



THE VISUALIZATION PROJECT

For Google Play Store

Xinzhuo Yu
28559134

Table of Contents

1. Introduction.....	2
2. Design.....	2
3. Implementation:.....	2
changes.....	3
Implementation.....	3
Libraries.....	3
Code	3
4. User guide	5
5. Conclusion	9
6. References.....	9
7. Appendix: 5 design sheets	9

1. Introduction

Data visualization has become a common part of every individual's daily life, with the rising usage of big data. Data visualization is not only showing data in different colours and shapes, they need a story--something to show, or, more aptly, to "tell" visually--and finding this tale isn't always obvious when digging through data ("Data Visualization and Data Storytelling: A Visual Revolution.", 2018). Data visualization tells stories.

Among the varies data visualizations, the interactive data visualization is the most trendy and acceptable visualization. The visualizations that are introduced in this report use interactive visualization technique to show a series data in Google Play Store. In these visualizations, we aimed to display the distribution of application rating and reviews and their relationship, the application categories and types distribution, and rating content key words.

The intend audience can be every application user who is interested in application data or the analyst in relative area. However, the main audience are the application developing companies, who will treat the data visualization as an important document for further application developing projects.

2. Design

The first stage of the design is getting and filtering ideas. In the Google Play Store datasets, there are multiple columns that can be analysed, but in the visualizations, the columns that will be selected should be meaningful and informative. In this stage, the data columns: "Category", "Rating", "Content Rating", "Review", "Type" are selected to generate ideal visualizations. For the visualization shapes, bar chart, box plot, scatter plot, pie chart, and word cloud are selected (see Appendix: Sheet 1).

The second stage is to design and draw draft layout and decide what interactives the visualizations may include. In this stage, three design sheets are created for different versions of the layout and interaction design. Sheet 2 is the rough draft that shows only basic layout. Sheet 3 has been improved from sheet 2, added the use of varies colours. Sheet 4 is the most completed draft, except the colours, the shape design is also considered (e.g. the centre radius of pie chart, see Appendix: Sheet 2, 3, 4).

The last stage is to decide the final version. In the last version, bar chart, box plot, pie chart and word cloud are used, also, except word cloud, all other visualizations are interactive. If the user clicks a part of certain visualizations, all other visualizations except word cloud will change according to the data that has be selected (clicked). The amount of visualizations will be six (see Appendix: Sheet 5).

3. Implementation:

changes

In this implementation, several small parts are changed from the final design version.

Firstly, the box plot is replaced by bar chart, because we are aiming at implementing the interaction between all visualization except word cloud. It will not be a problem to draw data in box plot shape, however, it will be a challenge to implement the interactive function, under the use of dc library. The scatter plot is replaced by bubble chart, since the bubble chart is more informative and more visual-friendly.

Secondly, the layout is slightly changed, the bubble chart is moved to another line, in case the visualizations are too crowded.

Lastly, the column for displaying pie chart and word cloud is changed because I found it is better to display “Category” as a pie chart. According to the design, the word cloud will filter all the words in category and pick the high frequency ones to display. However, the data has thousands of rows, every row’s category include multiple words. It will due to a significant delay when the page is loading, and it has negative influence on user experience. A backend database will be required for this need. Since it is not really possible to build a database here, the words that are displayed on the word cloud are pre-set. In the data wrangling part, I sorted the “Content Rating” column by frequency high to low and set the word size from large to small in word cloud.

Implementation

The implementation including the coding part for all visualizations and interactions.

Libraries

Crossfilter and dc: Exploring large multivariate datasets faster in the browser.

Bootstrap: Make the view clearer and keep the horizontal of visualizations in one row.

Cloud: Create word cloud.

Scale: Use a scale of colour.

Code

Below is the bubble chart part of the code.

```

dc.bubbleChart('#bubblechart')
// (_optional_) define chart transition duration, `default = 750`
.transitionDuration(1500)
.dimension(rateAndViewDimension)
.group(rateAndViewDimension.group())
.colorAccessor(function(d){

    if(d.key[0] < 2)
        return "000";
    else if (d.key[0] >= 2 && d.key[0] < 3)
        return "1000";
    else if (d.key[0] >= 3 && d.key[0] < 4)
        return "2000";
    else
        return "3000";
})
// `.keyAccessor` - the `X` value will be passed to the `'.x()` scale to determine pixel location
.keyAccessor(function (p) {
    return p.key[0];})
// `.valueAccessor` - the `Y` value will be passed to the `'.y()` scale to determine pixel location
.valueAccessor(function (p) {
    return p.key[1];})
// `.radiusValueAccessor` - the value will be passed to the `'.r()` scale to determine radius size;
.radiusValueAccessor(function (p) {
    return p.value/10;})
.maxBubbleRelativeSize(0.3)
.x(d3.scaleLinear().domain([1,5]))
.y(d3.scaleLinear().domain([100,200000]))
.r(d3.scaleLinear().domain([0, 20]))
//`.elasticY` and ` `.elasticX` determine whether the chart should rescale each axis to fit the data.
.elasticY(true)
.elasticX(true)
.yAxisPadding(20)
//.xAxisPadding(500)
// (_optional_) render horizontal grid lines, `default=false`
.renderHorizontalGridLines(true)
// (_optional_) render vertical grid lines, `default=false`
.renderVerticalGridLines(true)
// (_optional_) render an axis label below the x axis
.xAxisLabel('Rating')
// (_optional_) render a vertical axis lable left of the y axis
.yAxisLabel('Reviews')
.renderLabel(false)
.render();

