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

Отчет
Лабораторная работа № 2
По курсу «Методы машинного обучения»

Обработка признаков часть 1

ИСПОЛНИТЕЛЬ:
Попов Илья Андреевич
Группа ИУ5-23М

_____

"__"_____2022 г.

ПРЕПОДАВАТЕЛЬ:
Гапанюк Ю.Е.

_____

"__"_____2022 г.

Москва 2022

Задание:

1. Выбрать набор данных (датасет), содержащий категориальные и числовые признаки и пропуски в данных. Для выполнения следующих пунктов можно использовать несколько различных наборов данных (один для обработки пропусков, другой для категориальных признаков и т.д.) Просьба не использовать датасет, на котором данная задача решалась в лекции.
2. Для выбранного датасета (датасетов) на основе материалов лекций решить следующие задачи:
    i. устранение пропусков в данных;
    ii. кодирование категориальных признаков;
    iii. нормализацию числовых признаков.

Выполнение

Popov I.A. IU5-23M lab2

In [52]:
```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.preprocessing import OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
import matplotlib.pyplot as plt
import scipy.stats as stats
```

In [53]:
```
raw_data = pd.read_csv('weatherAUS.csv', sep=',')
```

In [54]:
```
raw_data.head()
```

Out[54]:

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	...	Humidity9am	Humi
0	2008-12-01	Albury	13.4	22.9	0.6	NaN	NaN	W	44.0	W	...	71.0	
1	2008-12-02	Albury	7.4	25.1	0.0	NaN	NaN	WNW	44.0	NNW	...	44.0	
2	2008-12-03	Albury	12.9	25.7	0.0	NaN	NaN	WSW	46.0	W	...	38.0	
3	2008-12-04	Albury	9.2	28.0	0.0	NaN	NaN	NE	24.0	SE	...	45.0	
4	2008-12-05	Albury	17.5	32.3	1.0	NaN	NaN	W	41.0	ENE	...	82.0	

5 rows × 23 columns

In [55]:
```
raw_data.dtypes
```

Out[55]:
```
Date              object
Location          object
MinTemp          float64
MaxTemp          float64
Rainfall         float64
Evaporation      float64
Sunshine         float64
WindGustDir       object
WindGustSpeed    float64
WindDir9am        object
WindDir3pm        object
WindSpeed9am     float64
WindSpeed3pm     float64
Humidity9am      float64
Humidity3pm      float64
Pressure9am      float64
Pressure3pm      float64
Cloud9am         float64
Cloud3pm         float64
Temp9am          float64
Temp3pm          float64
RainToday         object
RainTomorrow      object
dtype: object
```

In [56]:
```
raw_data_with_na = [c for c in raw_data.columns if raw_data[c].isnull().sum() > 0]
raw_data_with_na
```

Out[56]:
```
['MinTemp',
 'MaxTemp',
 'Rainfall',
 'Evaporation',
 'Sunshine',
 'WindGustDir',
 'WindGustSpeed',
 'WindDir9am',
 'WindDir3pm',
 'WindSpeed9am',
 'WindSpeed3pm',
 'Humidity9am',
 'Humidity3pm',
 'Pressure9am',
```

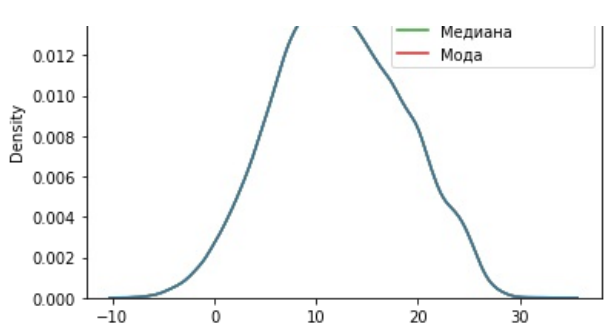
```
'Pressure3pm',
'Cloud9am',
'Cloud3pm',
'Temp9am',
'Temp3pm',
'RainToday',
'RainTomorrow']
```

