Лабораторная работа 2

Выполнил: Кузьмин Роман, ИУ5-25М

Датасет: Steam Store Data (https://www.kaggle.com/datasets/amanbarthwal/steam-store-data?select=steam-games.csv)

In [10]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
import scipy.stats as stats

import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
data = pd.read_csv('steam-games.csv', sep=",")
```

In [3]:
```
data.isnull().sum()
```

Out[3]:
```
app_id                    0
title                     0
release_date             57
genres                   87
categories               45
developer               190
publisher               211
original_price        37638
discount_percentage   37638
discounted_price        240
dlc_available             0
age_rating                0
content_descriptor    40122
about_description       138
win_support               0
mac_support               0
linux_support             0
awards                    0
overall_review         2477
overall_review_%       2477
overall_review_count   2477
recent_review         36994
recent_review_%       36994
recent_review_count   36994
dtype: int64
```

In [4]:
```
data.shape
```

Out[4]:
```
(42497, 24)
```

In [5]:
```
data.head()
```

Out[5]:

	app_id	title	release_date	genres	categories	developer	publisher
0	730	Counter-Strike 2	21 Aug, 2012	Action, Free to Play	Cross-Platform Multiplayer, Steam Trading Card...	Valve	Valve
1	570	Dota 2	9 Jul, 2013	Action, Strategy, Free to Play	Steam Trading Cards, Steam Workshop, SteamVR C...	Valve	Valve
2	2215430	Ghost of Tsushima DIRECTOR'S CUT	16 May, 2024	Action, Adventure	Single-player, Online Co-op, Steam Achievement...	Sucker Punch Productions	PlayStation PC LLC
3	1245620	ELDEN RING	24 Feb, 2022	Action, RPG	Single-player, Online PvP, Online Co-op, Steam...	FromSoftware Inc.	FromSoftware Inc.
4	1085660	Destiny 2	1 Oct, 2019	Action, Adventure, Free to Play	Single-player, Online PvP, Online Co-op, Steam...	Bungie	Bungie

5 rows × 24 columns

Пропуски в данных в столбцах с небольшим количеством пропусков можно обработать удалением - это единичные значения (по сравнению с размером датасета).

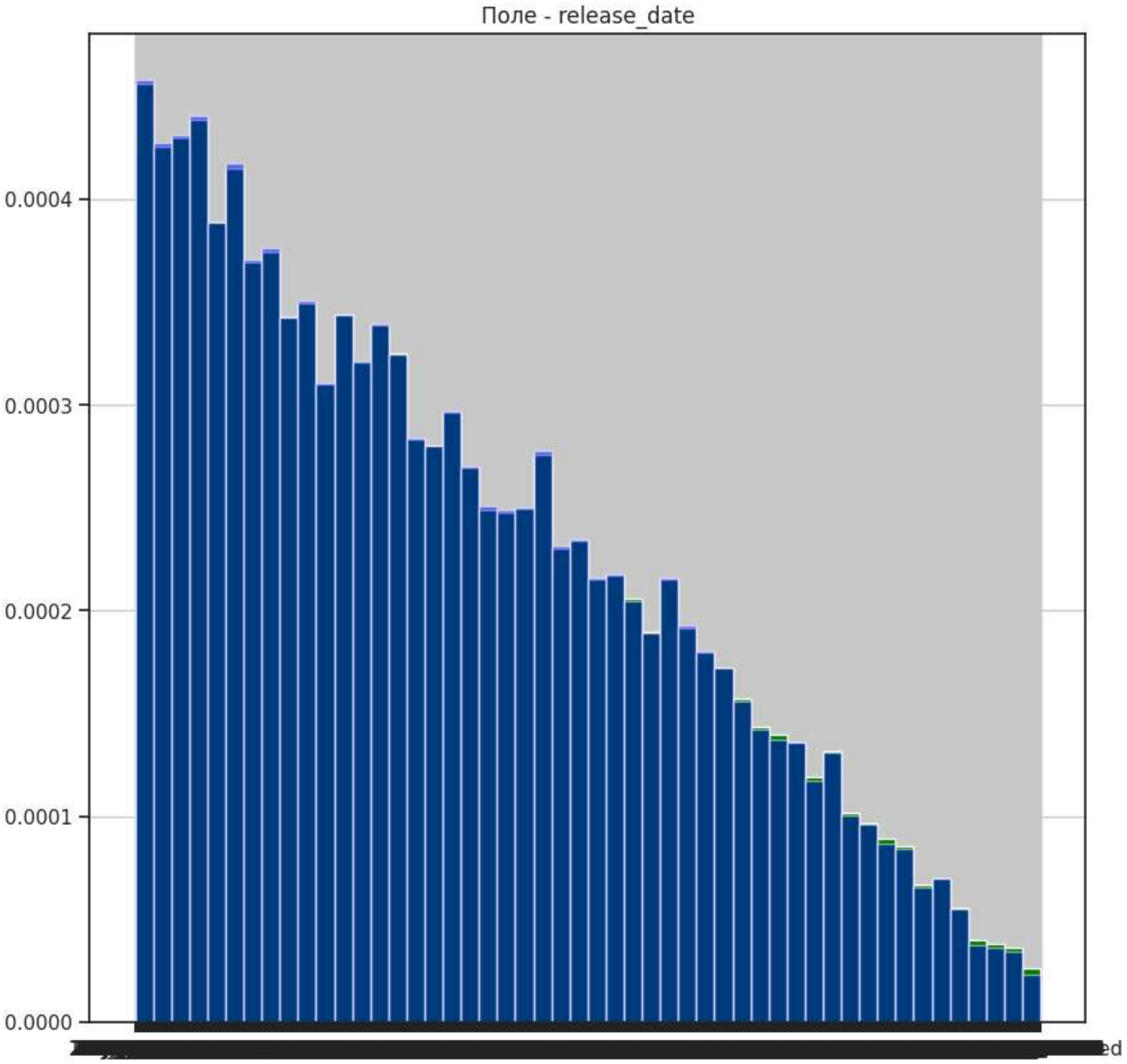
In [6]:
```
colsForDel = ['release_date', 'genres', 'categories', 'developer',
              'publisher', 'discounted_price', 'about_description']
data_drop_na = data[colsForDel].dropna()
data_drop_na.shape
```

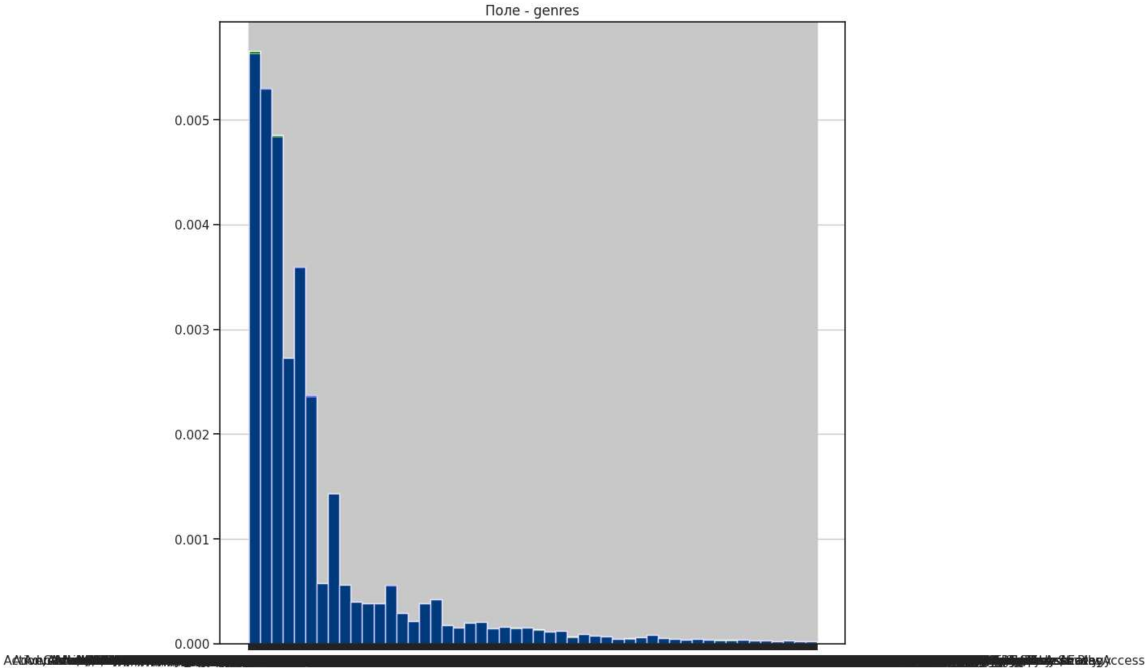
Out[6]:
```
(41975, 7)
```

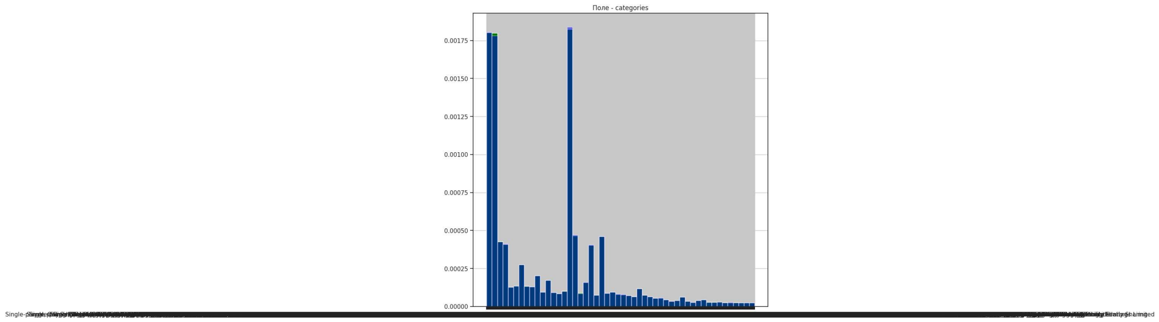
In [7]:
```
def plot_hist_diff(old_ds, new_ds, cols):
    """
    Разница между распределениями до и после устранения пропусков
    """
    for c in cols:
        fig, ax = plt.subplots(figsize=(10,10))
        ax.title.set_text('Поле - ' + str(c))
        old_ds[c].hist(bins=50, ax=ax, density=True, color='green')
        new_ds[c].hist(bins=50, ax=ax, color='blue', density=True, alpha=0.
5)
        plt.show()
```

