# Рубежный контроль № 1

## Задания

### Задача №9.

Для набора данных проведите устранение пропусков для одного (произвольного) числового признака с использованием метода заполнения "хвостом распределения".

### Задача №29.

Для набора данных проведите удаление константных и псевдоконстантных признаков.

## Доп. задание

Для произвольной колонки данных построить график "Ящик с усами (boxplot)"

In [1]:
```
import pandas as pd
import numpy as np
from sklearn.impute import MissingIndicator
from sklearn.impute import SimpleImputer
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as stats
from sklearn.datasets import load_boston
```

In [2]:
```
data = pd.read_csv('IPODataFull.csv', sep = ',')
```

```
/Users/lina/Documents/University Shit/Master/sem_2/ML/.venv/lib/python3.7/site
-packages/IPython/core/interactiveshell.py:3166: DtypeWarning: Columns (1342,1
425,1432,1543,1546,1549,1551,1552,1553,1562,1587,1588,1605,1608,1615,1619,162
0,1621,1622,1629,1630,1632,1633,1640,1641,1642,1643,1644,1646) have mixed type
s.Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

In [3]:
```
data.shape
```

Out[3]:
```
(3762, 1664)
```

In [4]:
```
data
```

Out[4]:

| | Symbol | DaysBetterThanSP | daysProfit | daysProfitGrouped | exactDiffernce | Year | Month |
|---|---|---|---|---|---|---|---|
| **0** | A | 122 | 249 | 200+ | NaN | 1999 | 1 |
| **1** | AAC | 131 | 262 | 200+ | 232.0 | 2014 | 10 |
| **2** | AAOI | 125 | 262 | 200+ | 6054.0 | 2013 | 9 |
| **3** | AAP | 128 | 261 | 200+ | NaN | 2001 | 1 |
| **4** | AAT | 123 | 127 | 100 - 149 | 181.0 | 2011 | |
| **...** | ... | ... | ... | ... | ... | ... | ... |

| | Symbol | DaysBetterThanSP | daysProfit | daysProfitGrouped | exactDiffernce | Year | Month |
|---|---|---|---|---|---|---|---|
| **3757** | ZUMZ | 139 | 261 | 200+ | NaN | 2005 | 5 |
| **3758** | ZUO | 5 | 7 | 0 - 49 | NaN | 2018 | 4 |
| **3759** | ZX | 102 | 25 | 0 - 49 | 1035.0 | 2011 | 5 |
| **3760** | ZYME | 115 | 19 | 0 - 49 | NaN | 2017 | 4 |
| **3761** | ZYNE | 104 | 42 | 0 - 49 | 3108.0 | 2015 | 8 |

3762 rows × 1664 columns

# Заполнение пропусков

In [5]:
```
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 [6]:
```
cols_with_na = [c for c in data.columns if data[c].isnull().sum() > 0]
cols_with_na
```

Out[6]:
```
['exactDiffernce',
 'highDay0',
 'openDay0',
 'lowDay0',
 'volumeDay0',
 'closeDay1',
 'highDay1',
 'openDay1',
 'lowDay1',
 'volumeDay1',
 'closeDay2',
 'highDay2',
 'openDay2',
 'lowDay2',
 'volumeDay2',
 'closeDay3',
 'highDay3',
 'openDay3',
 'lowDay3',
 'volumeDay3',
 'closeDay4',
 'highDay4',
 'openDay4',
 'lowDay4',
 'volumeDay4',
 'closeDay5',
 'highDay5',
 'openDay5',
 'lowDay5',
 'volumeDay5',
 'closeDay6',
 'highDay6',
 'openDay6',
 'lowDay6',
 'volumeDay6',
```

