DEPARTAMENTO
DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

# TP Estadística Clásica

5 de septiembre de 2017 Ciencia de Datos


| Integrante | LU | Correo electrónico |
|---|---|---|
| Cairo, Gustavo Juan | 89/13 | gj.cairo@gmail.com |
| Miño, Santiago | 866/11 | santiago.m92@hotmail.com |
| Zimenspitz, Ezequiel | 155/13 | ezeqzim@gmail.com |


| Instancia | Docente | Nota |
|---|---|---|
| Primera entrega | | |
| Segunda entrega | | |




**Facultad de Ciencias Exactas y Naturales**
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2160 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

# 1. Introducción

Se tienen doce atletas que entrenan para una competencia de 100m llanos. El entrenamiento se realiza aún en condiciones climáticas adversas. A partir de un archivo que contiene los tiempos en segundos de cada atleta para un entrenamiento en un día soleado, en un día nublado, y en un día de lluvia intensa, realizaremos un análisis de los datos por medio de tests y gráficos para presentar conclusiones a partir de estos estudios.

# 2. Exploración preliminar y gráficos

Como primer experimento, levantamos los datos con `numpy` y realizamos un scatter plot para visualizar la distribución de los mismos.

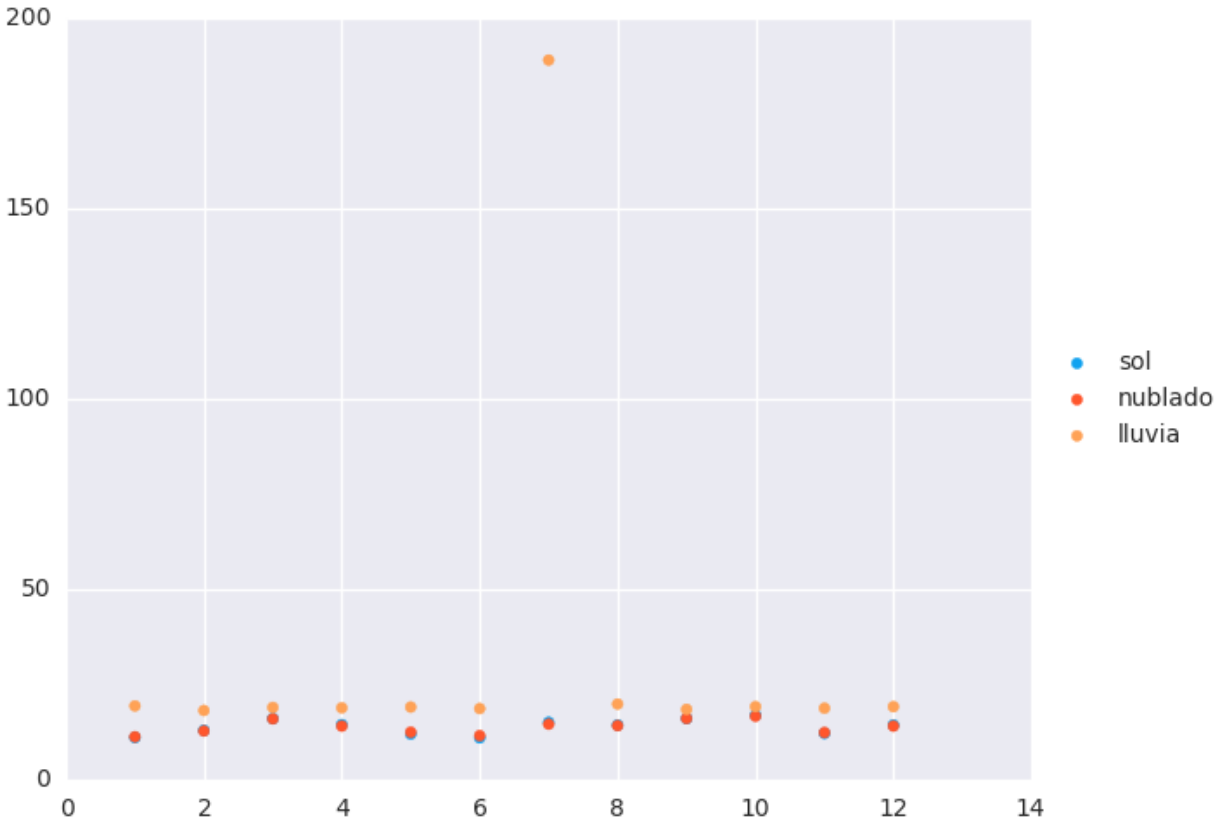


Figura 1: Scatter plot del dataset crudo

En este primer gráfico notamos un punto muy aislado del resto (atleta 7, valor de lluvia). Pensamos que es debido a un error humano al armar el dataset. Como posibilidad para mitigar el problema, nos planteamos tres alternativas:

1. Dejarlo y asumir que no hubo errores humanos: probablemente incorrecto, dado que el ruido es demasiado grande y obvio respecto de los otros valores.
2. Modificar el valor incorrecto a uno razonable, agregando una coma en la posición decimal que nos parecía coherente observando la forma de los otros datos.
3. Eliminar el registro, lo cual tal vez hubiera tenido sentido en un dataset más grande (ya que modificando el dato manualmente podemos estar en realidad asignando un nuevo valor erróneo). El problema es que, con tan solo 12 datos, eliminar uno de ellos implica perder gran parte de las muestras, por lo que decidimos no seguir por este camino.

Creemos que la opción 2 es la mejor, porque (por lo explicado arriba), seguramente haya habido un error, y eliminar la muestra entera implicaría perder gran cantidad de datos. Por esta razón, realizamos dicha modificación en el dataset antes de seguir adelante con los experimentos.

