Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Лабораторная работа №2
«Обработка признаков. Часть 1»
по дисциплине «Методы машинного обучения»

Выполнил:
студент группы ИУ5-25М
Тураев Г.В.
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва 2023

Цель: изучение продвинутых способов предварительной обработки данных для дальнейшего формирования моделей.

Для данной лабораторной работы выберем датасет: house_sales.

Импортируем нужные нам библиотеки и выведем:

```
[1] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    import scipy.stats as stats
```

```
[2] data = pd.read_csv("./house_sales.csv")
```

```
[3] data = data.drop('Id', 1)
    data.head()
```

```
<ipython-input-3-c100a8de87ec>:1: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only.
  data = data.drop('Id', 1)
```

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	sale cond.	p
0	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Inside	...	0	NaN	NaN	NaN	0	2	2008	WD	Normal	208
1	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2	...	0	NaN	NaN	NaN	0	5	2007	WD	Normal	181
2	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Inside	...	0	NaN	NaN	NaN	0	9	2008	WD	Normal	223
3	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Corner	...	0	NaN	NaN	NaN	0	2	2006	WD	Abnorml	140
4	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	FR2	...	0	NaN	NaN	NaN	0	12	2008	WD	Normal	250

5 rows × 80 columns

Проверим, есть ли пропущенные значения:

```
[4] data_features = list(zip(
    # признаки
    [i for i in data.columns],
    zip(
        # типы колонок
        [str(i) for i in data.dtypes],
        [i for i in data.isnull().sum()]
    )))
    # Признаки с типом данных и количеством пропусков
    data_features
```

```
 ('Exterior1st', ('object', 1)),
 ('Exterior2nd', ('object', 1)),
 ('MasVnrType', ('object', 24)),
 ('MasVnrArea', ('float64', 23))
```

...

```
('Fence', ('object', 2348)),
('MiscFeature', ('object', 2814)),
('MiscVal', ('int64', 0)),
('MoSold', ('int64', 0)),
('YrSold', ('int64', 0)),
('SaleType', ('object', 1)),
('sale cond.', ('object', 0)),
('sale price ', ('float64', 1459))]
```

Устранение пропусков

Доля пропусков (в процентах):

```
[5] [(c, data[c].isnull().mean()) for c in data.columns]
```

```
[('MSSubClass', 0.0),
 ('MSZoning', 0.0013703323055841042),
 ('LotFrontage', 0.16649537512846865),
 ('LotArea', 0.0),
 ('Street', 0.0),
 ('Alley', 0.9321685508735869),
```

...

```
('MiscVal', 0.0),
('MoSold', 0.0),
('YrSold', 0.0),
('SaleType', 0.0003425830763960260 4),
('sale cond.', 0.0),
('sale price ', 0.499828708461802)]
```

Удалим колонки, которые содержат пустые значения:

```
[6] data.dropna(axis=1, how='any')
```

	MSSubClass	LotArea	Street	LotShape	LandContour	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	...	WoodDeckSF	OpenPorchSF	EnclosedPorch	3SsnPorch	ScreenPorch	PoolArea
0	60	8450	Pave	Reg	Lvl	Inside	Gtl	CollgCr	Norm	Norm	...	0	61	0	0	0	0
1	20	9600	Pave	Reg	Lvl	FR2	Gtl	Veenker	Feedr	Norm	...	298	0	0	0	0	0
2	60	11250	Pave	IR1	Lvl	Inside	Gtl	CollgCr	Norm	Norm	...	0	42	0	0	0	0
3	70	9550	Pave	IR1	Lvl	Corner	Gtl	Crawfor	Norm	Norm	...	0	35	272	0	0	0
4	60	14260	Pave	IR1	Lvl	FR2	Gtl	NoRidge	Norm	Norm	...	192	84	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2914	160	1936	Pave	Reg	Lvl	Inside	Gtl	MeadowV	Norm	Norm	...	0	0	0	0	0	0
2915	160	1894	Pave	Reg	Lvl	Inside	Gtl	MeadowV	Norm	Norm	...	0	24	0	0	0	0
2916	20	20000	Pave	Reg	Lvl	Inside	Gtl	Mitchel	Norm	Norm	...	474	0	0	0	0	0
2917	85	10441	Pave	Reg	Lvl	Inside	Gtl	Mitchel	Norm	Norm	...	80	32	0	0	0	0
2918	60	9627	Pave	Reg	Lvl	Inside	Mod	Mitchel	Norm	Norm	...	190	48	0	0	0	0

