Московский государственный технический университет им. Н.Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Автоматизированные системы обработки информации и управления»

“Методы машинного обучения”

Отчет по лабораторной работе №1

“Разведочный анализ данных. Исследование и визуализация данных”

Выполнил:
Буклин С.В.
Группа ИУ5-21м

Москва 2018

1. Задание

- Выбрать набор данных (датасет).
- Создать ноутбук, который содержит следующие разделы:
  - Текстовое описание выбранного Вами набора данных.
  - Основные характеристики датасета.
  - Визуальное исследование датасета.
  - Информация о корреляции признаков.
- Сформировать отчет и разместить его в своем репозитории на github.

2. Текст программы и экранные формы

Описание набора данных

1) Текстовое описание набора данных

В качестве набора данных будет использоваться набор данных по мобильных приложениям в AppleStore - https://www.kaggle.com/ramamet4/app-store-apple-data-set-10k-apps

Эта задача является очень актуальной для создания "умных зданий", которые выполняют все требования по кондиционированию воздуха, температурным условиям, но при этом экономят электроэнергию в том случае, если людей в помещении нет.

Датасет состоит из 2 файлов:

1. AppleStore.csv - обучающая выборка
2. appleStore_description.csv

Импорт библиотек

2) Основные характеристики датасета

Импорт библиотек

```
[1]  !pip install -U seaborn
```

```
Requirement already up-to-date: seaborn in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied, skipping upgrade: numpy>=1.9.3 in /usr/local/lib/pyt
Requirement already satisfied, skipping upgrade: pandas>=0.15.2 in /usr/local/lib/p
Requirement already satisfied, skipping upgrade: matplotlib>=1.4.3 in /usr/local/li
Requirement already satisfied, skipping upgrade: scipy>=0.14.0 in /usr/local/lib/py
Requirement already satisfied, skipping upgrade: pytz>=2011k in /usr/local/lib/pyth
Requirement already satisfied, skipping upgrade: python-dateutil>=2 in /usr/local/l
Requirement already satisfied, skipping upgrade: cycler>=0.10 in /usr/local/lib/pyt
Requirement already satisfied, skipping upgrade: kiwisolver>=1.0.1 in /usr/local/li
Requirement already satisfied, skipping upgrade: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>
Requirement already satisfied, skipping upgrade: six>=1.5 in /usr/local/lib/python3
Requirement already satisfied, skipping upgrade: setuptools in /usr/local/lib/pytho
```

```
[2]  import numpy as np
     import pandas as pd
     import seaborn as sns
     import matplotlib.pyplot as plt
     %matplotlib inline
     sns.set(style="ticks")
```

Загрузка данных

Загрузка данных

```
[3] from google.colab import files
    uploaded = files.upload()
```

Выбрать файлы AppleStore.csv

- **AppleStore.csv**(application/vnd.ms-excel) - 837657 bytes, last modified: 10.06.2018 - 100% done

```
Saving AppleStore.csv to AppleStore (2).csv
```

```
[4] import io
    data = pd.read_csv(io.BytesIO(uploaded['AppleStore.csv']))
```

```
[5] # Первые 5 строк датасета
    data.head()
```

	Unnamed: 0	id	track_name	size_bytes	currency	price	rating_count_
0	1	281656475	PAC-MAN Premium	100788224	USD	3.99	21
1	2	281796108	Evernote - stay organized	158578688	USD	0.00	161
2	3	281940292	WeatherBug - Local Weather, Radar, Maps, Alerts	100524032	USD	0.00	188
3	4	282614216	eBay: Best App to Buy, Sell, Save! Online Shop...	128512000	USD	0.00	262
4	5	282935706	Bible	92774400	USD	0.00	985

```
[6] # Размер датасета
    data.shape
```

```
(7197, 17)
```

```
[7] total_count = data.shape[0]
    print('Всего строк: {}'.format(total_count))
```

```
Всего строк: 7197
```


```
[7] total_count = data.shape[0]
    print('Всего строк: {}'.format(total_count))
```

```
Всего строк: 7197
```


```
[8] # Список колонок
    data.columns
```

```
Index(['Unnamed: 0', 'id', 'track_name', 'size_bytes', 'currency', 'price',
       'rating_count_tot', 'rating_count_ver', 'user_rating',
       'user_rating_ver', 'ver', 'cont_rating', 'prime_genre',
       'sup_devices.num', 'ipadSc_urls.num', 'lang.num', 'vpp_lic'],
      dtype='object')
```

```
[9] # Список колонок с типами данных
    data.dtypes
```

```
Unnamed: 0            int64
id                    int64
track_name           object
size_bytes            int64
currency             object
price               float64
rating_count_tot      int64
rating_count_ver      int64
user_rating         float64
user_rating_ver     float64
ver                  object
cont_rating          object
prime_genre          object
sup_devices.num       int64
ipadSc_urls.num       int64
lang.num              int64
vpp_lic               int64
dtype: object
```