```

This part code is partly from an open source code, added appropriate logic and deleted unnecessary elements. Retrieved from <https://dc-js.github.io/dc.js/>.

There are 6 visualization in total, the code of bar charts, pie charts and bubble charts are quite similar, due to the limitation of this report, only the code of word cloud will be provided.

The code of word cloud is shown below.

```

    // set the dimensions and margins of the graph
var margin = {top: 0, right: 10, bottom: 10, left: 10},
    width = 450 - margin.left - margin.right,
    height = 450 - margin.top - margin.bottom;

// append the svg object to the body of the page
var svg = d3.select("#my_dataviz").append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
    .append("g")
    .attr("transform",
        "translate(" + margin.left + "," + margin.top + ")");

// Constructs a new cloud layout instance. It runs an algorithm to find the position of words that suits your
// requirements
// Wordcloud features that are different from one word to the other must be here
var layout = d3.layout.cloud()
    .size([width, height])
    .words(myWords.map(function(d) { return {text: d.word, size:d.size}; }))
    .padding(5)           //space between words
    .rotate(function() { return ~~(Math.random() * 2) * 90; })
    .fontSize(function(d) { return d.size; })      // font size of words
    .on("end", draw);
layout.start();
function draw(words) {
    svg
        .append("g")
        .attr("transform", "translate(" + layout.size()[0] / 2 + "," + layout.size()[1] / 2 + ")")
        .selectAll("text")
        .data(words)
        .enter().append("text")
        .style("font-size", function(d) { return d.size; })
        .style("fill", "#69b3a2")
        .attr("text-anchor", "middle")
        .style("font-family", "Impact")
        .attr("transform", function(d) {
            return "translate(" + [d.x, d.y] + ")rotate(" + d.rotate + ")";
        })
        .text(function(d) { return d.text; })
        .render();
}
}

```

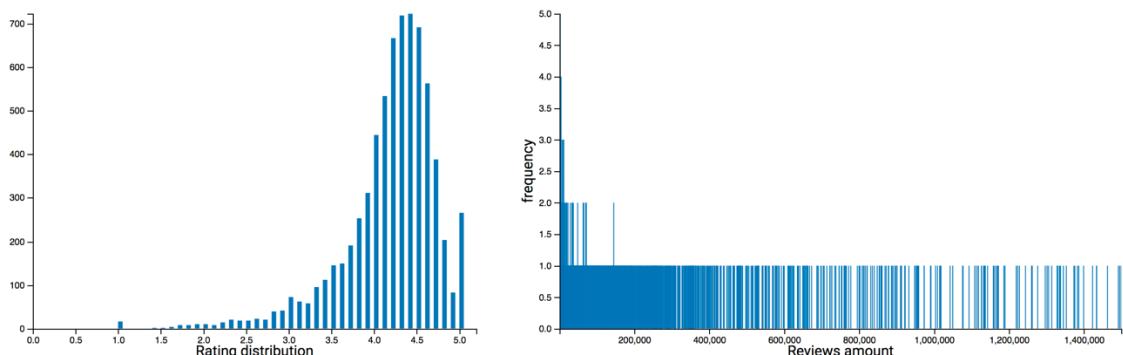
Partly retrieved from https://www.d3-graph-gallery.com/graph/wordcloud_size.html.