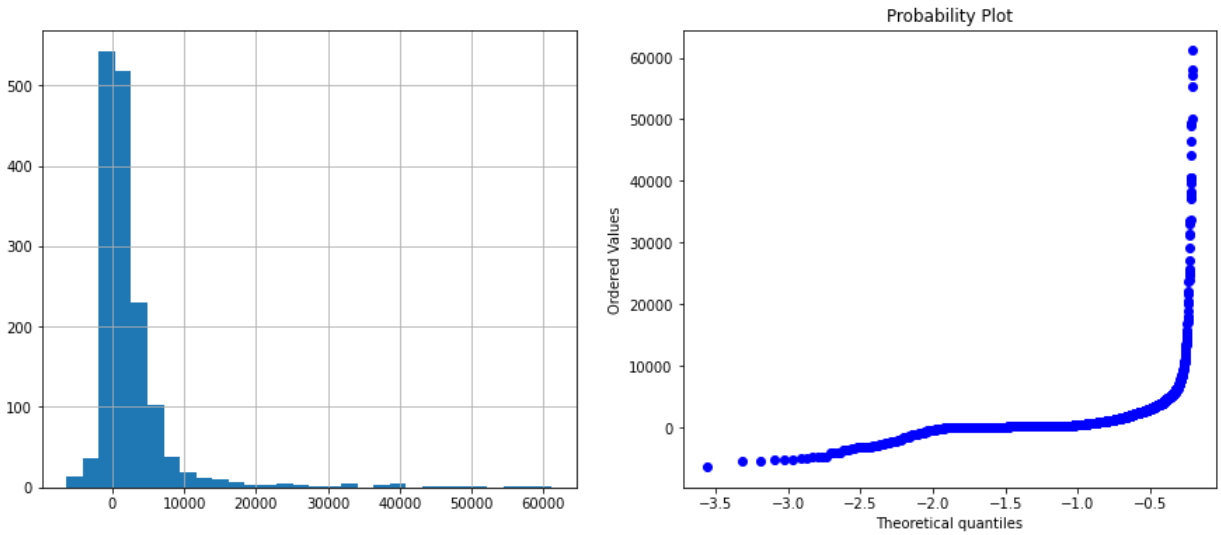
```
 ('volumeDay192', 196),
 ('closeDay193', 184),
 ('highDay193', 247),
 ('openDay193', 247),
 ('lowDay193', 276),
 ('volumeDay193', 197),
 ('closeDay194', 184),
 ('highDay194', 245),
 ('openDay194', 245),
 ('lowDay194', 270),
 ('volumeDay194', 197),
 ('closeDay195', 184),
 ('highDay195', 250),
 ('openDay195', 250),
 ('lowDay195', 280),
 ('volumeDay195', 197),
 ('closeDay196', 186),
 ('highDay196', 250),
 ('openDay196', 250),
 ('lowDay196', 281),
 ('volumeDay196', 199),
 ('closeDay197', 189),
 ('highDay197', 245),
 ('openDay197', 245),
 ('lowDay197', 274),
 ('volumeDay197', 202),
 ('closeDay198', 190),
 ('highDay198', 246),
 ('openDay198', 246),
 ('lowDay198', 274),
 ('volumeDay198', 203),
 ('closeDay199', 191),
 ('highDay199', 250),
 ('openDay199', 250),
 ('lowDay199', 279),
 ('volumeDay199', 204),
 ...]
```

In [8]:
```
sorted([(c, data[c].isnull().mean()) for c in cols_with_na], key=lambda x: x[
```

