DV Seminar 3

In [1]:
```python
import numpy as np, pandas as pd, matplotlib.pyplot as plt
```

```
c:\Users\Stefan\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\projections\__init__.py:63: UserWarning: Unable to i
mport Axes3D. This may be due to multiple versions of Matplotlib being installed (e.g. as a system package and as a pip package). As a re
sult, the 3D projection is not available.
  warnings.warn("Unable to import Axes3D. This may be due to multiple versions of "
```

In [2]:
```python
results = pd.read_csv('results.csv')
```

In [3]:
```python
results.sample(5)
```

Out[3]:

	id_alg	param_1	param_2	param_3	id_dataset	param_4	mean_ind	std_ind	ind_0	ind_1	ind_2	ind_3	ind_4	ind_5	ind_6	ind_7	i
482	OBLQ_1	-1	-	5	1	1	0.920000	0.060000	1.000000	0.900000	0.900000	0.900000	0.900000	0.900000	1.000000	1.000000	0.80
131	12	10	1	5	3	60	0.796515	0.076436	0.854545	0.759091	0.668182	0.709091	0.909091	0.763636	0.809091	0.861111	0.88
160	5	5	0	0	3	3	0.747399	0.112302	0.854545	0.750000	0.713636	0.904545	0.759091	0.854545	0.522727	0.680556	0.61
185	6	10	0	0	4	60	0.994252	0.006452	1.000000	1.000000	0.985224	1.000000	0.986842	0.983607	1.000000	1.000000	0.99
408	1	-1	0	5	7	4	0.762500	0.083437	0.733333	0.816667	0.591667	0.858333	0.800000	0.808333	0.841667	0.700000	0.65

In [4]:
```python
algo_ids = list(sorted(set(results['id_alg'].values)))
print(algo_ids)
```

```
['1', '10', '11', '12', '2', '3', '4', '5', '6', '7', '8', '9', 'OBLQ_1', 'OBLQ_2']
```

In [5]:
```python
dataset_ids = list(sorted(set(results['id_dataset'].values)))
print(dataset_ids)
```

```
[1, 2, 3, 4, 5, 6, 7, 8]
```

In [6]:
```python
results.groupby('id_alg').describe()
```

Out[6]:

	param_1								param_3		...	ind_8		ind_9							
	count	mean	std	min	25%	50%	75%	max	count	mean	...	75%	max	count	mean	std	min	25%	50%	75%	max
id_alg																					
1	40.0	-1.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	40.0	5.0	...	0.975814	1.0	40.0	0.789516	0.167548	0.312500	0.685985	0.816550	0.916230	1.0
10	40.0	10.0	0.0	10.0	10.0	10.0	10.0	10.0	40.0	5.0	...	0.976219	1.0	40.0	0.845433	0.152475	0.333333	0.724936	0.860111	0.988487	1.0
11	40.0	5.0	0.0	5.0	5.0	5.0	5.0	5.0	40.0	5.0	...	0.967564	1.0	40.0	0.820050	0.161333	0.416667	0.695644	0.866667	0.971689	1.0
12	40.0	10.0	0.0	10.0	10.0	10.0	10.0	10.0	40.0	5.0	...	0.977028	1.0	40.0	0.839014	0.141028	0.458333	0.714583	0.854215	0.986842	1.0
2	40.0	-1.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	40.0	5.0	...	0.973738	1.0	40.0	0.818730	0.165192	0.312500	0.705303	0.812383	0.992208	1.0
3	40.0	-1.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	40.0	10.0	...	0.986869	1.0	40.0	0.808657	0.160890	0.479167	0.694409	0.847917	0.966220	1.0
4	40.0	-1.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	40.0	10.0	...	0.970449	1.0	40.0	0.811330	0.163811	0.416667	0.690673	0.820251	0.995066	1.0
5	40.0	5.0	0.0	5.0	5.0	5.0	5.0	5.0	40.0	0.0	...	0.981935	1.0	40.0	0.831280	0.164435	0.312500	0.685433	0.872611	1.000000	1.0
6	40.0	10.0	0.0	10.0	10.0	10.0	10.0	10.0	40.0	0.0	...	0.986060	1.0	40.0	0.837082	0.146068	0.500000	0.699423	0.856250	0.992208	1.0
7	40.0	5.0	0.0	5.0	5.0	5.0	5.0	5.0	40.0	0.0	...	0.992208	1.0	40.0	0.808419	0.182123	0.291667	0.683144	0.872145	0.985224	1.0
8	40.0	10.0	0.0	10.0	10.0	10.0	10.0	10.0	40.0	0.0	...	0.982339	1.0	40.0	0.840062	0.145364	0.479167	0.747727	0.858586	0.981557	1.0
9	40.0	5.0	0.0	5.0	5.0	5.0	5.0	5.0	40.0	5.0	...	0.948365	1.0	40.0	0.829635	0.158177	0.458333	0.710000	0.847601	0.988082	1.0
OBLQ_1	40.0	-1.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	40.0	5.0	...	0.988082	1.0	40.0	0.803827	0.172445	0.447917	0.672500	0.835664	1.000000	1.0
OBLQ_2	40.0	-1.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	40.0	10.0	...	0.988082	1.0	40.0	0.803827	0.172445	0.447917	0.672500	0.835664	1.000000	1.0