In [57]:
```
[(c, raw_data[c].isnull().sum()) for c in raw_data_with_na]
```

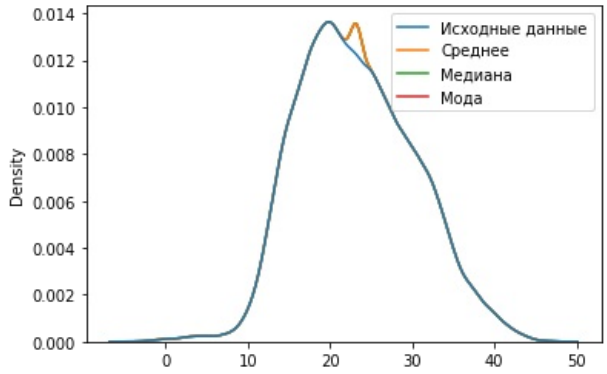
Out[57]:
```
[('MinTemp', 1485),
 ('MaxTemp', 1261),
 ('Rainfall', 3261),
 ('Evaporation', 62790),
 ('Sunshine', 69835),
 ('WindGustDir', 10326),
 ('WindGustSpeed', 10263),
 ('WindDir9am', 10566),
 ('WindDir3pm', 4228),
 ('WindSpeed9am', 1767),
 ('WindSpeed3pm', 3062),
 ('Humidity9am', 2654),
 ('Humidity3pm', 4507),
 ('Pressure9am', 15065),
 ('Pressure3pm', 15028),
 ('Cloud9am', 55888),
 ('Cloud3pm', 59358),
 ('Temp9am', 1767),
 ('Temp3pm', 3609),
 ('RainToday', 3261),
 ('RainTomorrow', 3267)]
```

In [58]:
```
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]
    new_data = dataset
    new_data[column] = all_data.reshape((size,))

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

def research_impute_numeric_column(dataset, num_column, const_value=None):
    strategy_params = ['mean', 'median', 'most_frequent', 'constant']
    strategy_params_names = ['Среднее', 'Медиана', 'Мода']
    strategy_params_names.append('Константа = ' + str(const_value))

    original_temp_data = dataset[[num_column]].values
    size = original_temp_data.shape[0]
    original_data = original_temp_data.reshape((size,))

    new_df = pd.DataFrame({'Исходные данные':original_data})

    for i in range(len(strategy_params)):
        strategy = strategy_params[i]
        col_name = strategy_params_names[i]
        if (strategy!='constant') or (strategy == 'constant' and const_value!=None):
            if strategy == 'constant':
                temp_data, _, _, _ = impute_column(dataset, num_column, strategy, fill_value_param=const_value)
            else:
                temp_data, _, _, _ = impute_column(dataset, num_column, strategy)
            new_df[col_name] = temp_data

    sns.kdeplot(data=new_df)
```

In [59]:
```
#Устранение пропусков
research_impute_numeric_column(raw_data, 'MinTemp')
```



In [60]:
```
research_impute_numeric_column(raw_data, 'MaxTemp')
```



In [67]:
```
#Для устранения пропусков в столбце MinTemp можно вставить среднее среди выборки
_,_,_,new_data = impute_column(raw_data, 'MinTemp', 'mean')
new_data
```

Out[67]:

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	...	Humidity9am
0	2008-12-01	Albury	13.4	22.900000	0.6	NaN	NaN	W	44.0	W	...	71.0
1	2008-12-02	Albury	7.4	25.100000	0.0	NaN	NaN	WNW	44.0	NNW	...	44.0
2	2008-12-03	Albury	12.9	25.700000	0.0	NaN	NaN	WSW	46.0	W	...	38.0
3	2008-12-04	Albury	9.2	28.000000	0.0	NaN	NaN	NE	24.0	SE	...	45.0
4	2008-12-05	Albury	17.5	32.300000	1.0	NaN	NaN	W	41.0	ENE	...	82.0
...	...	...	...	...	...	...	...	...	...	...	...	...
145455	2017-06-21	Uluru	2.8	23.400000	0.0	NaN	NaN	E	31.0	SE	...	51.0
145456	2017-06-22	Uluru	3.6	25.300000	0.0	NaN	NaN	NNW	22.0	SE	...	56.0
145457	2017-06-23	Uluru	5.4	26.900000	0.0	NaN	NaN	N	37.0	SE	...	53.0
145458	2017-06-24	Uluru	7.8	27.000000	0.0	NaN	NaN	SE	28.0	SSE	...	51.0
145459	2017-06-25	Uluru	14.9	23.221348	0.0	NaN	NaN	NaN	NaN	ESE	...	62.0

145460 rows × 23 columns

In [68]:
```
#В столбце MaxTemp лучше подойдёт замена на моду
_,_,_,new_data= impute_column(new_data, 'MaxTemp', 'most_frequent')
new_data
```

Out[68]:

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	...	Humidity9am
0	2008-12-01	Albury	13.4	22.900000	0.6	NaN	NaN	W	44.0	W	...	71.0
1	2008-12-02	Albury	7.4	25.100000	0.0	NaN	NaN	WNW	44.0	NNW	...	44.0
2	2008-12-03	Albury	12.9	25.700000	0.0	NaN	NaN	WSW	46.0	W	...	38.0
3	2008-12-04	Albury	9.2	28.000000	0.0	NaN	NaN	NE	24.0	SE	...	45.0

4	2008-12-05	Albury	17.5	32.300000	1.0	NaN	NaN	W	41.0	ENE	...	82.0
...	...	...	...	...	...	...	...	...	...	...	...	...
145455	2017-06-21	Uluru	2.8	23.400000	0.0	NaN	NaN	E	31.0	SE	...	51.0
145456	2017-06-22	Uluru	3.6	25.300000	0.0	NaN	NaN	NNW	22.0	SE	...	56.0
145457	2017-06-23	Uluru	5.4	26.900000	0.0	NaN	NaN	N	37.0	SE	...	53.0
145458	2017-06-24	Uluru	7.8	27.000000	0.0	NaN	NaN	SE	28.0	SSE	...	51.0
145459	2017-06-25	Uluru	14.9	23.221348	0.0	NaN	NaN	NaN	NaN	ESE	...	62.0

145460 rows × 23 columns

In [69]:
```
#Устранение пропуска в категориальном признаке
_,_,_,new_data =  impute_column(new_data, 'WindDir9am', 'most_frequent')
#new_data['WindDir9am'] = wind_dir9am_new
new_data
```

Out[69]:

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	...	Humidity9am
0	2008-12-01	Albury	13.4	22.900000	0.6	NaN	NaN	W	44.0	W	...	71.0
1	2008-12-02	Albury	7.4	25.100000	0.0	NaN	NaN	WNW	44.0	NNW	...	44.0
2	2008-12-03	Albury	12.9	25.700000	0.0	NaN	NaN	WSW	46.0	W	...	38.0
3	2008-12-04	Albury	9.2	28.000000	0.0	NaN	NaN	NE	24.0	SE	...	45.0
4	2008-12-05	Albury	17.5	32.300000	1.0	NaN	NaN	W	41.0	ENE	...	82.0
...	...	...	...	...	...	...	...	...	...	...	...	...
145455	2017-06-21	Uluru	2.8	23.400000	0.0	NaN	NaN	E	31.0	SE	...	51.0
145456	2017-06-22	Uluru	3.6	25.300000	0.0	NaN	NaN	NNW	22.0	SE	...	56.0
145457	2017-06-23	Uluru	5.4	26.900000	0.0	NaN	NaN	N	37.0	SE	...	53.0
145458	2017-06-24	Uluru	7.8	27.000000	0.0	NaN	NaN	SE	28.0	SSE	...	51.0
145459	2017-06-25	Uluru	14.9	23.221348	0.0	NaN	NaN	NaN	NaN	ESE	...	62.0