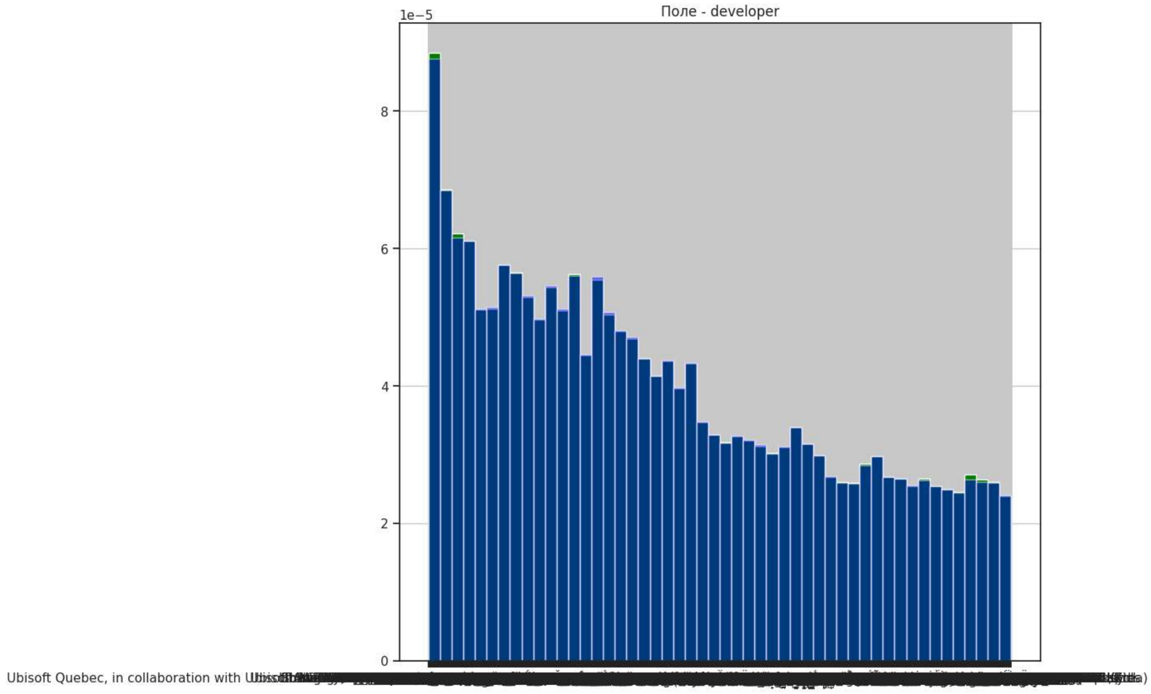
In [11]:

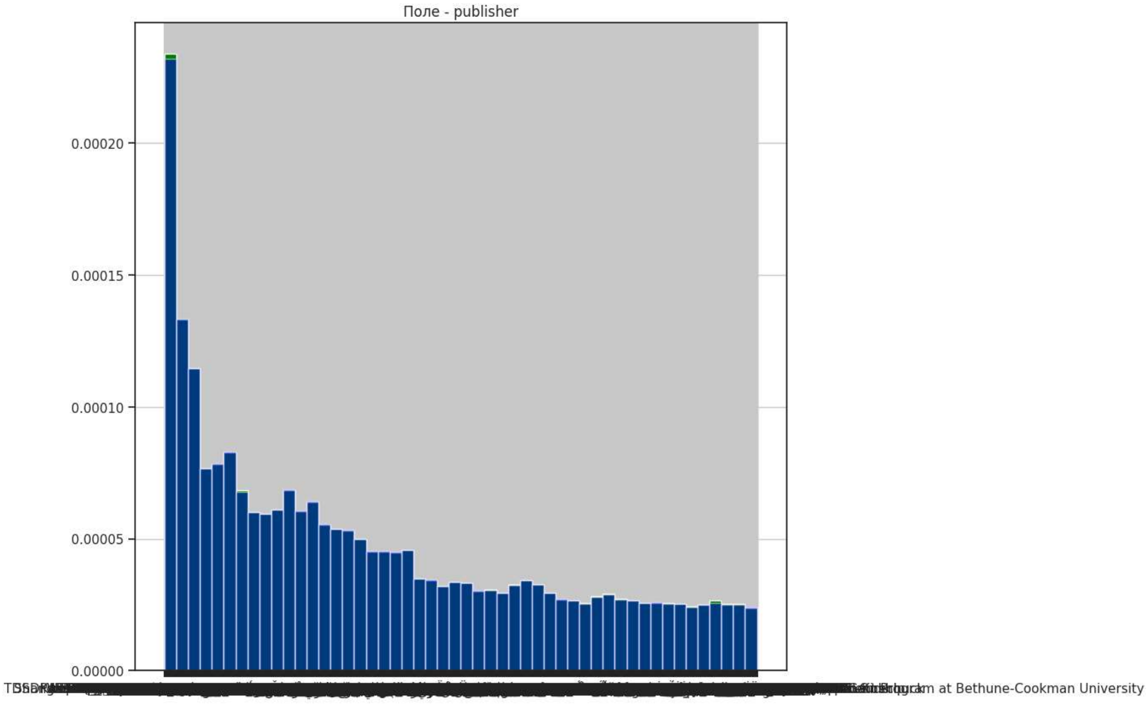
```
plot_hist_diff(data, data_drop_na, colsForDel)
```

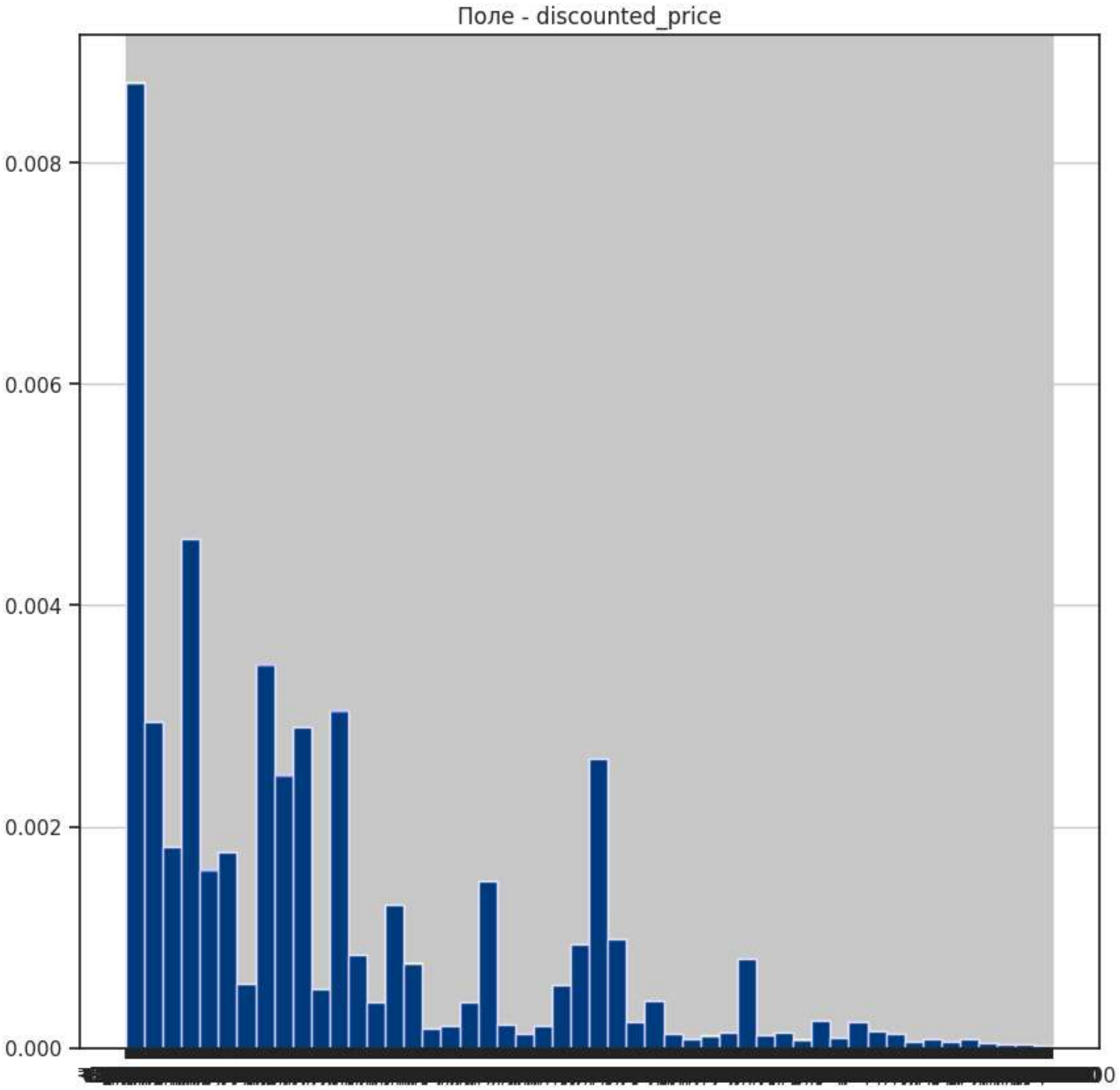

Поле - release_date
0.0004
0.0003
0.0002
0.0001
0.0000


Поле - genres
0.005
0.004
0.003
0.002
0.001
0.000


Поле - categories
0.00175
0.00150
0.00125
0.00100
0.00075
0.00050
0.00025
0.00000
Single-player
Limited


Поле - developer
1e−5
8
6
4
2
0
Ubisoft Quebec, in collaboration with Ubisoft


Поле - publisher
0.00020
0.00015
0.00010
0.00005
0.00000
at Bethune-Cookman University


Поле - discounted_price
0.008
0.006
0.004
0.002
0.000

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/IPython/core/formatters.py in __cal
l__(self, obj)
    339                 pass
    340             else:
--> 341                 return printer(obj)
    342             # Finally look for special method names
    343             method = get_real_method(obj, self.print_method)

/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py in print
_figure(fig, fmt, bbox_inches, base64, **kwargs)
    149         FigureCanvasBase(fig)
    150 
--> 151     fig.canvas.print_figure(bytes_io, **kw)
    152     data = bytes_io.getvalue()
    153     if fmt == 'svg':

/usr/local/lib/python3.10/dist-packages/matplotlib/backend_bases.py in prin
t_figure(self, filename, dpi, facecolor, edgecolor, orientation, format, bb
ox_inches, pad_inches, bbox_extra_artists, backend, **kwargs)
   2340                 )
   2341                 with getattr(renderer, "_draw_disabled", nullconte
xt)():
-> 2342                     self.figure.draw(renderer)
   2343 
   2344             if bbox_inches:

/usr/local/lib/python3.10/dist-packages/matplotlib/artist.py in draw_wrappe
r(artist, renderer, *args, **kwargs)
     93     @wraps(draw)
     94     def draw_wrapper(artist, renderer, *args, **kwargs):
---> 95         result = draw(artist, renderer, *args, **kwargs)
     96         if renderer._rasterizing:
     97             renderer.stop_rasterizing()

/usr/local/lib/python3.10/dist-packages/matplotlib/artist.py in draw_wrappe
r(artist, renderer)
     70                 renderer.start_filter()
     71 
---> 72             return draw(artist, renderer)
     73         finally:
     74             if artist.get_agg_filter() is not None:

/usr/local/lib/python3.10/dist-packages/matplotlib/figure.py in draw(self,
renderer)
   3138 
   3139             self.patch.draw(renderer)
-> 3140             mimage._draw_list_compositing_images(
   3141                 renderer, self, artists, self.suppressComposite)
   3142 

/usr/local/lib/python3.10/dist-packages/matplotlib/image.py in _draw_list_c
ompositing_images(renderer, parent, artists, suppress_composite)
    129     if not_composite or not has_images:
    130         for a in artists:
--> 131             a.draw(renderer)
    132     else:
    133         # Composite any adjacent images together