14 rows × 128 columns

In [7]:
```python
algos = {}
for id_alg in algo_ids:
    algos[id_alg] = results.query(f'id_alg=="{id_alg}"')[['ind_'+str(i) for i in range(10)]]
algos[id_alg].head(1)
```

Out[7]:

	ind_0	ind_1	ind_2	ind_3	ind_4	ind_5	ind_6	ind_7	ind_8	ind_9
481	0.8	0.9	0.8	1.0	1.0	0.8	0.9	0.9	1.0	1.0

In [9]:
```
from autorank import autorank, plot_stats, create_report
data = {id_alg:d['ind_0'].values for id_alg, d in algos.items()}

df = pd.DataFrame(data)
result = autorank(df, alpha=0.05, verbose=True)
plot_stats(result)

report = create_report(result)
print(report)
```

```
Fail to reject null hypothesis that data is normal for column 1 (p=0.005032>=0.003571)
Rejecting null hypothesis that data is normal for column 10 (p=0.001629<0.003571)
Rejecting null hypothesis that data is normal for column 11 (p=0.001985<0.003571)
Rejecting null hypothesis that data is normal for column 12 (p=0.000893<0.003571)
Fail to reject null hypothesis that data is normal for column 2 (p=0.009421>=0.003571)
Rejecting null hypothesis that data is normal for column 3 (p=0.002375<0.003571)
Fail to reject null hypothesis that data is normal for column 4 (p=0.008135>=0.003571)
Rejecting null hypothesis that data is normal for column 5 (p=0.000361<0.003571)
Rejecting null hypothesis that data is normal for column 6 (p=0.000843<0.003571)
Fail to reject null hypothesis that data is normal for column 7 (p=0.014942>=0.003571)
Rejecting null hypothesis that data is normal for column 8 (p=0.001126<0.003571)
Rejecting null hypothesis that data is normal for column 9 (p=0.001083<0.003571)
Fail to reject null hypothesis that data is normal for column OBLQ_1 (p=0.004042>=0.003571)
Fail to reject null hypothesis that data is normal for column OBLQ_2 (p=0.004042>=0.003571)
Using Levene's test for homoscedacity of non-normal data.
Fail to reject null hypothesis that all variances are equal (p=0.985669>=0.050000)
Using Friedman test as omnibus test
Rejecting null hypothesis that there is no difference between the distributions (p=0.002933)
Using Nemenyi post-hoc test. Differences are significant,if the distance between the mean ranks is greater than the critical distance.
The statistical analysis was conducted for 14 populations with 40 paired samples.
The family-wise significance level of the tests is alpha=0.050.
We rejected the null hypothesis that the population is normal for the populations 3 (p=0.002), 1 (p=0.002), 2 (p=0.001), 4 (p=0.001), 12
(p=0.001), 10 (p=0.000), 8 (p=0.001), and 9 (p=0.002). Therefore, we assume that not all populations are normal.
Because we have more than two populations and the populations and some of them are not normal, we use the non-parametric Friedman test as
omnibus test to determine if there are any significant differences between the median values of the populations. We use the post-hoc Neme
nyi test to infer which differences are significant. We report the median (MD), the median absolute deviation (MAD) and the mean rank (M
R) among all populations over the samples. Differences between populations are significant, if the difference of the mean rank is greater
than the critical distance CD=3.137 of the Nemenyi test.
We reject the null hypothesis (p=0.003) of the Friedman test that there is no difference in the central tendency of the populations 3 (MD
=0.814+-0.184, MAD=0.160, MR=8.850), 11 (MD=0.841+-0.174, MAD=0.151, MR=8.775), 1 (MD=0.809+-0.183, MAD=0.151, MR=8.662), 2 (MD=0.825+-0.
158, MAD=0.124, MR=8.350), 4 (MD=0.829+-0.168, MAD=0.130, MR=8.225), OBLQ_1 (MD=0.820+-0.162, MAD=0.139, MR=7.425), OBLQ_2 (MD=0.820+-0.1
62, MAD=0.139, MR=7.425), 7 (MD=0.817+-0.161, MAD=0.115, MR=7.350), 12 (MD=0.844+-0.155, MAD=0.141, MR=7.013), 10 (MD=0.848+-0.145, MAD=
0.133, MR=6.900), 8 (MD=0.892+-0.168, MAD=0.108, MR=6.737), 6 (MD=0.817+-0.157, MAD=0.141, MR=6.675), 9 (MD=0.846+-0.147, MAD=0.140, MR=
6.475), and 5 (MD=0.887+-0.170, MAD=0.109, MR=6.138). Therefore, we assume that there is a statistically significant difference between t
he median values of the populations.
Based on the post-hoc Nemenyi test, we assume that there are no significant differences within the following groups: 3, 11, 1, 2, 4, OBLQ
_1, OBLQ_2, 7, 12, 10, 8, 6, 9, and 5. All other differences are significant.
None
```