

Кольца Илья Вячеславович

Лабораторная работа № 2

Вариант 2

1. Загрузка данных и импорт библиотек

In [1]:

```
import pandas as pd
import warnings
import numpy as np

warnings.filterwarnings('ignore')

data = pd.read_excel('data.xlsx', sheet_name='RC')
data.head()
```

Out[1]:

	data	NGN	RUB	Brent
0	2008-12-31	0.007593	0.035522	43.405
1	2008-11-30	0.008502	0.036585	54.625
2	2008-10-31	0.008727	0.037941	71.490
3	2008-09-30	0.008493	0.039573	102.055
4	2008-08-31	0.008649	0.041402	114.720

NGN - цена нигерийской найры, в долларах США\ **RUB** - цена российского рубля, в долларах США\ **Brent** - цена барреля нефти марки **Brent**, в долларах США\ Данные представлены на период с января **2004** по декабрь **2008** года (**60** месяцев)

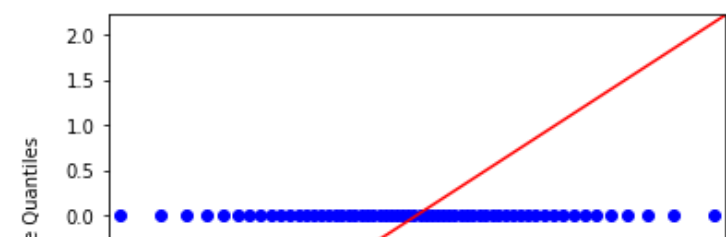
2. Построение pp и qq графиков

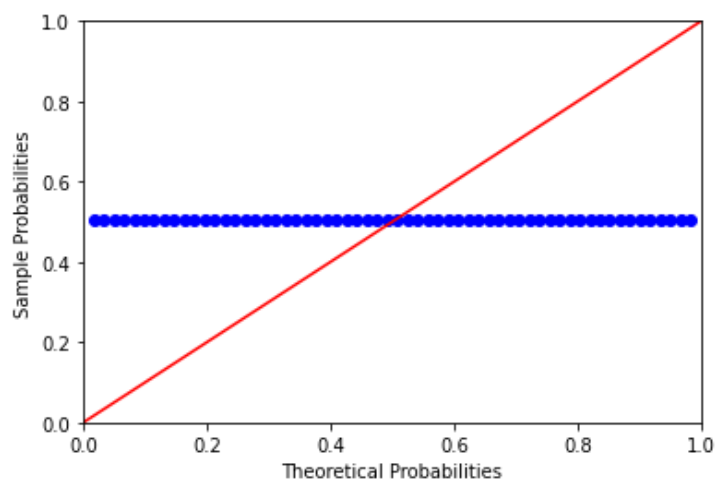
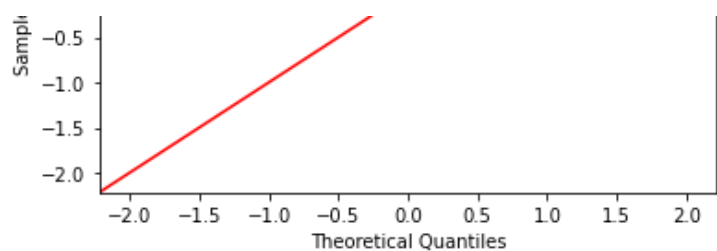
In [2]:

```
import statsmodels.api as sm
import matplotlib.pyplot as plt
%matplotlib inline

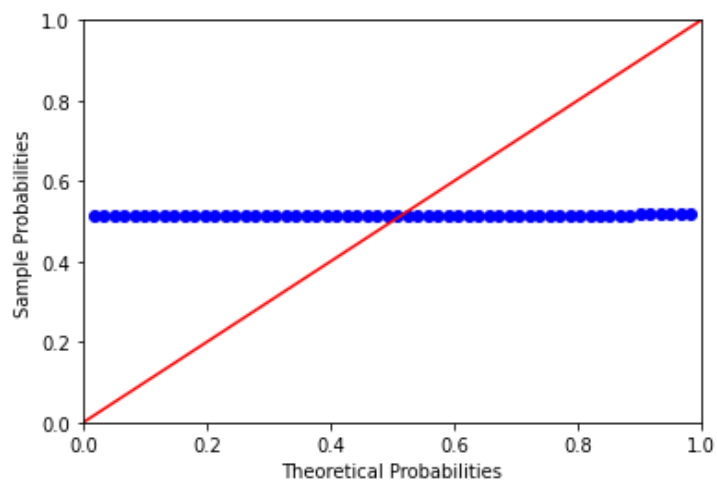
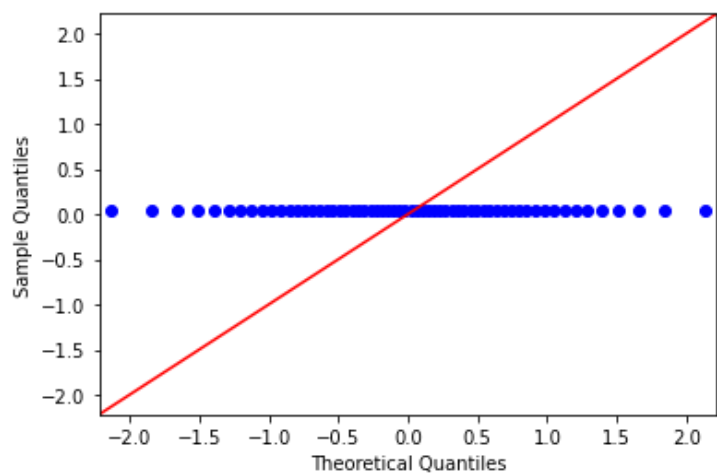
for val in ['NGN', 'RUB', 'Brent']:
    print(f"Графики qq и pp для {val}")
    pp = sm.ProbPlot(data[val])
    fig1 = pp.qqplot(line='45')
    fig2 = pp.ppplot(line='45')
    plt.show()
```

