Data Science

Introduction To Data Science Project

Google Play Store Dataset Analysis

- Manoj Pissay A

Dataset Chosen: Google Play Store

- Number of rows: 10,842 rows
- Number of columns: 13 columns

Categorical Data

- Category
- Type
- Content Rating
- Last Updated
- Current Ver
- Android Ver
- Genres

Numerical Data

- Installs
- Rating
- Reviews
- Size
- Price

Modules Used

sem3 > cleaning.py

cleaning

data.py | cleaning.py

```python
import re
import scipy.stats as s
import matplotlib.pyplot as plt
import matplotlib.colors as col
import numpy as np
import random
import csv
```

Data Cleaning

App

B	C	D	E	F	G
gory	Rating	Reviews	Size	Installs	Type
AND	4.1	159	19M	10,000+	Free
AND	3.9	967	14M	500,000+	Free
AND	4.7	87510	8.7M	5,000,000+	Free
AND	4.5	215644	25M	50,000,000	Free
AND	4.3	967	2.8M	100,000+	Free
AND	4.4	167	5.6M	50,000+	Free
AND	3.8	178	19M	50,000+	Free
AND	4.1	36815	29M	1,000,000+	Free
AND	4.4	13791	33M	1,000,000+	Free
AND	4.7	121	3.1M	10,000+	Free
AND	4.4	13880	28M	1,000,000+	Free
AND	4.4	8788	12M	1,000,000+	Free
AND	4.2	44829	20M	10,000,000	Free
AND	4.6	4326	21M	100,000+	Free
AND	4.4	1518	37M	100,000+	Free
AND	3.2	55	2.7M	5,000+	Free
AND	4.7	3632	5.5M	500,000+	Free
AND	4.5	27	17M	10,000+	Free
AND	4.3	194216	39M	5,000,000+	Free
AND	4.6	224399	31M	10,000,000	Free
AND	4	450	14M	100,000+	Free
AND	4.1	654	12M	100,000+	Free

B	C	D	E	F	G
Category	Rating	Reviews	Size	Installs	Type
ART_AND_	4.1	159	19	10000	Free
ART_AND_	3.9	967	14	500000	Free
ART_AND_	4.7	87510	8.7	5000000	Free
ART_AND_	4.5	215644	25	50000000	Free
ART_AND_	4.3	967	2.8	100000	Free
ART_AND_	4.4	167	5.6	50000	Free
ART_AND_	3.8	178	19	50000	Free
ART_AND_	4.1	36815	29	1000000	Free
ART_AND_	4.4	13791	33	1000000	Free
ART_AND_	4.7	121	3.1	10000	Free
ART_AND_	4.4	13880	28	1000000	Free
ART_AND_	4.4	8788	12	1000000	Free
ART_AND_	4.2	44829	20	10000000	Free
ART_AND_	4.6	4326	21	100000	Free
ART_AND_	4.4	1518	37	100000	Free
ART_AND_	3.2	55	2.7	5000	Free
ART_AND_	4.7	3632	5.5	500000	Free
ART_AND_	4.5	27	17	10000	Free
ART_AND_	4.3	194216	39	5000000	Free
ART_AND_	4.6	224399	31	10000000	Free
ART_AND_	4	450	14	100000	Free
ART_AND_	4.1	654	12	100000	Free