Основные статистические характеристики датасета

```
[11] # Основные статистические характеристки набора данных
     data.describe()
```

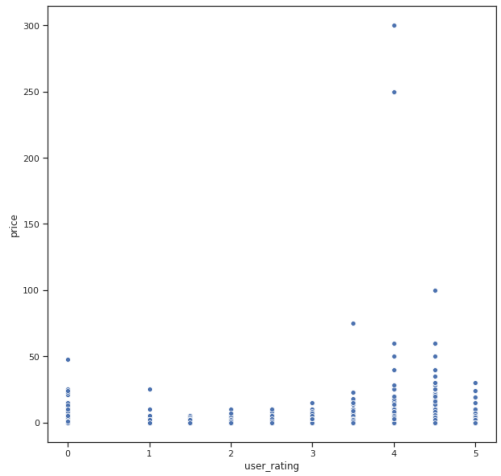
	Unnamed: 0	id	size_bytes	price	rating_count_tot	rating_count_ver	user_rating	user_rating_ver	sup_devices.num	ipadSc_urls.num	lang.num	vpp_lic
count	7197.000000	7.197000e+03	7.197000e+03	7197.000000	7.197000e+03	7197.000000	7197.000000	7197.000000	7197.000000	7197.000000	7197.000000	7197.000000
mean	4759.069612	8.631310e+08	1.991345e+08	1.726218	1.289291e+04	460.373906	3.526956	3.253578	37.361817	3.707100	5.434903	0.993053
std	3093.625213	2.712368e+08	3.592069e+08	5.833006	7.573941e+04	3920.455183	1.517948	1.809363	3.737715	1.986005	7.919593	0.083066
min	1.000000	2.816565e+08	5.898240e+05	0.000000	0.000000e+00	0.000000	0.000000	0.000000	9.000000	0.000000	0.000000	0.000000
25%	2090.000000	6.000937e+08	4.692275e+07	0.000000	2.800000e+01	1.000000	3.500000	2.500000	37.000000	3.000000	1.000000	1.000000
50%	4380.000000	9.781482e+08	9.715302e+07	0.000000	3.000000e+02	23.000000	4.000000	4.000000	37.000000	5.000000	1.000000	1.000000
75%	7223.000000	1.082310e+09	1.819249e+08	1.990000	2.793000e+03	140.000000	4.500000	4.500000	38.000000	5.000000	8.000000	1.000000
max	11097.000000	1.188376e+09	4.025970e+09	299.990000	2.974676e+06	177050.000000	5.000000	5.000000	47.000000	5.000000	75.000000	1.000000

Визуальное исследование датасета

Для визуального исследования могут быть использованы различные виды диаграмм, мы построим только некоторые варианты диаграмм, которые используются достаточно часто.

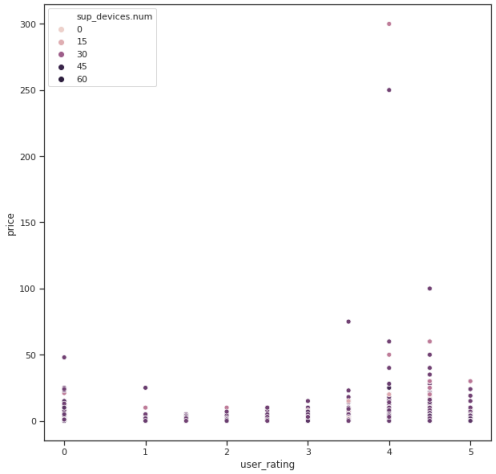
```
[16] fig, ax = plt.subplots(figsize=(10,10))
     sns.scatterplot(ax=ax, x='user_rating', y='price', data=data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f37fefadfd0>
```




```
[18] fig, ax = plt.subplots(figsize=(10,10))
     sns.scatterplot(ax=ax, x='user_rating', y='price', data=data, hue='sup_devices.num')
```

