PO_Blatt03

May 9, 2023

1 Praktische Optimierung Blatt 03

1.0.1 Tobias Lotz: 217856

1.0.2 Alexander van der Staay: 185444

[ ]:
```
from kompasssuche_test import kompasssuche_test
import numpy as np
from scipy.stats import ttest_ind, ranksums
import time
import matplotlib.pyplot as plt
import statsmodels.api as sm
```

1.0.3 Aufgabe 3.1

[ ]:
```
def f(x): return x[0]**2 + x[1]**2
```

[ ]:
```
np.random.seed(1)
```

[ ]:
```
x_sample = np.random.uniform(low=-10, high=10, size=500)
y_sample = np.random.uniform(low=-10, high=10, size=500)

sample = np.array(list(zip(x_sample, y_sample)))
```

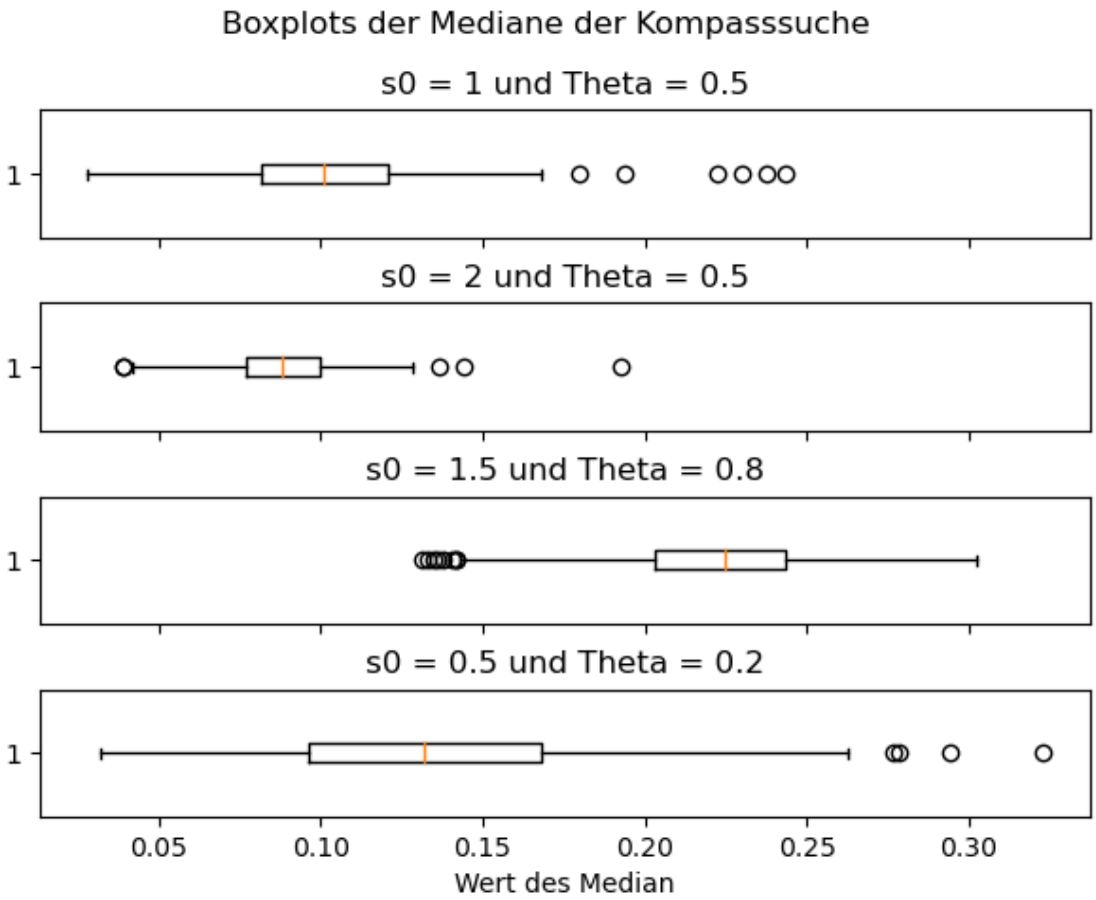
[ ]:
```
params = np.array([[1, 0.5], [2, 0.5], [1.5, 0.8], [0.5, 0.2]])
```

[ ]:
```
medians = []
for (s, t) in params:
    for (x, y) in sample:
        times = []
        for _ in range(100):
            start = time.perf_counter()
            kompasssuche_test(f, np.array([x, y]), s, t)
            exec_time_ms = (time.perf_counter() - start) * 1000
            times.append(exec_time_ms)
        times = np.array(times)
        medians.append(np.median(times, axis=0))
```

[ ]:
```
medians = np.array(medians)
```

[ ]:
```
fig, ax = plt.subplots(4, sharex=True)
fig.suptitle('Boxplots der Mediane der Kompasssuche')
fig.tight_layout(pad=1.5)
plt.xlabel('Wert des Median')
ax[0].set_title('s0 = 1 und Theta = 0.5')
ax[0].boxplot(medians[:500], vert=False)#
ax[1].set_title('s0 = 2 und Theta = 0.5')
ax[1].boxplot(medians[500:1000], vert=False)
ax[2].set_title('s0 = 1.5 und Theta = 0.8')
ax[2].boxplot(medians[1000:1500], vert=False)
ax[3].set_title('s0 = 0.5 und Theta = 0.2')
ax[3].boxplot(medians[1500:2000], vert=False)
```

[ ]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fe4a9a70b80>,
  <matplotlib.lines.Line2D at 0x7fe4a9a70e20>],
 'caps': [<matplotlib.lines.Line2D at 0x7fe4a9a7e100>,
  <matplotlib.lines.Line2D at 0x7fe4a9a7e3a0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fe4a9a708e0>],
 'medians': [<matplotlib.lines.Line2D at 0x7fe4a9a7e640>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fe4a9a7e8e0>],
 'means': []}
```



1.0.4 Interpretation

Das Zentrum des Plots der Parameter (2) liegt am weitesten links. Daher scheint die Optimierung mit diesen Parametern im Mittel am schnellsten zu sein. Außerdem ist die Streuung der Werte geringer, als die der anderen Parameterkombinationen. Ein bisschen langsamer scheint die Kombination (1), außerdem liegt eine höhere Streuung vor und es gibt größere Ausreißerpunkte. An dritter Stelle liegt die Kombination (4), die ein gutes Stück langsamer scheint als (1). Die Streuung ist hier am größten, dafür gibt es nicht so viele Ausreißer. Die Langsamste Parameterkombination ist hier (3). Die Verkleinerung von $\theta = 0.8$ scheint zu klein für die initiale Schrittgröße von $s_0 = 1.5$ zu sein. $\theta = 0.5$ führt schneller zur Konvergenz selbst bei Schrittgrößen $s_0 = 1$ und $s_0 = 2$. Die initiale Schrittgröße von $s_0 = 0.5$ scheint etwas zu klein zu sein, da sie hier langsamer konvergiert als eine größere Schrittgröße mit entsprechendem Reduktionsfaktor θ.

1.0.5 Tests

$(i) H_0 : \mu_0 \geq \mu_1 vs. H_1 : \mu_0 < \mu_1$

[ ]:

```
# Welch-Test
welch_val, welch_pval = ttest_ind(medians[:500], medians[1000:1500],␣
 ↪alternative='less', equal_var=False)
welch_reject_h0 = welch_pval < 0.05

# Wilcoxon-Test
wil_val, wil_pval = ranksums(medians[:500], medians[1000:1500],␣
 ↪alternative='less')
wil_reject_h0 = wil_pval < 0.05
```

[ ]:
```
print(f'Test:\t\tStatistik:\t\tP-Wert:\t\t\t\tH_0 verwerfen:')
print(f'Welch\t\t{welch_val : .2f}\t\t\t{welch_pval}\t\t\t\t{welch_reject_h0}')
print(f'Wilcoxon\t{wil_val : .2f}\t\t\t{wil_pval}\t\t{wil_reject_h0}')
```

```
Test:           Statistik:              P-Wert:                         H_0
verwerfen:
Welch           -63.98                  0.0                             True
Wilcoxon        -26.96                  2.2100372300782535e-160         True
```

Bei einem Signifikanzniveau $\alpha = 0.05$ verwerfen die beiden Tests die Nullhypothese H_0. Somit können wir zum Signifikanzniveau von 5% sagen, dass die Kompasssuche mit $s_0 = 1$ und $\theta = 0.5$ im Mittel weniger Zeit benötigt, als die Kompasssuche mit $s_0 = 1.5$ und $\theta = 0.8$

$(ii) H_0 : \mu_0 \geq \mu_1 vs. H_1 : \mu_0 < \mu_1$

[ ]:
```
# Welch-Test
welch_val, welch_pval = ttest_ind(medians[1000:1500], medians[1500:2000],␣
 ↪alternative='less', equal_var=False)
welch_reject_h0 = welch_pval < 0.05

# Wilcoxon-Test
wil_val, wil_pval = ranksums(medians[1000:1500], medians[1500:2000],␣
 ↪alternative='less')
wil_reject_h0 = wil_pval < 0.05
```

[ ]:
```
print(f'Test:\t\tStatistik:\t\tP-Wert:\t\tH_0 verwerfen:')
print(f'Welch\t\t{welch_val : .2f}\t\t\t{welch_pval}\t\t{welch_reject_h0}')
print(f'Wilcoxon\t{wil_val : .2f}\t\t\t{wil_pval}\t\t{wil_reject_h0}')
```

```
Test:           Statistik:              P-Wert:         H_0 verwerfen:
Welch            33.43                  1.0             False
Wilcoxon         23.48                  1.0             False
```

Bei einem Signifikanzniveau $\alpha = 0.05$ verwirft keiner der beiden Tests die Nullhypothese H_0. Somit können wir zum Signifikanzniveau von 5% nicht sagen, dass die Kompasssuche mit $s_0 = 1.5$ und $\theta = 0.8$ im Mittel weniger Zeit benötigt, als die Kompasssuche mit $s_0 = 0.5$ und $\theta = 0.2$

$(iii) H_0 : \mu_0 \geq \mu_1 vs. H_1 : \mu_0 < \mu_1$

```
[ ]: # Welch-Test
welch_val, welch_pval = ttest_ind(medians[500:1000], medians[1000:1500],␣
 ↪alternative='less', equal_var=False)
welch_reject_h0 = welch_pval < 0.05

# Wilcoxon-Test
wil_val, wil_pval = ranksums(medians[500:1000], medians[1000:1500],␣
 ↪alternative='less')
wil_reject_h0 = wil_pval < 0.05
```

```
[ ]: print(f'Test:\t\tStatistik:\t\tP-Wert:\t\t\t\tH_0 verwerfen:')
print(f'Welch\t\t{welch_val : .2f}\t\t\t{welch_pval}\t\t\t\t{welch_reject_h0}')
print(f'Wilcoxon\t{wil_val : .2f}\t\t\t{wil_pval}\t\t{wil_reject_h0}')
```

```
Test:           Statistik:              P-Wert:                         H_0
verwerfen:
Welch           -87.09                  0.0                             True
Wilcoxon        -27.35                  4.949810664609923e-165          True
```

Bei einem Signifikanzniveau $\alpha = 0.05$ verwerfen die beiden Tests die Nullhypothese H_0. Somit können wir zum Signifikanzniveau von 5% sagen, dass die Kompasssuche mit $s_0 = 2$ und $\theta = 0.5$ im Mittel weniger Zeit benötigt, als die Kompasssuche mit $s_0 = 1.5$ und $\theta = 0.8$

$(iv) H_0 : \mu_0 \geq \mu_1 vs. H_1 : \mu_0 < \mu_1$

```
[ ]: # Welch-Test
welch_val, welch_pval = ttest_ind(medians[:500], medians[1500:2000],␣
 ↪alternative='less', equal_var=False)
welch_reject_h0 = welch_pval < 0.05

# Wilcoxon-Test
wil_val, wil_pval = ranksums(medians[:500], medians[1500:2000],␣
 ↪alternative='less')
wil_reject_h0 = wil_pval < 0.05
```

```
[ ]: print(f'Test:\t\tStatistik:\t\tP-Wert:\t\t\t\tH_0 verwerfen:')
print(f'Welch\t\t{welch_val : .2f}\t\t\t{welch_pval}\t\t{welch_reject_h0}')
print(f'Wilcoxon\t{wil_val : .2f}\t\t\t{wil_pval}\t\t{wil_reject_h0}')
```