Графики qq и pp для NGN

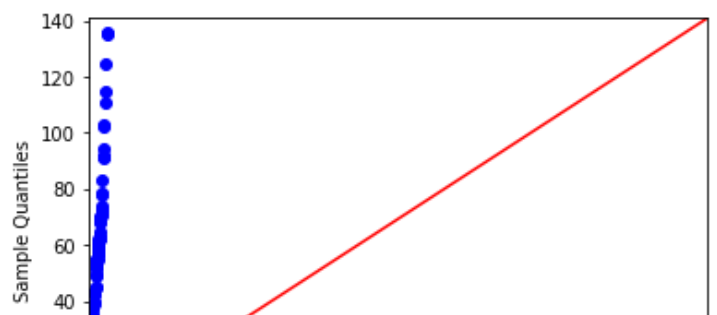


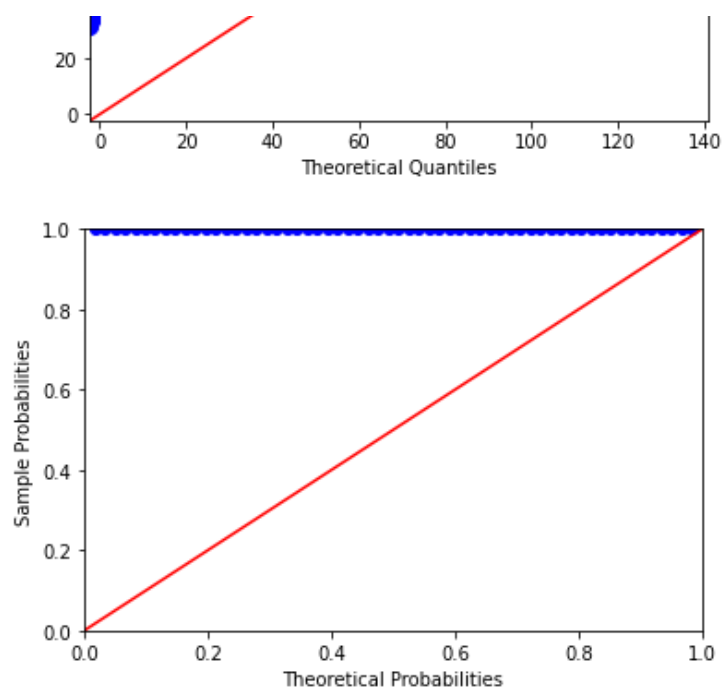


Графики qq и pp для RUB



Графики qq и pp для Brent





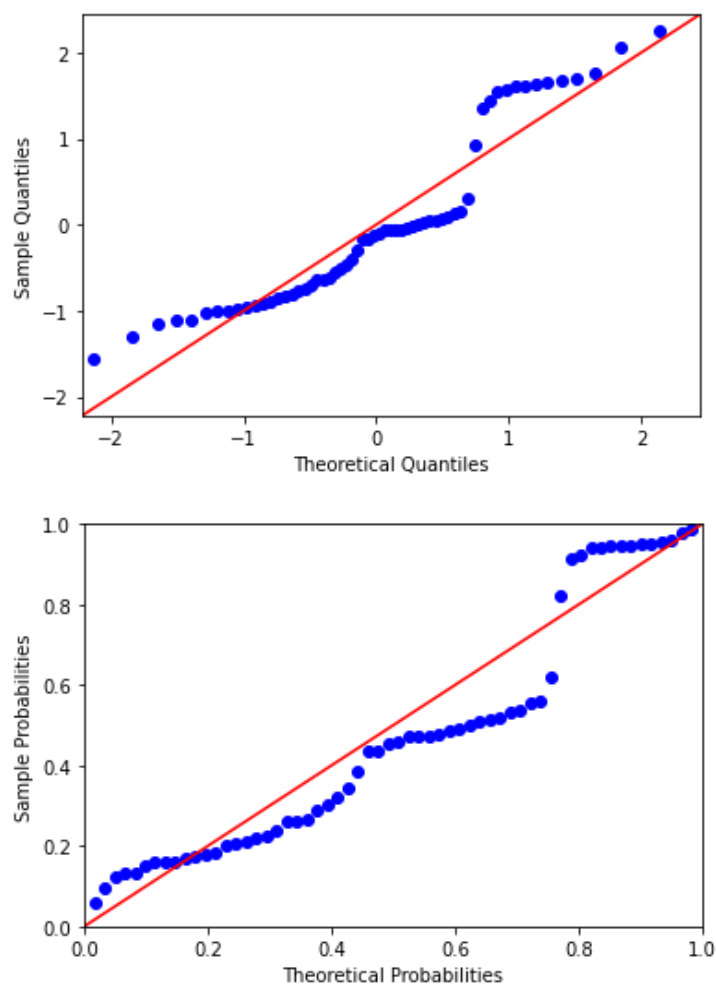
In [3]:

```
from sklearn.preprocessing import StandardScaler, normalize

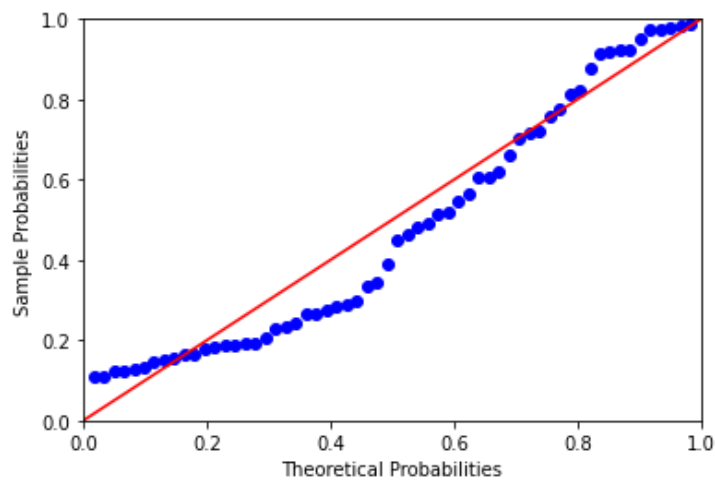
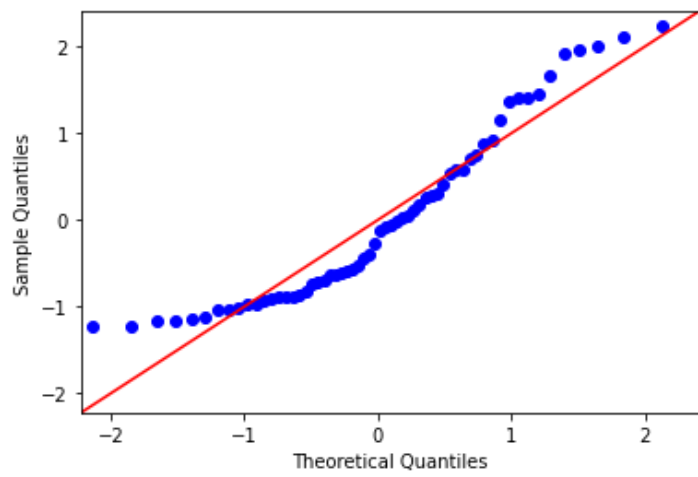
scaler = StandardScaler()
scaled = scaler.fit_transform(data.iloc[:, 1:].values)

for idx, val in enumerate(['NGN', 'RUB', 'Brent']):
    print(f"Графики qq и pp для стандартизованных {val}")
    pp = sm.ProbPlot(scaled[:, idx])
    fig1 = pp.qqplot(line='45')
    fig2 = pp.ppplot(line='45')
    plt.show()
```