145460 rows × 23 columns

In [70]:
```
#Проверка
[(c, raw_data[c].isnull().sum()) for c in raw_data_with_na]
```

Out[70]:
```
[('MinTemp', 0),
 ('MaxTemp', 0),
 ('Rainfall', 3261),
 ('Evaporation', 62790),
 ('Sunshine', 69835),
 ('WindGustDir', 10326),
 ('WindGustSpeed', 10263),
 ('WindDir9am', 0),
 ('WindDir3pm', 4228),
 ('WindSpeed9am', 1767),
 ('WindSpeed3pm', 3062),
 ('Humidity9am', 2654),
 ('Humidity3pm', 4507),
 ('Pressure9am', 15065),
 ('Pressure3pm', 15028),
 ('Cloud9am', 55888),
 ('Cloud3pm', 59358),
 ('Temp9am', 1767),
 ('Temp3pm', 3609),
 ('RainToday', 3261),
 ('RainTomorrow', 3267)]
```

In [32]:
```
#Кодирование категориальных признаков
ohe = OneHotEncoder()
cat_enc_ohe = ohe.fit_transform(new_data[['WindDir9am']])
cat_enc_ohe
```

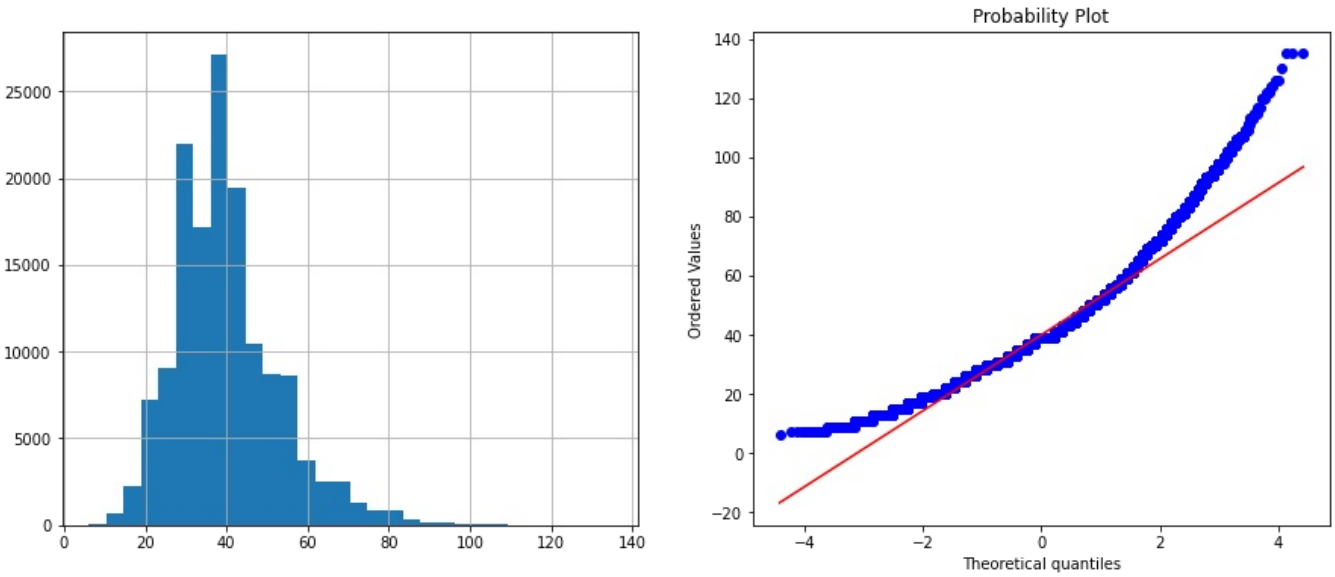
Out[32]:
```
<145460x16 sparse matrix of type '<class 'numpy.float64'>'
        with 145460 stored elements in Compressed Sparse Row format>
```