```
Test:           Statistik:              P-Wert:                         H_0
verwerfen:
Welch           -12.08                  2.7610239965363137e-31          True
Wilcoxon        -10.62                  1.2047173441657044e-26          True
```

Bei einem Signifikanzniveau $\alpha = 0.05$ verwerfen die beiden Tests die Nullhypothese H_0. Somit können wir zum Signifikanzniveau von 5% sagen, dass die Kompasssuche mit $s_0 = 1$ und $\theta = 0.5$ im Mittel weniger Zeit benötigt, als die Kompasssuche mit $s_0 = 0.5$ und $\theta = 0.2$

Die beiden Tests haben alle "Entscheidungen" identisch getroffen, allerdings haben sich die P-Werte

ein wenig von einander Unterschieden. Diese Unterschiede waren aber so minimal, dass es zu keinen unterschiedlichen Entscheidungen geführt hat. Die Schlussfolgerungen, die sich hier ergeben sind: (1) ist schneller als (3) Über (3) und (4) können wir keine Aussage machen, außer das wir uns nicht sicher sind, ob (3) schneller als (4) ist (2) ist schneller als (3) (1) ist schneller als (4)

Diese Erkenntnisse stimmen mit denen der Boxplots überein, denn die Zentren der Boxplots von (1) und (2) liegen weiter links als die von (3) und (4) und sind somit im Mittel schneller als (3) und (4). Das Zentrum des (3) Plots liegt ein gutes Stück rechts von (4) und ist somit im Mittel nicht schneller als (4).

1.0.6 Annahmen Welch-Test

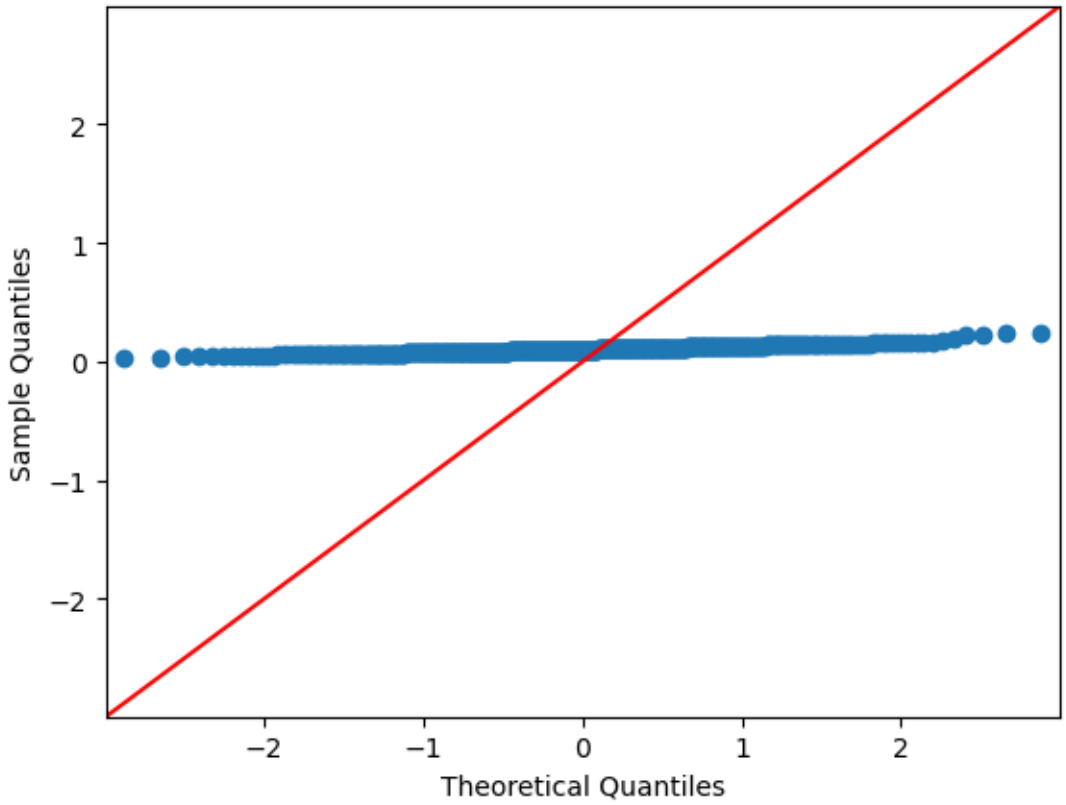
- Normalverteilung der Grundgesamtheit
- Ungleiche Varianz der Verteilungen

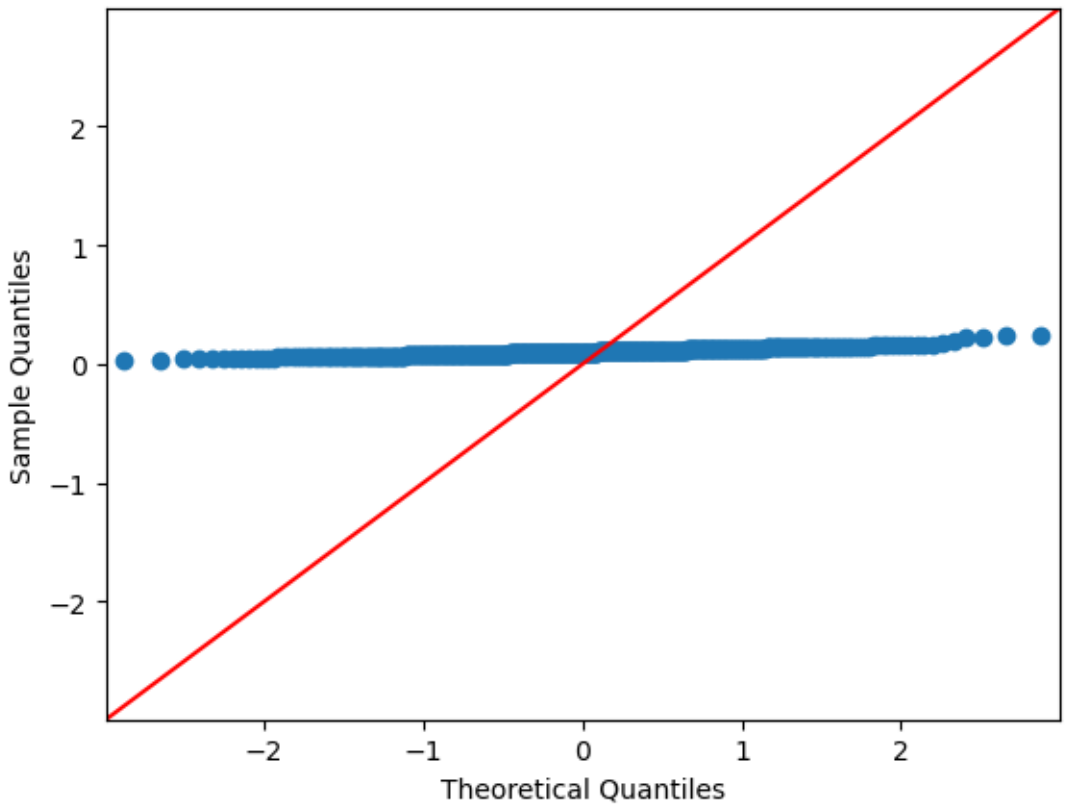
1.0.7 Annahmen Wilcoxon-Test

- $D_i = X_{i,1} - X_{i,2}$ unabhängig, identisch verteilt, stetig und symmetrisch

[ ]:
```
sm.qqplot(medians[:500], line='45')
```

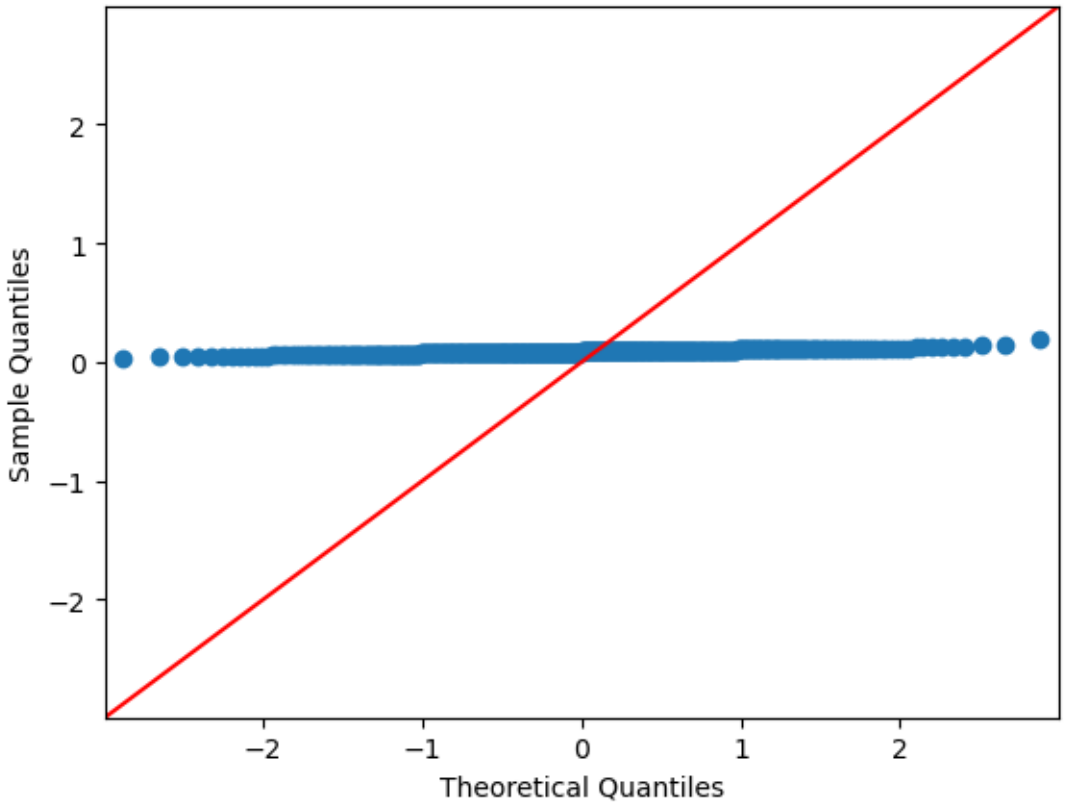