Out[8]:
```
[('Fiscal_year_ends_in_October_USDYearBeforeIPO', 0.9997341839447103),
 ('Pensions_and_other_postretirement_benefitsYearBeforeIPO',
  0.9997341839447103),
 ('Fiscal_year_ends_in_December_BRLYearBeforeIPO', 0.9997341839447103),
 ('Fiscal_year_ends_in_December_CADYearBeforeIPO', 0.9997341839447103),
 ('Fiscal_year_ends_in_December_JPYYearBeforeIPO', 0.9997341839447103),
 ('Fiscal_year_ends_in_May_USDYearBeforeIPO', 0.9997341839447103),
 ('Fiscal_year_ends_in_December_CLPYearBeforeIPO', 0.9997341839447103),
 ('Regulatory_liabilitiesYearBeforeIPO', 0.9997341839447103),
 ('Prepaid_pension_costsYearBeforeIPO', 0.9997341839447103),
 ('Sales_of_intangiblesYearBeforeIPO', 0.9997341839447103),
 ('Derivative_liabilitiesYearBeforeIPO', 0.9997341839447103),
 ('Derivative_assetsYearBeforeIPO', 0.9997341839447103),
 ('Fiscal_year_ends_in_March_CADYearBeforeIPO', 0.9997341839447103),
 ('Deferred_tax_liabilitiesYearBeforeIPO', 0.9997341839447103),
 ('Fiscal_year_ends_in_August_CNYYearBeforeIPO', 0.9997341839447103),
 ('Loans,_totalYearBeforeIPO', 0.9997341839447103),
 ('Trading_securitiesYearBeforeIPO', 0.9997341839447103),
 ('Loans_issuedYearBeforeIPO', 0.9997341839447103),
 ('(Gains)_loss_on_disposition_of_businessesYearBeforeIPO',
  0.9997341839447103),
 ('Fiscal_year_ends_in_March_CNYYearBeforeIPO', 0.9997341839447103),
 ('Fiscal_year_ends_in_September_EURYearBeforeIPO', 0.9997341839447103),
 ('Fiscal_year_ends_in_December_DKKYearBeforeIPO', 0.9997341839447103),
 ('Redemption_of_preferred_stockYearBeforeIPO', 0.9997341839447103),
 ('Other_equityYearBeforeIPO', 0.9997341839447103),
 ('Change_in_federal_funds_sold_and_securities_purchased_under_resale_agreemen
tsYearBeforeIPO',
```

```
('lowDay61', 0.045454545454545456),
('closeDay181', 0.045188729399255716),
('openDay121', 0.045188729399255716),
('highDay121', 0.045188729399255716),
('openDay120', 0.045188729399255716),
('highDay120', 0.045188729399255716),
('lowDay75', 0.045188729399255716),
('lowDay71', 0.045188729399255716),
('lowDay65', 0.045188729399255716),
('volumeDay160', 0.044922913343965976),
('volumeDay159', 0.044922913343965976),
('volumeDay158', 0.044922913343965976),
('volumeDay157', 0.044922913343965976),
('lowDay77', 0.044922913343965976),
('lowDay63', 0.044922913343965976),
('closeDay180', 0.044657097288676235),
('closeDay179', 0.044657097288676235),
('closeDay178', 0.044657097288676235),
('volumeDay156', 0.044657097288676235),
('openDay119', 0.044657097288676235),
('highDay119', 0.044657097288676235),
('lowDay96', 0.044657097288676235),
('lowDay64', 0.044657097288676235),
('lowDay62', 0.044657097288676235),
...]
```

In [9]:
```
diagnostic_plots(data, 'exactDiffernce')
```




In [10]:
```
exactDiffernce = data['exactDiffernce'].mean() + 1.5*data['exactDiffernce'].s
exactDiffernce
```