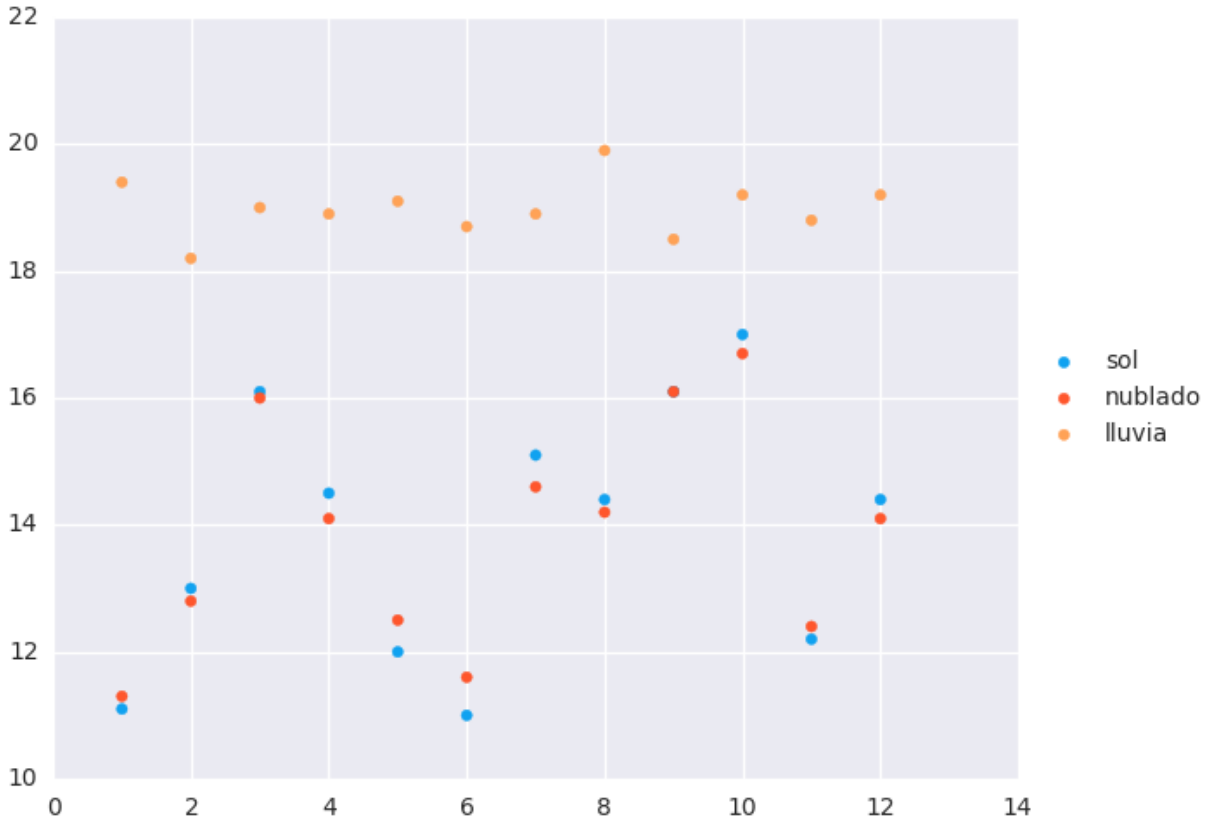


Figura 2: Scatter plot del dataset luego de las modificaciones

# 3. Tests

En esta sección presentaremos los tests que nos parecen adecuados para el análisis de los datos. Para cada uno de ellos, vamos a explicar por qué decidimos integrarlo (o no) para el análisis, y en la próxima sección señalaremos las conclusiones a partir de dicho estudio.

## 3.1. Test Binomial

Descartamos este test pues los datos no responden a experimentos binarios.

## 3.2. T-test de Student

Nos gustaría realizar tests de muestras apareadas para analizar la relación de los tiempos de cada corredor en los distintos climas.

### 3.2.1. Hipótesis para T-test

Para realizar los T-test de Student, debemos validar las siguientes hipótesis:

- *Las variables están distribuidas normalmente*

  Con el test de Shapiro tenemos como $H_0$: Las variables tienen distribución normal.

| Variable | P-Value | Rechazo $H_0$ |
|---|---|---|
| Sol | 0.475 | No |
| Nublado | 0.529 | No |
| Lluvia | 0.976 | No |

Por lo tanto, no podemos rechazar que las variables no tengan distribución normal.

- *Las variables tienen igual varianza*

  Con el F-test tenemos como $H_0$: Las variables tienen igual varianza.

  Este test chequea la igualdad de las varianzas, por lo tanto haremos un análisis a dos colas: queremos analizar si la varianza de la variable A es mayor a la de B, y también el caso en que la varianza de A es menor a la de B.

| Variable vs Variable | P-Value | Rechazo $H_0$ |
|---|---|---|
| Sol vs Lluvia | 1.47e-05 | Sí |
| Nublado vs Lluvia | 5.146e-05 | Sí |
| Sol vs Nublado | 0.687 | No |

- *Las variables son independientes*

  Dada la naturaleza del dataset, vamos a asumir que los datos son independientes.

#### 3.2.2. Muestras independientes

Descartamos realizar t-tests para muestras independientes debido a que éste chequea si dos muestras independientes tienen la misma media. Sin embargo, en este problema particular, sabemos que las muestras tienen un factor común: el corredor. Por lo tanto resulta más robusto utilizar la noción del factor común y utilizar el t-test para muestras apareadas.

No obstante, como la hipótesis de la varianza no se cumple para los pares que incluyen lluvia, no podemos realizar los tests de muestras apareadas con éstos. Por este motivo, decidimos correr el *Welch t-test* (test de muestras independientes para muestras con no necesariamente la misma varianza).

#### 3.2.3. 1 muestra

Descartamos el t-test para muestra única debido a que estamos comparando muestras entre sí y no encontramos la necesidad de comparar una muestra contra un valor fijo.

#### 3.2.4. Muestras apareadas

Dado que tenemos un factor común entre las muestras para cada clima, usamos t-test para muestras apareadas para comparar los distintos climas de a pares *que cumplen las hipótesis*, ésto es, sólo el par sol-nublado. Para los otros pares, utilizaremos Welch t-test.