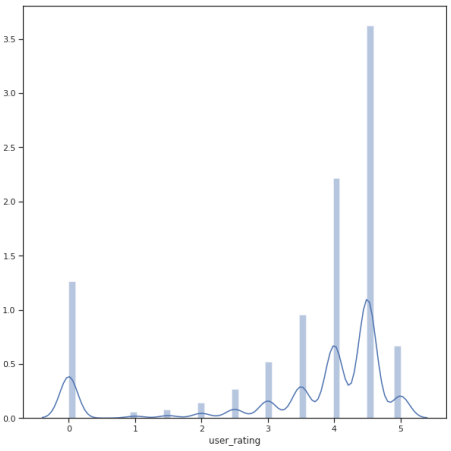
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f37feed1da0>
```



Гистограммы

```
[19] fig, ax = plt.subplots(figsize=(10,10))
     sns.distplot(data['user_rating'])
```

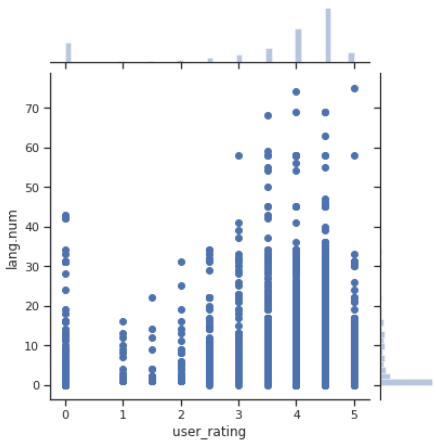
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f37feedcac8>
```



Joinplot

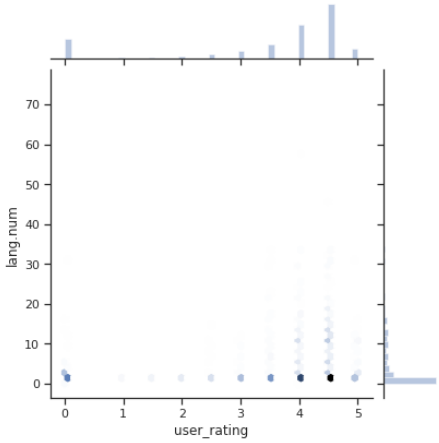
```
[29] sns.jointplot(x='user_rating', y='lang.num', data=data)
```

```
<seaborn.axisgrid.JointGrid at 0x7f37fe2149e8>
```



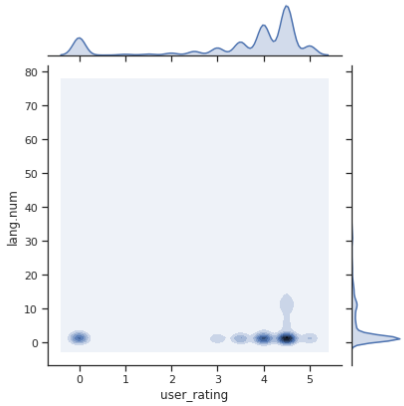

```
[28] sns.jointplot(x='user_rating', y='lang.num', data=data, kind='hex')
```

```
<seaborn.axisgrid.JointGrid at 0x7f37fe82a240>
```




```
[30] sns.jointplot(x='user_rating', y='lang.num', data=data, kind='kde')
```