[ ]:

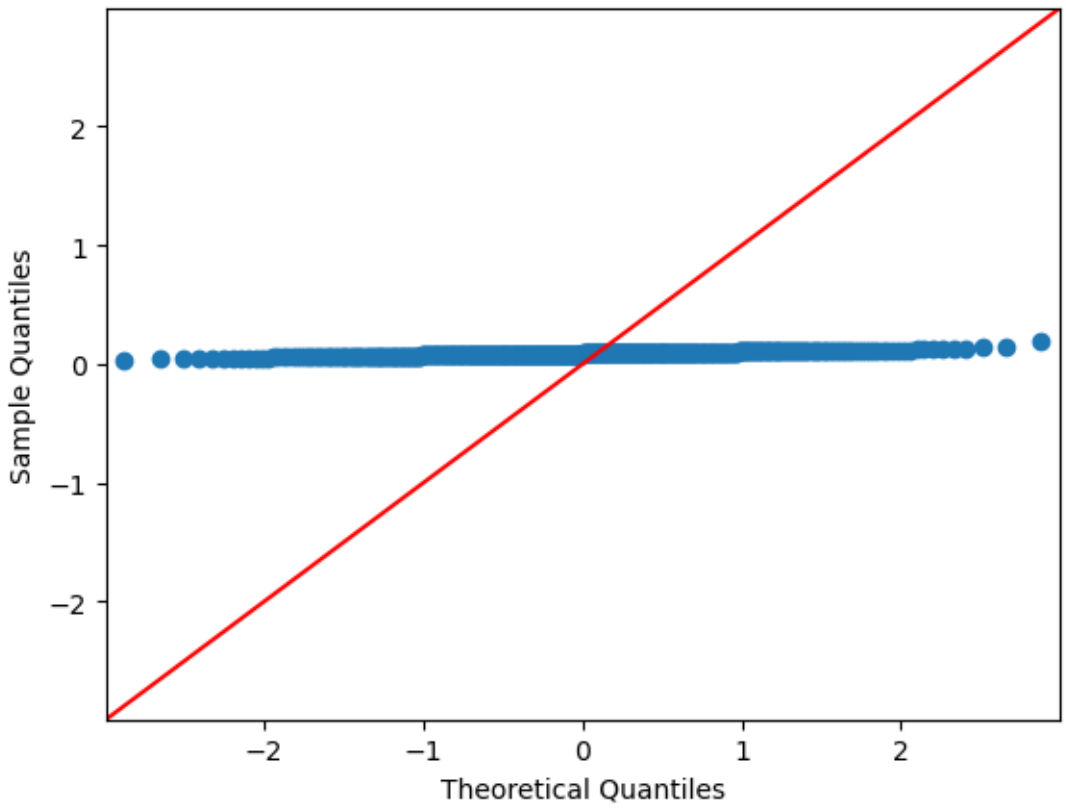




[ ]:
```
sm.qqplot(medians[500:1000], line='45')
```

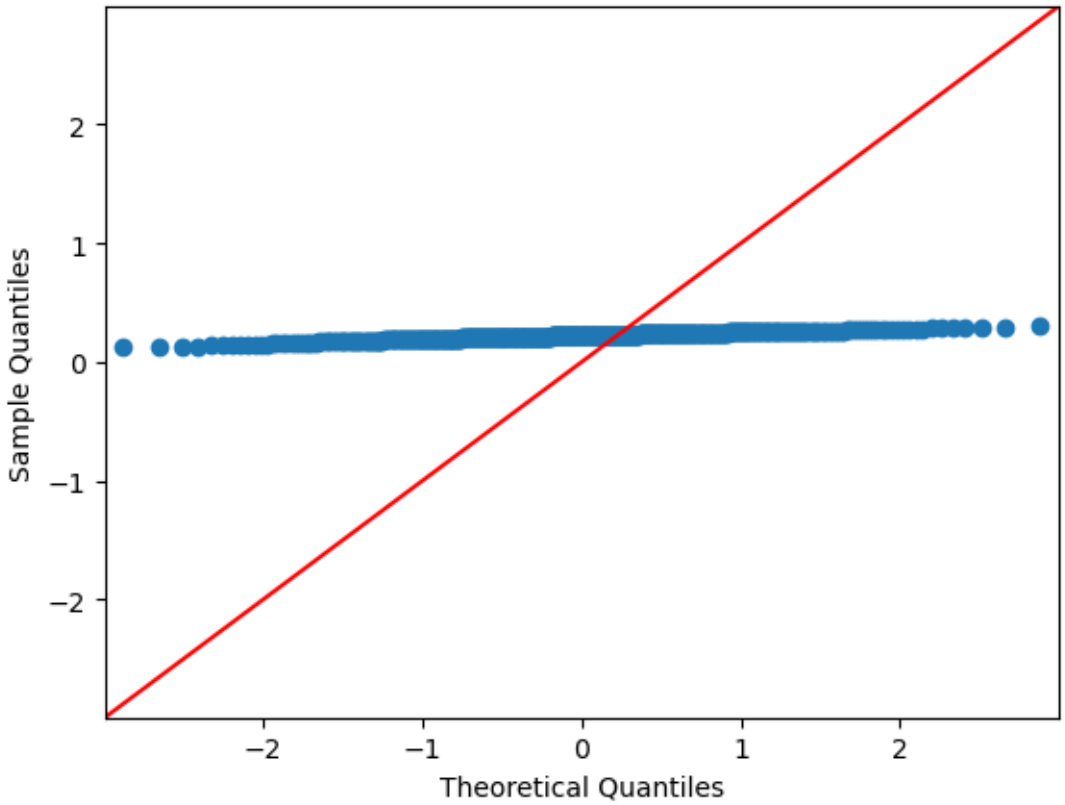
[ ]:

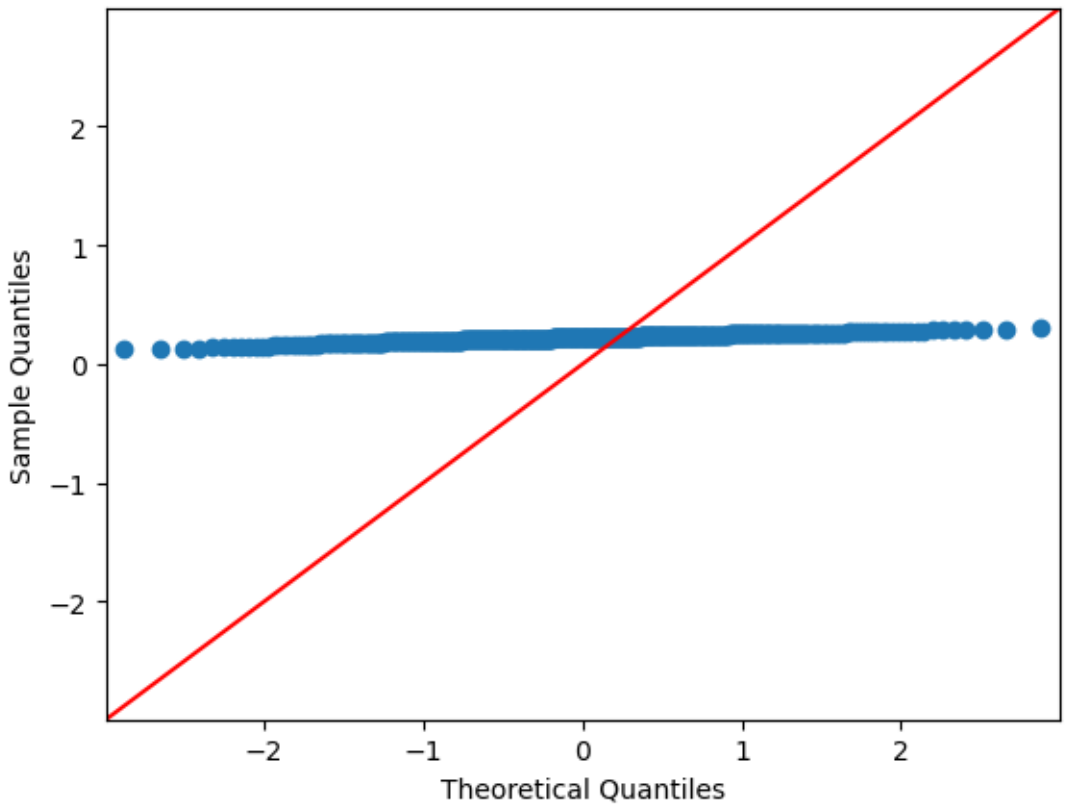

Sample Quantiles
Theoretical Quantiles
2
1
0
−1
−2
−2
−1
0
1
2



[ ]:
```
sm.qqplot(medians[1000:1500], line='45')
```

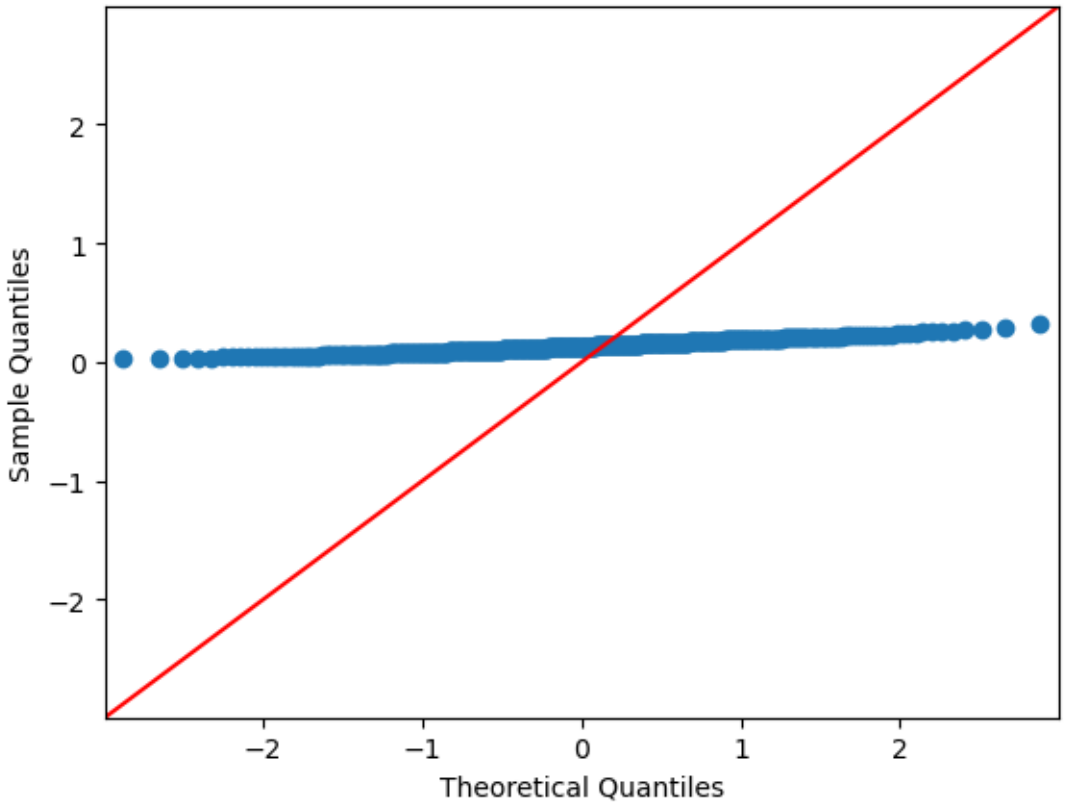
[ ]:

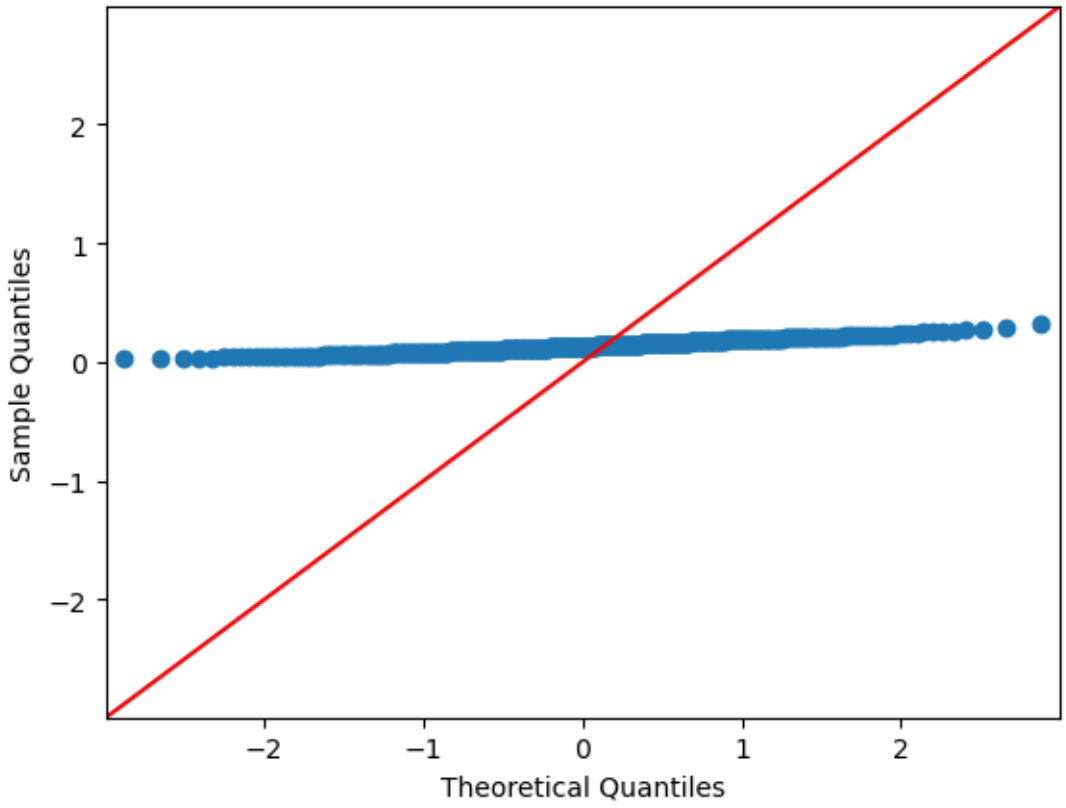

Sample Quantiles
Theoretical Quantiles
2
1
0
−1
−2
−2
−1
0
1
2



[ ]:
```
sm.qqplot(medians[1500:2000], line='45')
```

[ ]:


Sample Quantiles
Theoretical Quantiles
2
1
0
−1
−2
−2
−1
0
1
2



Die Mediane sind nicht normalverteilt, also ist die Normalverteilungsannahme des Welch-Test nicht erfüllt.

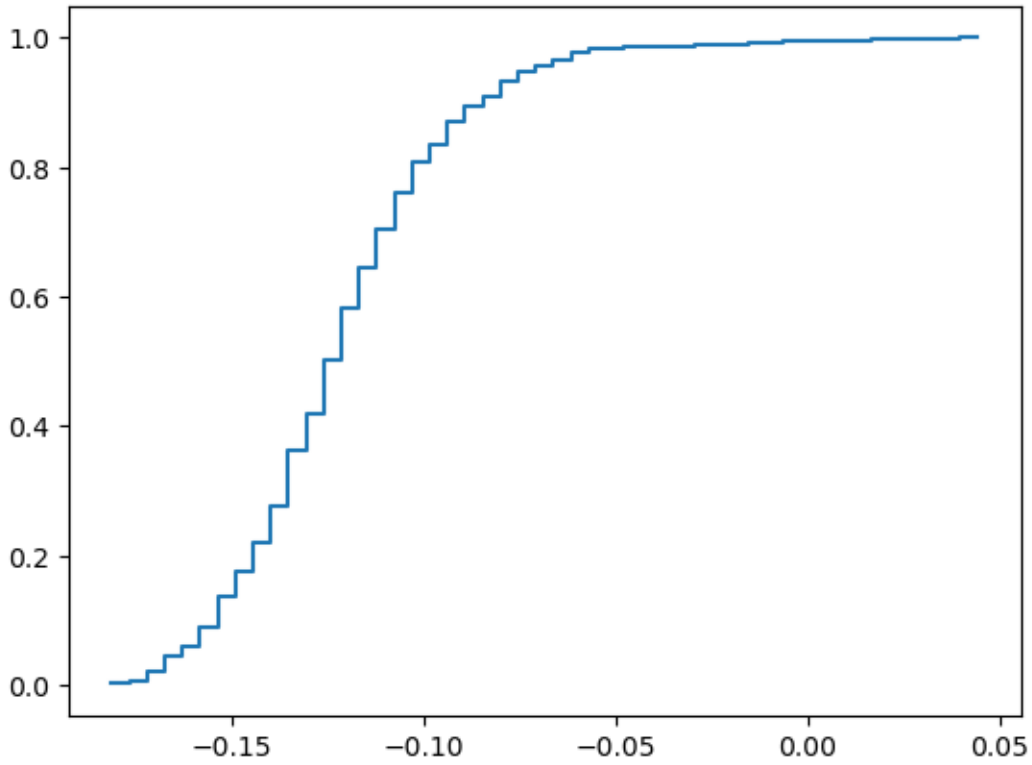
[ ]:
```
x0 = medians[:500]
x1 = medians[500:1000]
x2 = medians[1000:1500]
x3 = medians[1500:2000]