Con este test tenemos como $H_0$: Las muestras vienen de distribuciones con la misma media.

| Variable vs Variable | P-Value | Rechazo $H_0$ |
|---|---|---|
| Sol vs Nublado | 0.688 | No |

#### 3.2.5. Welch T-test

Con este test tenemos como $H_0$: Las muestras vienen de distribuciones con la misma media.

| Variable vs Variable | P-Value | Rechazo $H_0$ |
|---|---|---|
| Sol vs Lluvia | 2.03e-06 | Sí |
| Nublado vs Lluvia | 4.403e-07 | Sí |

## 3.3. Test de Permutaciones

Para el planteo de este test consideramos la siguiente $H_0$: Los tiempos de los corredores en climas soleados son mayores a los tiempos de los mismos en climas lluviosos.

Para testearlo, pensamos en *shufflear* las etiquetas de sol y lluvia entre el total de etiquetas, pero de esta manera perdemos la información relacionada con cada atleta. Es decir, para cada atleta tenemos un par de tiempos (tiempo de corrida en un día lluvioso, y tiempo de corrida en un día soleado), pero si simplemente reasignamos etiquetas al azar, perdemos la relación *atleta→(sol, lluvia)*.

Para utilizar la información de la que disponemos y hacer más robusto nuestro test, nos conviene explotar esta información de contexto y mantener la relación de par de tiempos para cada atleta. Para esto, pensamos a las permutaciones como un conjunto de flips (con una probabilidad de 0.5) entre la etiqueta sol y lluvia para cada atleta. De esta manera mantenemos la información de cada competidor durante todo el test, ya que sólo reasignaríamos, para cada atleta, su tiempo en día de lluvia como tiempo en día de sol y viceversa.