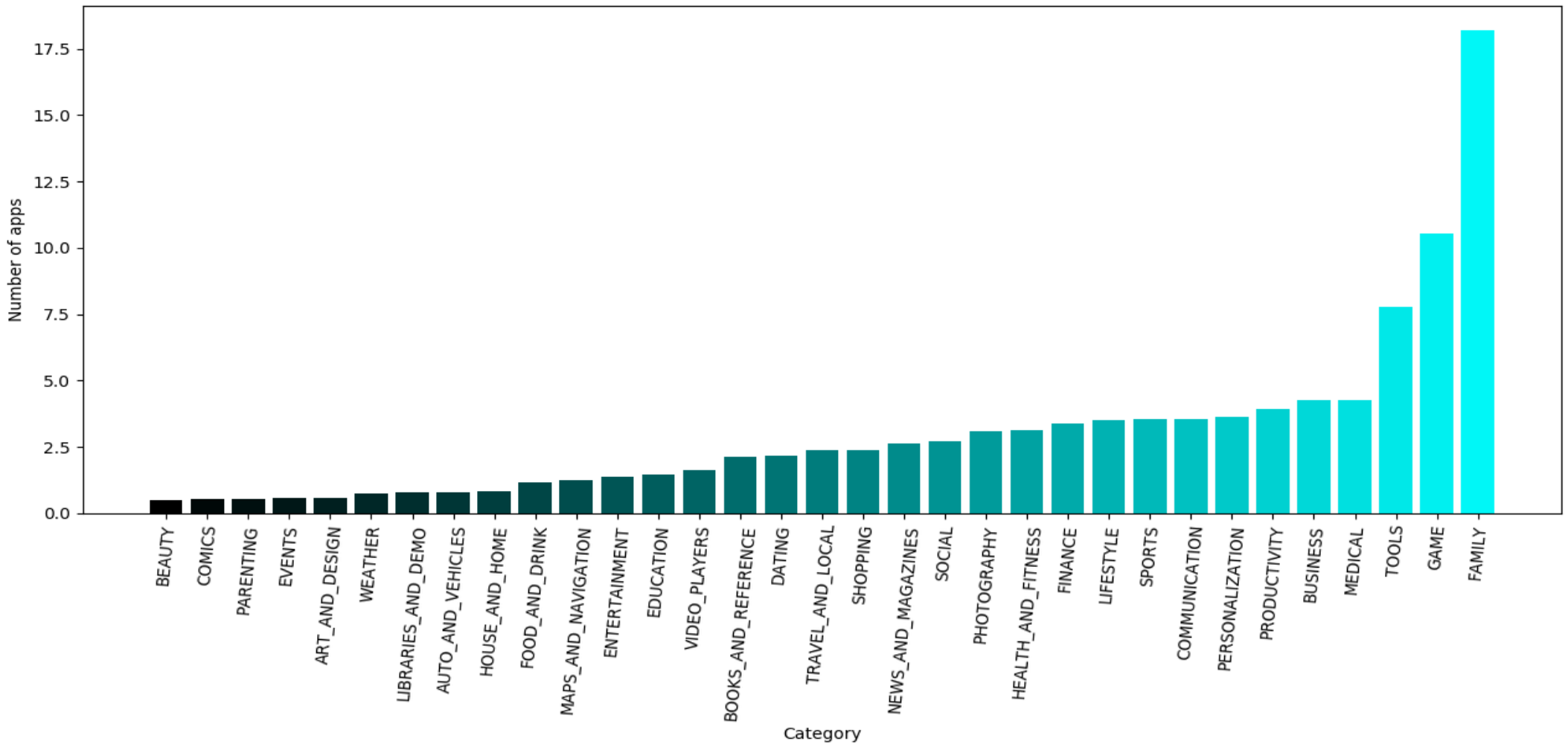
O12

	A	B	C	D	E	F
112	Sweet Self	BEAUTY	4.3	601	35M	100,000+
113	Colors of v	BEAUTY	4.5	36	6.7M	10,000+
114	Selfie Cam	BEAUTY	4.1	187	30M	50,000+
115	Wrinkles a	BEAUTY	NaN	182	5.7M	100,000+
116	Eyes Make	BEAUTY	4.2	30	2.9M	10,000+
117	Photo Edit	BEAUTY	4.5	134	17M	10,000+
118	Step By Ste	BEAUTY	4.4	74	2.9M	10,000+
119	Beauty Ca	BEAUTY	4	113715	Varies with	10,000,000
120	Girls Hairst	BEAUTY	4.1	3595	Varies with	500,000+
121	Mirror Car	BEAUTY	4.1	9315	2.6M	1,000,000+
122	Beauty Tip	BEAUTY	4.4	75	4.2M	50,000+
123	Haircut Tu	BEAUTY	4.6	38	7.1M	10,000+
124	Sephora: S	BEAUTY	4.5	26834	57M	1,000,000+
125	Manicure -	BEAUTY	NaN	119	3.7M	50,000+
126	Sticker Car	BEAUTY	3.9	2277	22M	500,000+
127	Filters for	BEAUTY	4.4	2280	24M	500,000+
128	Skin Care a	BEAUTY	NaN	654	7.4M	100,000+
129	Facial Wrir	BEAUTY	4.6	184	21M	10,000+
130	Makeup Vi	BEAUTY	3.8	9	3.4M	5,000+
131	Secrets of	BEAUTY	NaN	77	2.9M	10,000+
132	Recipes an	BEAUTY	NaN	35	3.1M	10,000+
133	Discover C	BEAUTY	4	364	6.4M	100,000+
134	Eyeliner st	BEAUTY	4.3	18	3.2M	5,000+
135	Dresses Id	BEAUTY	4.5	473	8.2M	100,000+
136	Lady advis	BEAUTY	NaN	30	9.9M	10,000+
137	Step By Ste	BEAUTY	4.1	66	2.9M	10,000+
138	Rainbow C	BEAUTY	3.7	3871	23M	1,000,000+
139	Methods c	BEAUTY	4.7	257	4.6M	50,000+
140	Girls hairst	BEAUTY	4.2	62	3.1M	10,000+