D_Test_i = np.asarray(x0)-np.asarray(x2)
ecdf = sm.distributions.ECDF(D_Test_i)

x = np.linspace(min(D_Test_i), max(D_Test_i))
y = ecdf(x)

plt.step(x, y)
```

[ ]:
```
[<matplotlib.lines.Line2D at 0x7fe4a979ac70>]
```

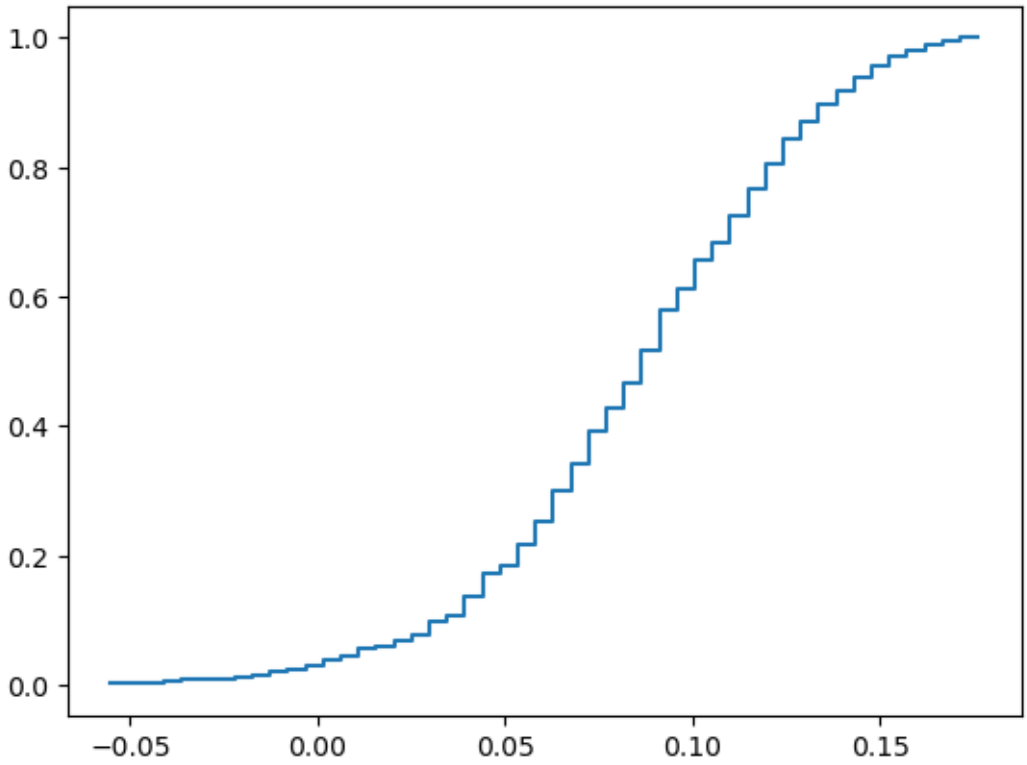


[ ]:
```
D_Test_i = np.asarray(x2)-np.asarray(x3)
ecdf = sm.distributions.ECDF(D_Test_i)