/usr/local/lib/python3.10/dist-packages/matplotlib/artist.py in draw_wrappe
```

```
r(artist, renderer)
     70                 renderer.start_filter()
     71 
---> 72             return draw(artist, renderer)
     73         finally:
     74             if artist.get_agg_filter() is not None:

/usr/local/lib/python3.10/dist-packages/matplotlib/axes/_base.py in draw(se
lf, renderer)
   3062             _draw_rasterized(self.figure, artists_rasterized, rend
erer)
   3063 
-> 3064         mimage._draw_list_compositing_images(
   3065             renderer, self, artists, self.figure.suppressComposite)
   3066 

/usr/local/lib/python3.10/dist-packages/matplotlib/image.py in _draw_list_c
ompositing_images(renderer, parent, artists, suppress_composite)
    129     if not_composite or not has_images:
    130         for a in artists:
--> 131             a.draw(renderer)
    132     else:
    133         # Composite any adjacent images together

/usr/local/lib/python3.10/dist-packages/matplotlib/artist.py in draw_wrappe
r(artist, renderer)
     70                 renderer.start_filter()
     71 
---> 72             return draw(artist, renderer)
     73         finally:
     74             if artist.get_agg_filter() is not None:

/usr/local/lib/python3.10/dist-packages/matplotlib/axis.py in draw(self, re
nderer, *args, **kwargs)
   1375 
   1376         ticks_to_draw = self._update_ticks()
-> 1377         tlb1, tlb2 = self._get_ticklabel_bboxes(ticks_to_draw, ren
derer)
   1378 
   1379         for tick in ticks_to_draw:

/usr/local/lib/python3.10/dist-packages/matplotlib/axis.py in _get_ticklabe
l_bboxes(self, ticks, renderer)
   1302         if renderer is None:
   1303             renderer = self.figure._get_renderer()
-> 1304         return ([tick.label1.get_window_extent(renderer)
   1305                  for tick in ticks if tick.label1.get_visible()],
   1306                 [tick.label2.get_window_extent(renderer)

/usr/local/lib/python3.10/dist-packages/matplotlib/axis.py in <listcomp>(.
0)
   1302         if renderer is None:
   1303             renderer = self.figure._get_renderer()
-> 1304         return ([tick.label1.get_window_extent(renderer)
   1305                  for tick in ticks if tick.label1.get_visible()],
   1306                 [tick.label2.get_window_extent(renderer)

/usr/local/lib/python3.10/dist-packages/matplotlib/text.py in get_window_ex
tent(self, renderer, dpi)
    957 
    958         with cbook._setattr_cm(self.figure, dpi=dpi):
```

```
--> 959             bbox, info, descent = self._get_layout(self._renderer)
    960             x, y = self.get_unitless_position()
    961             x, y = self.get_transform().transform((x, y))

/usr/local/lib/python3.10/dist-packages/matplotlib/text.py in _get_layout(s
elf, renderer)
    384             clean_line, ismath = self._preprocess_math(line)