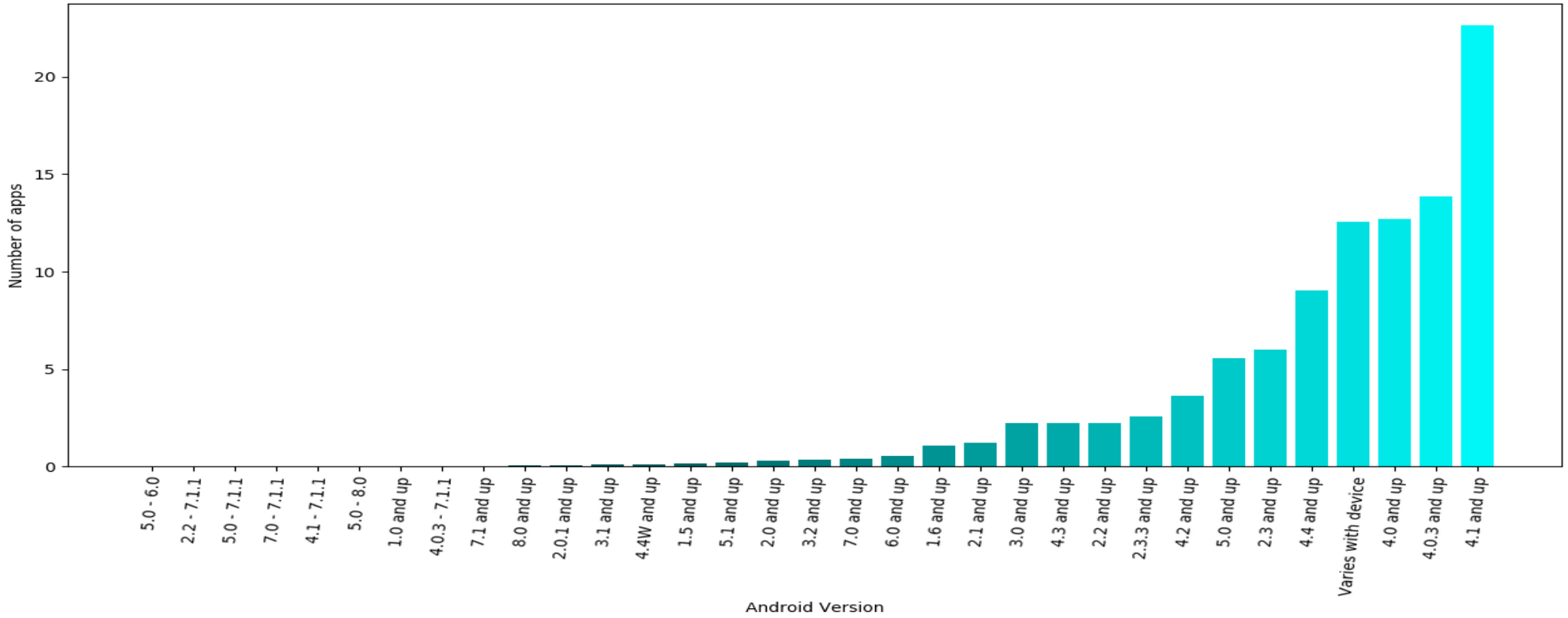
E140 3.1

	A	B	C	D	E
112	Sweet Self	BEAUTY	4.3	601	35
113	Colors of v	BEAUTY	4.5	36	6.7
114	Selfie Cam	BEAUTY	4.1	187	30
115	Wrinkles a	BEAUTY	**4.2**	182	5.7
116	Eyes Make	BEAUTY	4.2	30	2.9
117	Photo Edit	BEAUTY	4.5	134	17
118	Step By Ste	BEAUTY	4.4	74	2.9
119	Beauty Ca	BEAUTY	4	113715	**22.3**
120	Girls Hairst	BEAUTY	4.1	3595	**22.3**
121	Mirror Car	BEAUTY	4.1	9315	2.6
122	Beauty Tip	BEAUTY	4.4	75	4.2
123	Haircut Tu	BEAUTY	4.6	38	7.1
124	Sephora: S	BEAUTY	4.5	26834	57
125	Manicure -	BEAUTY	4.5	119	3.7
126	Sticker Car	BEAUTY	3.9	2277	22
127	Filters for	BEAUTY	4.4	2280	24
128	Skin Care a	BEAUTY	**4.2**	654	7.4
129	Facial Wrir	BEAUTY	4.6	184	21
130	Makeup Vi	BEAUTY	3.8	9	3.4
131	Secrets of	BEAUTY	**4.2**	77	2.9
132	Recipes an	BEAUTY	**4.2**	35	3.1
133	Discover C	BEAUTY	4	364	6.4
134	Eyeliner st	BEAUTY	4.3	18	3.2
135	Dresses Id	BEAUTY	4.5	473	8.2
136	Lady advis	BEAUTY	**4.2**	30	9.9
137	Step By Ste	BEAUTY	4.1	66	2.9
138	Rainbow C	BEAUTY	3.7	3871	23
139	Methods c	BEAUTY	4.7	257	4.6
140	Girls hairst	BEAUTY	4.2	62	3.1