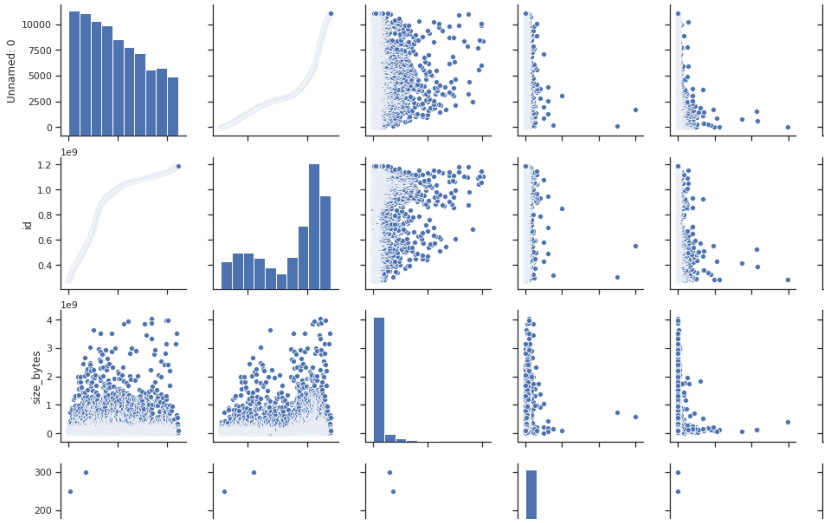
```
<seaborn.axisgrid.JointGrid at 0x7f37fe06e128>
```



Парные диаграммы

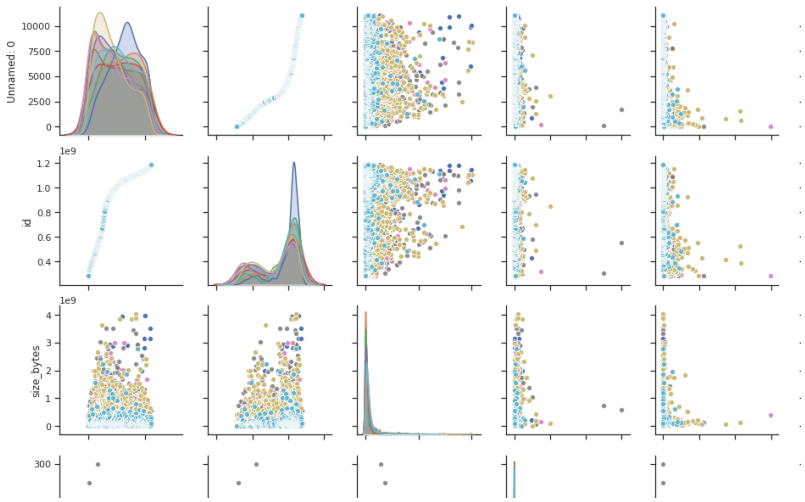
```
[31] sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x7f37fe280710>
```



```
[32] sns.pairplot(data, hue="user_rating")
```

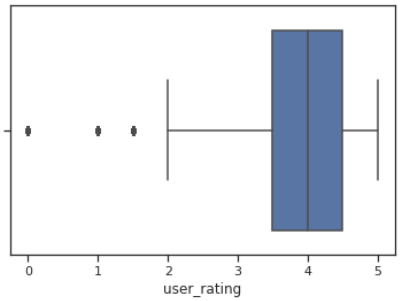
```
/usr/local/lib/python3.6/dist-packages/statsmodels/nonparametric/kde.py:494: RuntimeWarning: invalid value encountered in tru
  binned = fast_linbin(X,a,b,gridsize)/(delta*nobs)
/usr/local/lib/python3.6/dist-packages/statsmodels/nonparametric/kdetools.py:34: RuntimeWarning: invalid value encountered in
  FAC1 = 2*(np.pi*bw/RANGE)**2
<seaborn.axisgrid.PairGrid at 0x7f37fb29c1d0>
```



Ящик с усами

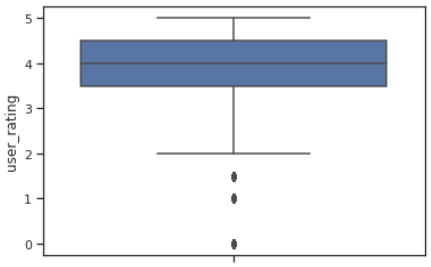
```
[33] sns.boxplot(x=data['user_rating'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f37f6b14160>
```