    385             if clean_line:
--> 386                 w, h, d = _get_text_metrics_with_cache(
    387                     renderer, clean_line, self._fontproperties,
    388                     ismath=ismath, dpi=self.figure.dpi)

/usr/local/lib/python3.10/dist-packages/matplotlib/text.py in _get_text_met
rics_with_cache(renderer, text, fontprop, ismath, dpi)
     95     # Cached based on a copy of fontprop so that later in-place mu
tations of
     96     # the passed-in argument do not mess up the cache.
---> 97     return _get_text_metrics_with_cache_impl(
     98         weakref.ref(renderer), text, fontprop.copy(), ismath, dpi)
     99 

/usr/local/lib/python3.10/dist-packages/matplotlib/text.py in _get_text_met
rics_with_cache_impl(renderer_ref, text, fontprop, ismath, dpi)
    103         renderer_ref, text, fontprop, ismath, dpi):
    104     # dpi is unused, but participates in cache invalidation (via th
e renderer).
--> 105     return renderer_ref().get_text_width_height_descent(text, font
prop, ismath)
    106 
    107 

/usr/local/lib/python3.10/dist-packages/matplotlib/backends/backend_agg.py
in get_text_width_height_descent(self, s, prop, ismath)
    228         if ismath:
    229             ox, oy, width, height, descent, font_image = \
--> 230                 self.mathtext_parser.parse(s, self.dpi, prop)
    231             return width, height, descent
    232 

/usr/local/lib/python3.10/dist-packages/matplotlib/mathtext.py in parse(sel
f, s, dpi, prop)
    224         # text._get_text_metrics_with_cache for a similar case).
    225         prop = prop.copy() if prop is not None else None
--> 226         return self._parse_cached(s, dpi, prop)
    227 
    228     @functools.lru_cache(50)

/usr/local/lib/python3.10/dist-packages/matplotlib/mathtext.py in _parse_ca
ched(self, s, dpi, prop)
    245             self.__class__._parser = _mathtext.Parser()
    246 
--> 247         box = self._parser.parse(s, fontset, fontsize, dpi)
    248         output = _mathtext.ship(box)
    249         if self._output_type == "vector":

/usr/local/lib/python3.10/dist-packages/matplotlib/_mathtext.py in parse(se
lf, s, fonts_object, fontsize, dpi)
   1993         except ParseBaseException as err:
   1994             # explain becomes a plain method on pyparsing 3 (err.ex
plain(0)).
-> 1995             raise ValueError("\n" + ParseException.explain(err,
0)) from None
   1996         self._state_stack = None
   1997         self._in_subscript_or_superscript = False

ValueError: 
Another story of Bloodlust, this time told through the eyes of Ravenblood;
part Ghost... part Vampire... and 100% bada$$. His journey in search of los
t strength and revenge will take you through the dark tombs and dungeons of
the Vampirem.

^
ParseException: Expected end of text, found '$'  (at char 118), (line:1, co
l:119)
```

```
<Figure size 1000x1000 with 1 Axes>
```