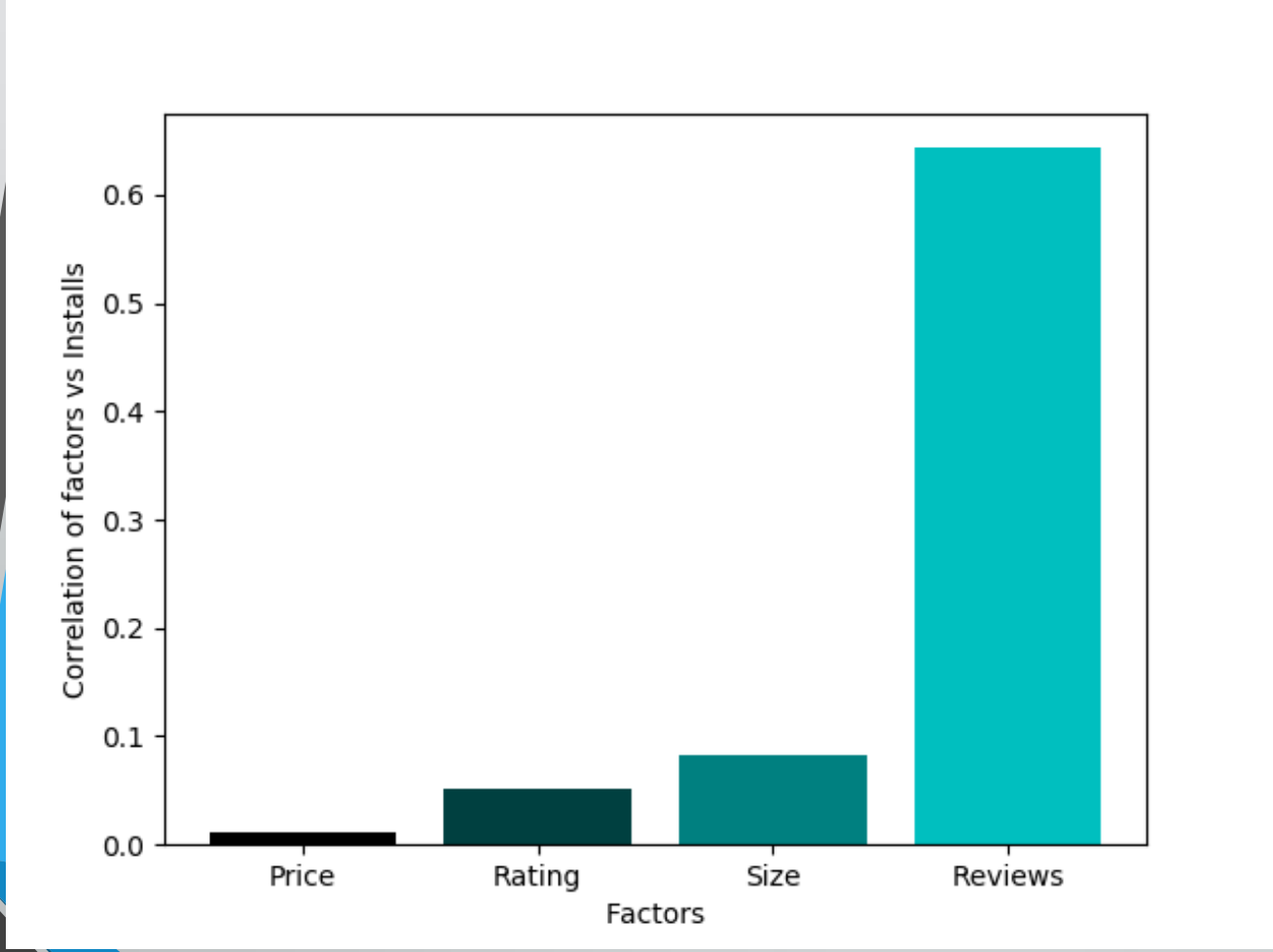
Graph Visualizations



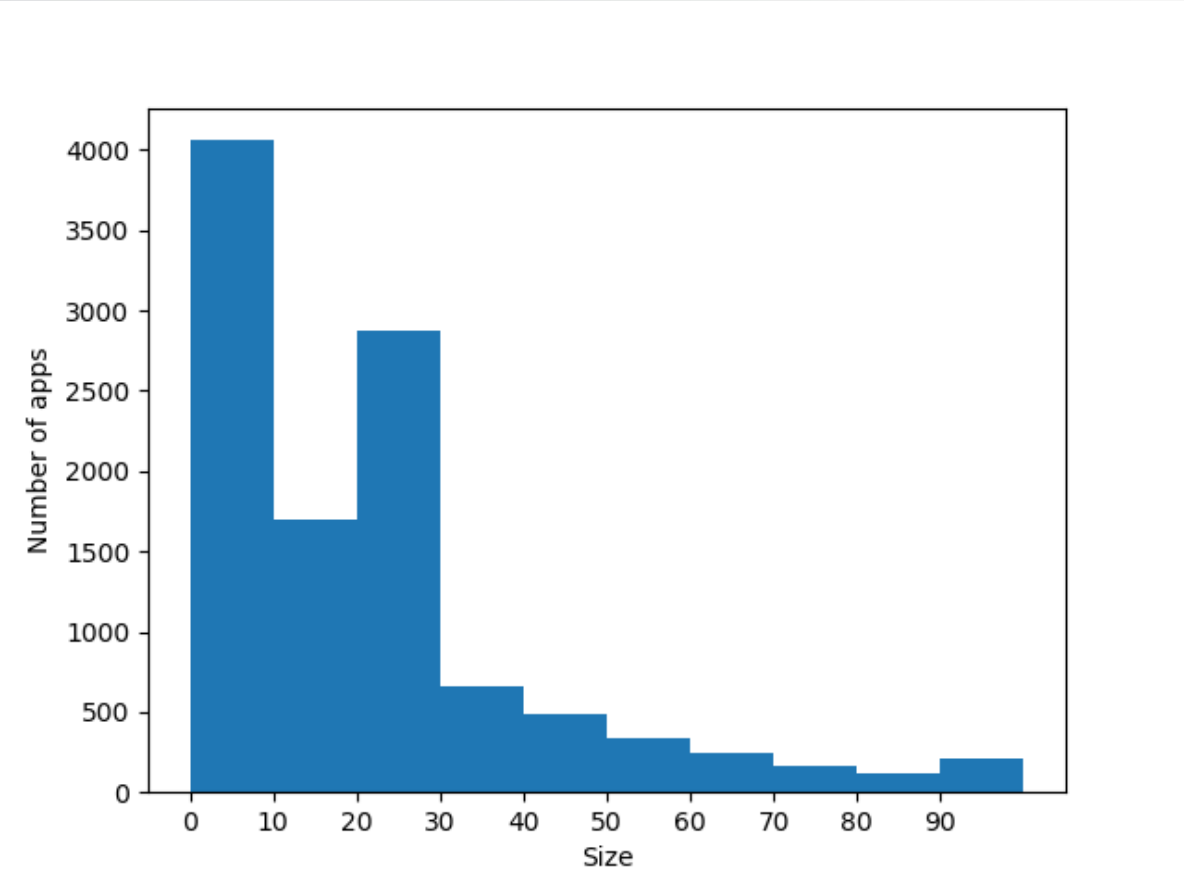
- The most number of apps downloaded are FAMILY based.
- Beauty, Comics, Parenting, Events, Art and Design are the least downloaded apps.