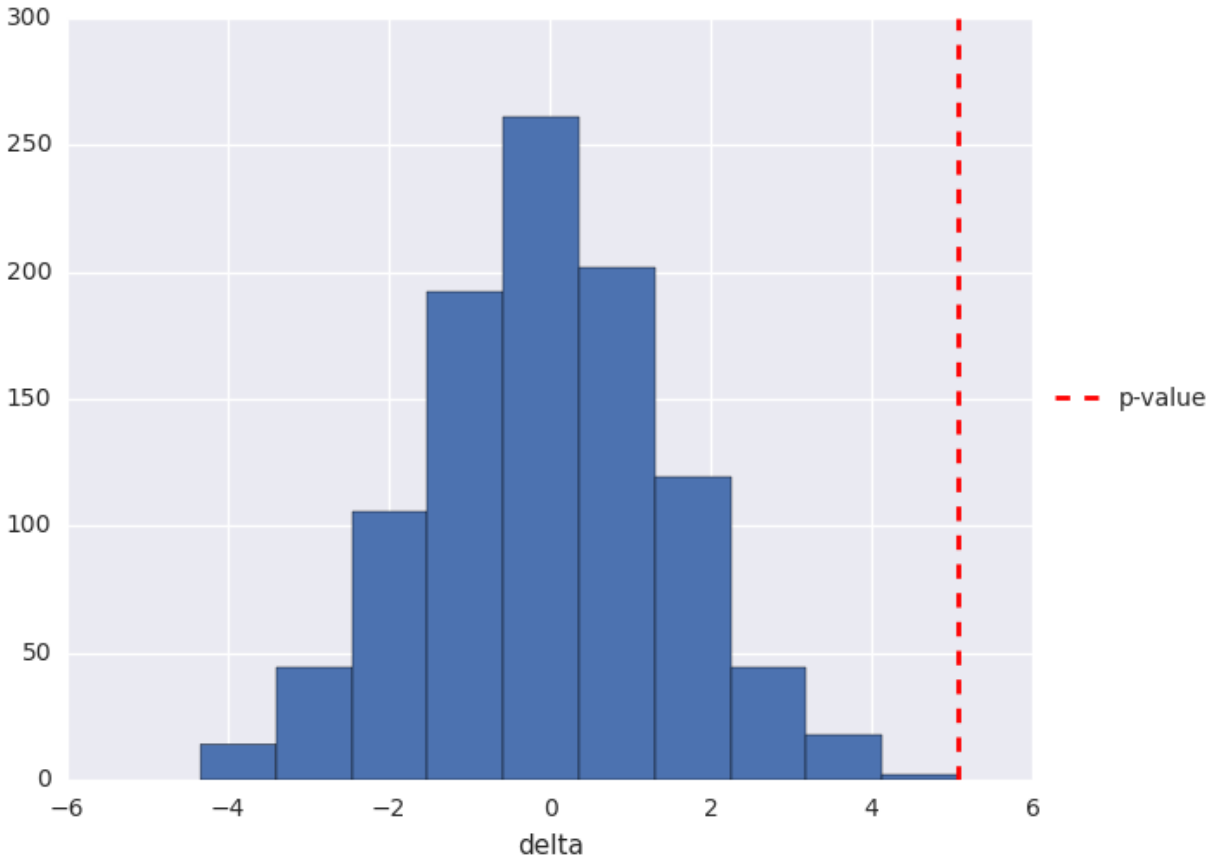


Figura 3: Test de permutaciones. P-Value = 0

Como puede observarse, el *p-valor* es cero, por lo que rechazamos $H_0$.

## 3.4. Test de Wilcoxon

Dado que no pudimos realizar el t-test de muestras apareadas para los pares de variables que incluían a lluvia porque no cumplen las hipótesis de t-test, tenemos la posibilidad de correr en su lugar el test de Wilcoxon.

Con este test tenemos como $H_0$: Las muestras vienen de la misma distribución.

| **Variable vs Variable** | **P-Value** | **Rechazo $H_0$** |
|---|---|---|
| Sol vs Lluvia | 0.002 | Sí |
| Nublado vs Lluvia | 0.002 | Sí |

### 3.5. Correlación de Pearson

Con este test medimos el coeficiente de correlación de las muestras. Un valor cercano a 1 indica alta correlación, cercano a -1 alta anti-correlación, cercano a 0 no están correlacionados. El *p-valor* indica la probabilidad de que el índice de correlación haya sido resultado del azar, es decir, cuanto más bajo, mayor es la probabilidad de que el índice de correlación sea significativo.

| **Variable vs Variable** | **Índice de correlación** | **P-Value** | **Correlacionadas** |
|---|---|---|---|
| Sol vs Nublado | 0.99 | 5.045e-10 | Altamente correlacionados |
| Sol vs Lluvia | 0.052 | 0.871 | No correlacionados |
| Nublado vs Lluvia | 0.042 | 0.896 | No correlacionados |