x = np.linspace(min(D_Test_i), max(D_Test_i))
y = ecdf(x)

plt.step(x, y)
```

[ ]:
```
[<matplotlib.lines.Line2D at 0x7fe4a96a3eb0>]
```

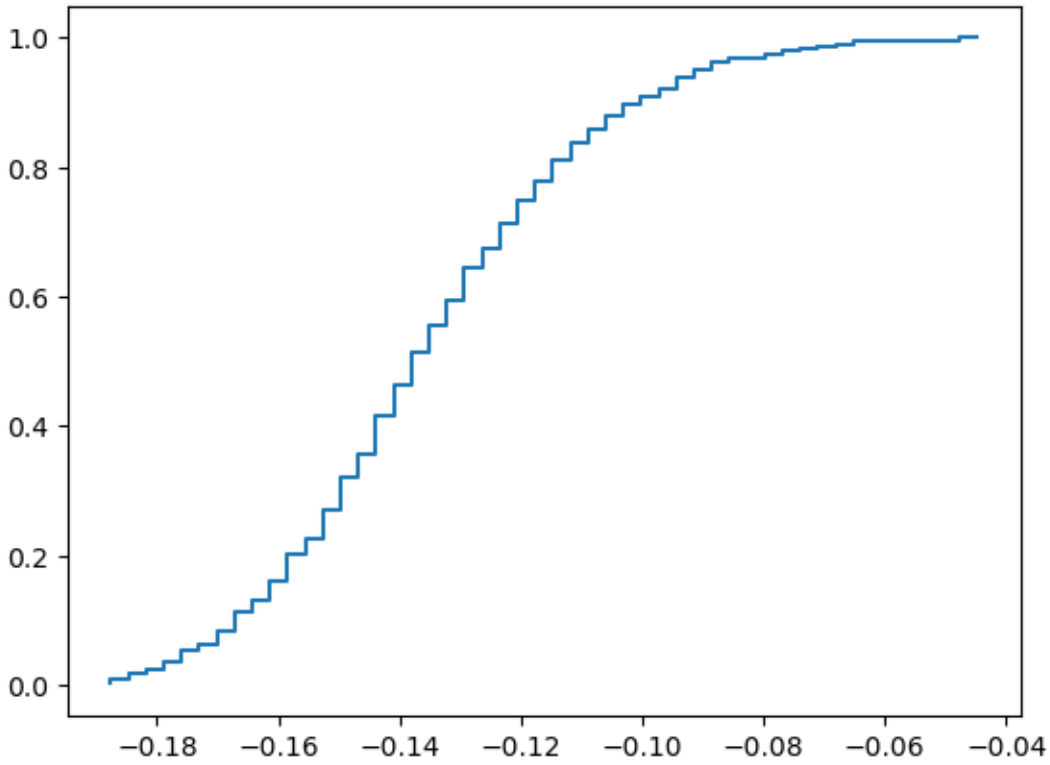


[ ]:
```
D_Test_i = np.asarray(x1)-np.asarray(x2)
ecdf = sm.distributions.ECDF(D_Test_i)