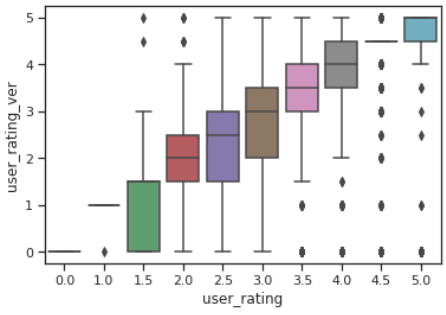

```
[34] # По вертикали
     sns.boxplot(y=data['user_rating'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f37f69e7e10>
```




```
[36] # Распределение параметра user_rating_ver сгруппированные по user_rating.
     sns.boxplot(x='user_rating', y='user_rating_ver', data=data)
```

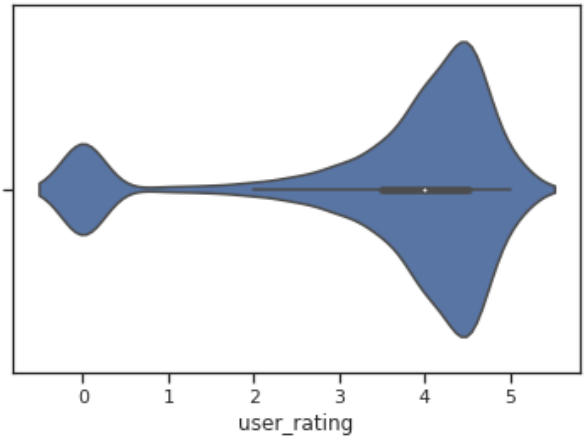
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f37f6f59be0>
```



Violin plot

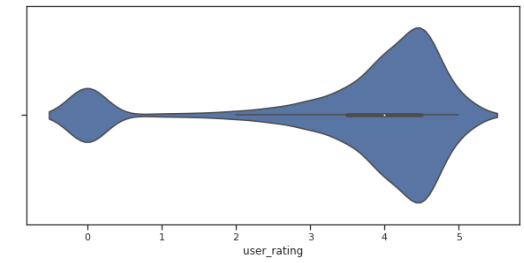
```
[38]
sns.violinplot(x=data['user_rating'])
```

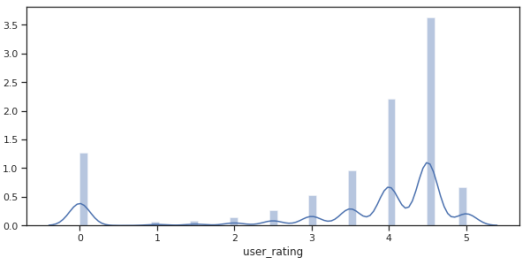
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f37f75b6128>
```




```
[40] fig, ax = plt.subplots(2, 1, figsize=(10,10))
     sns.violinplot(ax=ax[0], x=data['user_rating'])
     sns.distplot(data['user_rating'], ax=ax[1])
```

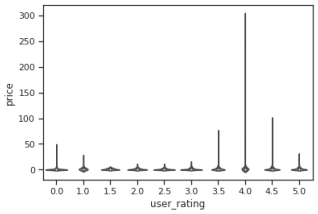
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f37f5884208>
```





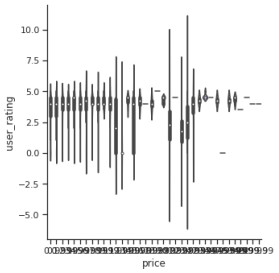

```
[41] # Распределение параметра price сгруппированные по user_rating.
     sns.violinplot(x='user_rating', y='price', data=data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f37f57c60f0>
```




```
sns.catplot(y='user_rating', x='price', data=data, kind="violin", split=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f37f2131b70>
```



Корреляционные матрицы