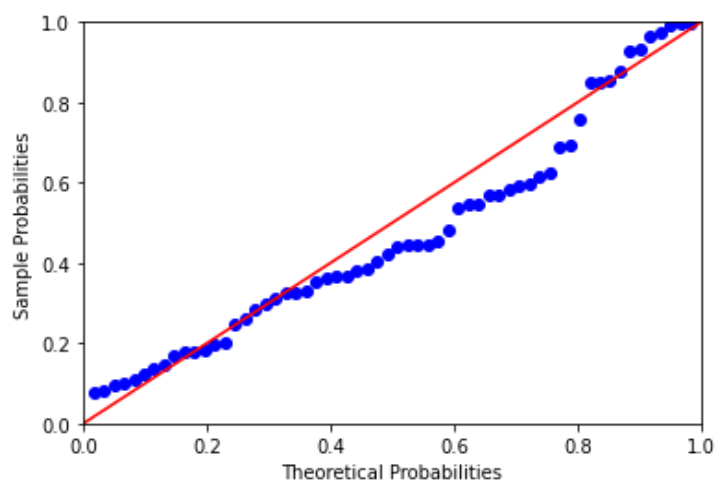
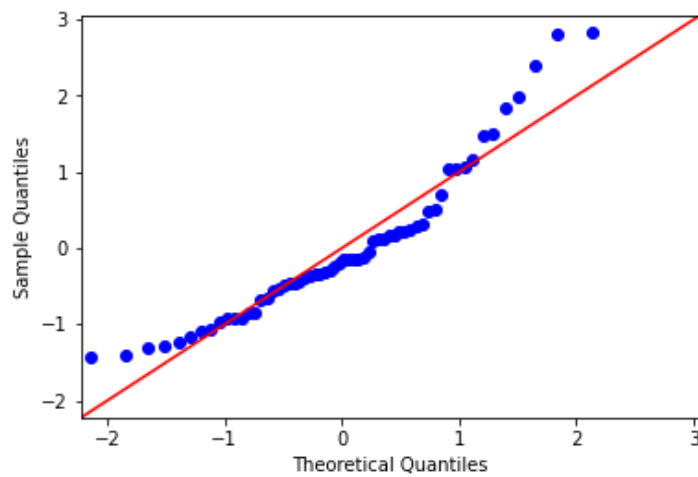
Графики qq и pp для стандартизованных NGN



Графики qq и pp для стандартизованных RUB



Графики qq и pp для стандартизованных Brent

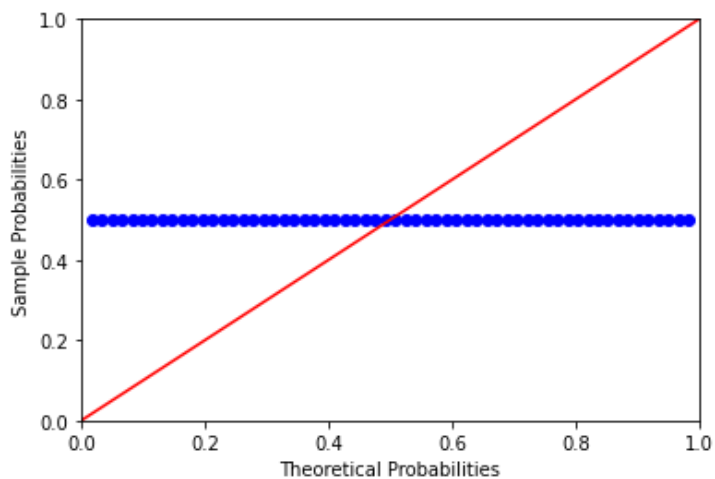
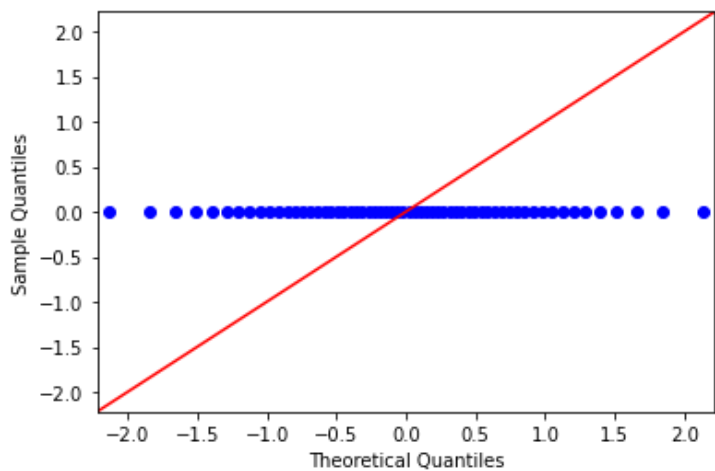


