Подключение библиотек, загрузка данных

In [47]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import random
```

In [48]:
```
df_apple = pd.read_csv('AppleStore.csv', index_col='Unnamed: 0')
df_google = pd.read_csv('googleplaystore.csv')
```

Подготовка данных

In [49]:
```
df_apple.head()
```

Out[49]:

	id	track_name	size_bytes	currency	price	rating_count_tot	rating_count_ver	user_rating	u
1	281656475	PAC-MAN Premium	100788224	USD	3.99	21292	26	4.0	
2	281796108	Evernote - stay organized	158578688	USD	0.00	161065	26	4.0	
3	281940292	WeatherBug - Local Weather, Radar, Maps, Alerts	100524032	USD	0.00	188583	2822	3.5	
4	282614216	eBay: Best App to Buy, Sell, Save! Online Shop...	128512000	USD	0.00	262241	649	4.0	
5	282935706	Bible	92774400	USD	0.00	985920	5320	4.5	

In [50]:
```
df_google.head()
```

Out[50]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Ge
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & De
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Design;Pre
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & De
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & De
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Design;Crea

In [51]:
```
df_apple.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7197 entries, 1 to 11097
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   id                7197 non-null   int64  
 1   track_name        7197 non-null   object 
 2   size_bytes        7197 non-null   int64  
 3   currency          7197 non-null   object 
 4   price             7197 non-null   float64
 5   rating_count_tot  7197 non-null   int64  
 6   rating_count_ver  7197 non-null   int64  
 7   user_rating       7197 non-null   float64
 8   user_rating_ver   7197 non-null   float64
 9   ver               7197 non-null   object 
 10  cont_rating       7197 non-null   object 
 11  prime_genre       7197 non-null   object 
 12  sup_devices.num   7197 non-null   int64  
 13  ipadSc_urls.num   7197 non-null   int64  
 14  lang.num          7197 non-null   int64  
 15  vpp_lic           7197 non-null   int64  
dtypes: float64(3), int64(8), object(5)
memory usage: 815.3+ KB
```

In [52]:
```
df_google.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   App             10841 non-null  object 
 1   Category        10841 non-null  object 
 2   Rating          9367 non-null   float64
 3   Reviews         10841 non-null  object 
 4   Size            10841 non-null  object 
 5   Installs        10841 non-null  object 
 6   Type            10840 non-null  object 
 7   Price           10841 non-null  object 
 8   Content Rating  10840 non-null  object 
 9   Genres          10841 non-null  object 
 10  Last Updated    10841 non-null  object 
 11  Current Ver     10833 non-null  object 
 12  Android Ver     10838 non-null  object 
dtypes: float64(1), object(12)
memory usage: 592.9+ KB
```

Создаю столбец "Type" для df_apple, в котором будет хранится информация о типе приложения(платное или бесплатное), как в df_google:

In [53]:
```
bins = [-np.inf, 0, np.inf]
labels = ['Free','Paid']
df_apple['Type'] = pd.cut(df_apple['price'], bins=bins, labels=labels)
```

In [54]:
```
df_google.Type.fillna(value = 'Free', inplace = True)
```

In [55]:
```
df_google.loc[df_google['Type'] == '0','Type'] = 'Free'
```

Создаю новый столбец, в который заношу "новую" цену за приложение (типа float64, удобнее анализировать):

In [56]:
```
df_google['NewPrice'] = pd.to_numeric(df_google.Price.str.replace('$', ''), errors='coerce')
```

In [57]:
```
df_google.NewPrice.fillna(value = 0.0, inplace = True)
```

- Столбец для A.S., показывающий, является ли приложение игрой
- Аналогичный столбец для G.P.S.

In [58]:
```
df_apple['Games'] = df_apple.prime_genre.isin(['Games'])
df_google['Games'] = df_google.Category.isin(['GAME'])
```

Создаю столбец(float64) с кол-вом загрузок для G.P.S.

In [59]:
```
df_google['NewInstalls'] = df_google.Installs.str.replace(',', '')
df_google['NewInstalls'] = pd.to_numeric(df_google.NewInstalls.str.replace('+',''),errors='coerce')
```

Создаю столбец NewRating(float64) с рейтингом приложений и столбец NewReviews(float64) с количеством оценок в G.P.S.

In [60]:
```
df_google.Rating.fillna(value = '0',inplace = True)
df_google['NewRating'] = pd.to_numeric(df_google.Rating,errors='coerce')
df_google['NewReviews'] = pd.to_numeric(df_google.Reviews,errors='coerce')
```

Остальное:

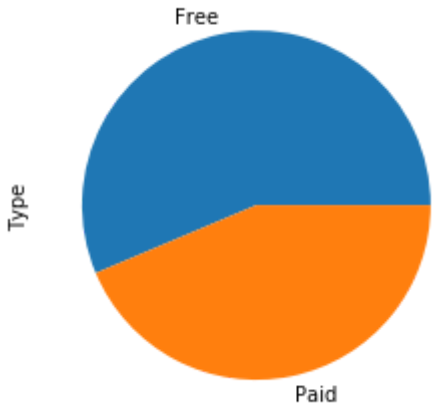
In [61]:
```
df_google.loc[[10472], ['Category']] = 'TOOLS'
```

Описательная статистика + гипотезы

Примечание: A.S. - Apple Store, G.P.S. - Google Play Store

In [62]:
```
df_apple.Type.value_counts().plot.pie()
```

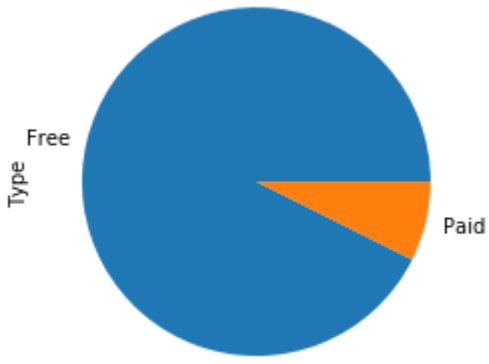
Out[62]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xba4dfa0>
```