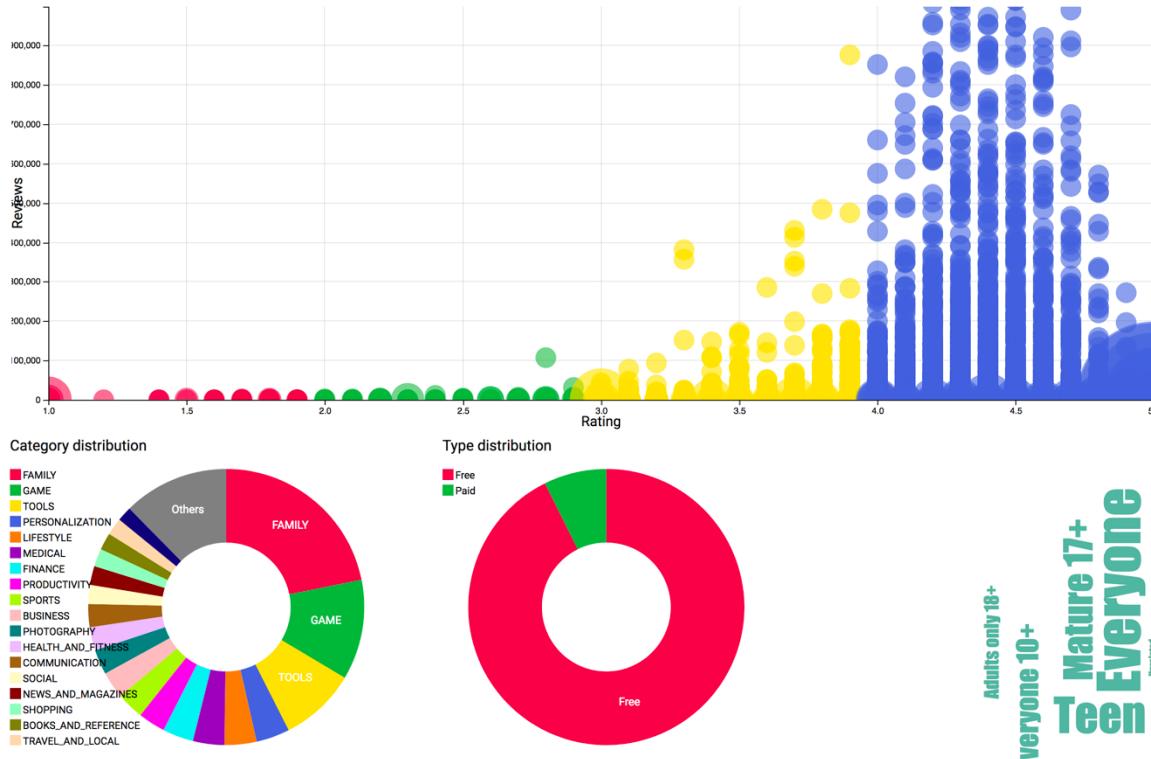
4. User guide

Instructions for viewing and exploring the narrative visualisation using a standard web browser and images showing how the visualization works.

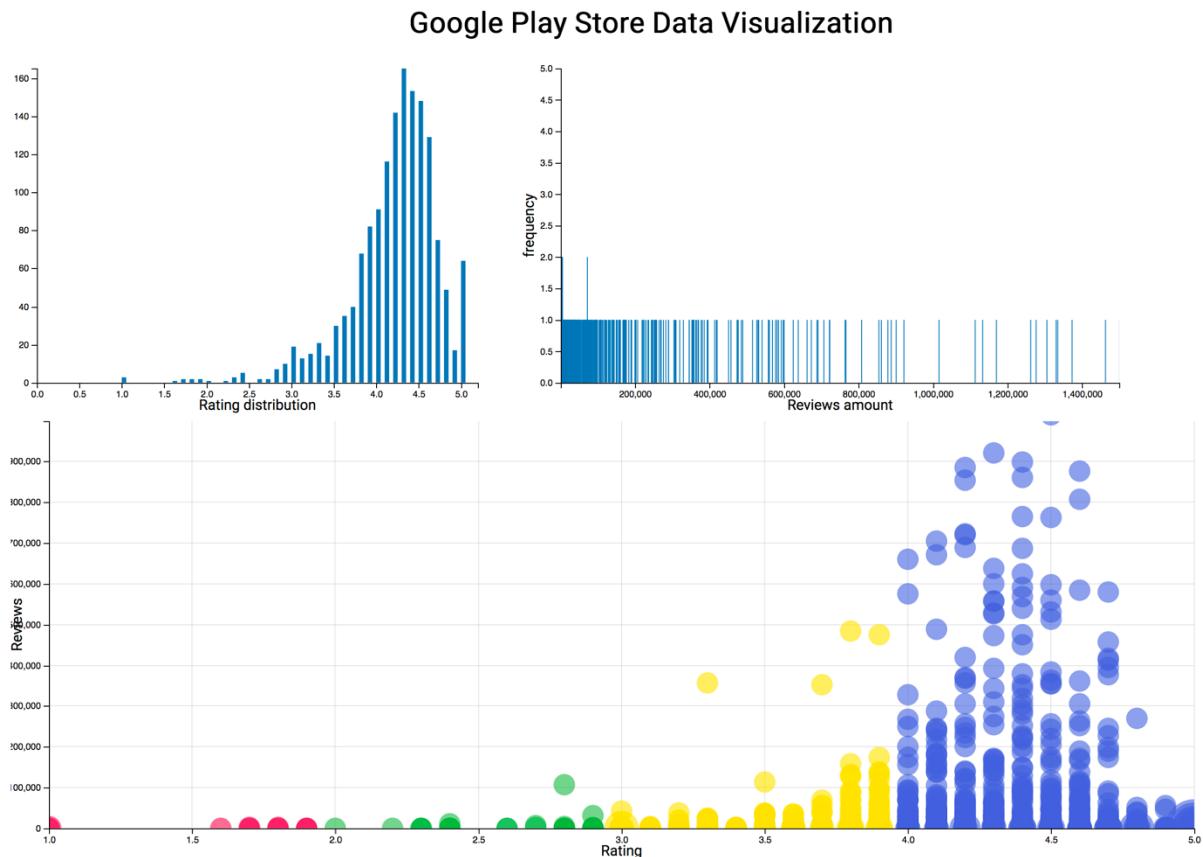
The js code works perfectly with Firefox browser. When it is run, the page layout is display as below.