In [12]:

```
data.dtypes
```

Out[12]:

```
app_id                    int64
title                    object
release_date             object
genres                   object
categories               object
developer                object
publisher                object
original_price           object
discount_percentage      object
discounted_price         object
dlc_available             int64
age_rating                int64
content_descriptor       object
about_description        object
win_support                bool
mac_support                bool
linux_support              bool
awards                    int64
overall_review           object
overall_review_%        float64
overall_review_count    float64
recent_review            object
recent_review_%         float64
recent_review_count     float64
dtype: object
```

In [13]:

```
data = data.dropna(subset=colsForDel)
data.shape
```

Out[13]:

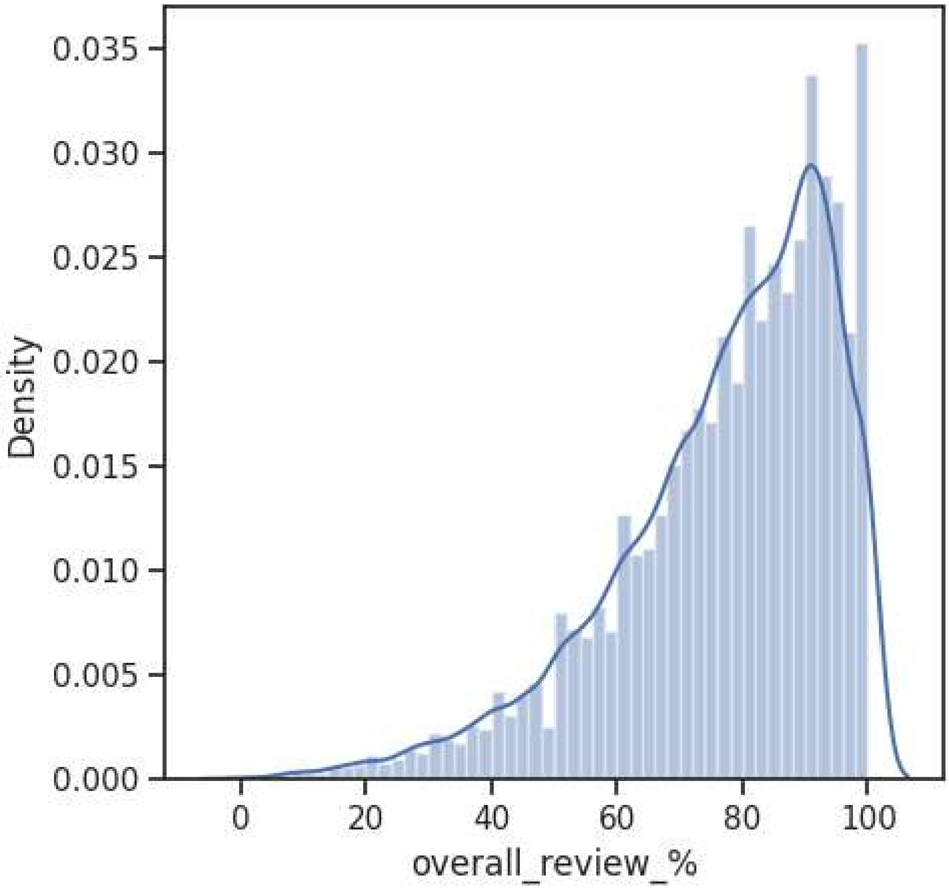
```
(41975, 24)
```

In [14]:
```
data.isnull().sum()
```

Out[14]:
```
app_id                       0
title                        0
release_date                 0
genres                       0
categories                   0
developer                    0
publisher                    0
original_price           37144
discount_percentage      37144
discounted_price             0
dlc_available                0
age_rating                   0
content_descriptor       39634
about_description            0
win_support                  0
mac_support                  0
linux_support                0
awards                       0
overall_review            2340
overall_review_%          2340
overall_review_count      2340
recent_review            36510
recent_review_%          36510
recent_review_count      36510
dtype: int64
```

In [15]:
```
fig, ax = plt.subplots(figsize=(5,5))
sns.distplot(data['overall_review_%'])
```

Out[15]:
```
<Axes: xlabel='overall_review_%', ylabel='Density'>
```



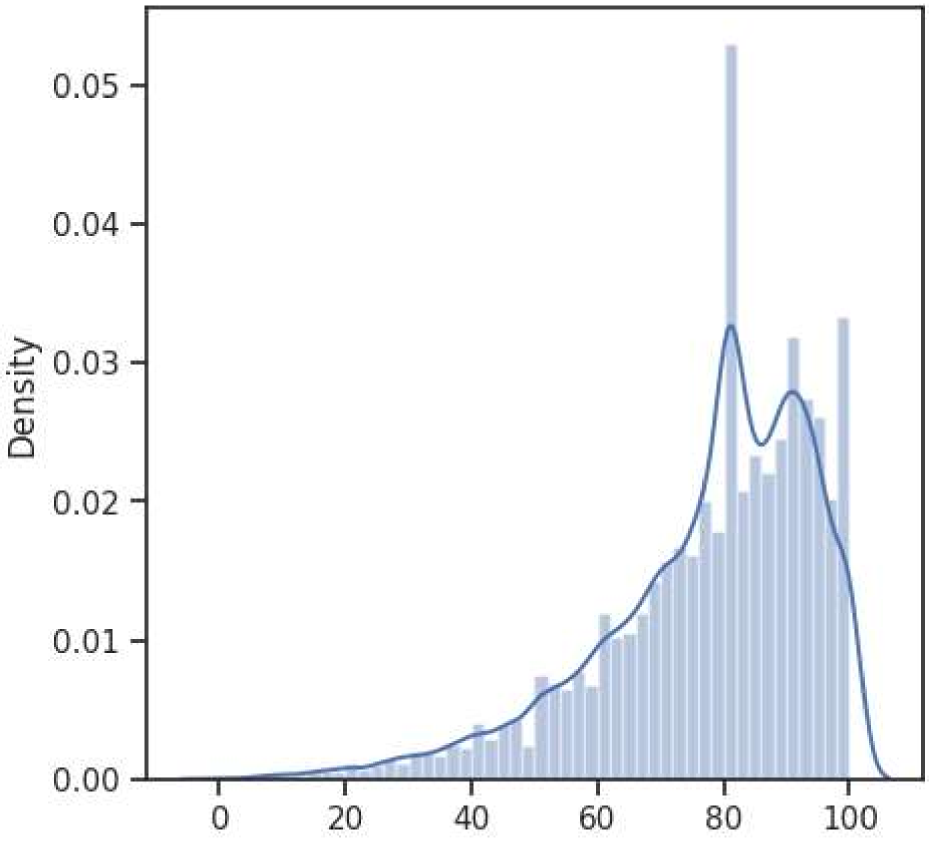
Заполним overall*review*%

```
In [16]: def impute_column(dataset, column, strategy_param, fill_value_param=None):
             """
             Заполнение пропусков в одном признаке
             """
             temp_data = dataset[[column]].values
             size = temp_data.shape[0]

             indicator = MissingIndicator()
             mask_missing_values_only = indicator.fit_transform(temp_data)

             imputer = SimpleImputer(strategy=strategy_param,
                                     fill_value=fill_value_param)
             all_data = imputer.fit_transform(temp_data)

             missed_data = temp_data[mask_missing_values_only]
             filled_data = all_data[mask_missing_values_only]

             return all_data.reshape((size,)), filled_data, missed_data
```

```
In [17]: filled_data, _, _ = impute_column(data, 'overall_review_%', 'median')
```

```
In [18]: fig, ax = plt.subplots(figsize=(5,5))
         sns.distplot(filled_data)
```

```
Out[18]: <Axes: ylabel='Density'>
```



```
In [19]: filled_data
```

```
Out[19]: array([87., 81., 89., ..., 81., 81., 81.])
```