2919 rows × 45 columns

Удалим колонки с высоким процентом (более 50%) пропусков:

```
[7] data.dropna(axis=1, thresh=730)
```

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	...	EnclosedPorch	3SsnPorch	ScreenPorch	PoolArea	MiscVal	MoSold	YrSold
0	60	RL	65.0	8450	Pave	Reg	Lvl	AllPub	Inside	Gtl	...	0	0	0	0	0	2	2008
1	20	RL	80.0	9600	Pave	Reg	Lvl	AllPub	FR2	Gtl	...	0	0	0	0	0	5	2007
2	60	RL	68.0	11250	Pave	IR1	Lvl	AllPub	Inside	Gtl	...	0	0	0	0	0	9	2008
3	70	RL	60.0	9550	Pave	IR1	Lvl	AllPub	Corner	Gtl	...	272	0	0	0	0	2	2006
4	60	RL	84.0	14260	Pave	IR1	Lvl	AllPub	FR2	Gtl	...	0	0	0	0	0	12	2008
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2914	160	RM	21.0	1936	Pave	Reg	Lvl	AllPub	Inside	Gtl	...	0	0	0	0	0	6	2006
2915	160	RM	21.0	1894	Pave	Reg	Lvl	AllPub	Inside	Gtl	...	0	0	0	0	0	4	2006
2916	20	RL	160.0	20000	Pave	Reg	Lvl	AllPub	Inside	Gtl	...	0	0	0	0	0	9	2006
2917	85	RL	62.0	10441	Pave	Reg	Lvl	AllPub	Inside	Gtl	...	0	0	0	0	700	7	2006
2918	60	RL	74.0	9627	Pave	Reg	Lvl	AllPub	Inside	Mod	...	0	0	0	0	0	11	2006

Заполним пропуски возраста средними значениями:

```
[8] def impute_na(df, variable, value):
        df[variable].fillna(value, inplace=True)
    impute_na(data, 'LotFrontage', data['LotFrontage'].mean())
```

Удостоверимся, что признак LotFrontage не имеет пустых значений:

```
[9] data.isnull().sum()

MSSubClass        0
MSZoning          4
LotFrontage       0
LotArea           0
Street            0
                 ...
MoSold            0
YrSold            0
SaleType          1
sale cond.        0
sale price     1459
Length: 80, dtype: int64
```

Кодирование категориальных признаков

```
[10] from sklearn.preprocessing import LabelEncoder
```

```
[13] le = LabelEncoder()
     cat_enc_le = le.fit_transform(data['sale cond.'])
```

```
[14] data['sale cond.'].unique()
```

```
array(['Normal', 'Abnorml', 'Partial', 'AdjLand', 'Alloca', 'Family'],
      dtype=object)
```

```
[15] np.unique(cat_enc_le)
```

```
array([0, 1, 2, 3, 4, 5])
```

```
[16] le.inverse_transform([0, 1, 2, 3, 4, 5])
```

```
array(['Abnorml', 'AdjLand', 'Alloca', 'Family', 'Normal', 'Partial'],
      dtype=object)
```