Google Play Store Data Visualization

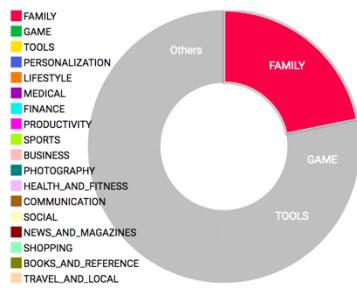




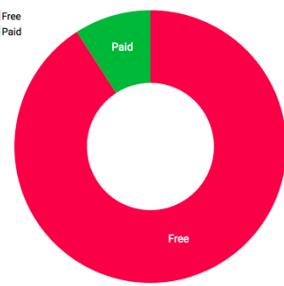
Every visualization except word cloud is clickable. For example, I clicked “Famliy” part in Category pie chart, then the data set will be narrowed down to only family category. The visualizations change as below.



Category distribution



Type distribution

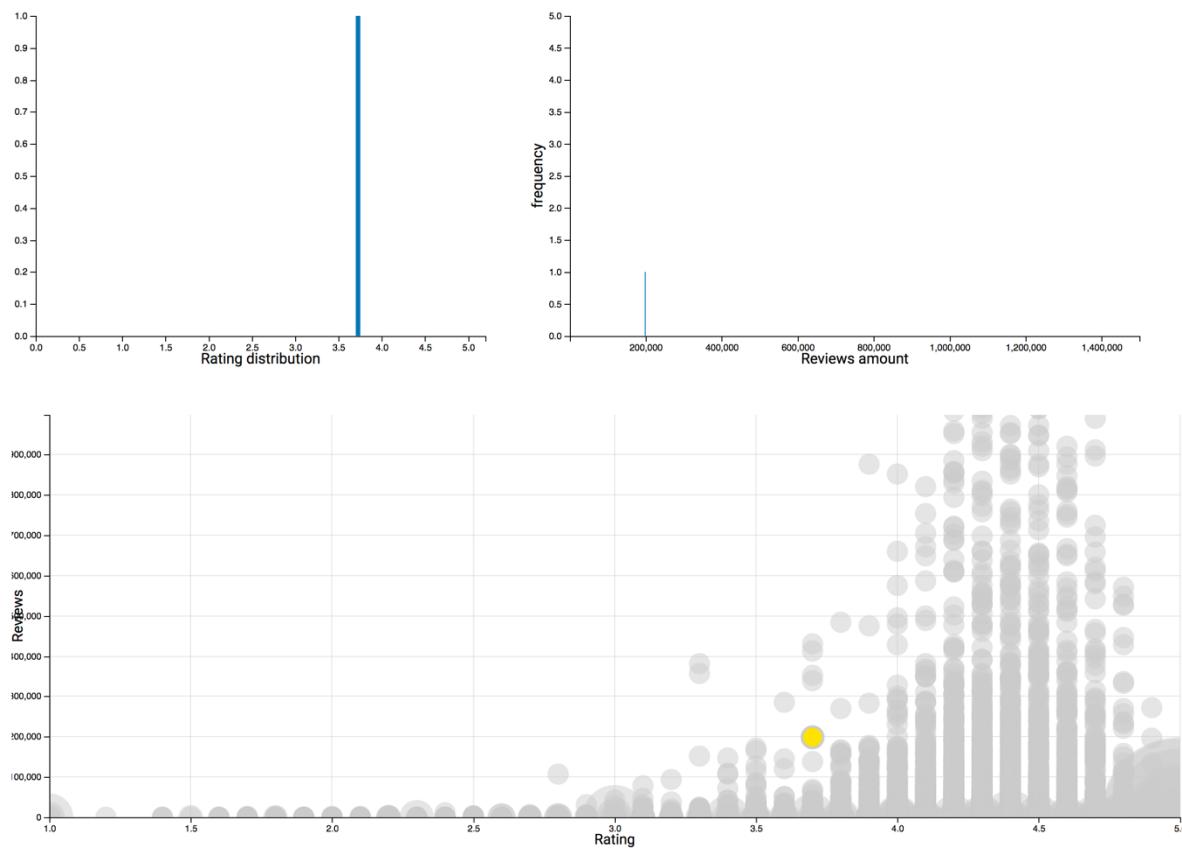


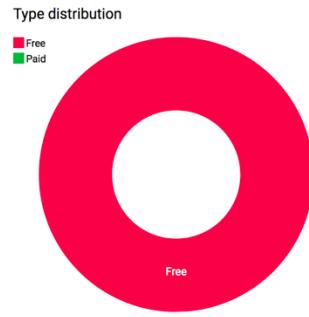
Everyone
Teen
Mature 17+
Everyone 10+
Adults only 18+
Unrated

