The Journal of Open Source Software

Statmanager-kr: A user-friendly statistical package for Python in Pandas

Changseok Lee [1]

1 DYPHI Research Institute, DYPHI Inc., Daejeon, Republic of Korea

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 15 January 2024
Published: unpublished

Summary

Python is one of the most popular and easiest programming languages. Many researchers use Python for data preprocessing and statistical analysis. However, there are few statistical packages that inherit Python's simple, user-friendly characteristics. Many researchers who are not familiar with programming may not know how to utilize various methods for different types of analysis and adjust parameters effectively. Consequently, people who possess statistical knowledge but lack familiarity with programming languages continue to rely on other costly software.

The `statmanager-kr` was designed to provide easy-to-use statistical functions for people with little knowledge of programming languages. Because many people are already familiar with data in table format, such as that in Microsoft Excel, `statmanager-kr` was designed to be compatible with `Pandas.DataFrame`. In addition, the `statmanager-kr` was designed so that the analysis is performed using only one method and utilizes as few parameters as possible.

Additionally, `statmanager-kr` relies on `scipy` and `statsmodels` for accurate and valid statistical analysis. The `statmanager-kr` provides functions related to testing for normality and homoscedasticity assumptions, comparing between-group and within-group differences, conducting regression analysis, and data visualization.

Statement of need

The `statmanager-kr` is a statistical package for Python in `Pandas`. This package provides functions commonly used for null hypothesis significance testing (NHST), which is of interest to researchers in various fields of research (Moon, 2020). The `statmanager-kr` provides statistical analysis functions to test the researcher's or student's hypothesis. It is also possible to check whether the assumption of normality or equivariance is met. For example, the Shapiro-Wilk Test, the Levene Test or the Fmax test can be used.

Most statistical software available today is difficult to use and expensive. One of the difficulties university students face in statistics course was the use of software (Murtonen & Lehtinen, 2003). Although there are basic statistical libraries in Python, such as Scipy (Seabold & Perktold, 2010) and Statsmodels (Virtanen et al., 2020), they are quite difficult and complex. While some studies involve complex and detailed statistical modelling and analysis, there are also many studies that require only a few hypothesis tests. It would therefore be of great benefit to these researchers if a statistical package could be developed that is easy to use.

To achieve this goal, the `statmanager-kr` has been designed to allow running analyses with only three lines of code: 1. read data as a `Pandas.DataFrame`, 2. create a `Stat_Manager` object, 3. execute the `.progress()` method. Therefore, users can use the `statmanager-kr` as long as they know the Pandas methods to read the data, such as `.read_csv()` or `.read_excel()`. It also includes functions to visualize the results depending on the analysis method.

The Journal of Open Source Software

Related Work

Recent advances in the statistical field have been accomplished through the emergence of user-friendly packages such as `Pingouin`(Vallat, 2018). The `Pingouin` is designed to be an easy-to-use statistical package that offers a wide range of tests. The purpose of tthe `statmanager-kr` has been similar to the goal of the `Pingouin` project. Similar to the `Pingouin`, the `statmanager-kr` provides various statistical functions. However, while sharing the common goal of user-friendly, the `statmanager-kr` differs in several key aspects.

The `statmanager-kr` was developed with a focus on researchers with limited programming experience. This means that in the simplified workflow, there are differences between the `statamanager-kr` and the `Pingouin`. The `statmanager-kr` offers a streamlined workflow that allows users to perform a wide range of statistical analyses by simply specifying the analysis type, variables, or group variables. Users can perform various statistical tests always using the same `.progress()` method, making it accessible to users unfamiliar with programming concepts in the `statmanager-kr` This design philosophy reduces user complexity, which is especially beneficial for those in fast-paced research environments who need fast and reliable results. Although the `Pingouin` is also easy to use, it aimed more at researchers with some programming experience. Therefore in terms of workflow, the `Pingouin` offers a more comprehensive set of statistical analysis and fine-tuning capabilities, but it requires users to learn about the analysis-specific methods to use. However, the `Pingouin` has the advantage of providing more detailed analysis results. Due to the separated functionality, the `Pingouin` supports a more extensive range of statistical analysis with more diverse and highly cutomizable than the `statmanager-kr`. Also, the `statmanager-kr` only works with `Pandas.DataFrame` for convenience, while the `Pingouin` has the advantage of being compatible with a wider range of datasets.

The `statmanager-kr` and the `Pingouin` also differ in the visualization of results and the post-hoc. The `statmanager-kr` performs post-hoc by adding the `posthoc` parameter in the `.progress()`. In addition, it is possible to visualize the results by using `.figure()` as a chain method. While the `Pingouin` does not support the ability to directly visualize the results of the analysis. However, the `Pingouin` offers more useful functions to generate graphs than the `statmanager-kr`, such as paired plot, shift plot, and plot for circle mean. In addition, the Pingouin has the advantage of supporting a wider range of post-hoc tests than the `statmanager-kr`.

Therefore, the `statmanager-kr` and the `pingouin` may be applied differently depending on the user's familiarity with programming. Researchers who are comfortable with programming and are in a situation may be better suited to the `pingouin`, which supports a wider range of analysis methods and customization. On the other hand, the `statmanager-kr` is designed to be used by researchers who are not familiar with programming and coding, but need to get fast, quick results.

Features

The `statmanager-kr` was designed to be compatible with the wide range form of `pandas.DataFrame`.

User-friendly Features

Set Language

It is possible to change the language by adjusting the `language` parameter when creating an object of the Stat_Manager class. The supported languages are Korean ("kor") and English ("eng"), and the default is Korean.

```
sm = Stat_Manager(df, language = 'eng')
```

Other Methods

Users can search for a specific usage by calling the `.howtouse()` method. It can also change the language with `.set_language()`, or change the dataframe by running `.change_dataframe()`.

Statistical Test

The implementation of analysis in statmanager-kr can be summarized as follows.

Objective	Analysis
Check the normality assumption	Kolmogorov-Smirnov Test, Shapiro-Wilks Test, Z-Skeweness & Z-Kurtosis Test
Check the homoskedasticity assumption	Levenve Test, Fmax Test
Frequency analysis	Chi-Squared Test, Fisher's Exact Test
Check the reliability of the scale	Calculating Cronbach's Alpha
Correlation analysis	Pearson's r, Spearman's rho, Kendall's tau
Comparison between groups	Independent Samples T-test, Yuen's T-test, Welch's T-test, Mann-whitney U test, Brunner-Munzel Test, One-way ANOVA, Kruskal Wallis Test, One-way ANCOVA
Comparison within group	Dependent Samples T-test, Wilcoxon-Signed Rank Test, One-way Repeated Measures ANOVA, Friedman Test, Repeated Measures ANCOVA,
Comparison by multiple ways	N-way ANOVA, N-way Mixed Repeated Measures ANOVA
Regression analysis	Linear Regression, Logistic Regression
etc	Bootstrapping percentile method

Each analysis method has its own "key" that allows it to be used in the `.progress()` method. The analysis is performed by passing the key for each analysis method to the `method` parameter in the `.progress()` method, the variables to be analyzed to the `vars` parameter, and the group variables to the `group_vars` parameter.

```
import pandas as pd
from stamanager import Stat_Manager

# 1. Reading the data
df = pd.read_csv(r'../testdata.csv', index_col = 'name')

# 2. Creating object of Stat_Manager class
sm = Stat_Manager(df)

# 3. Running: check the difference in weight by sex
sm.progress(method = 'ttest_ind', vars = 'weight', group_vars = 'sex')
```

Also, if a post-hoc test is required, as in the case of a one-way ANOVA (key of one-way ANOVA is `f_oneway`), it can be conducted by simply providing `True` to the `posthoc` parameter.

```
sm.progress(method = 'f_oneway', vars = 'income', group_vars = 'condition', posthoc = Tr
```

Keys and Related Informations

The method-specific information needed to use the `.progress()` method can be found by using the `.howtouse()` method. The detailed information is summarized in the table below:

Key	Analysis	Required Parameters	Optional Parameters
`kstest`	Kolmogorov-Smirnov Test	`vars`	`group_vars`
`shapiro`	Shapiro-Wilks Test	`vars`	`group_vars`
`z_normal`	Z-skeweness & z-kurtosis test	`vars`	`group_vars`
`levene`	Levene Test	`vars`, `group_vars`	
`fmax`	Fmax Test	`vars`, `group_vars`	
`chi2_contingency`	Chi-squared Test	`vars`	
`fisher`	Fisher's Exact Test	`vars`	
`pearsonr`	Pearson's r	`vars`	
`spearmanr`	Spearman's rho	`vars`	
`kendallt`	Kendall's tau	`vars`	
`ttest_ind`	Independent Samples T-test	`vars`, `group_vars`	
`ttest_rel`	Dependent Samples T-test	`vars`	
`ttest_ind_trim`	Yuen's Two Samples T-test	`vars`, `group_vars`	
`ttest_ind_welch`	Welch's Two Samples T-test	`vars`, `group_vars`	
`mannwhitneyu`	Mann-Whitney U Test	`vars`, `group_vars`	
`brunner`	Brunner-Munzel Test	`vars`, `group_vars`	
`wilcoxon`	Wilcoxon-Signed Rank Test	`vars`	
`bootstrap`	Boostrap Percentile Method	`vars`	`group_vars`
`f_oneway`	One-way ANOVA	`vars`, `group_vars`	`posthoc`, `posthoc_method`
`f_oneway_rm`	One-way Repeated Measures ANOVA	`vars`	`posthoc`, `posthoc_method`
`kruskal`	Kruskal-Wallis Test	`vars`, `group_vars`	`posthoc`, `posthoc_method`
`friedman`	Friedman Test	`vars`	`posthoc`, `posthoc_method`
`f_nway`	N-way ANOVA	`vars`, `group_vars`	`posthoc`, `posthoc_method`
`f_nway_rm`	N-way Mixed Repeated Measures ANOVA	`vars`, `group_vars`	`posthoc`, `posthoc_method`
`linearr`	Linear Regression	`vars`	
`hier_linearr`	Hierarchical Linear Regression	`vars`	
`logisticr`	Logistic Regression	`vars`	
`oneway_ancova`	One-way ANCOVA	`vars`, `group_vars`	
`rm_ancova`	One-way Repeated Measures ANCOVA	`vars`	
`cronbach`	Calculating Cronbach's Alpha	`vars`	

Also the `statmanager-kr` provides two posthoc methods. It can be run by providing the key

of the `posthoc_method` parameter as follows:

Key of `posthoc_method`	Method
`bonf`	Bonferroni Correction
`tukey`	Tukey HSD

Visualization

A figure is automatically generated for the results of the analysis when a `.figure()` is run as a chain method against a `.progress()`.

```
# Running: check the difference in weight by sex with figure
sm.progress(method = 'ttest_ind', vars = 'weight', group_vars = 'sex').figure()
```

Acknowledgements

Author declares no conflicts of interests.

References

Moon, S. M. (2020). *Statistics for the social sciences: Moving toward an integrated approach*. Cognella Academic Publishing. ISBN: 978-1516519613

Murtonen, M., & Lehtinen, E. (2003). Difficulties experienced by education and sociology students in quantitative methods courses. *Studies in Higher Education*, *28*(2), 171–185. https://doi.org/10.1080/0307507032000058064

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. *Proceedings of the 9th Python in Science Conference*, *57*(61), 10–25080. https://doi.org/10.25080/Majora-92bf1922-011

Vallat, R. (2018). Pingouin: Statistics in python. *Journal of Open Source Software*, *3*(31), 1026. https://doi.org/10.21105/joss.01026

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., & others. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2