In [20]:

```
knnimpute_hdata = data[['overall_review_%', 'overall_review_count']].copy()
knnimpute_hdata.head()
from sklearn.impute import KNNImputer
knnimputer = KNNImputer(
    n_neighbors=5,
    weights='distance',
    metric='nan_euclidean',
    add_indicator=False,
)
knnimpute_hdata_imputed_temp = knnimputer.fit_transform(knnimpute_hdata)
knnimpute_hdata_imputed = pd.DataFrame(knnimpute_hdata_imputed_temp, columns=knnimpute_hdata.columns)
knnimpute_hdata_imputed.head()
```

Out[20]:

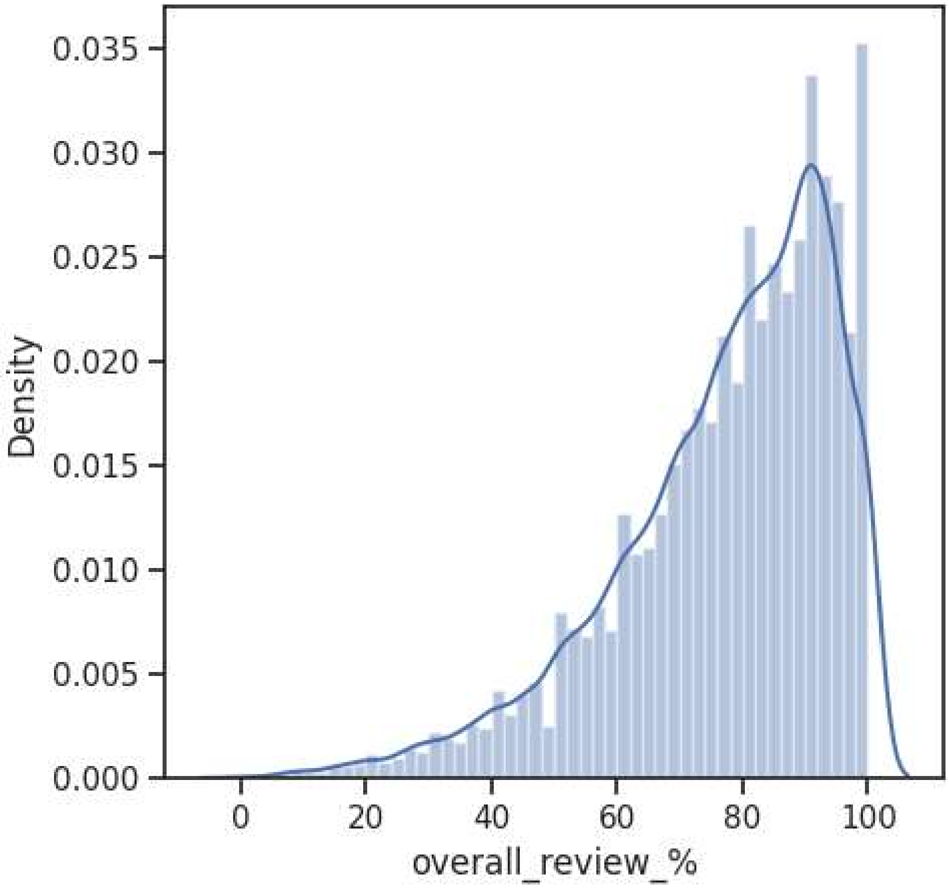
	overall_review_%	overall_review_count
0	87.0	8062218.0
1	81.0	2243112.0
2	89.0	12294.0
3	93.0	605191.0
4	80.0	594713.0

In [21]:

```
fig, ax = plt.subplots(figsize=(5,5))
sns.distplot(knnimpute_hdata['overall_review_%'])
```

Out[21]:

```
<Axes: xlabel='overall_review_%', ylabel='Density'>
```



С помощью импьютации сохранили форму распределения, не создав пиков.

Кодирование категориальных признаков

In [22]:
```
data1 = pd.read_csv('steam-games.csv', sep=",")
```

In [23]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

In [24]:
```
data1.head()
```

Out[24]:

	app_id	title	release_date	genres	categories	developer	publisher
0	730	Counter-Strike 2	21 Aug, 2012	Action, Free to Play	Cross-Platform Multiplayer, Steam Trading Card...	Valve	Valve
1	570	Dota 2	9 Jul, 2013	Action, Strategy, Free to Play	Steam Trading Cards, Steam Workshop, SteamVR C...	Valve	Valve
2	2215430	Ghost of Tsushima DIRECTOR'S CUT	16 May, 2024	Action, Adventure	Single-player, Online Co-op, Steam Achievement...	Sucker Punch Productions	PlayStation PC LLC
3	1245620	ELDEN RING	24 Feb, 2022	Action, RPG	Single-player, Online PvP, Online Co-op, Steam...	FromSoftware Inc.	FromSoftware Inc.
4	1085660	Destiny 2	1 Oct, 2019	Action, Adventure, Free to Play	Single-player, Online PvP, Online Co-op, Steam...	Bungie	Bungie

5 rows × 24 columns

In [25]:
```
data1['overall_review'].unique()
```

Out[25]:
```
array(['Very Positive', 'Overwhelmingly Positive', 'Mixed',
       'Mostly Positive', 'Mostly Negative', 'Overwhelmingly Negative',
       nan, 'Positive', 'Very Negative', 'Negative'], dtype=object)
```

In [26]:
```
cat_enc_le = le.fit_transform(data1['overall_review'])
```

In [27]:
```
np.unique(cat_enc_le)
```

Out[27]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [28]:
```
le.inverse_transform([0, 1, 2, 3])
```

Out[28]:
```
array(['Mixed', 'Mostly Negative', 'Mostly Positive', 'Negative'],
      dtype=object)
```

In [29]:
```
pd.get_dummies(data1[['overall_review']]).head()
```

Out[29]:

	overall_review_Mixed	overall_review_Mostly Negative	overall_review_Mostly Positive	overall_review_Negative
0	False	False	False	False
1	False	False	False	False
2	False	False	False	False
3	False	False	False	False
4	False	False	False	False