If I click a bubble on bubble chart, which represent a certain row of data, the relative rating, review amount, category and type will be shown, and all other rows of data on the bubble chart will be grey out.

The visualization now is shown as below.

Google Play Store Data Visualization

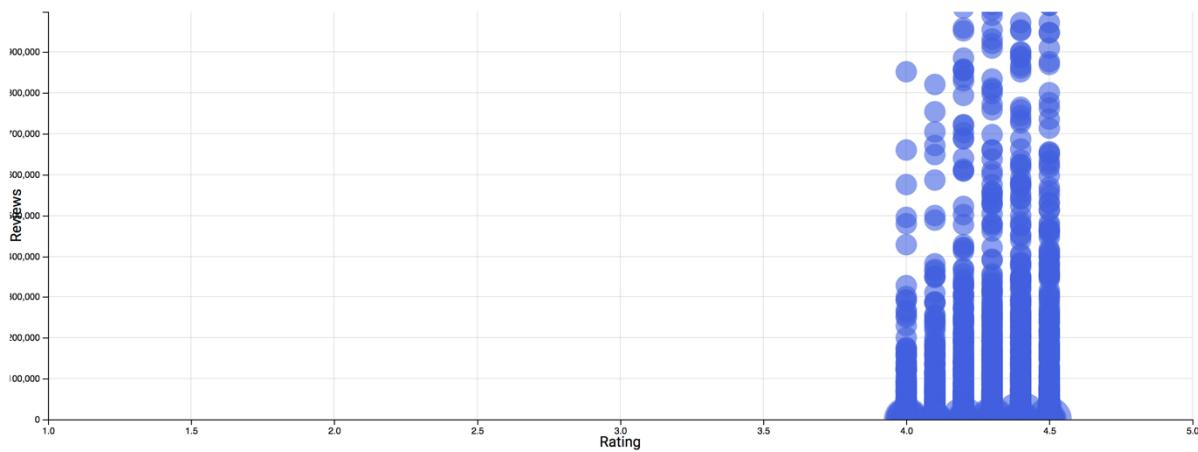
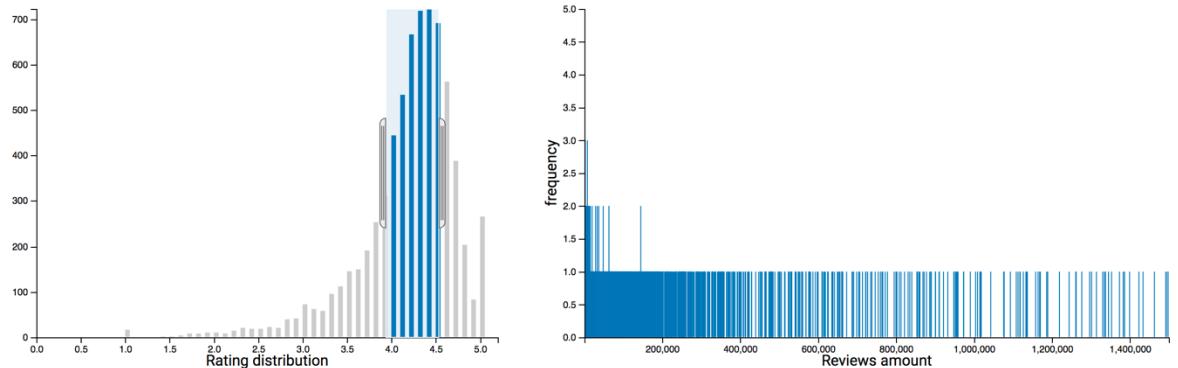




Mature 17+
Everyone
Teen
Unrelated
Adults only 18+
Everyone 10+

Ranges are also selectable on bar charts. For example, I select rating range from 4 to 4.5, then the visualization displays as shown below.

Google Play Store Data Visualization



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

If user want to cancel selection, just click again when selecting a certain part, or click elsewhere when selecting a range, or even refresh the page, it will be reset to the original data.

5. Conclusion

In this project, the data of Google Play Store has been deeply analysed and interactive visualizations have been created. It is a project that implemented well, and I will be proud of. During the design and implement part in this project, I was inspired by many beautiful code and skilful people online and in the real words. This project still has multiple aspects that can be improve, and I will keep learning more technique that will make every future project better.