```
[45] data.corr(method='pearson')
```

	Unnamed: 0	id	size_bytes	price	rating_count_tot	rating_count_ver	user_rating	user_rating_ver	sup_devices.num	ipadSc_urls.num	lang.num	vpp_lic
Unnamed: 0	1.000000	0.910714	0.041277	-0.074326	-0.168640	-0.064717	-0.204867	-0.127580	0.024648	0.014589	-0.148487	0.000501
id	0.910714	1.000000	0.077486	-0.075247	-0.201976	-0.072784	-0.185178	-0.109849	0.033605	0.052082	-0.128932	0.017743
size_bytes	0.041277	0.077486	1.000000	0.182392	0.004486	0.006337	0.066256	0.086075	-0.118347	0.152697	0.004614	-0.150418
price	-0.074326	-0.075247	0.182392	1.000000	-0.039044	-0.018012	0.046601	0.025173	-0.115361	0.066100	-0.006713	-0.029942
rating_count_tot	-0.168640	-0.201976	0.004486	-0.039044	1.000000	0.163645	0.083310	0.088744	0.008832	0.015734	0.137675	-0.000982
rating_count_ver	-0.064717	-0.072784	0.006337	-0.018012	0.163645	1.000000	0.068754	0.077840	0.037951	0.024333	0.013287	0.006460
user_rating	-0.204867	-0.185178	0.066256	0.046601	0.083310	0.068754	1.000000	0.774140	-0.042451	0.265671	0.170976	0.069816
user_rating_ver	-0.127580	-0.109849	0.086075	0.025173	0.088744	0.077840	0.774140	1.000000	-0.018901	0.275737	0.175580	0.050094
sup_devices.num	0.024648	0.033605	-0.118347	-0.115361	0.008832	0.037951	-0.042451	-0.018901	1.000000	-0.037728	-0.041681	-0.037109
ipadSc_urls.num	0.014589	0.052082	0.152697	0.066100	0.015734	0.024333	0.265671	0.275737	-0.037728	1.000000	0.088378	0.071901
lang.num	-0.148487	-0.128932	0.004614	-0.006713	0.137675	0.013287	0.170976	0.175580	-0.041681	0.088378	1.000000	0.032477
vpp_lic	0.000501	0.017743	-0.150418	-0.029942	-0.000982	0.006460	0.069816	0.050094	-0.037109	0.071901	0.032477	1.000000

```
[47] data.corr(method='kendall')
```

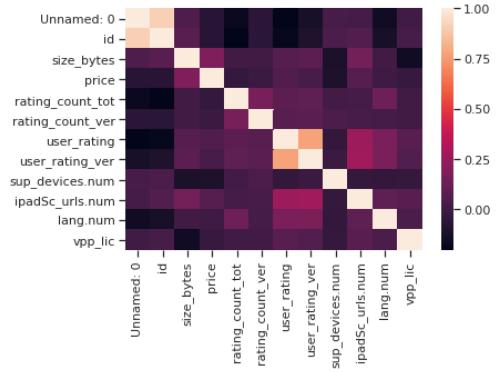
	Unnamed: 0	id	size_bytes	price	rating_count_tot	rating_count_ver	user_rating	user_rating_ver	sup_devices.num	ipadSc_urls.num	lang.num	vpp_lic
Unnamed: 0	1.000000	1.000000	0.046355	-0.111122	-0.325483	-0.140087	-0.089297	-0.076096	0.034404	0.011097	-0.127586	0.005013
id	1.000000	1.000000	0.046355	-0.111122	-0.325483	-0.140087	-0.089297	-0.076096	0.034404	0.011097	-0.127586	0.005013
size_bytes	0.046355	0.046355	1.000000	0.047233	0.107414	0.110773	0.116266	0.105794	0.015110	0.279588	0.065207	-0.024916
price	-0.111122	-0.111122	0.047233	1.000000	-0.062836	0.027679	0.068234	0.078965	-0.005798	0.102258	0.008775	-0.031300
rating_count_tot	-0.325483	-0.325483	0.107414	-0.062836	1.000000	0.586412	0.397702	0.367031	-0.061228	0.147709	0.185850	0.023638
rating_count_ver	-0.140087	-0.140087	0.110773	0.027679	0.586412	1.000000	0.385948	0.444056	0.060876	0.175720	0.112699	0.021002
user_rating	-0.089297	-0.089297	0.116266	0.068234	0.397702	0.385948	1.000000	0.640035	0.014741	0.197511	0.138074	0.055051
user_rating_ver	-0.076096	-0.076096	0.105794	0.078965	0.367031	0.444056	0.640035	1.000000	-0.005660	0.198487	0.129746	0.037570
sup_devices.num	0.034404	0.034404	0.015110	-0.005798	-0.061228	0.060876	0.014741	-0.005660	1.000000	0.052610	-0.093839	-0.037934
ipadSc_urls.num	0.011097	0.011097	0.279588	0.102258	0.147709	0.175720	0.197511	0.198487	0.052610	1.000000	0.090673	0.061790
lang.num	-0.127586	-0.127586	0.065207	0.008775	0.185850	0.112699	0.138074	0.129746	-0.093839	0.090673	1.000000	0.039328
vpp_lic	0.005013	0.005013	-0.024916	-0.031300	0.023638	0.021002	0.055051	0.037570	-0.037934	0.061790	0.039328	1.000000