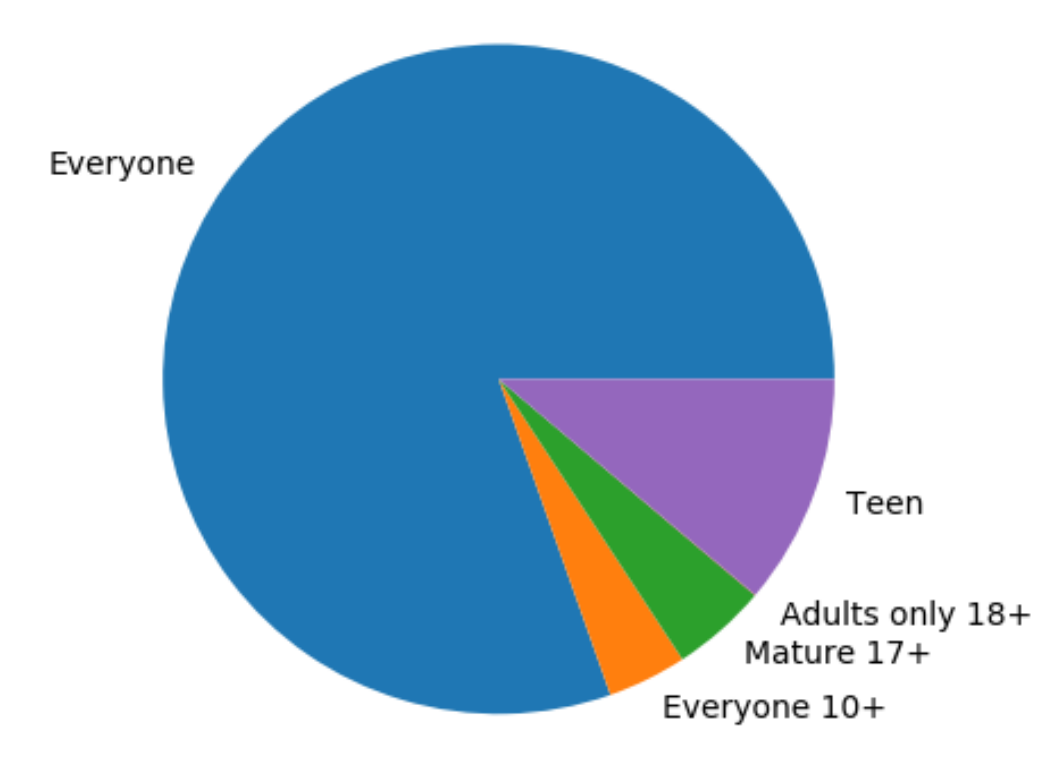
- Most of the apps are designed for android versions of "4.1 and up".
- Most of the apps don't support very recent or very old versions.