6. References

d3/d3-scale. (2019). Retrieved from <https://github.com/d3/d3-scale>

DC.js Bubble Chart. (2019). Retrieved from
https://www.tutorialspoint.com/dcjs/dcjs_bubble_chart.html

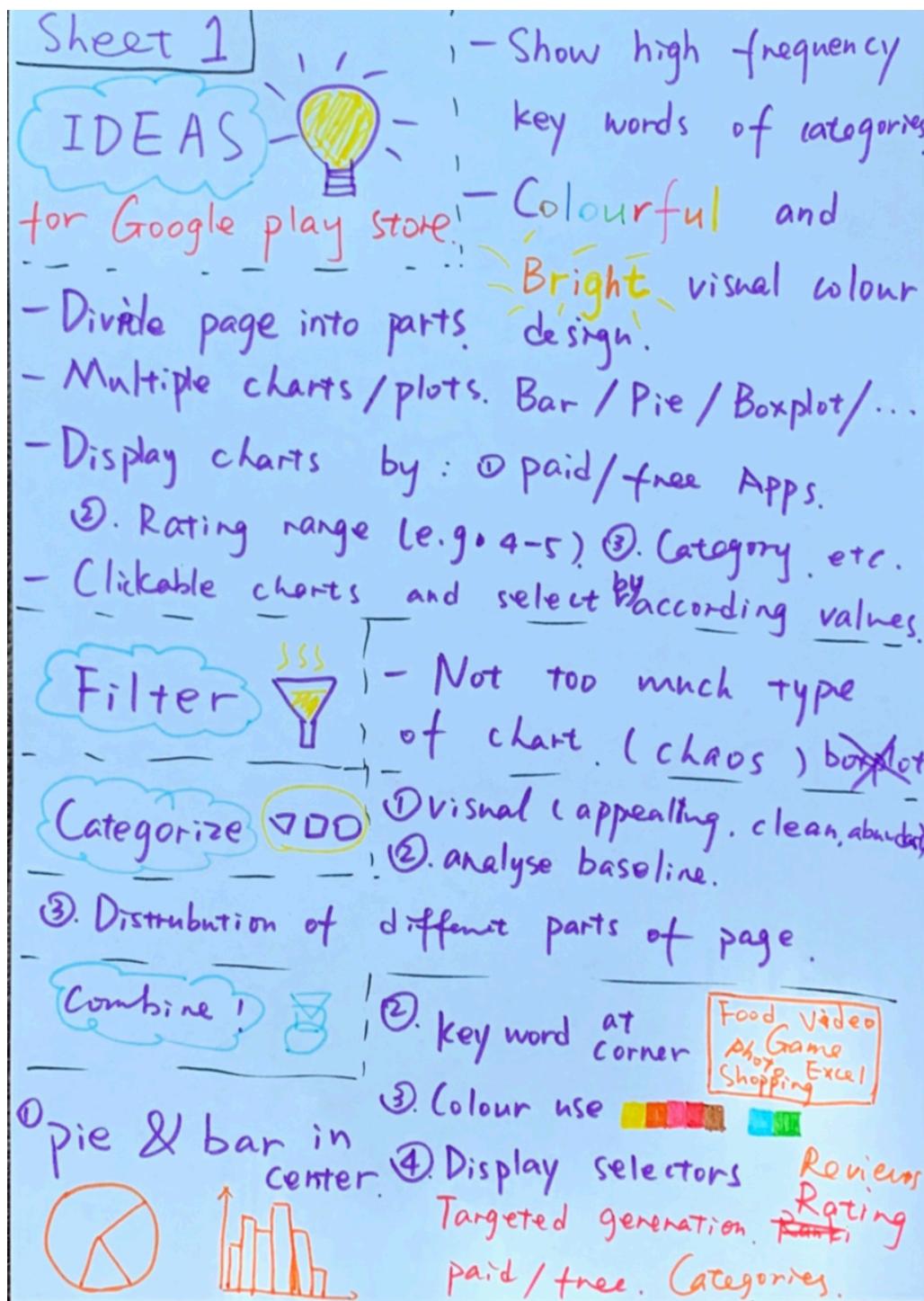
dc.js - Dimensional Charting Javascript Library. (2019). Retrieved from <https://dc-js.github.io/dc.js/>

Holtz, Y. (2019). Wordcloud with proportional font size in d3.js. Retrieved from
https://www.d3-graph-gallery.com/graph/wordcloud_size.html