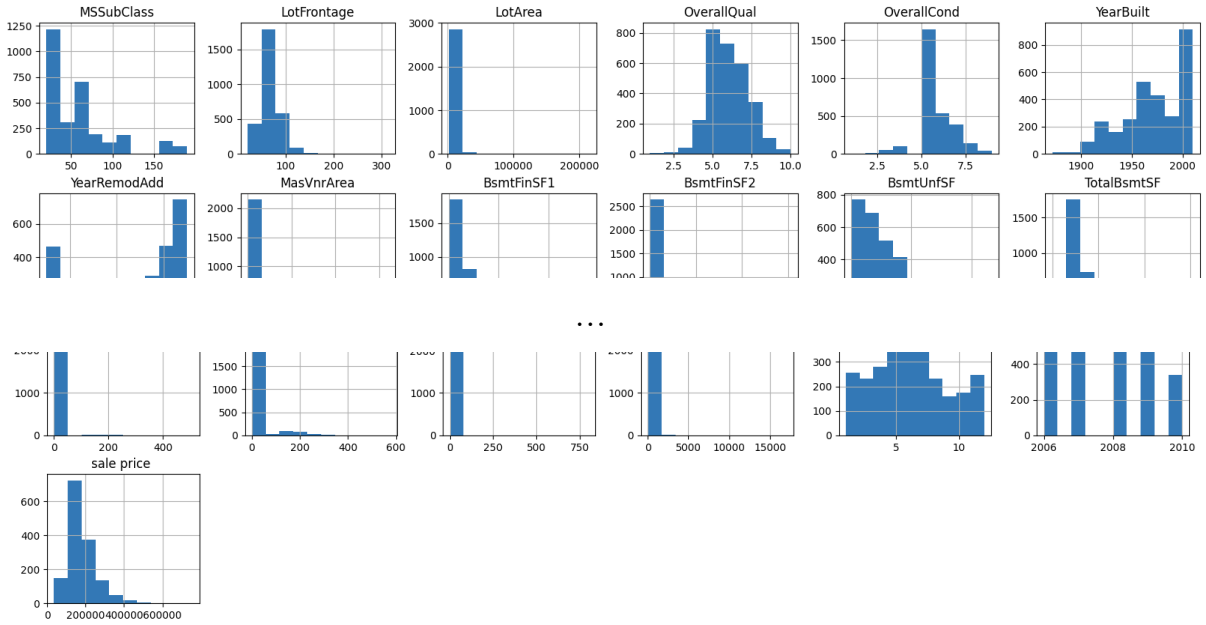
```
[17] data['LotConfig'].unique()
```

```
array(['Inside', 'FR2', 'Corner', 'CulDSac', 'FR3'], dtype=object)
```

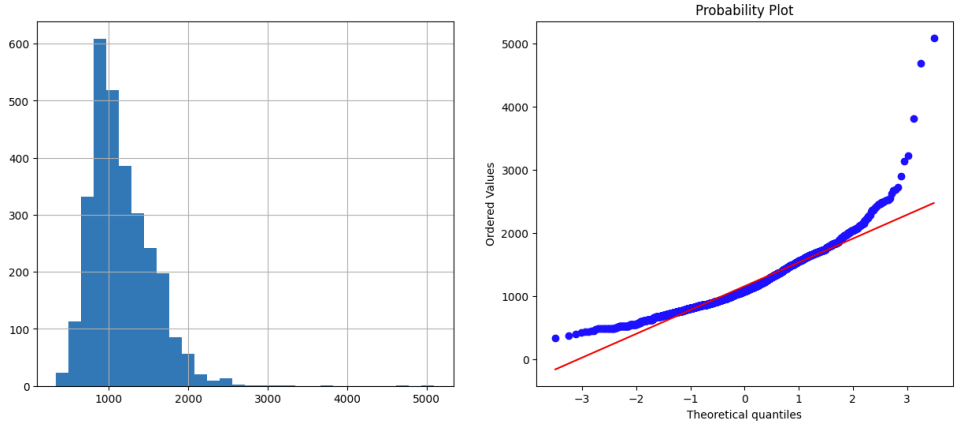
Нормализация числовых признаков

```
[20] 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()
```

```
[21] data.hist(figsize=(20,20))
     plt.show()
```

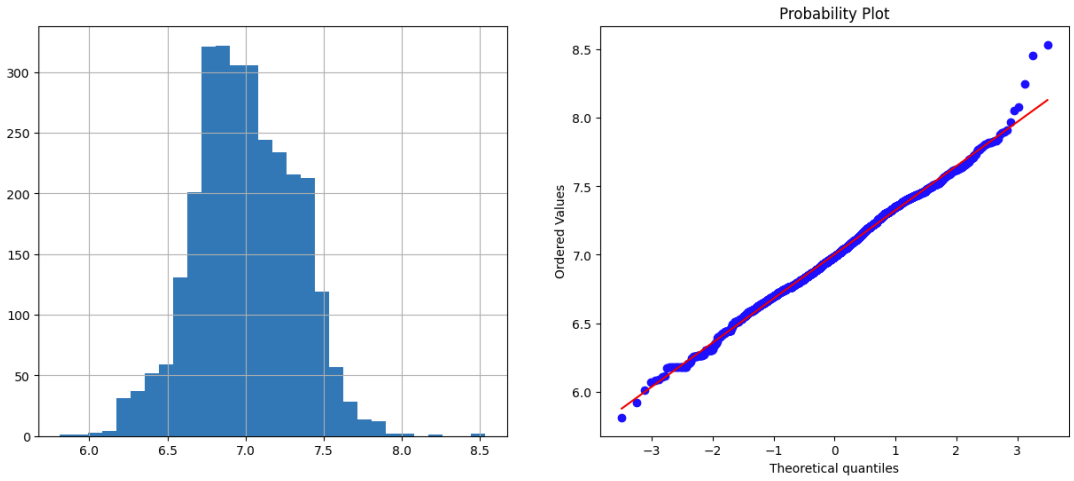



```
[22] diagnostic_plots(data, '1stFlrSF')
```