x = np.linspace(min(D_Test_i), max(D_Test_i))
y = ecdf(x)

plt.step(x, y)
```

[ ]:
```
[<matplotlib.lines.Line2D at 0x7fe4a96bc8e0>]
```

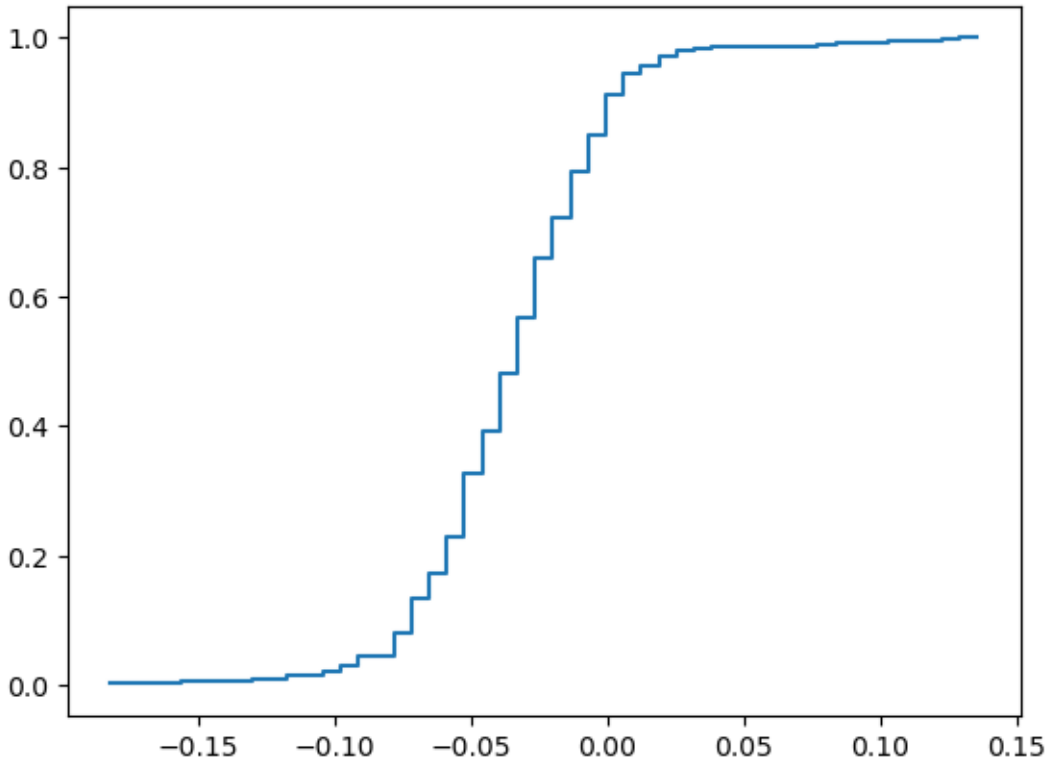


[ ]:
```
D_Test_i = np.asarray(x0)-np.asarray(x3)
ecdf = sm.distributions.ECDF(D_Test_i)

x = np.linspace(min(D_Test_i), max(D_Test_i))
y = ecdf(x)

plt.step(x, y)
```

[ ]:
```
[<matplotlib.lines.Line2D at 0x7fe4a9655370>]
```



Die unabhängigkeit der Verteilungen der Mediane ist auch nicht gegeben. Also ist auch diese Bedingung des Wilcoxon-Test nicht erfüllt.

1.0.8 Aufgabe 3.2

Es gilt $\theta = \theta_0$. Außerdem gilt, dass H_0 für ein $\theta \neq \theta_0$ mit einer Wahrscheinlichkeit von maximal α verworfen wird. $\Rightarrow$ für ein belibiges $\epsilon > 0$ gilt, $G_\varphi(\theta - \epsilon) \leq \alpha$ und $G_\varphi(\theta + \epsilon) \leq \alpha$. Da G_φ stetig ist, gilt also auch $G_\varphi(\theta) \leq \alpha$

Inhaltlich bedeutet das, dass die Gütefunktion auf den Intervallen $\theta < \theta_0$ und $\theta > \theta_0$ den Wert α nicht überschreitet (Def. Signifikanzniveau). Um also an der Stelle $\theta = \theta_0$ einen größeren Wert annehmen zu können, müsste die Funktion an der Stelle einen Sprung machen. Da diese aber immer stetig ist, ist das nicht möglich. Man kann also keinen Test konstruieren, dessen Gütefunktion in genau einem Punkt einen hohen Wert (nahe 1.0) annimmt und in allen anderen Werten unterhalb des Signifikantsniveaus α bleibt.

1.0.9 Aufgabe 3.3

(a) Nein, da stets nur eine signifikante Aussage getroffen werden kann, wenn H_0 verworfen wird. Ist dies der Fall, kann man sich sicher sein, dass die Stichprobe unter Annahme H_0 sehr unwahrscheinlich ist. Wird H_0 nicht verworfen heißt dies aber noch nicht, dass H_0 wahrscheinlich richtig ist. (b) Nein. Die Stichprobe wird nicht verworfen, also ist keine signifikante Aussage über die Hypothesen möglich, außerdem liegt der p-wert von 0.08 nur knapp über dem Signifikanzniveau von 0.05, also ist diese Stichprobe unter Annahme einer Normalverteilung immernoch recht unwahrscheinlich. Auch

hier kann man sich nicht sicher sein, dass die Stichprobe normalverteilt ist, da der Test die Nullhypothese nicht verworfen hat. (*c*) Nein. Die Signifikanz einer Aussage trifft keinerlei Aussage über die Relevanz einer Aussage. Ein Beispiel hierfür wäre der Vergleich von zwei objektiv schlechten Algorithmen. Trifft man eine signifikante Aussage darüber, dass Algorithmus A schneller ist als Algorithmus B, aber es ist bekannt das Algorithmus C deutlich schneller ist als A und B, so hat diese Aussage keine wirkliche Relevanz.