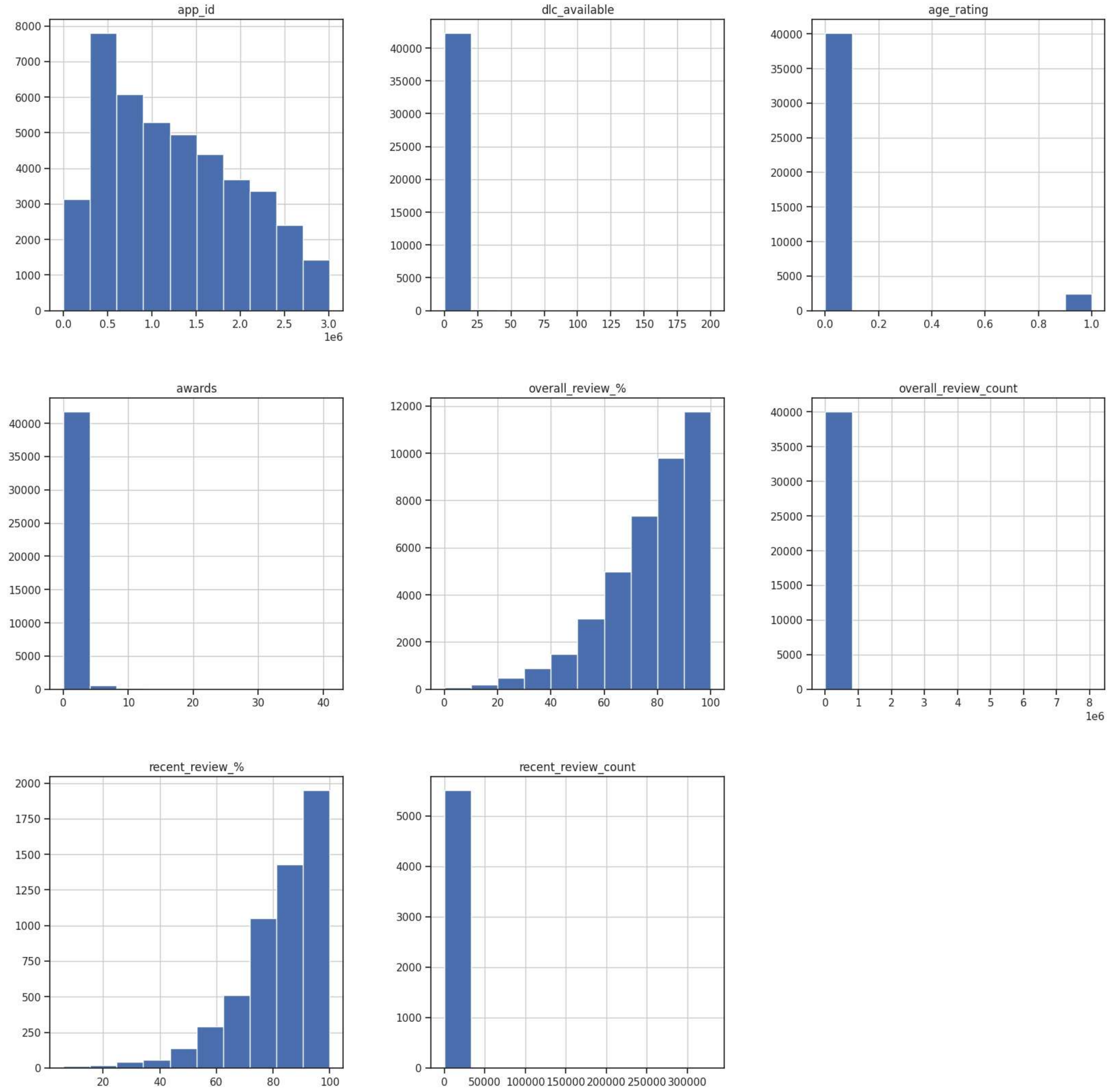
Нормализация числовых признаков

In [30]:
```
data2 = pd.read_csv('steam-games.csv', sep=",")
def diagnostic_plots(df, variable):
    plt.figure(figsize=(15,6))
    # гистограмма
    plt.subplot(1, 2, 1)
    df[variable].hist(bins=30)
    ## Q-Q plot
    plt.subplot(1, 2, 2)
    stats.probplot(df[variable], dist="norm", plot=plt)
    plt.show()
```

In [31]:
```
data2.dtypes
```

Out[31]:
```
app_id                    int64
title                    object
release_date             object
genres                   object
categories               object
developer                object
publisher                object
original_price           object
discount_percentage      object
discounted_price         object
dlc_available             int64
age_rating                int64
content_descriptor       object
about_description        object
win_support                bool
mac_support                bool
linux_support              bool
awards                    int64
overall_review           object
overall_review_%        float64
overall_review_count    float64
recent_review            object
recent_review_%         float64
recent_review_count     float64
dtype: object
```

In [32]:
```
data2.hist(figsize=(20,20))
plt.show()
```



In [33]:

```
from sklearn.preprocessing import MinMaxScaler
# Обучаем StandardScaler на всей выборке и масштабируем
cs31 = MinMaxScaler()
data_cs31_scaled_temp = cs31.fit_transform(data2[['overall_review_count']])
# формируем DataFrame на основе массива
data_scaled =pd.DataFrame(data_cs31_scaled_temp, columns=['overall_review_count'])
data_scaled.describe()
```

Out[33]:

	overall_review_count
count	40020.000000
mean	0.000309
std	0.006063
min	0.000000
25%	0.000001
50%	0.000006
75%	0.000034
max	1.000000

In [34]:

```
data_scaled.loc[data_scaled['overall_review_count']==0]
```

Out[34]:

	overall_review_count
6185	0.0
7037	0.0
7210	0.0
7386	0.0
7613	0.0
...	...
40392	0.0
40415	0.0
40442	0.0
40833	0.0
40837	0.0

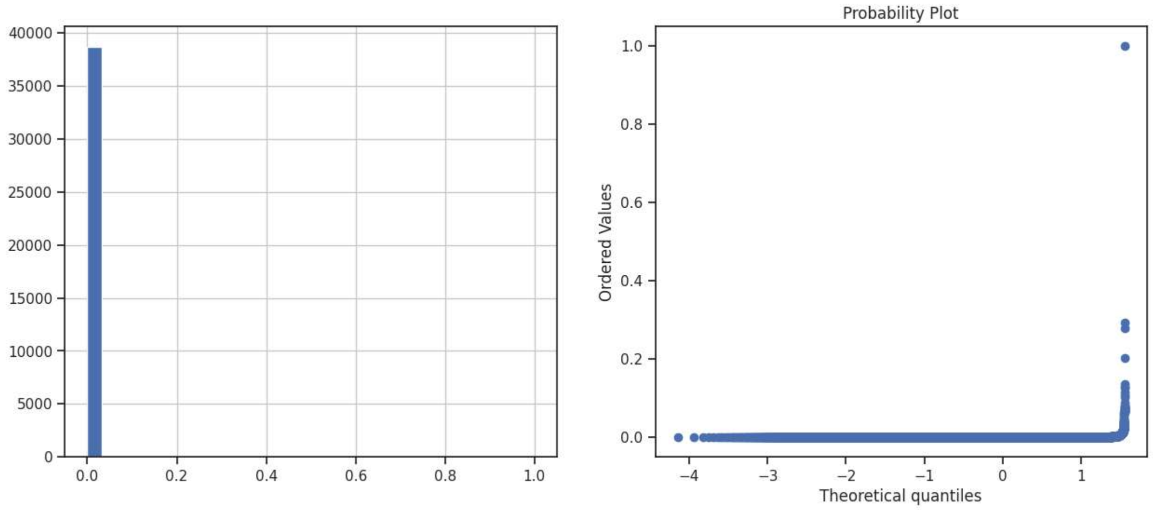
1299 rows × 1 columns

In [35]:

```
data_scaled = data_scaled.loc[data_scaled['overall_review_count']!=0]
```