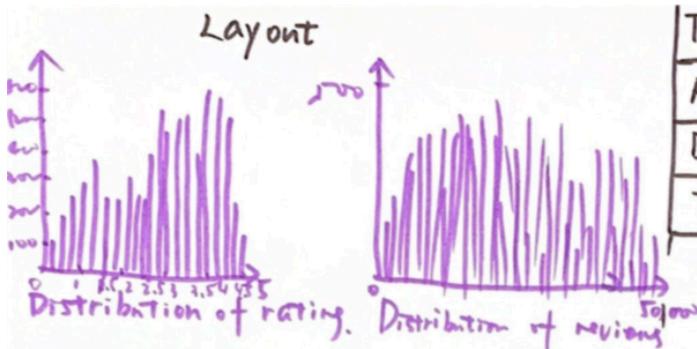
"Data Visualization and Data Storytelling: A Visual Revolution." Big Data Quarterly, Spring 2018, p. 41. Computer Database,
<http://link.galegroup.com/apps/doc/A536398907/CDB?u=monash&sid=CDB&xid=edb72167>. Accessed 8 June 2019.

7. Appendix: 5 design sheets

Sheet 1



Sheet 2



TITLE	Google play store.
Author	Xinzhuo Yu 28559139
Date	30/May Sheet 2
Task	Comphote version 1



Focus / zoom

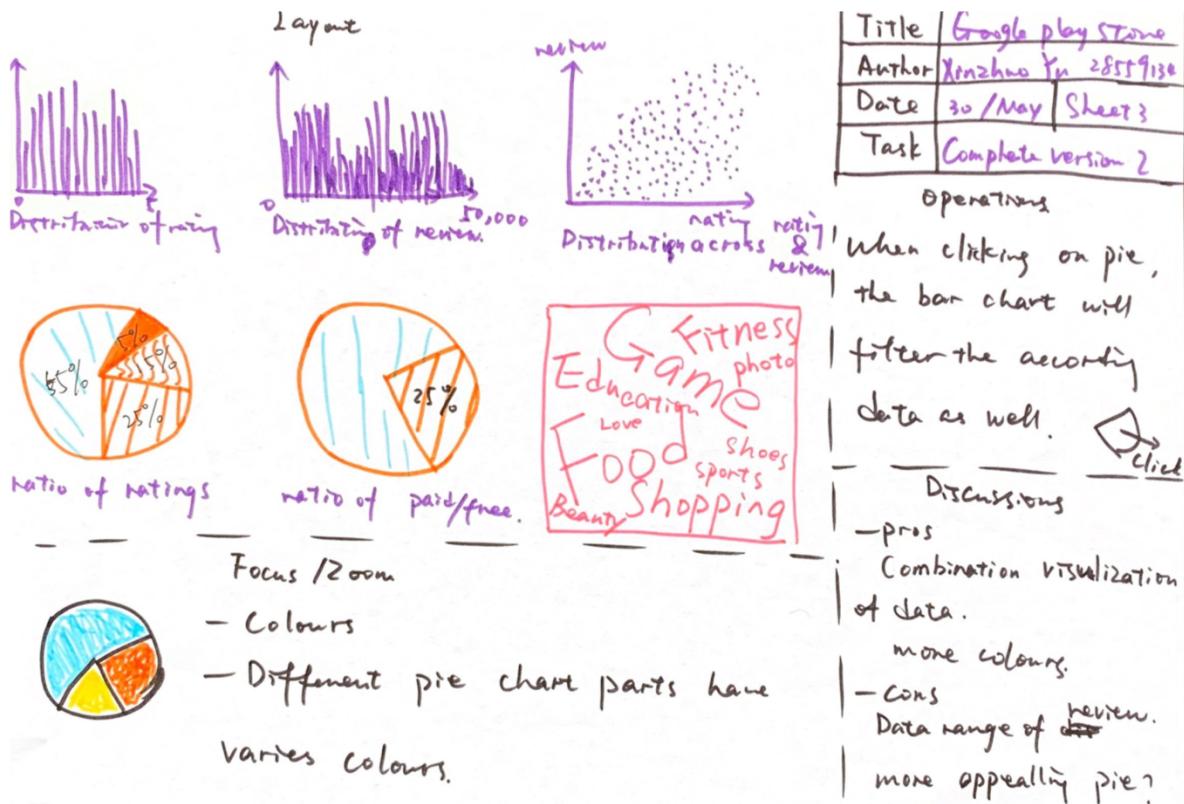
- bar chart.
the ratio will display
on the part of pie chart
accordingly.

- The key words
with high frequency
will be longer.