```
[48] data.corr(method='spearman')
```

	Unnamed: 0	id	size_bytes	price	rating_count_tot	rating_count_ver	user_rating	user_rating_ver	sup_devices.num	ipadSc_urls.num	lang.num	vpp_lic
Unnamed: 0	1.000000	1.000000	0.069115	-0.149284	-0.472812	-0.205097	-0.126441	-0.105574	0.048324	0.014683	-0.173242	0.006139
id	1.000000	1.000000	0.069115	-0.149284	-0.472812	-0.205097	-0.126441	-0.105574	0.048324	0.014683	-0.173242	0.006139
size_bytes	0.069115	0.069115	1.000000	0.067273	0.161708	0.163161	0.161290	0.149332	0.018922	0.362570	0.090474	-0.030514
price	-0.149284	-0.149284	0.067273	1.000000	-0.087833	0.033926	0.082622	0.096503	-0.008989	0.118101	0.011450	-0.034180
rating_count_tot	-0.472812	-0.472812	0.161708	-0.087833	1.000000	0.762076	0.507648	0.486164	-0.081478	0.189967	0.250726	0.028722
rating_count_ver	-0.205097	-0.205097	0.163161	0.033926	0.762076	1.000000	0.491654	0.569864	0.080762	0.223652	0.149575	0.025208
user_rating	-0.126441	-0.126441	0.161290	0.082622	0.507648	0.491654	1.000000	0.711775	0.018424	0.232260	0.174390	0.061501
user_rating_ver	-0.105574	-0.105574	0.149332	0.096503	0.486164	0.569864	0.711775	1.000000	-0.006035	0.235104	0.165961	0.042367
sup_devices.num	0.048324	0.048324	0.018922	-0.008989	-0.081478	0.080762	0.018424	-0.006035	1.000000	0.060939	-0.113684	-0.041181
ipadSc_urls.num	0.014683	0.014683	0.362570	0.118101	0.189967	0.223652	0.232260	0.235104	0.060939	1.000000	0.108038	0.065479
lang.num	-0.173242	-0.173242	0.090474	0.011450	0.250726	0.149575	0.174390	0.165961	-0.113684	0.108038	1.000000	0.043870
vpp_lic	0.006139	0.006139	-0.030514	-0.034180	0.028722	0.025208	0.061501	0.042367	-0.041181	0.065479	0.043870	1.000000

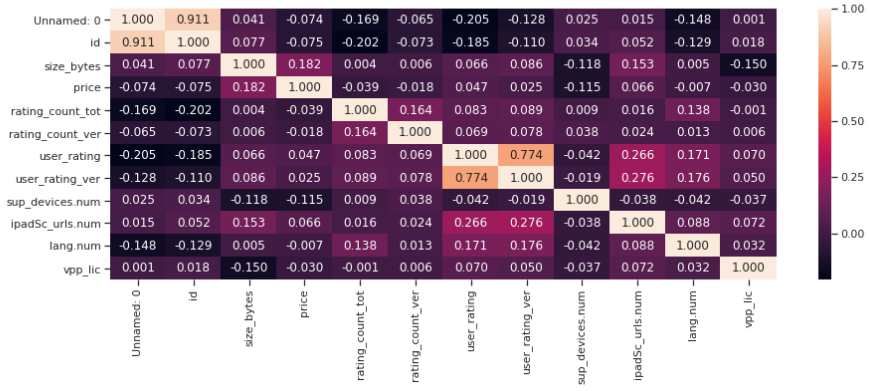
```
[49] sns.heatmap(data.corr())
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f37f5371e10>
```



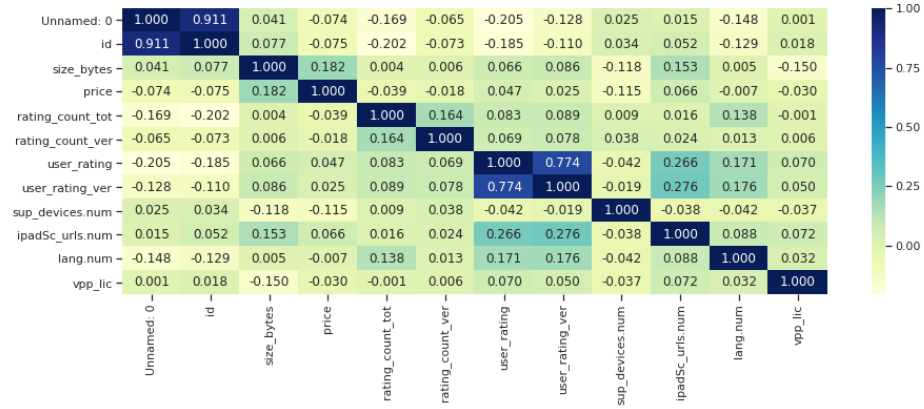