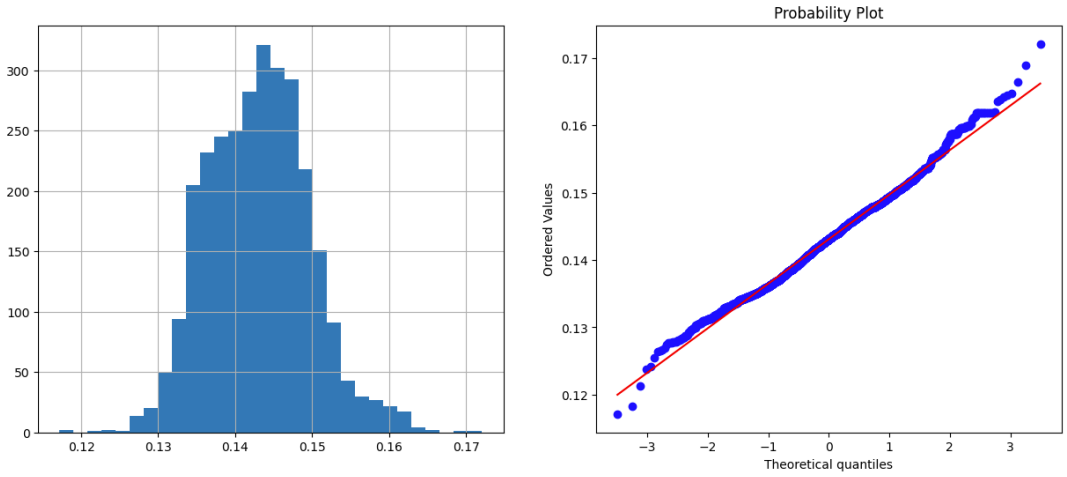
Произведем логарифмическое преобразование:

```
[23] data['1stFlrSF'] = np.log(data['1stFlrSF'])
     diagnostic_plots(data, '1stFlrSF')
```



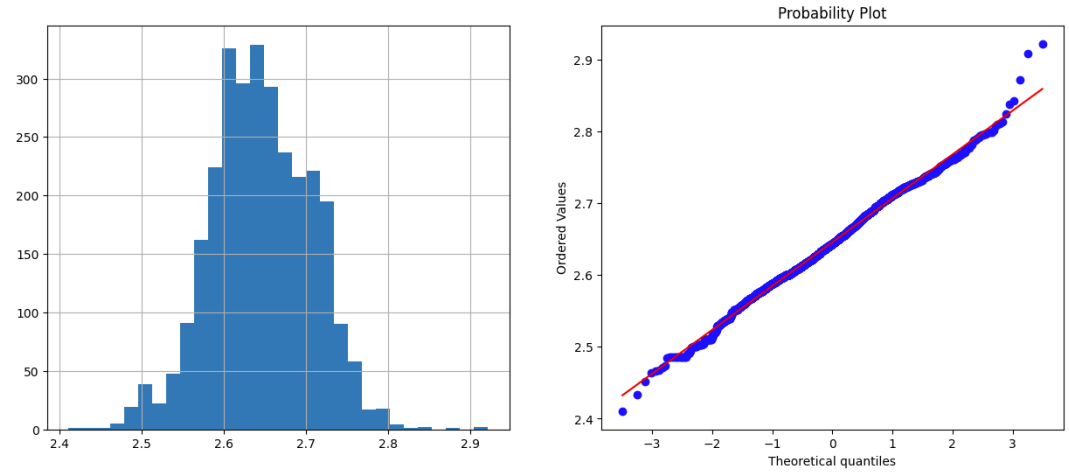
Обратное преобразование:

```
[24] data['1stFlrSF_reciprocal'] = 1 / (data['1stFlrSF'])
     diagnostic_plots(data, '1stFlrSF_reciprocal')
```