In [4]:

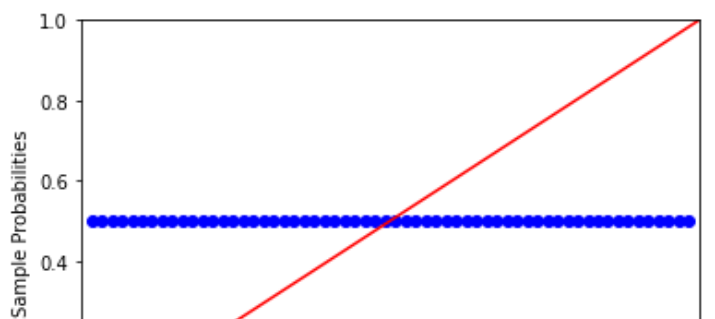
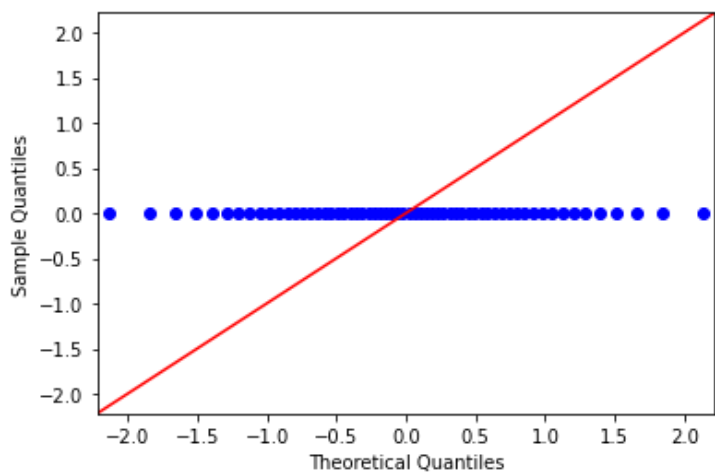
```
normalized = normalize(data.iloc[:, 1:].values)
```

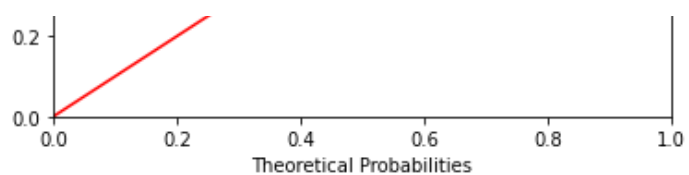
```
for idx, val in enumerate(['NGN', 'RUB', 'Brent']):  
    print(f"Графики qq и pp для нормализованных {val}")  
    pp = sm.ProbPlot(normalized[:, idx])  
    fig1 = pp.qqplot(line='45')  
    fig2 = pp.ppplot(line='45')  
    plt.show()
```

Графики qq и pp для нормализованных NGN

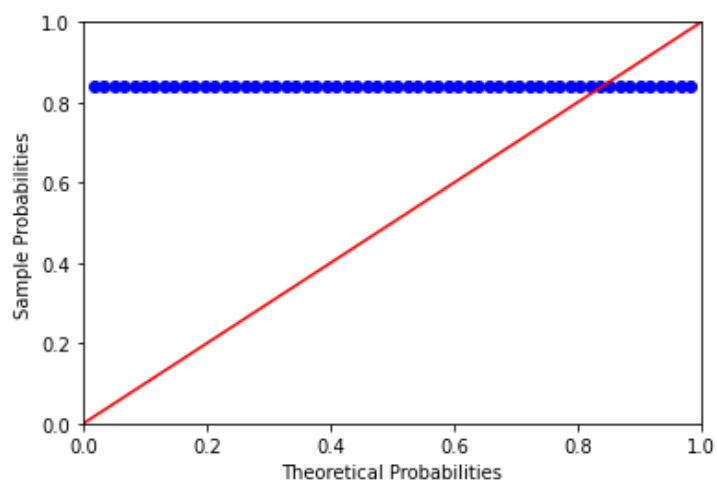
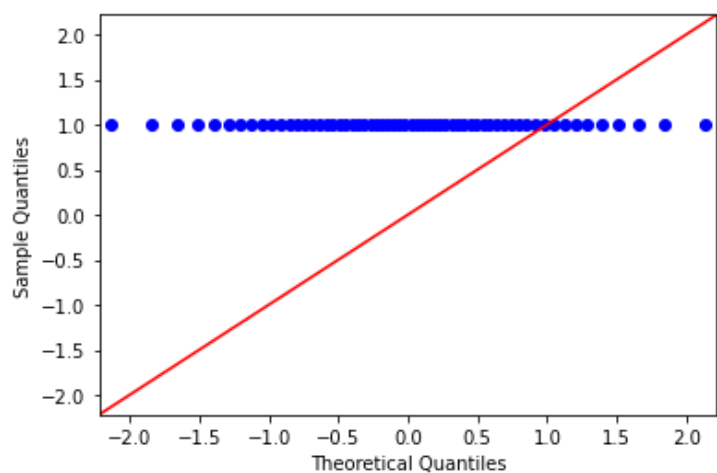