```
[58] # Вывод значений в ячейках
     fig, ax = plt.subplots(figsize=(15,5))
     sns.heatmap(data.corr(), annot=True, fmt='.3f')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f37f2ec70b8>
```



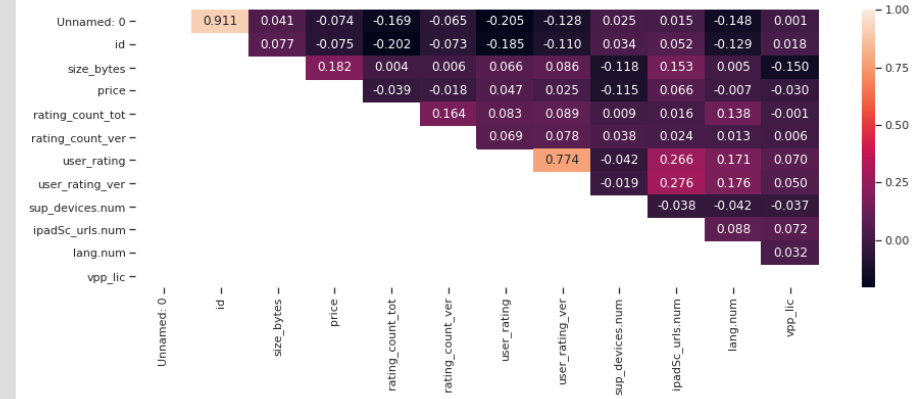
[59]
```
# Изменение цветовой гаммы
fig, ax = plt.subplots(figsize=(15,5))
sns.heatmap(data.corr(), cmap='YlGnBu', annot=True, fmt='.3f')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f37f2caebe0>
```



[60]
```
# Треугольный вариант матрицы
fig, ax = plt.subplots(figsize=(15,5))
mask = np.zeros_like(data.corr(), dtype=np.bool)
# чтобы оставить нижнюю часть матрицы
# mask[np.triu_indices_from(mask)] = True
# чтобы оставить верхнюю часть матрицы
mask[np.tril_indices_from(mask)] = True
sns.heatmap(data.corr(), mask=mask, annot=True, fmt='.3f')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f37f2b18fd0>
```



[55]
```
fig, ax = plt.subplots(1, 3, sharex='col', sharey='row', figsize=(35,5))
sns.heatmap(data.corr(method='pearson'), ax=ax[0], annot=True, fmt='.2f')
sns.heatmap(data.corr(method='kendall'), ax=ax[1], annot=True, fmt='.2f')
sns.heatmap(data.corr(method='spearman'), ax=ax[2], annot=True, fmt='.2f')
fig.suptitle('Корреляционные матрицы, построенные различными методами')
ax[0].title.set_text('Pearson')
ax[1].title.set_text('Kendall')
ax[2].title.set_text('Spearman')
```

Корреляционные матрицы, построенные различными методами