# Autorank - a simple python package

CMSC 6950 - Computer Based Research Tools and Applications

Term Project

6th August, 2020

Submitted by

Anne Odeh

*Memorial University of Newfoundland*
*St. John's, Canada.*

# Abstract

Reproducibility is seen as one of the pillars of the entire scientific method, a criterion on which to measure the efficacy of an experiment. This paper presents how autorank can be reproduced by understanding how it works and the necessary tools that are required to mimic its exact work.

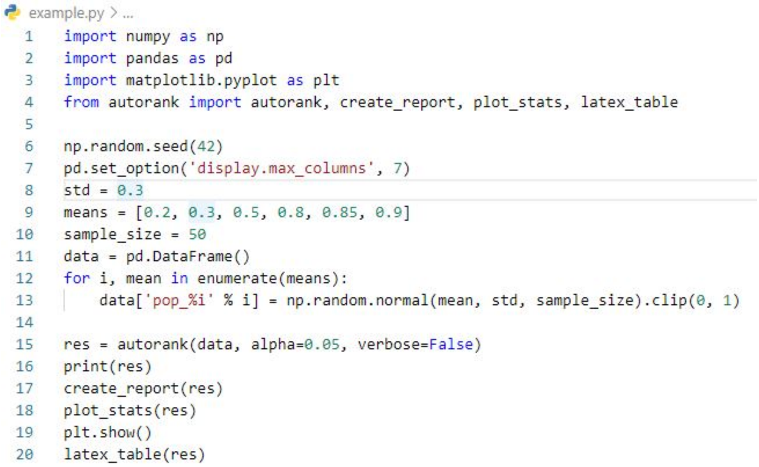
## 1 Introduction

Autorank is a simple Python package with one task: simplify the comparison between (multiple) paired populations. This is, for example, required if the performance different machine learning algorithms or simulations should be compared on multiple data sets. The performance measures on each data set are then the paired samples, the difference in the central tendency (e.g., the mean or median) can be used to rank the different algorithms. This problem is not new and how such tests could be done was already described in 2006 in the well-known article Janez Demšar. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. J. Mach. Learn. Res. 7 (December 2006), 1–30.

Regardless, the correct use of Demšar guidelines is hard for non-experts in statistics. Correct use of the guidelines requires the decision of whether a paired t-test, a Wilcoxon's rank sum test, repeated measures ANOVA with Tukey's HSD as post-hoc test, or Friedman's tests and Nemenyi's post-hoc test to determine an answer to the question if there are differences. For this, the distribution of the populations must be analyzed with the Shapiro-Wilk test for normality and, depending on the normality with Levene's test or Bartlett's tests for homogeneity of the data. All this is already quite complex. This does not yet account for the adjustment of the significance level in case of repeated tests to achieve the desired family-wise significance. Additionally, not only the tests should be conducted, but good reporting of the results also include confidence intervals, effect sizes, and the decision of whether it is appropriate to report the mean value and standard deviation, or whether the median value and the median absolute deviation is more appropriate.

# 2 Reproducing autorank

I was able to reproduce autorank by running the example code provided:

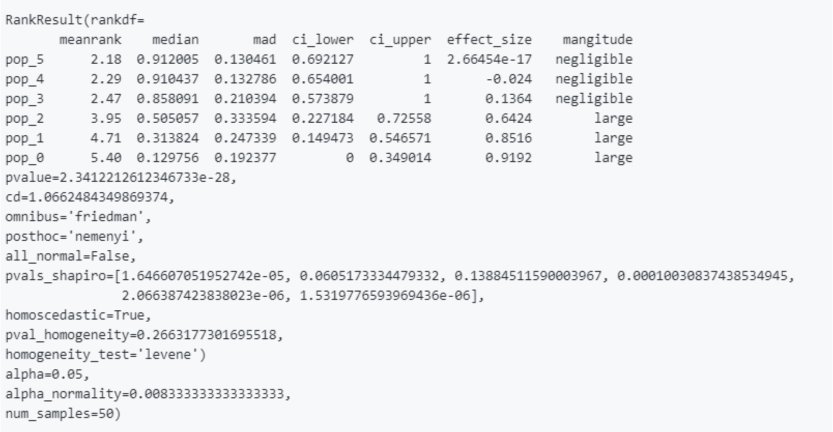


```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from autorank import autorank, create_report, plot_stats, latex_table

np.random.seed(42)
pd.set_option('display.max_columns', 7)
std = 0.3
means = [0.2, 0.3, 0.5, 0.8, 0.85, 0.9]
sample_size = 50
data = pd.DataFrame()
for i, mean in enumerate(means):
    data['pop_%i' % i] = np.random.normal(mean, std, sample_size).clip(0, 1)

res = autorank(data, alpha=0.05, verbose=False)
print(res)
create_report(res)
plot_stats(res)
plt.show()
latex_table(res)
```