Графики qq и pp для нормализованных RUB



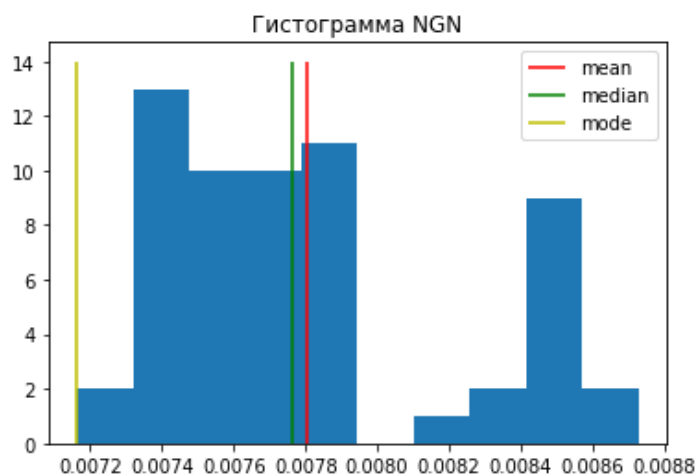


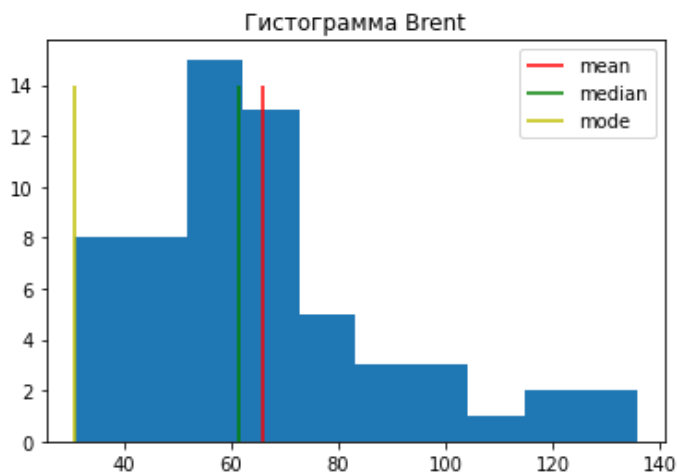
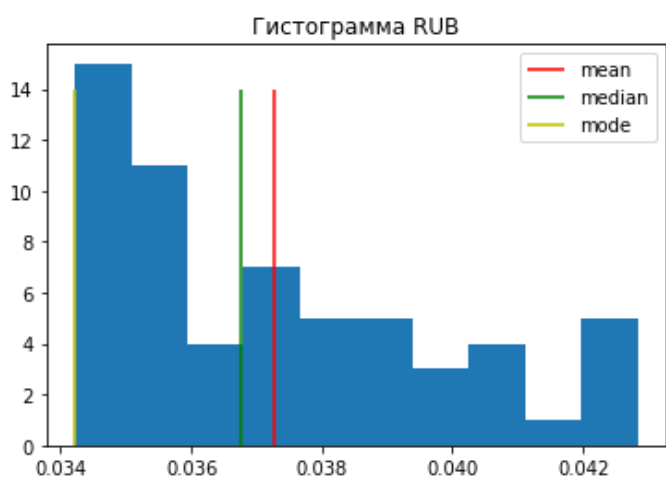
Графики qq и pp для нормализованных Brent



In [5]:

```
for val in ['NGN', "RUB", "Brent"]:
    mean = data[val].mean()
    med = data[val].median()
    mode = data[val].mode()[0]
    plt.title(f"Гистограмма {val}")
    plt.hist(data[val])
    plt.vlines(mean, 0, 14, colors='r', label='mean')
    plt.vlines(med, 0, 14, colors='g', label='median')
    plt.vlines(mode, 0, 14, colors='y', label='mode')
    plt.legend()
    plt.show()
```





In [6]:

```
import scipy.stats as sts

print('Проверка гипотезы о равенстве медиан для NGN и RUB', sts.median_test(data['NGN'],
data['RUB'])[1])
print('Проверка гипотезы о равенстве медиан для NGN и Brent', sts.median_test(data['NGN'],
data['Brent'])[1])
print('Проверка гипотезы о равенстве медиан для Brent и RUB', sts.median_test(data['Brent'],
data['RUB'])[1])
```

Проверка гипотезы о равенстве медиан для NGN и RUB 4.67374541395913e-27
 Проверка гипотезы о равенстве медиан для NGN и Brent 4.67374541395913e-27
 Проверка гипотезы о равенстве медиан для Brent и RUB 4.67374541395913e-27

