

## Librerías y funciones de análisis estadístico

### Tema: Estadística inferencial

#### Guía de referencia Estadística Inferencial

##### Importar las librerías

```
from scipy import stats
import statsmodels.api as sm
```

##### Funciones básicas - SciPy

```
stats.sem
Error estándar de la muestra
stats.levene
Prueba de Levene
stats.probplot
Cuantiles de gráfica P-P
```

##### Funciones básicas - statsmodels

sm.stats.proportion_confint	Intervalo de confianza de una proporción
sm.stats.DescrStatsW(s).zconfint_mean()	Intervalo de confianza de una media
sm.stats.binom_test	Test de un experimento de Bernoulli
sm.stats.proportions_ztest	z-test de una proporción
sm.stats.ztest	z-test de una media
sm.stats.ttest_ind	t-test de muestras independientes
sm.stats.CompareMeans(x,y).ztest_ind()	z-test de muestras independientes
sm.graphics.gofplots.qqplot	Gráfica q-q

##### Distribuciones de probabilidad

###### Distribución T de Student

stats.distributions.t.cdf	Función de densidad acumulada
stats.distributions.t.ppf	Función de punto porcentual

###### Distribución normal

stats.distributions.norm.cdf	Función de densidad acumulada
stats.distributions.norm.ppf	Función de punto porcentual

##### Tests paramétricos

stats.ttest_1samp	t-test de la media de 1 muestra
stats.ttest_ind	t-test de 2 muestras independientes
stats.ttest_rel	t-test de 2 muestras emparejadas
stats.f_oneway	ANOVA unidireccional

##### Tests no paramétricos

stats.mannwhitneyu	Test de Mann-Whitney
stats.wilcoxon	Test de Wilcoxon
stats.kruskal	Test de Kuskal-Wallis
stats.friedmanchisquare	Test de Friedman

### Tema: Modelado estadístico

#### Guía de referencia: Correlación y regresión

##### Importar las librerías

```
import pandas as pd
import numpy as np
from scipy import stats
import statsmodels.api as sm
import statsmodels.formula.api as smf
from sklearn.linear_model import
LinearRegression
```

##### Correlación básica

```
np.corrcoef
pd.DataFrame.corr
pd.Series.corr
pd.DataFrame.corrwith
```

##### Correlación con significancia

###### Correlaciones con p-valor

```
stats.pearsonr    r de Pearson
stats.spearmanr   ρ de Spearman
stats.kendalltau   τ de Kendall
```

##### Regresión lineal simple

###### Regresión lineal simple con SciPy

```
pend, interc, corr, p_valor,
error_std = stats.linregress(x, y)
```

###### Modelo usando el método mínimos cuadrados ordinarios (OLS)

```
model = sm.OLS(y, x)
```

##### Regresión con statsmodels formula

###### Regresión lineal con mínimos cuadrados ordinarios

```
model = smf.ols(formula="V_D ~ V_I1 + V_I2 + V_I3",
data=datos).fit()
```

###### Regresión logística

```
model = smf.logit(formula="V_D ~ V_I1 + V_I2",
data=datos).fit()
```

##### Ejecución del modelo de regresión

###### Calcula los parámetros del modelo

```
model.fit()
```

###### Muestra los datos del modelo

```
model.summary()
```

Hacer predicciones usando datos nuevos predictions = model.predict(x\_nuevos)

##### Regresión lineal múltiple

###### Con 2 variables independientes

```
model = sm.OLS.from_formula
("V_DEP ~ V_I1 + V_I2",
data=datos)
```

###### Con 3 variables independientes

```
model = sm.OLS.from_formula
("V_D ~ V_I1 + V_I2 + V_I3",
data=datos)
```

##### Regresión lineal con scikit-learn

```
lm= LinearRegression()
```

###### Calcular los parámetros del modelo

```
lm.fit(x, y)    Ajustar el modelo
lm.coef_       Coeficientes
lm.intercept_  Intercepto
lm.score(x, y) Coeficiente
de determinación
```

##### Regresión logística

###### Regresión logística

```
model = sm.GLM.from_formula
("V_DEP ~ V_IND1",
family=sm.families.Binomial(),
data=da)
```

###### Regresión logística con 2 variables independientes

```
model = sm.GLM.from_formula
("V_DEP ~ V_IND1 + V_IND2",
family = sm.families.Binomial(),
data = da)
```

##### Regresión lineal con scikit-learn

Hacer predicciones usando nuevos datos  
y\_pred = lm.predict(X\_nuevos)