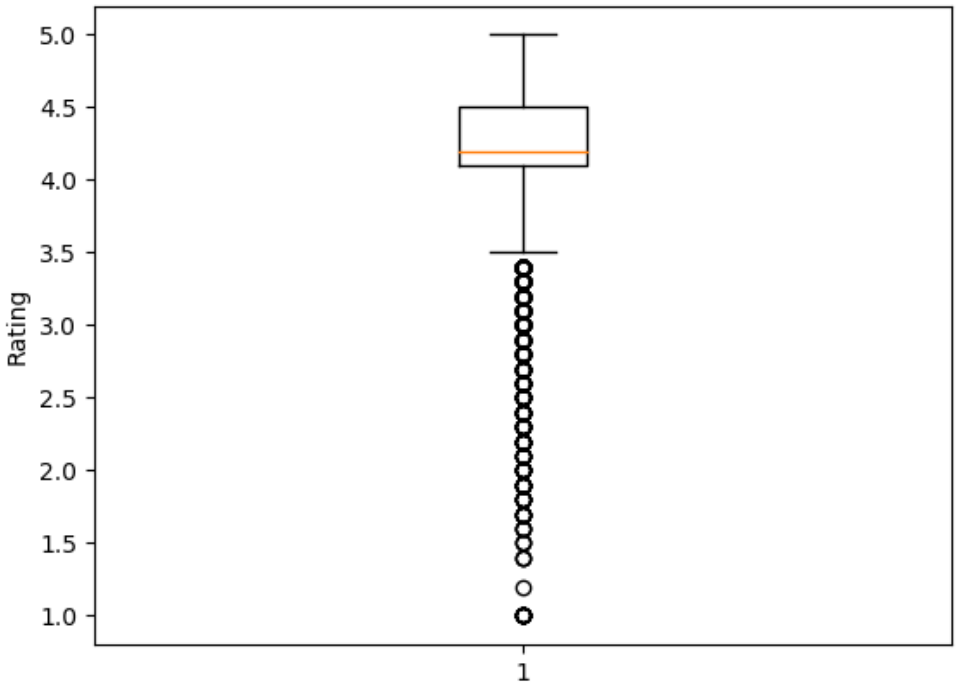
- Reviews and the number of installs are positively correlated.
- The number of installs is independent of the price.



- Most of the apps are in the range of 0-30Mb
- Apps in the range 70-100Mb are few in number.



- More than 75% of the apps are designed for everyone.
- Considerable number of apps target the teenagers.



- The median of the app rating is around 4.2.
- Clearly, a lot of outliers are present in the dataset.