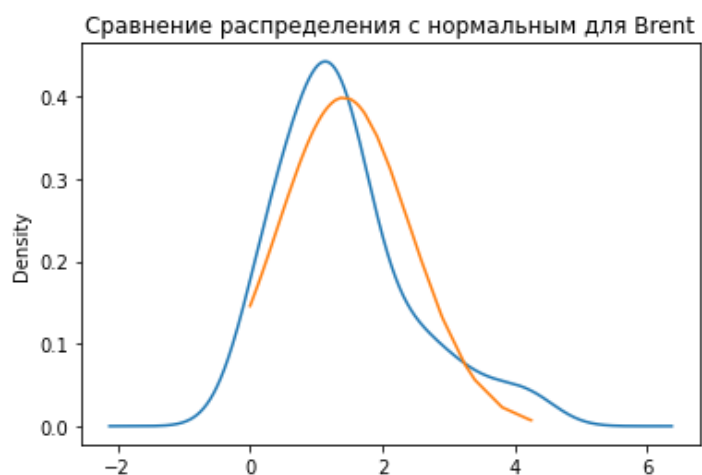
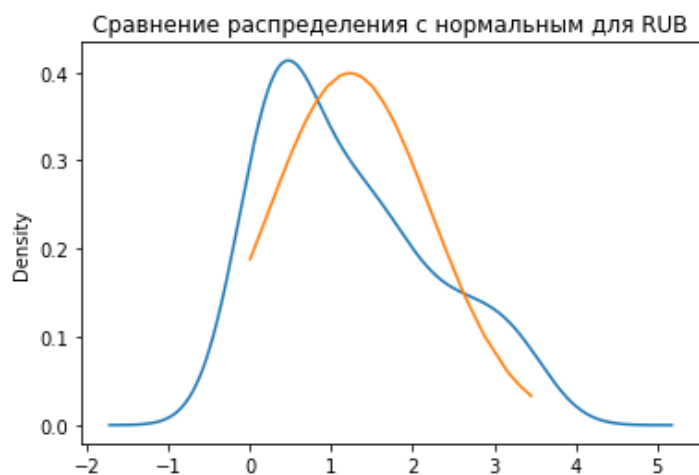
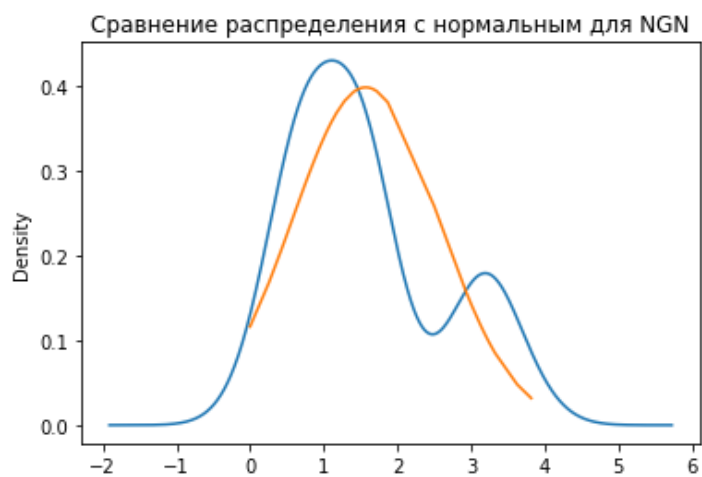
In [7]:

```
print('Проверка гипотезы о равенстве средних для NGN и RUB', sts.ttest_ind(data['NGN'], d
ata['RUB'])[1])
print('Проверка гипотезы о равенстве средних для NGN и Brent', sts.ttest_ind(data['NGN'],
data['Brent'])[1])
print('Проверка гипотезы о равенстве средних для Brent и RUB', sts.ttest_ind(data['Brent'
], data['RUB'])[1])
```

Проверка гипотезы о равенстве средних для NGN и RUB 2.172258671920374e-110
 Проверка гипотезы о равенстве средних для NGN и Brent 1.0302418634894988e-40
 Проверка гипотезы о равенстве средних для Brent и RUB 1.07374126801427e-40

In [8]:

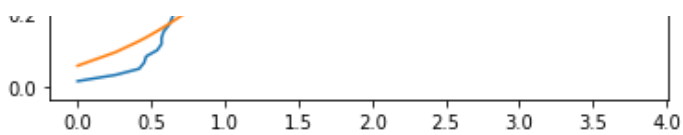
```
scaled_df = pd.DataFrame(scaled, columns=['NGN', "RUB", "Brent"])
for val in ['NGN', "RUB", "Brent"]:
    x = scaled_df[val]
    plt.title(f'Сравнение распределения с нормальным для {val}')
    (x - x.min()).plot.kde()
    x = sorted(x - x.min())
    plt.plot(x, sts.norm.pdf(x, np.mean(x), 1))
    plt.show()
```



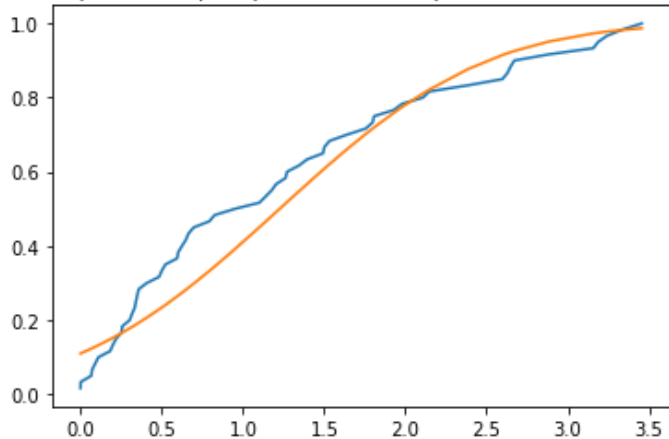
In [9]:

```
for val in ['NGN', "RUB", "Brent"]:
    x = scaled_df[val]
    plt.title(f'Сравнение распределения с нормальным для {val}')
    x = np.sort(x - x.min())
    y = np.arange(1, len(x) + 1) / len(x)
    plt.plot(x, y)
    plt.plot(x, sts.norm.cdf(x, np.mean(x), 1))
    plt.show()
```

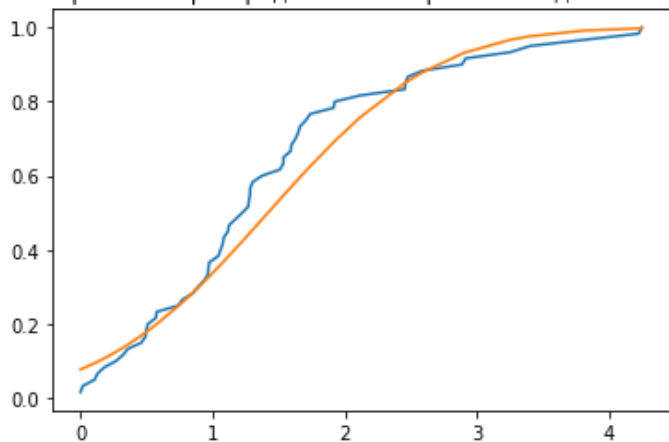




Сравнение распределения с нормальным для RUB



Сравнение распределения с нормальным для Brent



In [10]:

```
for val in ['NGN', "RUB", "Brent"]:
    print(f"Проверка гипотезы на нормальность распределения тестом Шапиро-Уилка для {val}
:")
    z, p = sts.shapiro(scaled_df[val])
    print(f"Статистика критерия {z}, p_value {p}")
    print()
```