### 3.6. Varianzas

Presentamos los valores de las varianzas para las muestras.

| **Variable** | **Varianza** |
|---|---|
| Sol | 3.762 |
| Nublado | 2.94 |
| Lluvia | 0.175 |

## 4. Conclusiones

- *Los atletas son más lentos en días de lluvia que en días soleados*

  Podemos extraer esta conclusión a partir de los tests realizados.

  Como primer análisis, el Welch T-test entre las variables de Sol y Lluvia, nos permitió rechazar la hipótesis de que las distribuciones tienen la misma media. Por otro lado, el test de permutaciones nos permitió rechazar la hipótesis de que los tiempos de los corredores en climas soleados son mayores a los de climas lluviosos. Por lo tanto, como las medias son distintas y la velocidad de los corredores no es mayor con lluvia (si no con sol), podemos deducir que los atletas son más lentos en días de lluvia.

- *El cielo nublado no influye en los tiempos de los atletas*

  No tenemos información suficiente para rechazar ni extraer esta conclusión.

  Considerando que los tiempos normales son los medidos en climas soleados, el test de muestras apareadas entre las variables Sol y Nublado nos muestra que no podemos rechazar que las muestras tengan la misma media. Por otro lado el índice de correlación de Pearson nos dice que las muestras están muy altamente correlacionadas. Por lo tanto, la distribución de ambas variables sería en principio muy parecida.