Out[10]:
```
12224.959963645306
```

In [11]:
```
data['exactDiffernce'].isnull().sum()
```

Out[11]:
```
2194
```

In [12]:
```
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
```

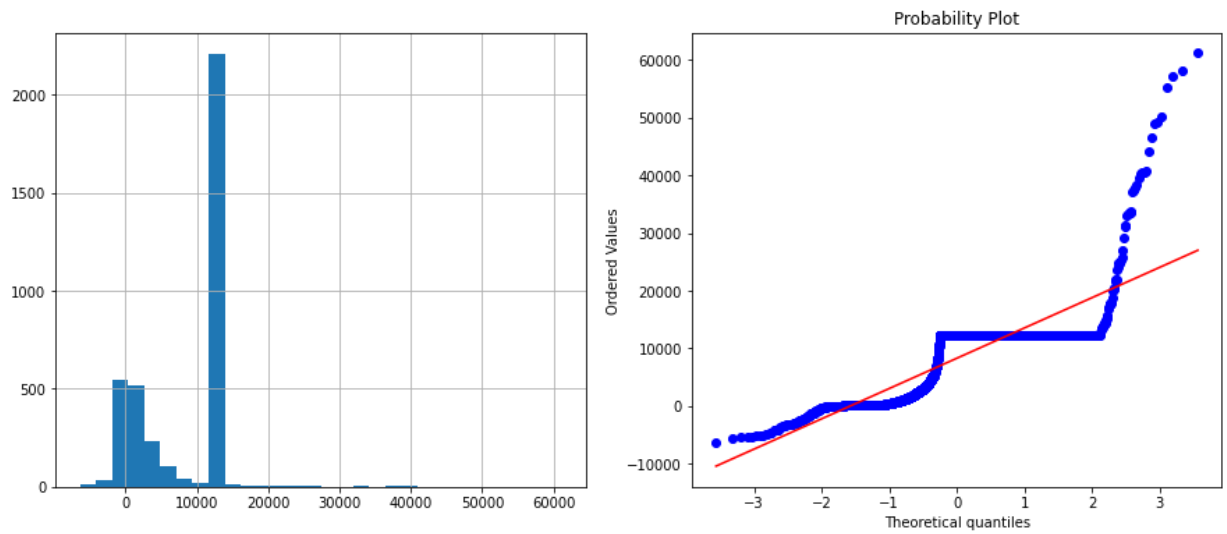
In [13]:
```
new_data, x, y = impute_column(data, 'exactDiffernce', 'constant', exactDiffe
```

In [14]:
```
data['exactDiffernce']=new_data
```

In [15]:
```
data['exactDiffernce'].isnull().sum()
```

Out[15]: 0

In [16]:
```
diagnostic_plots(data, 'exactDiffernce')
```




## Удаление константных и псевдоконстантных признаков

In [19]:
```
from sklearn.feature_selection import VarianceThreshold
```

In [136…
```
data = pd.read_csv('dataset.csv', sep = ',')
print(data.shape)
```

```
(345, 20)
```

In [138…
```
data.head()
```

Out[138…

| | f1 | f2 | f3 | f4 | f5 | f6 | f7 | f8 | f9 | f10 | f11 | f12 | f13 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.341 | 0.325 | 0.764 | 0.771 | 0.201 | 0 | 0.138 | 0.247 | 0.406 | 0.905 | 0.213 | 0.695 | 0.352 | 0. |

| | f1 | f2 | f3 | f4 | f5 | f6 | f7 | f8 | f9 | f10 | f11 | f12 | f13 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0.840 | 0.845 | 0.617 | 0.142 | 0.269 | 0 | 0.466 | 0.918 | 0.885 | 0.042 | 0.796 | 0.353 | 0.405 | 0.5 |
| **2** | 0.251 | 0.263 | 0.842 | 0.020 | 0.960 | 0 | 0.218 | 0.441 | 0.560 | 0.282 | 0.587 | 0.193 | 0.642 | 0. |
| **3** | 0.447 | 0.125 | 0.028 | 0.889 | 0.771 | 0 | 0.735 | 0.249 | 0.284 | 0.365 | 0.125 | 0.796 | 0.770 | 0.5 |
| **4** | 0.696 | 0.736 | 0.177 | 0.976 | 0.782 | 0 | 0.483 | 0.560 | 0.470 | 0.458 | 0.484 | 0.331 | 0.797 | 0. |