- Доля платных приложений в A.S.

In [63]:
```
df_google.Type.value_counts().plot.pie()
```

Out[63]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xfcdd48>
```



- Доля платных приложений в G.P.S.

Средняя цена за приложение в A.S. :

In [64]:
```
np.average(df_apple[~df_apple.price.isin(['0'])].price)
```

Out[64]:
```
3.955297675899396
```

Средняя цена за приложение в G.P.S. :

In [65]:
```
np.average(df_google[~df_google.NewPrice.isin(['0'])].NewPrice)
```

Out[65]:
```
13.920837500000003
```

В Apple Store намного больше платных приложений (почти половина) чем в Google Play Store (примерно 10%). Однако в G.P.S. средняя цена за приложение почти в 4 раза выше. Следовательно можно сделать вывод: В A.S. хочешь, не хочешь все равно придется отдать копеечку, хоть и небольшую, за продукт, а в G.P.S. приложения в большинстве своем бесплатные, но за платный продукт придется раскошелится.

In [66]:
```
df_apple.Games.value_counts().plot.pie(labels=['Games','Other'], colors=['tomato','blue'])
```

Out[66]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xba6c640>
```

In [67]:
```
df_google.Games.value_counts().plot.pie(labels=['Other','Games'], colors=['blue','tomato'])
```