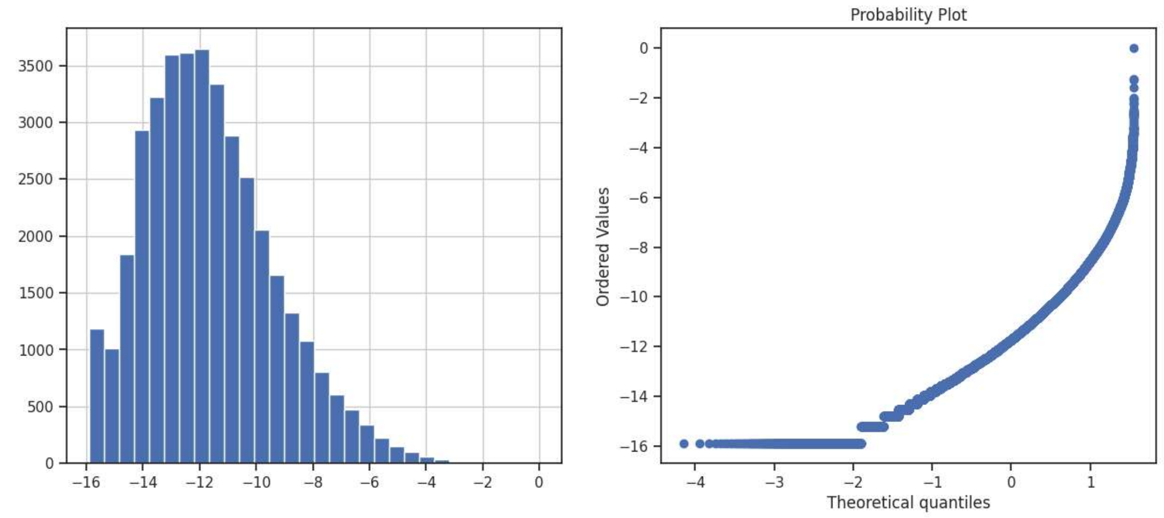
In [36]:

```
diagnostic_plots(data_scaled, 'overall_review_count')
```



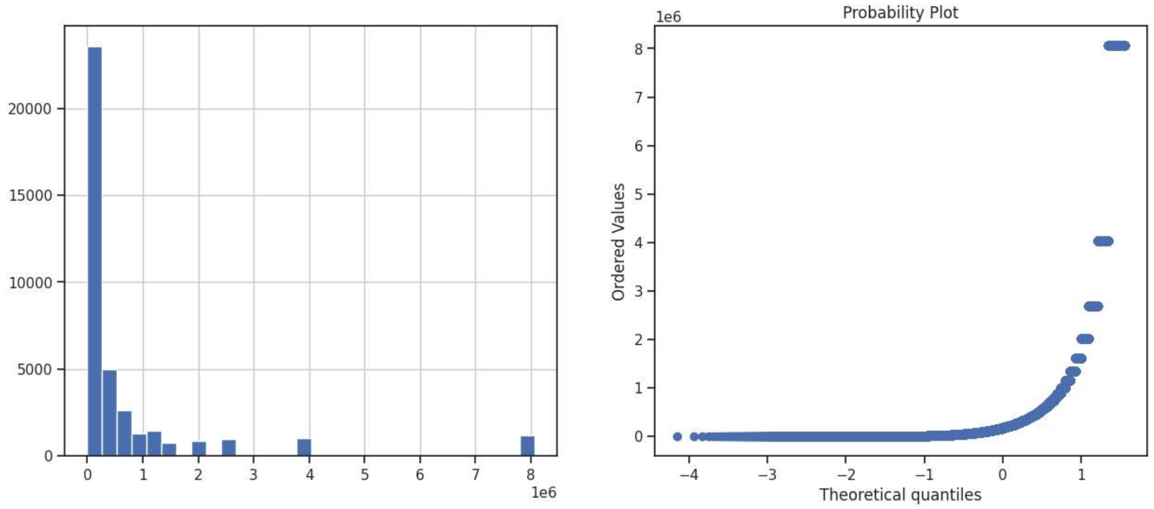
In [37]:

```
# логарифмическое
data_scaled['norm_log'] = np.log(data_scaled['overall_review_count'])
diagnostic_plots(data_scaled, 'norm_log')
```



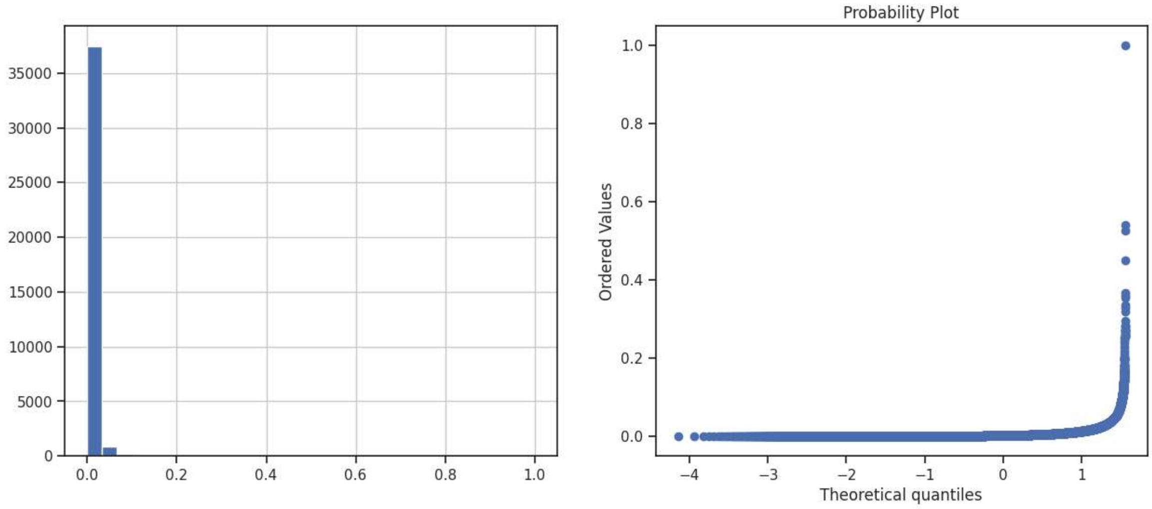
In [38]:

```
# обратное
data_scaled['norm_reciprocal'] = 1 / (data_scaled['overall_review_count'])
diagnostic_plots(data_scaled, 'norm_reciprocal')
```



In [42]:

```
# root
data_scaled['norm_sqr'] = data_scaled['overall_review_count']**(1/2)
diagnostic_plots(data_scaled, 'norm_sqr')
```


Probability Plot
Ordered Values
Theoretical quantiles