In [141…

```
selector = VarianceThreshold(threshold=0.15)
selector.fit(data)
selector.variances_
# Оно не работает, поэтому я сделаю другим способом
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-141-d186ae3cc664> in <module>
      1 selector = VarianceThreshold(threshold=0.15)
----> 2 selector.fit(data)
      3 selector.variances_
      4 # Оно не работает, поэтому я сделаю другим способом

~/Documents/University Shit/Master/sem_2/ML/.venv/lib/python3.7/site-packages/
sklearn/feature_selection/_variance_threshold.py in fit(self, X, y)
     91             if X.shape[0] == 1:
     92                 msg += " (X contains only one sample)"
---> 93             raise ValueError(msg.format(self.threshold))
     94
     95         return self

ValueError: No feature in X meets the variance threshold 0.15000
```

In [145…

```
[(c, len(data[c].unique())) for c in data]
```

Out[145…

```
[('f1', 290),
 ('f2', 287),
 ('f3', 292),
 ('f4', 278),
 ('f5', 296),
 ('f6', 1),
 ('f7', 291),
 ('f8', 288),
 ('f9', 289),
 ('f10', 293),
 ('f11', 302),
 ('f12', 295),
 ('f13', 291),
 ('f14', 277),
 ('f15', 292),
 ('f16', 302),
 ('f17', 294),
 ('f18', 3),
 ('f19', 288),
 ('f20', 291)]
```

In [147…

```
columns = [c for c in data if len(data[c].unique())>3]
```

In [148…

```
data=data[columns]
```

In [150…

```
data
```

Out[150…

| | f1 | f2 | f3 | f4 | f5 | f7 | f8 | f9 | f10 | f11 | f12 | f13 | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.341 | 0.325 | 0.764 | 0.771 | 0.201 | 0.138 | 0.247 | 0.406 | 0.905 | 0.213 | 0.695 | 0.352 | 0.0 |
| **1** | 0.840 | 0.845 | 0.617 | 0.142 | 0.269 | 0.466 | 0.918 | 0.885 | 0.042 | 0.796 | 0.353 | 0.405 | 0.5 |
| **2** | 0.251 | 0.263 | 0.842 | 0.020 | 0.960 | 0.218 | 0.441 | 0.560 | 0.282 | 0.587 | 0.193 | 0.642 | 0.1 |
| **3** | 0.447 | 0.125 | 0.028 | 0.889 | 0.771 | 0.735 | 0.249 | 0.284 | 0.365 | 0.125 | 0.796 | 0.770 | 0.5 |
| **4** | 0.696 | 0.736 | 0.177 | 0.976 | 0.782 | 0.483 | 0.560 | 0.470 | 0.458 | 0.484 | 0.331 | 0.797 | 0.4 |
| **…** | … | … | … | … | … | … | … | … | … | … | … | … | |
| **340** | 0.440 | 0.945 | 0.147 | 0.564 | 0.334 | 0.292 | 0.742 | 0.521 | 0.572 | 0.488 | 0.347 | 0.917 | 0.5 |
| **341** | 0.371 | 0.236 | 0.417 | 0.371 | 0.047 | 0.491 | 0.868 | 0.783 | 0.360 | 0.755 | 0.931 | 0.842 | 0.1 |
| **342** | 0.548 | 0.012 | 0.963 | 0.048 | 0.241 | 0.436 | 0.832 | 0.371 | 0.029 | 0.928 | 0.004 | 0.083 | 0.8 |
| **343** | 0.860 | 0.086 | 0.185 | 0.763 | 0.842 | 0.317 | 0.468 | 0.702 | 0.269 | 0.519 | 0.331 | 0.930 | 0.8 |
| **344** | 0.919 | 0.819 | 0.806 | 0.626 | 0.174 | 0.257 | 0.085 | 0.100 | 0.265 | 0.394 | 0.019 | 0.360 | 0.7 |

345 rows × 18 columns

## Ящик с усами

In [151…

```
sns.boxplot(x=data['f9'])
```

Out[151… <AxesSubplot:xlabel='f9'>