  Trabajando con los datos que nos dieron, podríamos concluir que el clima nublado no afecta a la velocidad de los corredores, pero no tenemos evidencia suficiente para mostrar que Sol y Nublado tienen la misma media, y por lo tanto, que son prácticamente indistinguibles una de la otra. Si la tuviéramos, podríamos extraer esta conclusión ya que significaría que los tiempos de los atletas son casi idénticos haya sol o esté nublado: las nubes no influyen.

- *La velocidad en días de lluvia es independiente del atleta*

  Podemos extraer esta conclusión a partir de los tests realizados.

  La varianza de la variable Lluvia es mucho menor a la de las otras variables, lo que nos da un indicio de que, no importa qué tan bueno o qué tan malo fuera un atleta con sol o nubes, esa diferencia va a ser mucho menor cuando llueva.

  Por otro lado, las posiciones relativas de los atletas no se mantienen cuando llueve (el índice de correlación de Pearson entre Lluvia y los otros climas es muy bajo y con un *p-valor* muy alto, que indica que la poca correlación encontrada probablemente sea producto del azar), lo cual también nos habla de que no importa cuál sea la *performance* del atleta con nubes o sol, poco tendrá que ver con sus tiempos cuando llueva.

  Además, por el test de Wilcoxon, podemos decir que es altamente probable que tanto Lluvia y Sol como Lluvia y Nublado tengan distintas distribuciones, lo cual parecería indicar que los tiempos de los corredores en estos climas no tienen relación alguna.

- *El clima influye en la velocidad de los atletas*

  No tenemos información suficiente para rechazar ni extraer esta conclusión.

  Como vimos en el *permutation test*, los tiempos en climas soleados siempre son menores (o iguales) a los tiempos en días lluviosos, por lo que podríamos concluir que la lluvia afecta al rendimiento de los atletas.

  Por otro lado, dijimos que no podemos rechazar que el clima nublado no produce cambios de velocidad en los atletas respecto a días soleados. Si tuviésemos evidencia de esto, de aquí podríamos deducir que los atletas son más lentos en días lluviosos que en días nublados. Por lo tanto, sólo el clima de lluvia afectaría a la velocidad de los atletas.

## 5. Consejos al entrenador

Como pudimos ver, la velocidad de los atletas en días de lluvia es independiente del atleta, lo que significa que no tiene sentido entrenar en estos días, porque no importa qué tan bueno o qué tan malo un corredor sea en un día normal, esa performance no se mantendrá en un día de lluvia, y todos pasaran a tener tiempos más parecidos.

En resumen, los tiempos de los atletas no mejorarían por entrenar un día de lluvia.

## 6. Código

```
import numpy as np
import scipy as sc
import matplotlib.pyplot as plt
import seaborn as sns

atleta, sol, nublado, lluvia = np.loadtxt('tiempos.txt', skiprows =
   1, unpack = True)

# Primera vista de los datos
colors = plt.cm.rainbow(np.linspace(0, 1, 10))
ax = plt.subplot(111)
ax.scatter(atleta, sol, label = 'sol', color=colors[2])
ax.scatter(atleta, nublado, label = 'nublado', color=colors[8])
ax.scatter(atleta, lluvia, label = 'lluvia', color=colors[7])
box = ax.get_position()
ax.set_position([box.x0, box.y0, box.width * 0.9, box.height])
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.show()

# Shapiro test for normality
w, p = sc.stats.shapiro(sol)
print 'shapiro sol'
print w, p

w, p = sc.stats.shapiro(nublado)
print 'shapiro nublado'
print w, p

w, p = sc.stats.shapiro(lluvia)
print 'shapiro lluvia'
print w, p

# F-Test
FSolLluvia = np.var(sol) / np.var(lluvia)
FLluviaSol = np.var(lluvia) / np.var(sol)
FNubladoLluvia = np.var(nublado) / np.var(lluvia)
FLluviaNublado = np.var(lluvia) / np.var(nublado)
FSolNublado = np.var(sol) / np.var(nublado)
FNubladoSol = np.var(nublado) / np.var(sol)
```

```
39
40 dfSol = len(sol) - 1
41 dfLluvia = len(lluvia) - 1
42 dfNublado = len(nublado) - 1
43
44 p = sc.stats.f.sf(FSolLluvia, dfSol, dfLluvia)
45 p += sc.stats.f.cdf(FLluviaSol, dfLluvia, dfSol)
46 print 'f-test sol-lluvia dos colas'
47 print p
48
49 p = sc.stats.f.sf(FNubladoLluvia, dfNublado, dfLluvia)
50 p += sc.stats.f.cdf(FLluviaNublado, dfLluvia, dfNublado)
51 print 'f-test nublado-lluvia dos colas'
52 print p
53
54 p = sc.stats.f.sf(FSolNublado, dfSol, dfNublado)
55 p += sc.stats.f.cdf(FNubladoSol, dfNublado, dfSol)
56 print 'f-test sol-nublado dos colas'
57 print p
58
59 # Welch T-Test
60 t, p = sc.stats.ttest_ind(sol, lluvia, equal_var = False)
61 print 't-test welch sol-lluvia'
62 print t, p
63
64 t, p = sc.stats.ttest_ind(lluvia, nublado, equal_var = False)
65 print 't-test welch lluvia-nublado'
66 print t, p
67
68 # T-Test muestras apareadas
69 t, p = sc.stats.ttest_rel(sol, nublado)
70 print 't-test apareados sol-nublado'
71 print t, p
72
73 # Wilcoxon
74 t, p = sc.stats.wilcoxon(sol, lluvia)
75 print 'wilcoxon sol-lluvia'
76 print t, p
77
78 t, p = sc.stats.wilcoxon(lluvia, nublado)
79 print 'wilcoxon lluvia-nublado'
80 print t, p
81
82 # Test de correlacion Pearson
83 t, p = sc.stats.pearsonr(sol, nublado)
84 print 'pearsonr sol-nublado'
85 print t, p
86
87 t, p = sc.stats.pearsonr(sol, lluvia)
88 print 'pearsonr sol-lluvia'
89 print t, p
90
91 t, p = sc.stats.pearsonr(lluvia, nublado)
92 print 'pearsonr lluvia-nublado'
93 print t, p
94
95 # Varianzas
96 print 'varianza sol'
97 print np.var(sol)
98 print 'varianza nublado'
99 print np.var(nublado)
100 print 'varianza lluvia'
```

```
101 print np.var(lluvia)
102
103 # Permutation Test
104 # Si resta > 0 entonces lluvia > sol
105 # Si resta < 0 entonces lluvia < sol
106 delta0 = np.mean(lluvia) - np.mean(sol)
107 deltas = [delta0]
108
109 for _ in range(1001):
110   flips = np.random.choice([True, False], len(sol))
111   solC = 0
112   lluviaC = 0
113   for i in range(len(flips)):
114     if flips[i]:
115       solC += lluvia[i]
116       lluviaC += sol[i]
117     else:
118       solC += sol[i]
119       lluviaC += lluvia[i]
120   solC /= float(len(sol))
121   lluviaC /= float(len(lluvia))
122   deltas.append(lluviaC - solC)
123
124 ax = plt.subplot(111)
125 values, bins, _ = ax.hist(deltas)
126 areaTotal = sum(np.diff(bins)*values)
127 indice = next(i for i in range(len(bins)) if (lambda y : y >= delta0)
      (bins[i]))
128
129 newBins = [delta0]
130 newBins += bins[indice:]
131 newValues = values[indice:]
132
133 areaDerecha = sum(np.diff(newBins)*newValues)
134
135 print areaDerecha / areaTotal
136
137 box = ax.get_position()
138 plt.axvline(delta0, color='red', linestyle='dashed', linewidth=2,
      label = 'p-value')
139 ax.set_position([box.x0, box.y0, box.width * 0.9, box.height])
140 plt.xlabel('delta')
141 ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
142 plt.show()
```