Проверка гипотезы на нормальность распределения тестом Шапиро-Уилка для NGN:
Статистика критерия 0.8878504037857056, p_value 4.882759822066873e-05

Проверка гипотезы на нормальность распределения тестом Шапиро-Уилка для RUB:
Статистика критерия 0.9053937792778015, p_value 0.00020803295774385333

Проверка гипотезы на нормальность распределения тестом Шапиро-Уилка для Brent:
Статистика критерия 0.9147350192070007, p_value 0.0004738709540106356

In [11]:

```
quant = scaled_df.quantile((0.05, 0.95))
f_scaled_df = scaled_df.apply(lambda x: x[(x > quant.loc[0.05, x.name]) & (x < quant.loc
[0.95, x.name])], axis=0)
f_scaled_df.dropna(inplace=True)
```

In [12]:

```
for val in ['NGN', "RUB", "Brent"]:
    print(f"Проверка гипотезы на нормальность распределения тестом Шапиро-Уилка для {val}
с удалением выбросов:")
    z, p = sts.shapiro(scaled_df[val])
    print(f"Статистика критерия {z}, p_value {p}")
```

```
print()
```

Проверка гипотезы на нормальность распределения тестом Шапиро-Уилка для NGN с удалением выбросов:

Статистика критерия 0.8878504037857056, p_value 4.882759822066873e-05

Проверка гипотезы на нормальность распределения тестом Шапиро-Уилка для RUB с удалением выбросов:

Статистика критерия 0.9053937792778015, p_value 0.00020803295774385333

Проверка гипотезы на нормальность распределения тестом Шапиро-Уилка для Brent с удалением выбросов:

Статистика критерия 0.9147350192070007, p_value 0.0004738709540106356

In [13]:

```
print('Диаграмма рассеяния для NGN и RUB')
scaled_df.plot.scatter(x='NGN', y='RUB')
plt.show()
print('Диаграмма рассеяния для NGN и Brent')
scaled_df.plot.scatter(x='NGN', y='Brent')
plt.show()
print('Диаграмма рассеяния для Brent и RUB')
scaled_df.plot.scatter(x='Brent', y='RUB')
plt.show()
```

Диаграмма рассеяния для NGN и RUB

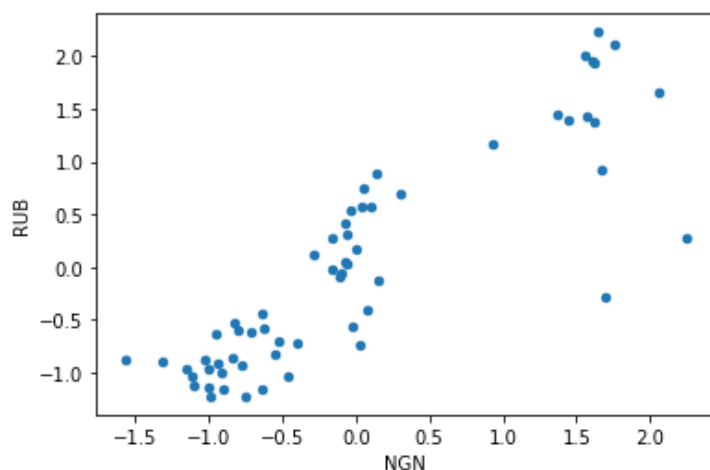


Диаграмма рассеяния для NGN и Brent

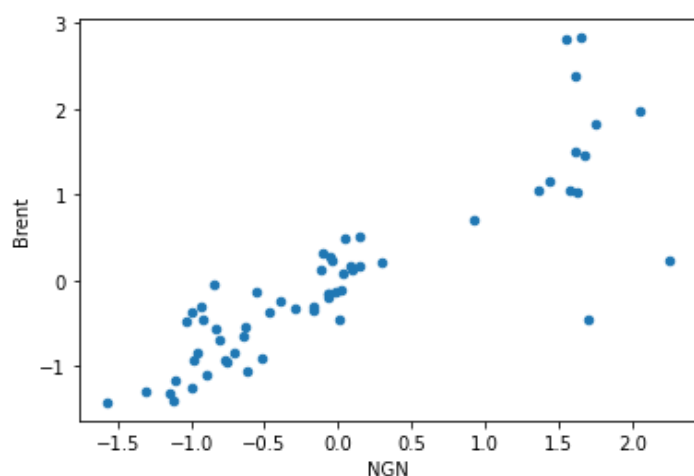
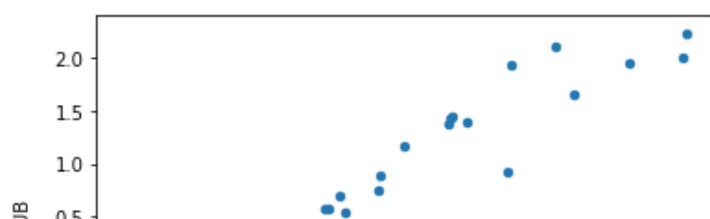
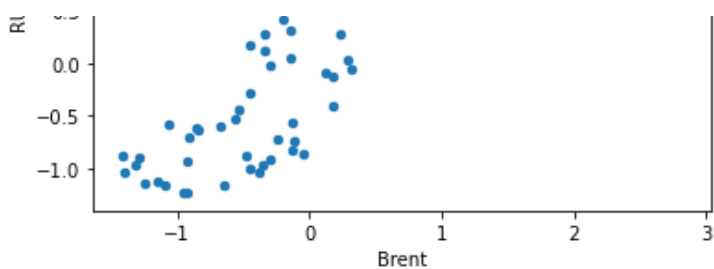