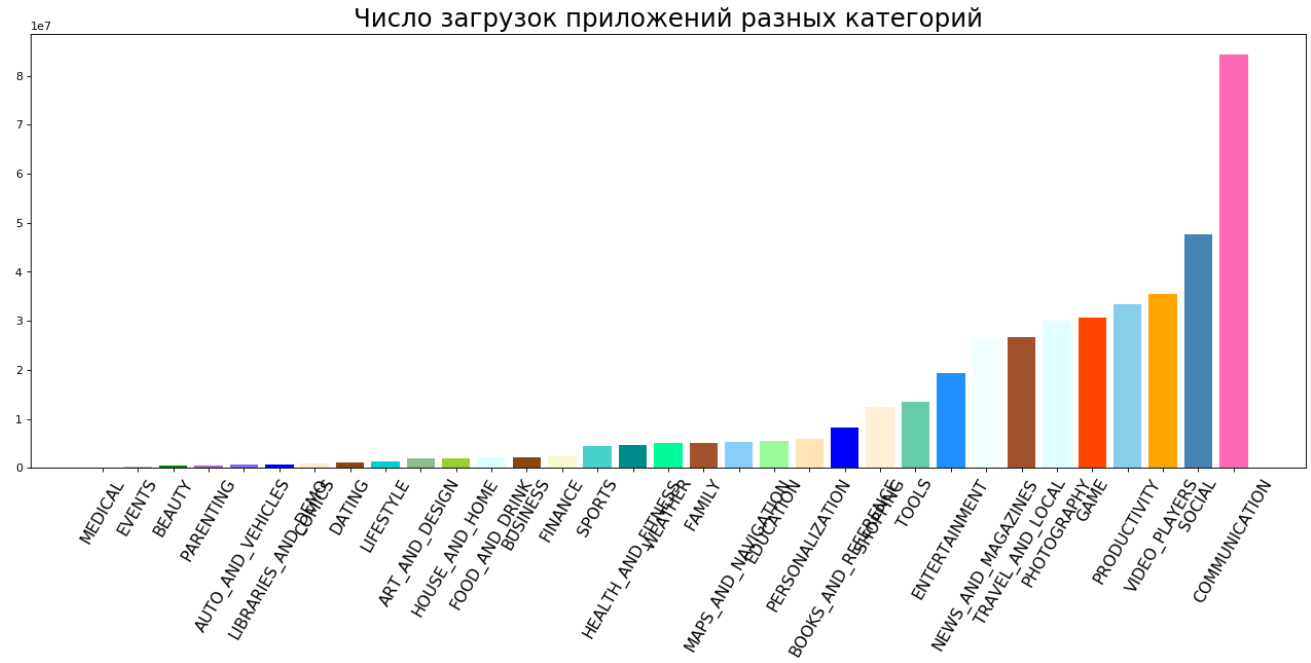
Out[67]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xb8f09d0>
```

Большая часть приложений в A.S. - игры. В то время, как в G.P.S. они составляют чуть больше 10% от общего количества

Как говорится, без спроса нет предложения. Владельцам iOS лишь бы в игрульки поиграть...

Рассмотрим G.P.S. глубже

In [68]:
```
# Prepare Data
df = df_google.groupby('Category', as_index=False).mean()
df.sort_values('NewInstalls', inplace=True)
n = df['Category'].unique().__len__()+1
all_colors = list(plt.cm.colors.cnames.keys())
random.seed(3)