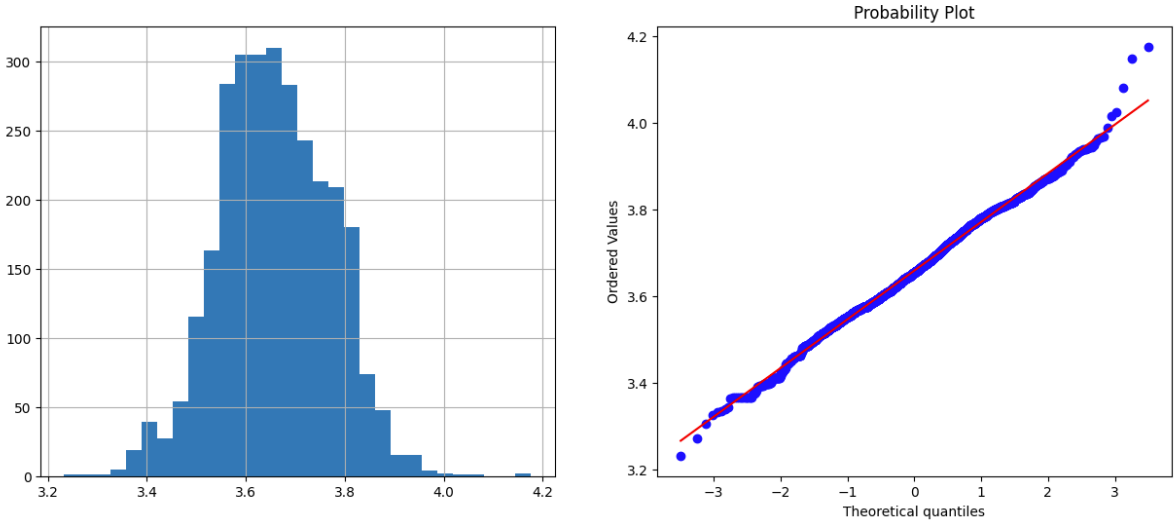
Извлечение квадратного корня:

```
[25] data['1stFlrSF_sqr'] = data['1stFlrSF']**(1/2)
     diagnostic_plots(data, '1stFlrSF_sqr')
```

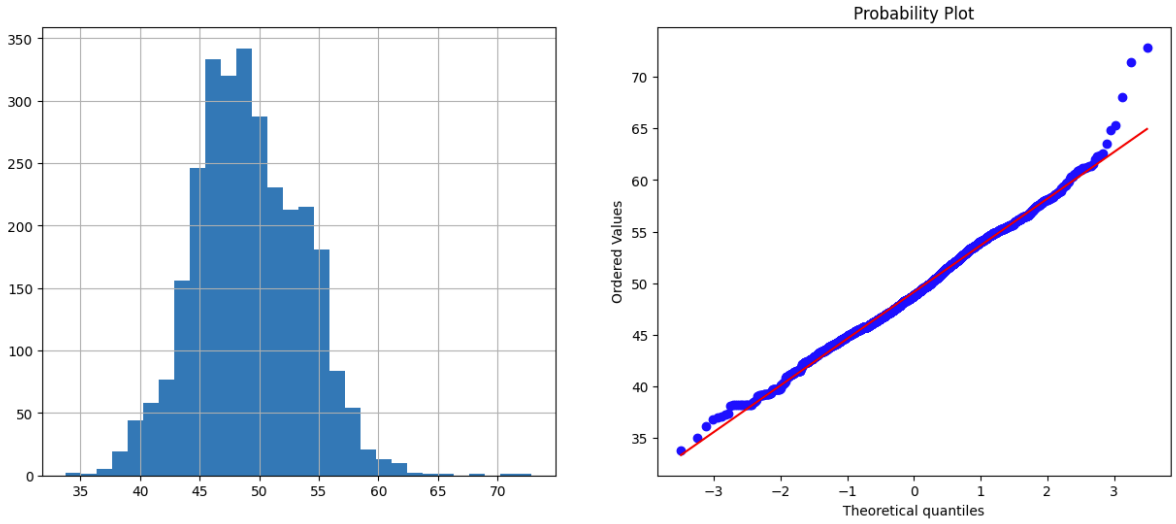


Возведение в степень:

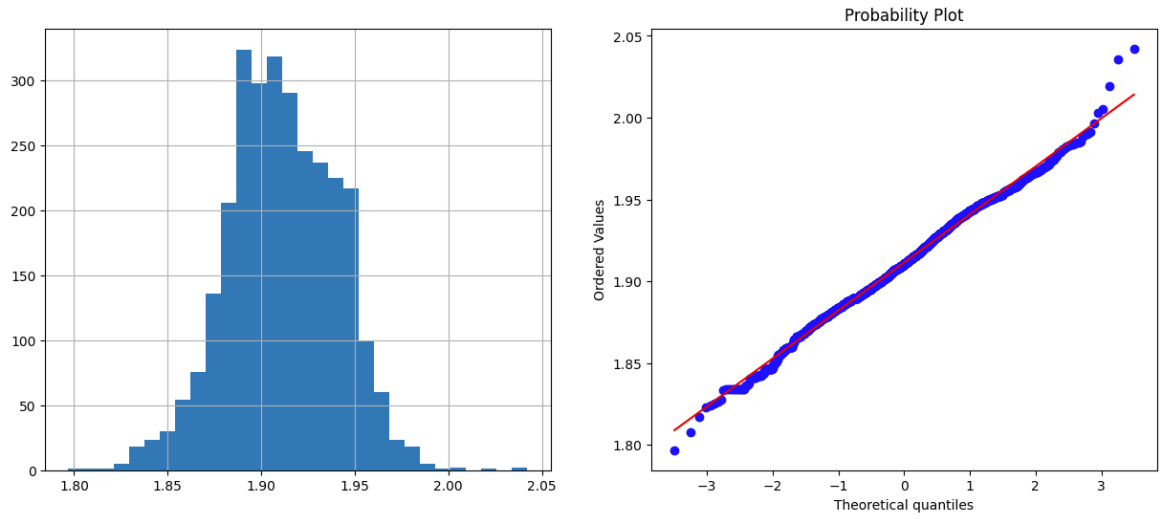
```
[26] data['1stFlrSF_exp1'] = data['1stFlrSF']**(1/1.5)
     diagnostic_plots(data, '1stFlrSF_exp1')
```




```
[27] data['1stFlrSF_exp2'] = data['1stFlrSF']**(2)
     diagnostic_plots(data, '1stFlrSF_exp2')
```



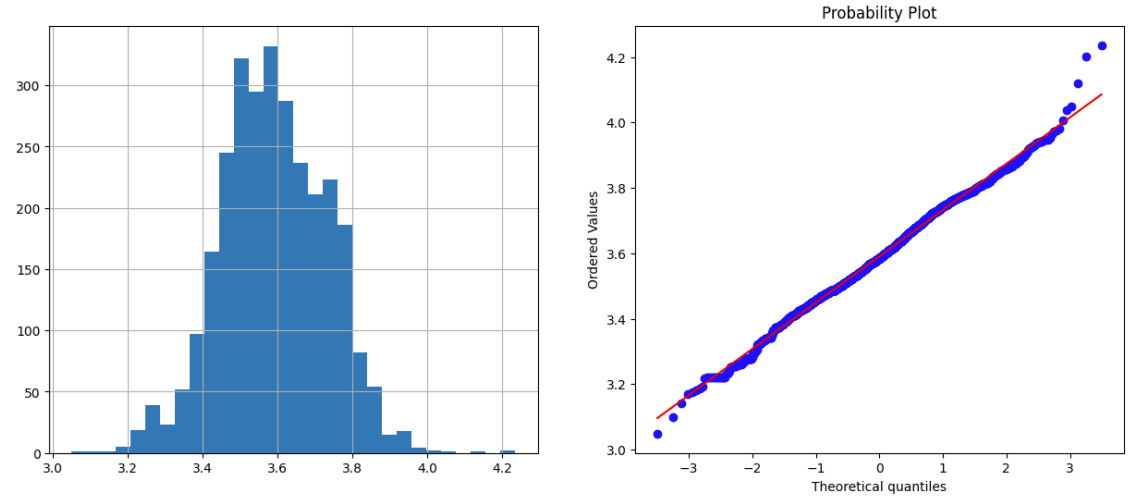

```
[28] data['1stFlrSF_exp3'] = data['1stFlrSF']**(0.333)
     diagnostic_plots(data, '1stFlrSF_exp3')
```



Произведем преобразование Бокса-Кокса:

```
[29] data['1stFlrSF_boxcox'], param = stats.boxcox(data['1stFlrSF'])
     print('Оптимальное значение λ = {}'.format(param))
     diagnostic_plots(data, '1stFlrSF_boxcox')
```

```
Оптимальное значение λ = 0.5764205521187881
```



Вывод: в рамках данной части лабораторной работы были решены следующие задачи:

- устранение пропусков в данных;
- кодирование категориальных признаков;
- нормализация числовых признаков.