Диаграмма рассеяния для Brent и RUB





In [14]:

```
print("Корреляция Спирмена для для NGN и RUB: ", sts.spearmanr(scaled_df['NGN'], scaled_df['RUB']))
print("Корреляция Пирсона для для NGN и RUB: ", sts.pearsonr(scaled_df['NGN'], scaled_df['RUB']))
print("Корреляция Кендала для для NGN и RUB: ", sts.kendalltau(scaled_df['NGN'], scaled_df['RUB']))
```

Корреляция Спирмена для для NGN и RUB: SpearmanrResult(correlation=0.8637399277577107, pvalue=6.567557236137693e-19)
 Корреляция Пирсона для для NGN и RUB: (0.8682306574586346, 2.6514705256049466e-19)
 Корреляция Кендала для для NGN и RUB: KendalltauResult(correlation=0.6711864406779661, pvalue=3.536953306268572e-14)

In [15]:

```
print("Корреляция Спирмена для для NGN и Brent: ", sts.spearmanr(scaled_df['NGN'], scaled_df['Brent']))
print("Корреляция Пирсона для для NGN и Brent: ", sts.pearsonr(scaled_df['NGN'], scaled_df['Brent']))
print("Корреляция Кендала для для NGN и Brent: ", sts.kendalltau(scaled_df['NGN'], scaled_df['Brent']))
```

Корреляция Спирмена для для NGN и Brent: SpearmanrResult(correlation=0.8676854681856072, pvalue=2.965334821182811e-19)
 Корреляция Пирсона для для NGN и Brent: (0.8562293068155475, 2.7920578743092767e-18)
 Корреляция Кендала для для NGN и Brent: KendalltauResult(correlation=0.711864406779661, pvalue=9.267550828144194e-16)

In [16]:

```
print("Корреляция Спирмена для для Brent и RUB: ", sts.spearmanr(scaled_df['Brent'], scaled_df['RUB']))
print("Корреляция Пирсона для для Brent и RUB: ", sts.pearsonr(scaled_df['Brent'], scaled_df['RUB']))
print("Корреляция Кендала для для Brent и RUB: ", sts.kendalltau(scaled_df['Brent'], scaled_df['RUB']))
```

Корреляция Спирмена для для Brent и RUB: SpearmanrResult(correlation=0.8607390941928316, pvalue=1.1828013320320498e-18)
 Корреляция Пирсона для для Brent и RUB: (0.9086923527619712, 1.1336312476597168e-23)
 Корреляция Кендала для для Brent и RUB: KendalltauResult(correlation=0.688135593220339, pvalue=7.953621422745603e-15)

In [17]:

```
print('Диаграмма рассеяния для NGN и RUB')
x = scaled_df['NGN']
y = scaled_df['RUB']
a, b = np.polyfit(x, y, 1)
plt.plot(x, y, 'ro', x, a*x + b, 'b')
plt.show()
print('Диаграмма рассеяния для NGN и Brent')
x = scaled_df['NGN']
y = scaled_df['Brent']
a, b = np.polyfit(x, y, 1)
plt.plot(x, y, 'ro', x, a*x + b, 'b')
plt.show()
print('Диаграмма рассеяния для Brent и RUB')
x = scaled_df['Brent']
y = scaled_df['RUB']
```

```
a, b = np.polyfit(x, y, 1)
plt.plot(x, y, 'ro', x, a*x + b, 'b')
plt.show()
```

Диаграмма рассеяния для NGN и RUB

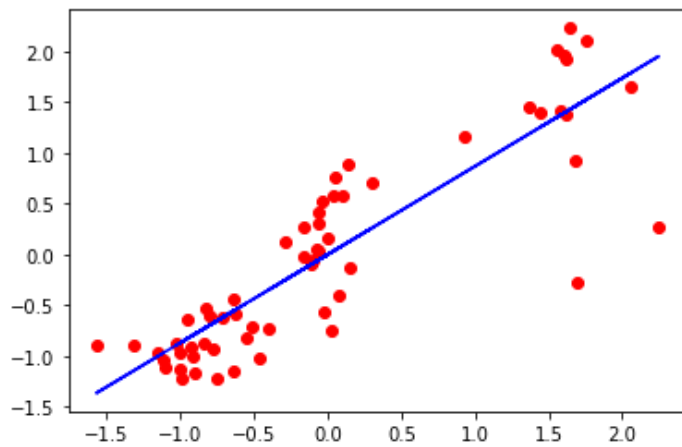


Диаграмма рассеяния для NGN и Brent

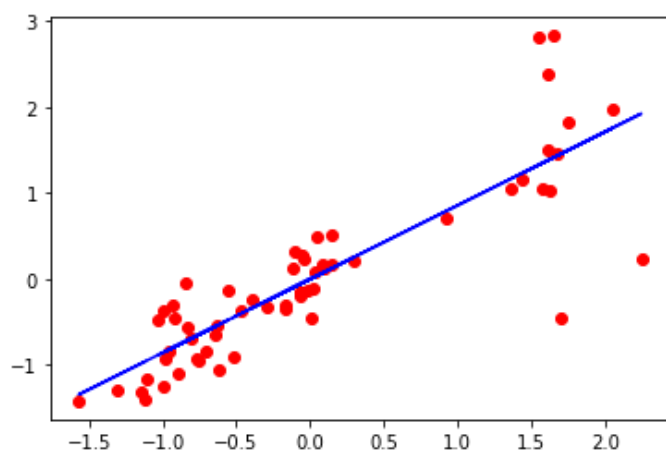


Диаграмма рассеяния для Brent и RUB