c = random.choices(all_colors, k=n)
# Plot Bars
plt.figure(figsize=(20,7), dpi= 80)
plt.bar(df['Category'], df['NewInstalls'], color=c, width=.8)
plt.gca().set_xticklabels(df['Category'], rotation=60, horizontalalignment= 'center',
fontsize=14)
plt.title("Число загрузок приложений разных категорий",fontsize=22)
plt.show()
```

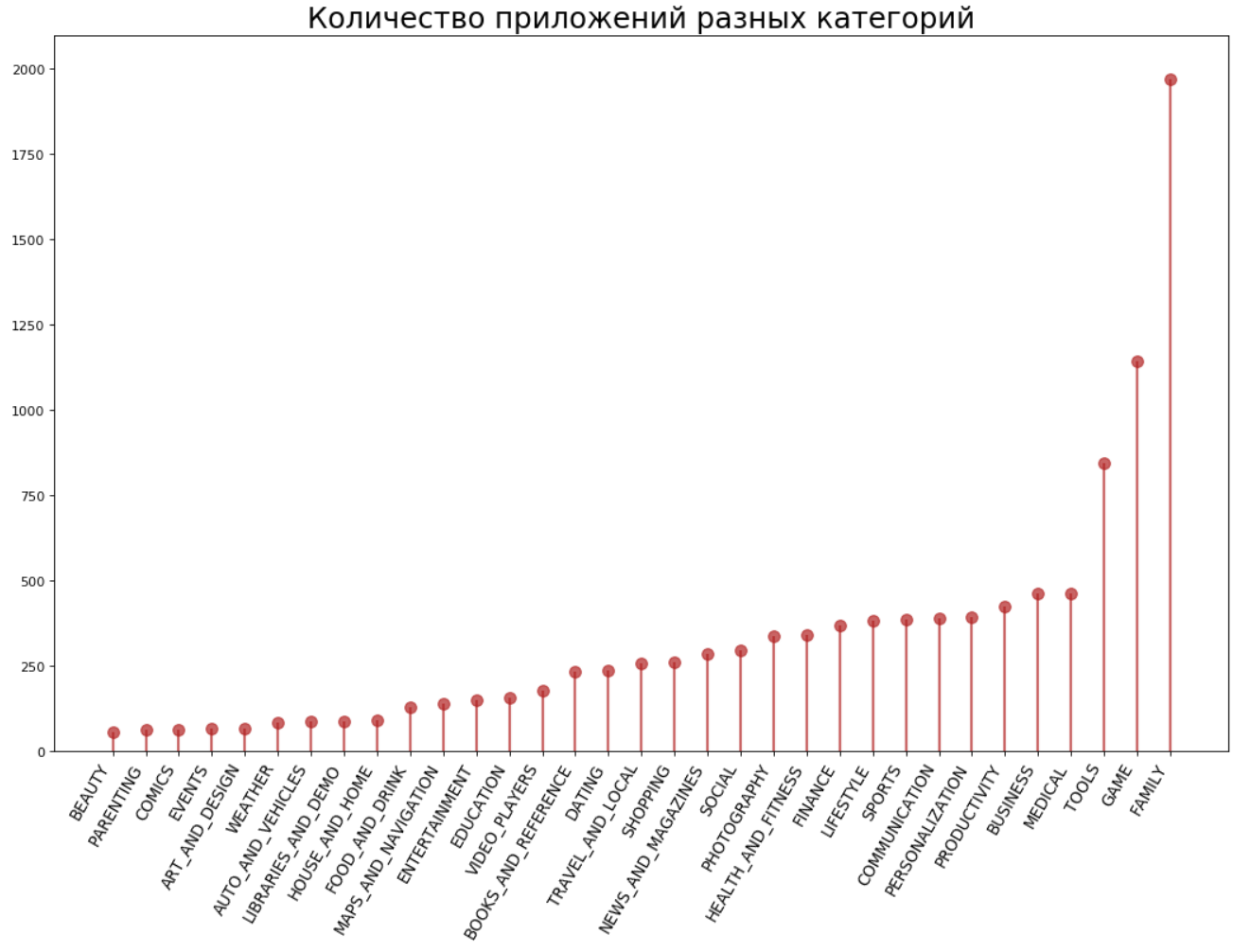


In [69]:
```
# Prepare Data
df = pd.DataFrame(df_google.Category.value_counts())
df.reset_index(inplace=True)
df.columns = ['Category','Value']
df.sort_values('Value', inplace=True)
df.reset_index(inplace=True)

# Draw plot
fig, ax = plt.subplots(figsize=(16,10), dpi= 80)
ax.vlines(x=df.index, ymin=0, ymax=df.Value, color='firebrick', alpha=0.7, linewidth=
2)
ax.scatter(x=df.index, y=df.Value, s=75, color='firebrick', alpha=0.7)

# Title, Label, Ticks and Ylim
ax.set_title('Количество приложений разных категорий', fontdict={'size':22})
ax.set_xticks(df.index)
ax.set_xticklabels(df.Category.str.upper(), rotation=60, fontdict={'horizontalalignme
nt': 'right', 'size':12})
ax.set_ylim(0, 2100)

plt.show()
```



Как мы видим, приложения категории "Communication" самые популярные, ведь имеют наибольшее количество загрузок с большим отрывом от других категорий, хотя по кол-ву приложений данной категории они находятся лишь на 8 месте. Скорее всего это связано с популярностью именно социальных сетей, которые, очевидно, входят в эту категорию.

Обратная ситуция с категорий "Family". Несмотря на то, что приложений данной категории очень много, они не могут похвастаться особой популярностью среди пользователей.

Рассмотрим рейтинги приложений и количество оценок на площадках

Нашел средний рейтинг приложений в G.P.S.

```
In [70]: np.average(df_google[~df_google.NewRating.isin(['0'])].NewRating)
```

```
Out[70]: 4.193338315362443
```

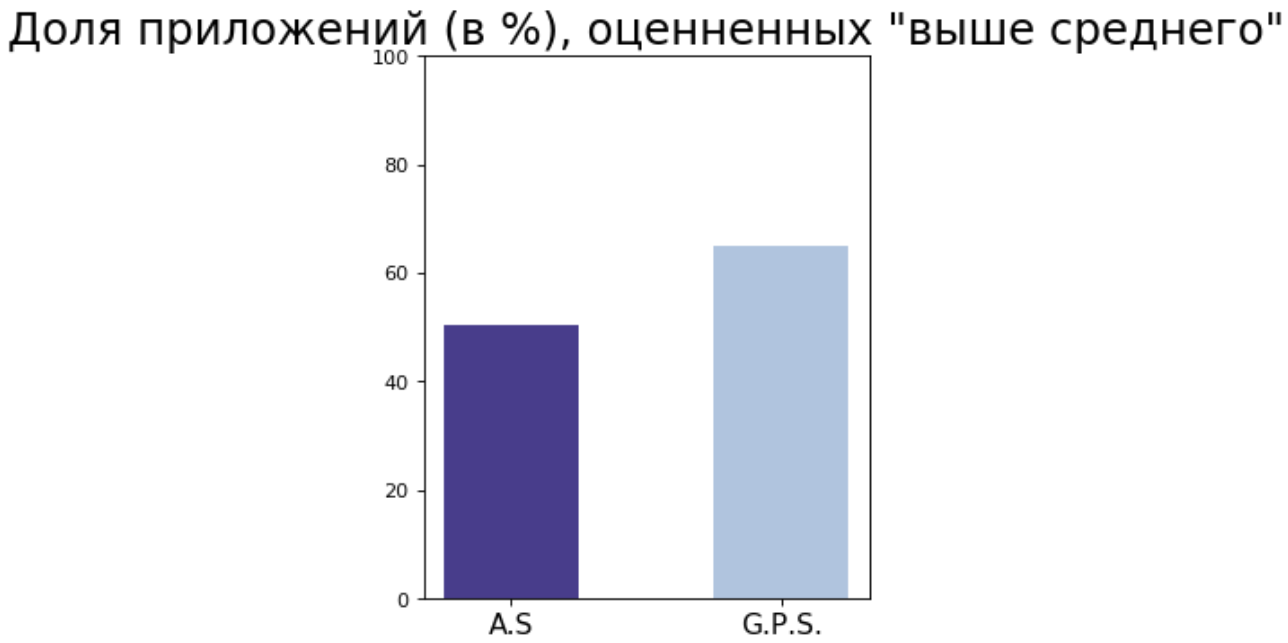
Будем считать рейтинг < 4.2 - "ниже среднего", а >= 4.2 - "выше среднего"

Аналогично для A.S.

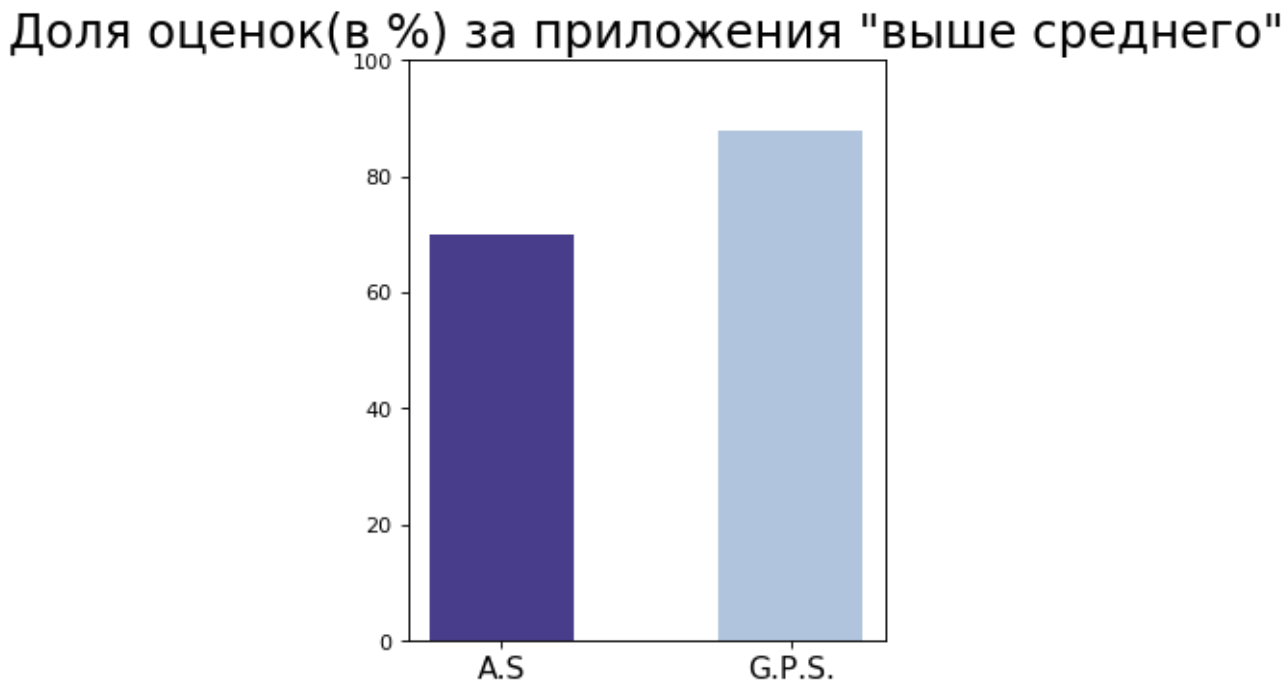
```
In [71]: np.average(df_apple[~df_apple.user_rating.isin(['0'])].user_rating)
```

```
Out[71]: 4.049696873005743
```

In [72]:
```
# Prepare Data
x = len(df_google[df_google.NewRating > 4.1]) / len(df_google[df_google.NewRating > 0
]) * 100
y = len(df_apple[df_apple.user_rating > 4.0]) / len(df_apple[df_apple.user_rating > 0
]) * 100
#----------
n = 2
all_colors = list(plt.cm.colors.cnames.keys())
random.seed(3)
c = random.choices(all_colors, k=n)
# Plot Bars
plt.figure(figsize=(4,5), dpi= 80)
plt.bar(['A.S.','G.P.S.'], [y,x], color=c, width=.5)
plt.ylim(0, 100)
# Decoration
plt.gca().set_xticklabels(['A.S','G.P.S.'], horizontalalignment= 'center', fontsize=1
4)
plt.title('Доля приложений (в %), оцененных "выше среднего"', fontsize=22)
plt.show()
```



In [73]:
```
# Prepare Data
x = (df_google[df_google.NewRating > 4.1].NewReviews.sum() / df_google[df_google.NewR
ating > 0].NewReviews.sum()) * 100
y = (df_apple[df_apple.user_rating > 4.0].rating_count_tot.sum() / df_apple[df_apple.
user_rating > 0].rating_count_tot.sum()) * 100
#------------------
n = 2
all_colors = list(plt.cm.colors.cnames.keys())
random.seed(3)
c = random.choices(all_colors, k=n)
# Plot Bars
plt.figure(figsize=(4,5), dpi= 80)
plt.bar(['A.S.','G.P.S.'], [y,x], color=c, width=.5)
plt.ylim(0, 100)
# Decoration
plt.gca().set_xticklabels(['A.S','G.P.S.'], horizontalalignment= 'center', fontsize=1
4)
plt.title('Доля оценок(в %) за приложения "выше среднего"', fontsize=22)
plt.show()
```



Из полученного можно сделать вывод, что люди, в основном, положительно оценивают приложения в этих магазинах, еще несмотря на то, что средние оценки приложений 4.1 и 4.0 (т.е. оценка 3.9 считалась плохой). Также люди чаще комментируют хорошие приложения, т.е. приложения с оценкой "выше среднего".