five categories with average scores to divide the equal intervals and check the boundaries of each interval. The key code is as follows:

```
# remove NaN, key can take the following valuesRating, Installs, Reviews
rm_rating_NaN_df = process_df[~process_df[key].str.contains(pat=r'NaN', regex=True)][['App'
# transform to numeric
rm_rating_NaN_df[key] = rm_rating_NaN_df[key].apply(pd.to_numeric)
# calculate the average score to get the top 5 categories of average score
rating_categories = rm_rating_NaN_df.groupby('Category')[key].mean().sort_values(ascending=
# b are intervals
a, b = pd.cut(rm_rating_NaN_df[rm_rating_NaN_df['Category'] == category][key], bins=3, retb
```

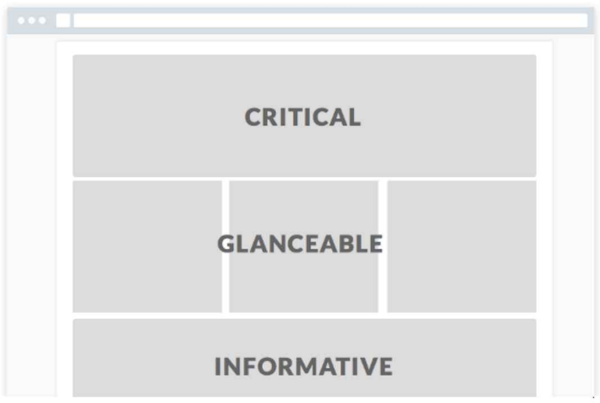
Category Rating Distribution

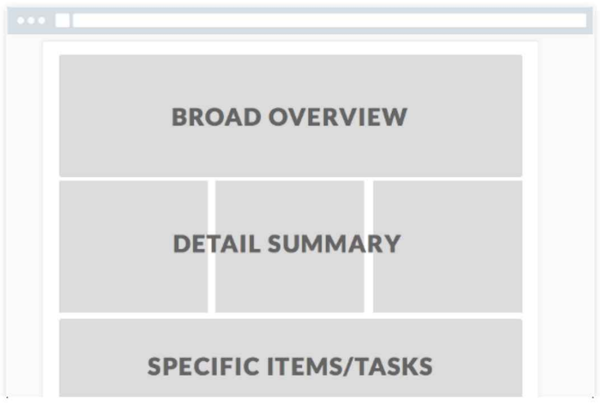
In this part, after removing the apps rated NaN, we extract the score distribution range of each category to view the data, take the maximum and minimum scores of a specific category, and then view the distribution. The key code is as follows:

```
# remove NaN
rm_rating_NaN_df = process_df[~process_df['Rating'].str.contains(pat=r'NaN', regex=True)][[
# select a specific kind of data
rm_rating_NaN_df = rm_rating_NaN_df[rm_rating_NaN_df['Category'] == catagory]
# partition interval
quartiles = pd.cut(rm_rating_NaN_df['Rating'], bins=bins)
```

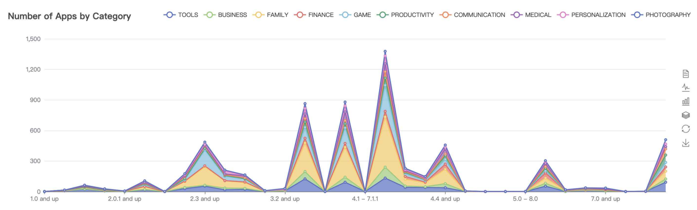
Describe the layout

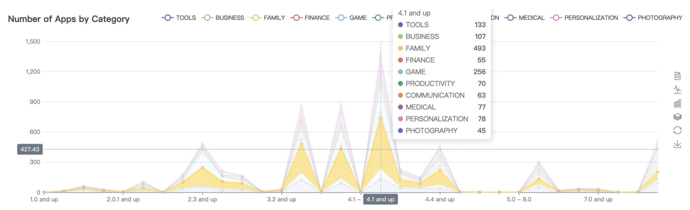
Layout as follows:



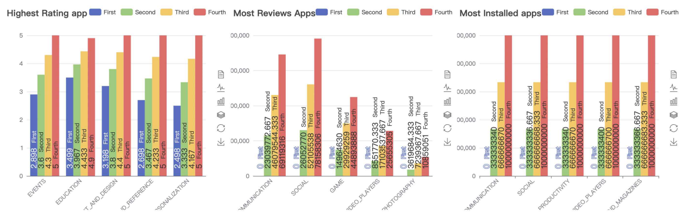


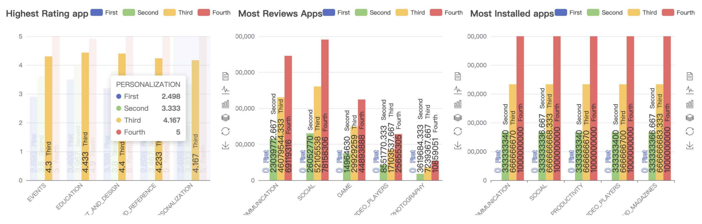
This section shows how the number of the top ten kinds of apps changes with Android Version. It can be seen that the types of apps in the intermediate version are the most, and the number of low version and high version apps is less.



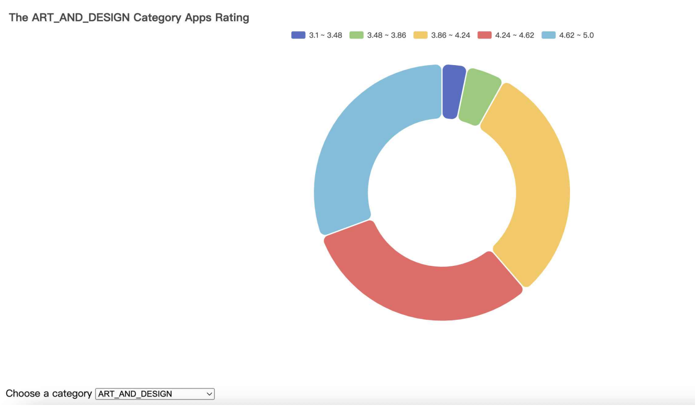


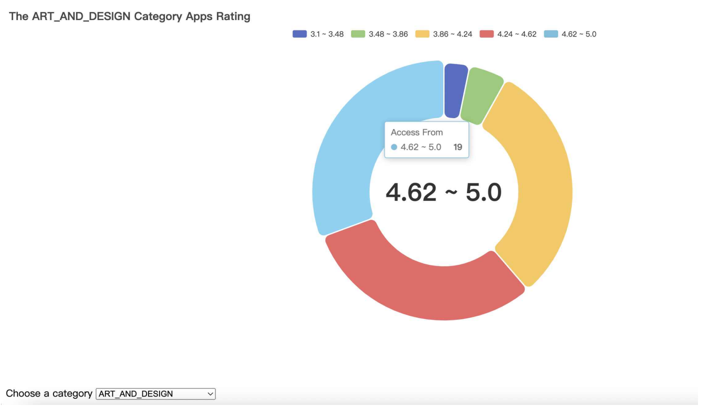
This part mainly shows the boundaries of each interval after the software category with the most rating / installs / reviews is divided into equal intervals, so that users can quickly understand the situation of multiple types of software.





This part mainly shows the score distribution of a specific kind of software, so that users can quickly understand the situation of this kind of software.


The ART_AND_DESIGN Category Apps Rating
3.1 ~ 3.48
3.48 ~ 3.86
3.86 ~ 4.24
4.24 ~ 4.62
4.62 ~ 5.0
Choose a category
ART_AND_DESIGN


The ART_AND_DESIGN Category Apps Rating
3.1 ~ 3.48
3.48 ~ 3.86
3.86 ~ 4.24
4.24 ~ 4.62
4.62 ~ 5.0
Access From
4.62 ~ 5.0 19
4.62 ~ 5.0
Choose a category
ART_AND_DESIGN