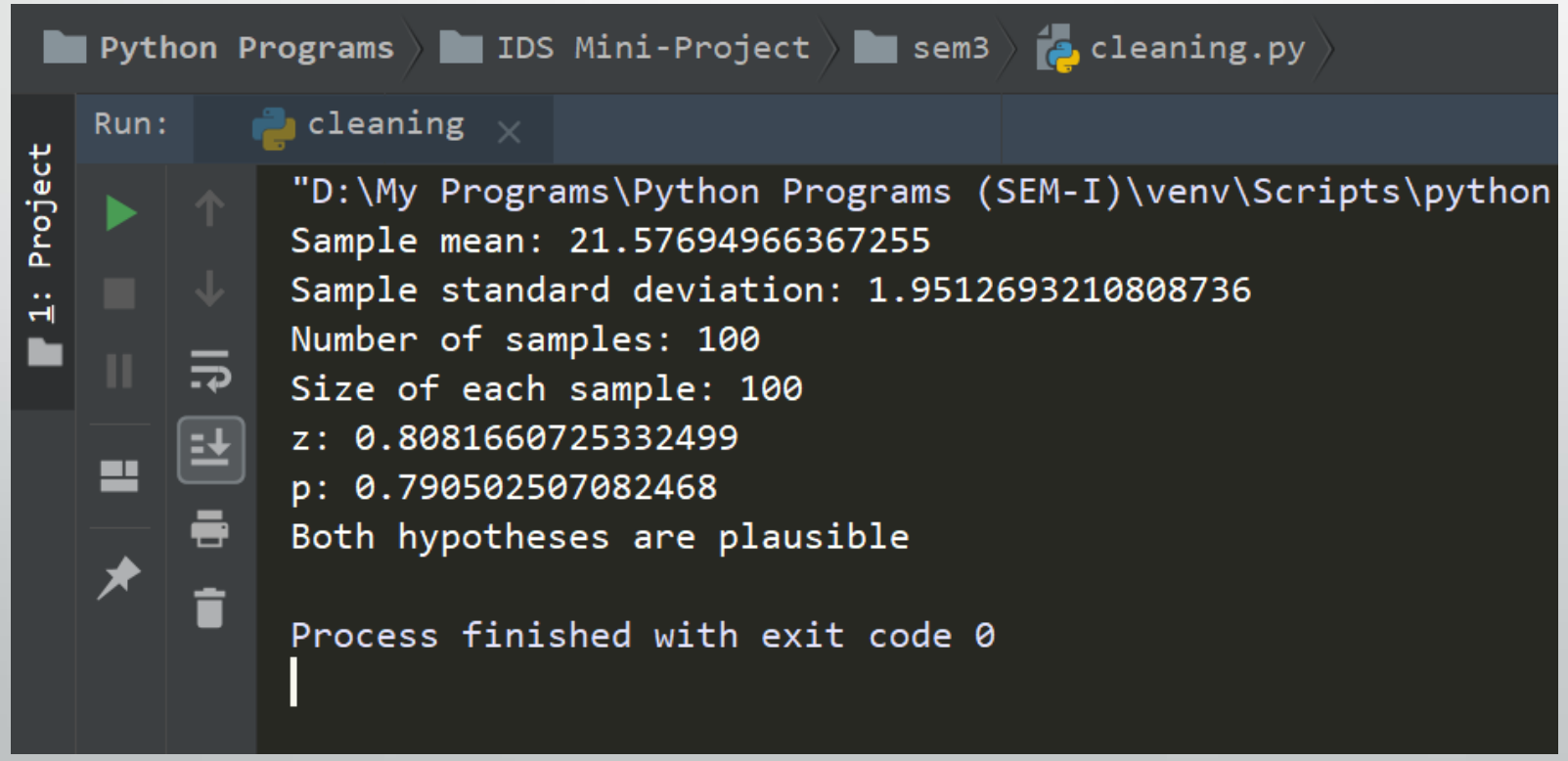
Hypothesis Testing

- Null Hypothesis H_0 : The mean size of an app is greater than 20Mb.

$$H_0 : \mu > 20$$

- Alternate Hypothesis: The mean size of an app is less than or equal to 20Mb.

$$H_a : \mu <= 20$$



Python Programs > IDS Mini-Project > sem3 > cleaning.py

data.py | cleaning.py

```python
def hypothesis_test(data, samples, null_mean, null_sign):
    sample = []
    while samples != 0:
        l = []
        for i in range(100):
            r = random.randint(0, len(data)-1)
            if r not in l:
                l.append(r)
        l = [data[i] for i in l]
        sample.append(np.mean(l))
        samples -= 1
    mean = np.mean(sample)
    std = np.std(sample)
    print("Sample mean:", mean)
    print("Sample standard deviation:", std)
    print("Number of samples:", len(sample))
    print("Size of each sample:", 100)

    if(null_sign == ">"):
        left_tail(mean, std, null_mean)
    elif(null_sign == "<"):
        right_tail(mean, std, null_mean)
    elif(null_sign == "="):
        two_sided(mean, std, null_mean)
```

Hypothesis Test Function