Figure 1: example.py

# 3 Results

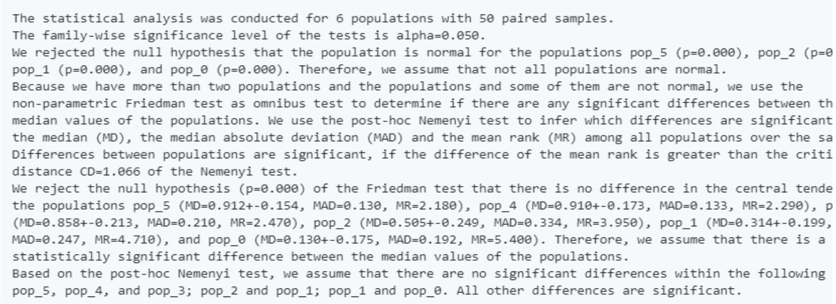
Firstly, the result of autorank shows the content of the dataset.



```
RankResult(rankdf=
       meanrank    median       mad  ci_lower  ci_upper  effect_size    mangitude
pop_5      2.18  0.912005  0.130461  0.692127         1  2.66454e-17   negligible
pop_4      2.29  0.910437  0.132786  0.654001         1       -0.024   negligible
pop_3      2.47  0.858091  0.210394  0.573879         1       0.1364   negligible
pop_2      3.95  0.505057  0.333594  0.227184   0.72558       0.6424        large
pop_1      4.71  0.313824  0.247339  0.149473  0.546571       0.8516        large
pop_0      5.40  0.129756  0.192377         0  0.349014       0.9192        large
pvalue=2.3412212612346733e-28,
cd=1.0662484349869374,
omnibus='friedman',
posthoc='nemenyi',
all_normal=False,
pvals_shapiro=[1.646607051952742e-05, 0.0605173334479332, 0.13884511590003967, 0.00010030837438534945,
               2.066387423838023e-06, 1.5319776593969436e-06],
homoscedastic=True,
pval_homogeneity=0.2663177301695518,
homogeneity_test='levene')
alpha=0.05,
alpha_normality=0.008333333333333333,
num_samples=50)
```

Figure 2: Contents of the dataset

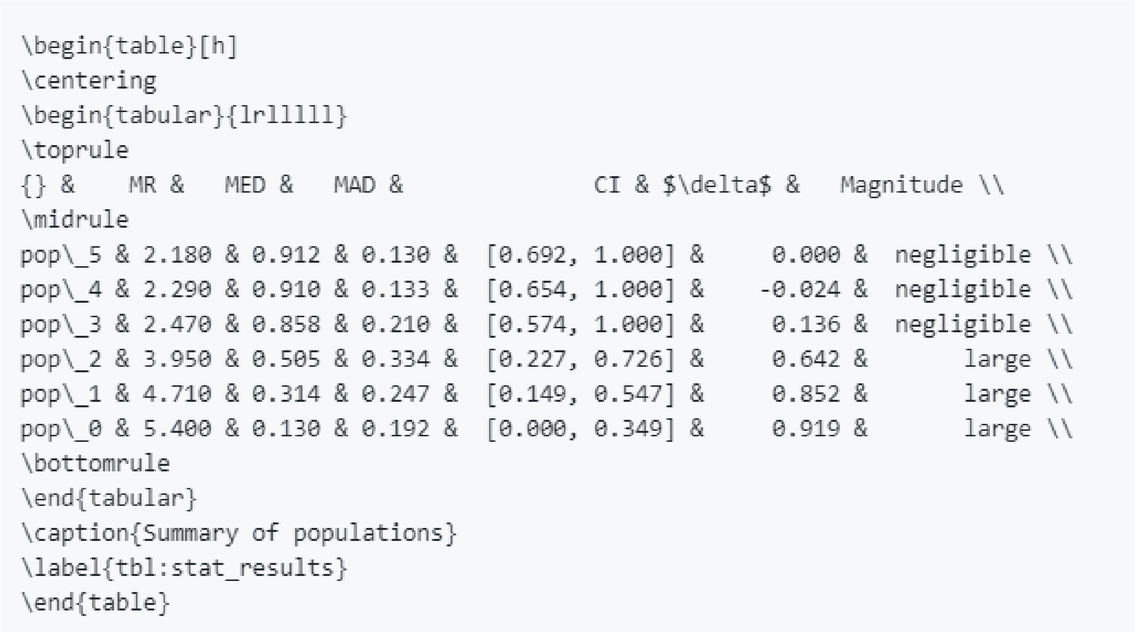
Secondly, the statistical analysis of the dataset.



```
The statistical analysis was conducted for 6 populations with 50 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 pop_5 (p=0.000), pop_2 (p=0
pop_1 (p=0.000), and pop_0 (p=0.000). 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 th
median values of the populations. We use the post-hoc Nemenyi test to infer which differences are significant
the median (MD), the median absolute deviation (MAD) and the mean rank (MR) among all populations over the sa
Differences between populations are significant, if the difference of the mean rank is greater than the criti
distance CD=1.066 of the Nemenyi test.
We reject the null hypothesis (p=0.000) of the Friedman test that there is no difference in the central tende
the populations pop_5 (MD=0.912+-0.154, MAD=0.130, MR=2.180), pop_4 (MD=0.910+-0.173, MAD=0.133, MR=2.290), p
(MD=0.858+-0.213, MAD=0.210, MR=2.470), pop_2 (MD=0.505+-0.249, MAD=0.334, MR=3.950), pop_1 (MD=0.314+-0.199,
MAD=0.247, MR=4.710), and pop_0 (MD=0.130+-0.175, MAD=0.192, MR=5.400). Therefore, we assume that there is a
statistically significant difference between the median values of the populations.
Based on the post-hoc Nemenyi test, we assume that there are no significant differences within the following
pop_5, pop_4, and pop_3; pop_2 and pop_1; pop_1 and pop_0. All other differences are significant.
```

Figure 3: Statistical Analysis

Lastly, it displays the latex format of the dataset.



```
\begin{table}[h]
\centering
\begin{tabular}{lrlllll}
\toprule
{} &    MR &   MED &   MAD &              CI & $\delta$ &   Magnitude \\
\midrule
pop\_5 & 2.180 & 0.912 & 0.130 &  [0.692, 1.000] &     0.000 &  negligible \\
pop\_4 & 2.290 & 0.910 & 0.133 &  [0.654, 1.000] &    -0.024 &  negligible \\
pop\_3 & 2.470 & 0.858 & 0.210 &  [0.574, 1.000] &     0.136 &  negligible \\
pop\_2 & 3.950 & 0.505 & 0.334 &  [0.227, 0.726] &     0.642 &       large \\
pop\_1 & 4.710 & 0.314 & 0.247 &  [0.149, 0.547] &     0.852 &       large \\
pop\_0 & 5.400 & 0.130 & 0.192 &  [0.000, 0.349] &     0.919 &       large \\
\bottomrule
\end{tabular}
\caption{Summary of populations}
\label{tbl:stat_results}
\end{table}
```

Figure 4: Latex format of the dataset

# 4 Data Exploration

## 4.1 Dataset Summary

This dataset has 17 columns and 11163 rows. Below is a detailed description of each column:

- age: the age of group of ban customers
- job: type of job (admin., blue collar, entrepreneur, housemaid, management, retired, 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown')
- marital: marital status (divorced, married, single, unknown; note: divorced means divorced or widowed)
- education: (primary, secondary, tertiary, and unknown)
- default: has credit in default? (no, yes, unknown)
- housing: has housing loan? (no, yes, unknown)
- loan: has personal loan? (no, yes, unknown)

- balance: Balance of the individual
- contact: contact communication type (cellular, telephone)
- month: last contact month of year
- day: last contact day of the week
- duration: last contact duration, in seconds (numeric).
- campaign: number of contacts performed during this campaign and for this client
- pdays: number of days that passed by after the client was last contacted from a previous campaign
- previous: number of contacts performed before this campaign and for this client
- outcome: outcome of the previous marketing campaign (failure, non-existent, success)
- deposit: has the client subscribed a term deposit? (yes or no)

## 4.2 Attributes Types

| **Attributes** | **Types** |
|---|---|
| age | numeric |
| job | categorical |
| marital | categorical |
| education | categorical |
| default | categorical |
| housing | categorical |
| loan | categorical |
| contact | categorical |
| balance | numeric |
| contact | unknown |
| month | categorical |
| day | numeric |
| duration | numeric |
| campaign | numeric |
| pdays | numeric |
| previous | numeric |
| poutcome | unknown |
| deposit | categorical |

Table 1: Attributes Types

## 4.3 Implementing autorank in my dataset

Banks use marketing campaigns as tools to focus on customer needs and their overall satisfaction strategically. However, different variables determine whether a marketing campaign will be successful or not. One of such avenue to develop a questionnaire during calls. Since duration of the call is the feature that most positively correlates with whether a potential client will open a term deposit or not.

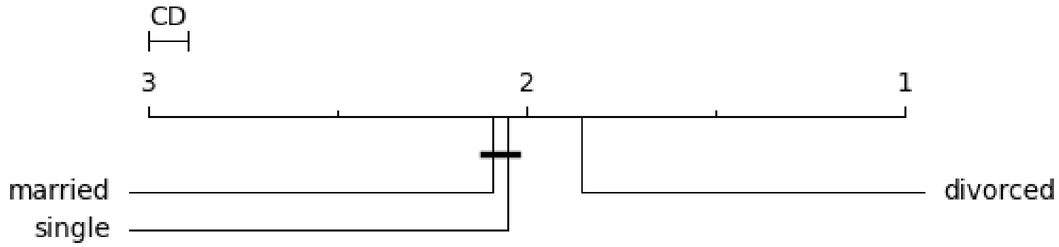


Figure 5: Autorank Plot

The plot shows the number of contacts with different subset of customers during the campaign.

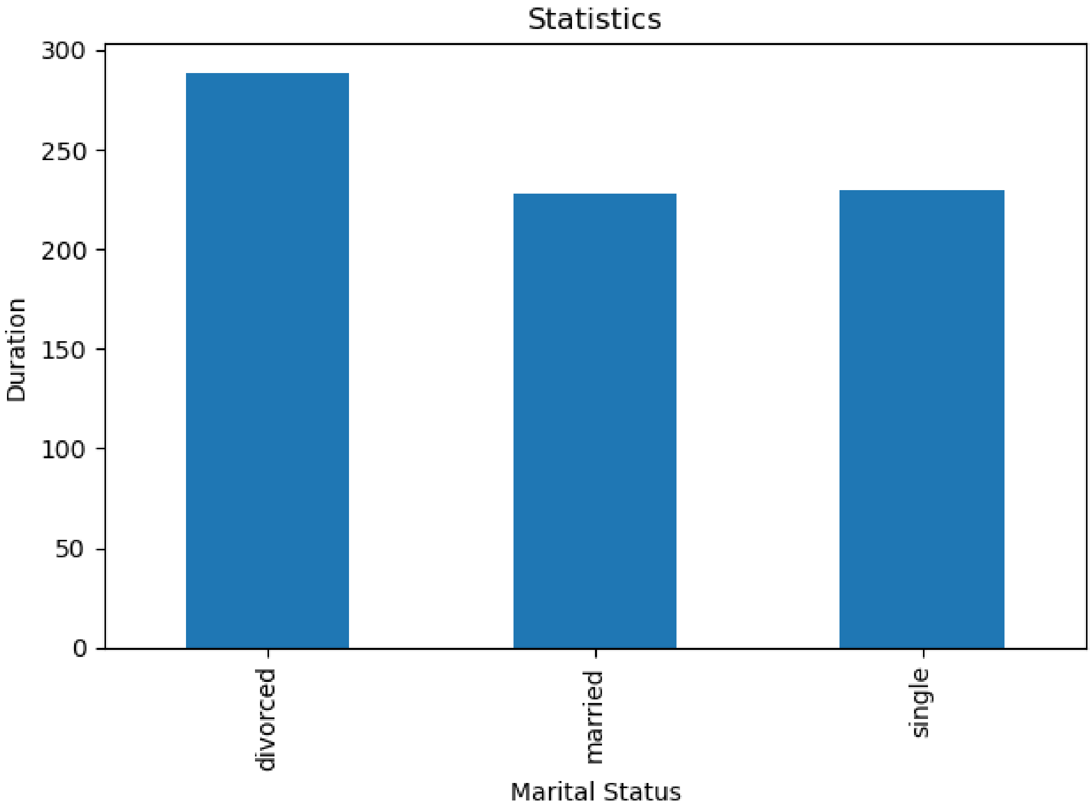


Figure 6: Marital Status/Duration

# 5 Conclusion

The goal of Autorank is to simplify the statistical analysis for non-experts. Autorank takes care of all of the above with a single function call. Additional functions allow the generation of appropriate plots, result tables, and even of a complete latex document. All that is required is the data about the populations is in a Pandas dataframe.Overall, autorank statistical analysis involves evaluating and then summarizing the data into a mathematical form that was easy to implement in my dataset.

# References

Martinez, J. (2017). Bank Marketing Dataset. Retrieved July 31, 2020, from `https://www.kaggle.com/janiobachmann/bank-marketing-dataset`

S. Herbold (2020). Autorank: A Python package for automated ranking of classifiers. Journal of Open Source Software. https://doi.org/10.21105/joss.02173