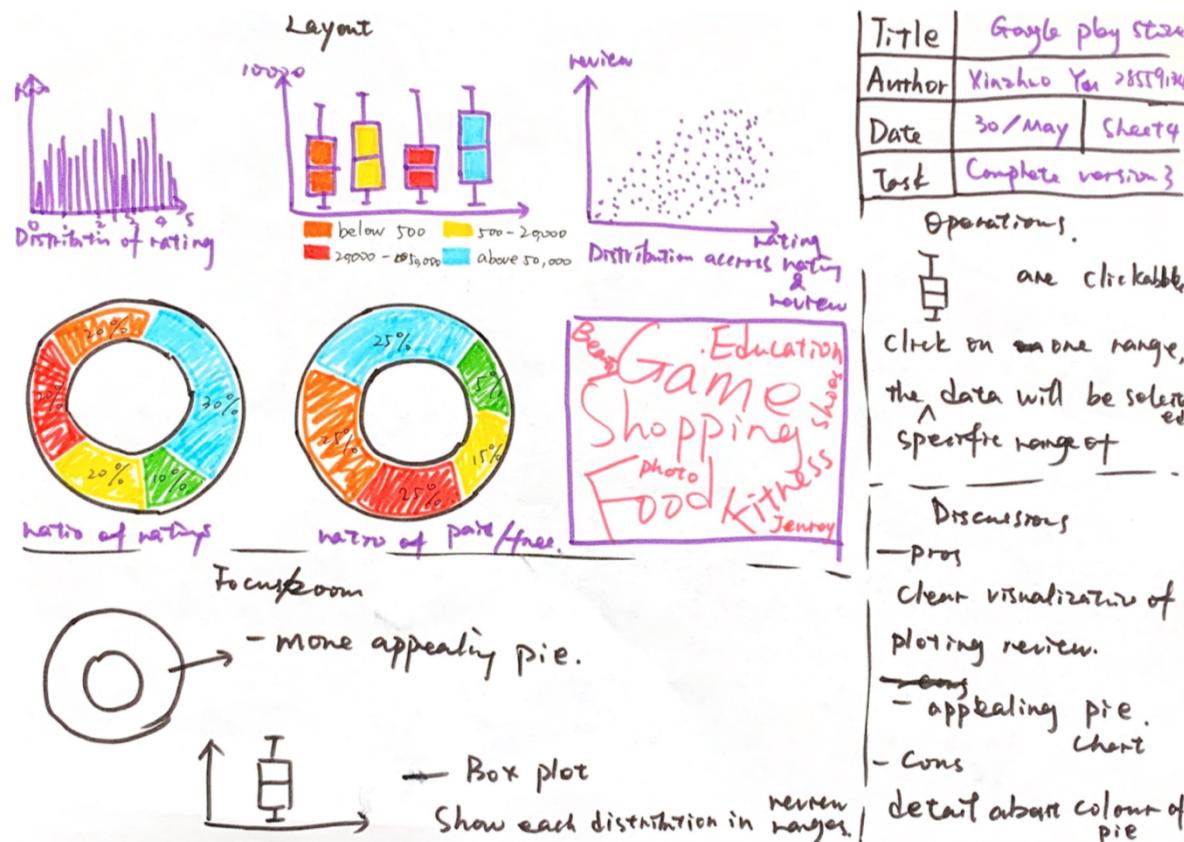
operations

- | if click on the a certain part of pie chart, the range for all bar/pie chart changes.
- | → clickable

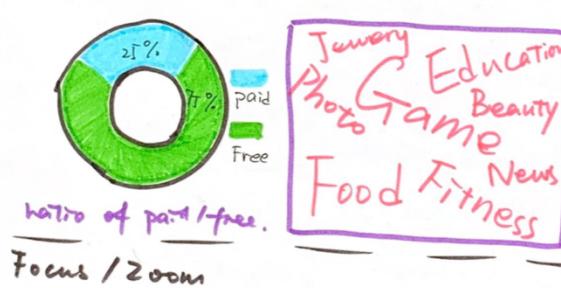
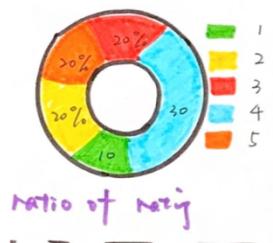
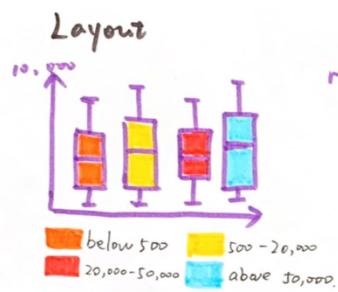
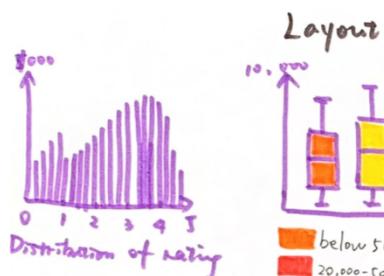
-
- | Discussion
- | - pros
interactive.
show data in different visualizations.
- | - cons
all ~~the~~ are ~~say~~
single data, no combination of different data.



Sheet 4



Sheet 5



Title	Google play store
Author	Xinheuo Yu > 11794
Date	30/May Sheet 5
Task	Complete final version

Operations

- Both box plot and pie chart are clickable.
- The name of key words will turn bigger when on.
- Detail
- Will be built in d3.
- May use resources in interaction part.
- Estimate time: 2 weeks.