In [33]:
```
cat_enc_ohe.todense()[0:10]
```

Out[33]:
```
matrix([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0.],
        [0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0.],
        [0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0.]])
```

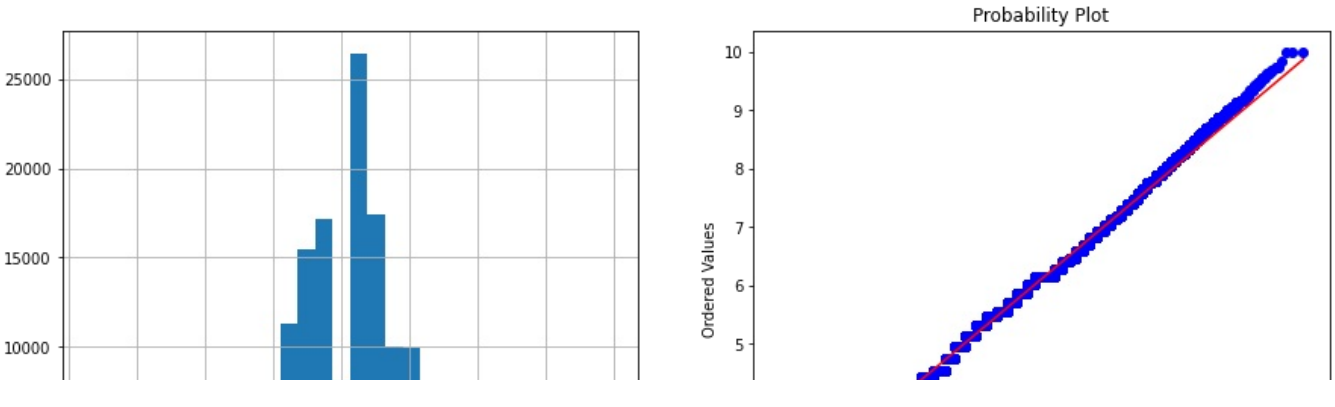
In [34]:
```
#Нормализация числовых признаков
def diagnostic_plots(df, variable):
    plt.figure(figsize=(15,6))
    # гистограмма
    plt.subplot(1, 2, 1)
    df[variable].hist(bins=30)
    ## Q-Q plot
    plt.subplot(1, 2, 2)
    stats.probplot(df[variable], dist="norm", plot=plt)
    plt.show()
```

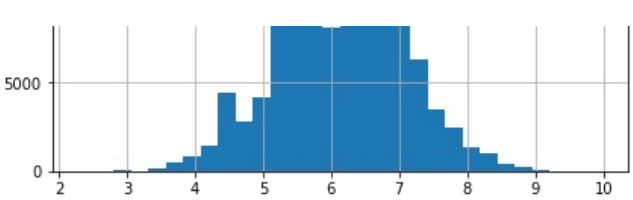
In [83]:
```
_,_,_, new_data_w = impute_column(new_data, 'WindGustSpeed', 'median')
diagnostic_plots(new_data_w, 'WindGustSpeed')
```

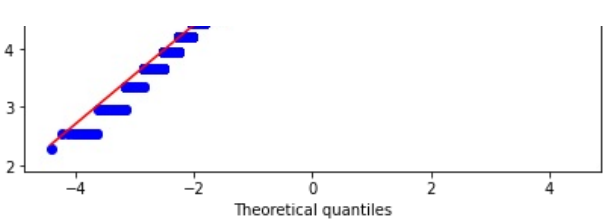


In [84]:
```
new_data['WindGustSpeed_boxcox'], param = stats.boxcox(new_data['WindGustSpeed'])
print('Оптимальное значение λ = {}'.format(param))
diagnostic_plots(new_data, 'WindGustSpeed_boxcox')
```

```
Оптимальное значение λ = 0.26194865714069987
```







In [ ]: