

Manual de la Biblioteca **Pingouin** en Python

Contents

1 Introducción

La librería Pingouin es un paquete estadístico para Python, diseñado para facilitar el análisis de datos de manera intuitiva, clara y con funciones de alto nivel. Proporciona herramientas para estadística descriptiva, pruebas de hipótesis, medidas de efecto, correlaciones, ANOVAs, regresión y análisis de confiabilidad. Su enfoque está en ser ligera, fácil de usar y producir resultados listos para ser interpretados.

2 Características Generales

- Escrita en Python puro, sin dependencias pesadas.
- Produce resultados en formato de tabla (`DataFrame`) de pandas.
- Funciones de estadística avanzada integradas en pocas líneas de código.
- Compatible con librerías como NumPy, SciPy y pandas.
- Resultados listos para informes científicos (incluyendo valores p, intervalos de confianza y tamaños de efecto).

3 Funciones Principales por Categoría

3.1 1. Estadísticos Descriptivos

- `pg.describe()`: genera estadísticas descriptivas de un conjunto de datos, como media, desviación estándar, error estándar, intervalo de confianza, asimetría y curtosis.
- **Ejemplo:**

```
pg.describe(data, dv='Nota', group='Grupo')
```

Resultados: tamaño de la muestra, media, desviación típica, IC95%, asimetría y curtosis por grupo.

3.2 2. Pruebas de Normalidad y Homogeneidad

- `pg.normality()`: aplica la prueba de Shapiro-Wilk u otras pruebas para verificar si los datos siguen una distribución normal.
- `pg.homoscedasticity()`: realiza la prueba de Levene o Bartlett para comprobar la igualdad de varianzas entre grupos.
- **Ejemplo:**

```
pg.normality(data, dv='Nota', group='Grupo')
pg.homoscedasticity(data, dv='Nota', group='Grupo')
```

3.3 3. Pruebas de Hipótesis

- `pg.ttest()`: prueba t de Student para muestras independientes o relacionadas, con opción de corrección de Welch.
- `pg.mwu()`: prueba de Mann-Whitney U (alternativa no paramétrica a la t-test).
- `pg.anova()`: análisis de varianza de un factor.
- `pg.pairwise_ttests()`: comparaciones múltiples con corrección por error tipo I.
- **Ejemplo:**

```
pg.ttest(dataA, dataB, correction='auto')
```

3.4 4. Medidas de Efecto

- `pg.compute_effsize()`: calcula el tamaño del efecto entre dos grupos (Cohen's d, Hedges' g, Glass' delta, etc.).
- `pg.partial_eta_sq()`: calcula el eta cuadrado parcial para ANOVA.
- **Ejemplo:**

```
pg.compute_effsize(dataA, dataB, eftype='cohen')
```

3.5 5. Correlaciones

- `pg.corr()`: calcula la correlación entre dos variables con diferentes métodos (Pearson, Spearman, Kendall).
- `pg.partial_corr()`: correlación parcial controlando variables adicionales.
- `pg.rcorr()`: calcula una matriz completa de correlaciones.
- **Ejemplo:**

```
pg.corr(data['X'], data['Y'], method='pearson')
```

3.6 6. ANOVA y Modelos Lineales

- `pg.anova()`: ANOVA de un factor o de medidas repetidas.
- `pg.rm_anova()`: ANOVA de medidas repetidas.
- `pg.mixed_anova()`: ANOVA de medidas mixtas (intra e inter sujetos).
- **Ejemplo:**

```
pg.anova(data=data, dv='Nota', between='Grupo')
```

3.7 7. Fiabilidad y Consistencia

- `pg.cronbach_alpha()`: calcula el alfa de Cronbach para medir consistencia interna.
- `pg.intraclass_corr()`: calcula la correlación intraclase para fiabilidad entre evaluadores.
- Ejemplo:

```
pg.cronbach_alpha(data)
```

3.8 8. Otras Funcionalidades Útiles

- `pg.rm_corr()`: correlación de medidas repetidas.
- `pg.logistic_regression()`: regresión logística.
- `pg.linear_regression()`: regresión lineal simple o múltiple.
- `pg.power_ttest()`: cálculo de potencia estadística en pruebas t.
- `pg.multicomp()`: corrección de comparaciones múltiples (Bonferroni, Holm, FDR, etc.).

4 Conclusiones

La librería Pingouin concentra en un solo paquete las funciones estadísticas más comunes para la investigación científica. Su diseño orientado a `pandas` y la facilidad de obtener resultados en tablas lo convierten en una herramienta ideal para psicología, neurociencia, biología, educación y áreas aplicadas que requieran análisis estadístico claro y reproducible.