

Guide 3 graphics

September 13, 2018

1 Computación II - 2018

Guía No. 3: Gráficos en python

Ingeniería en Estadística - Universidad de Valparaíso

Profesor: Eduardo Jorquera - eduardo.jorquera@postgrado.uv.cl

1.0.1 1. Histogramas

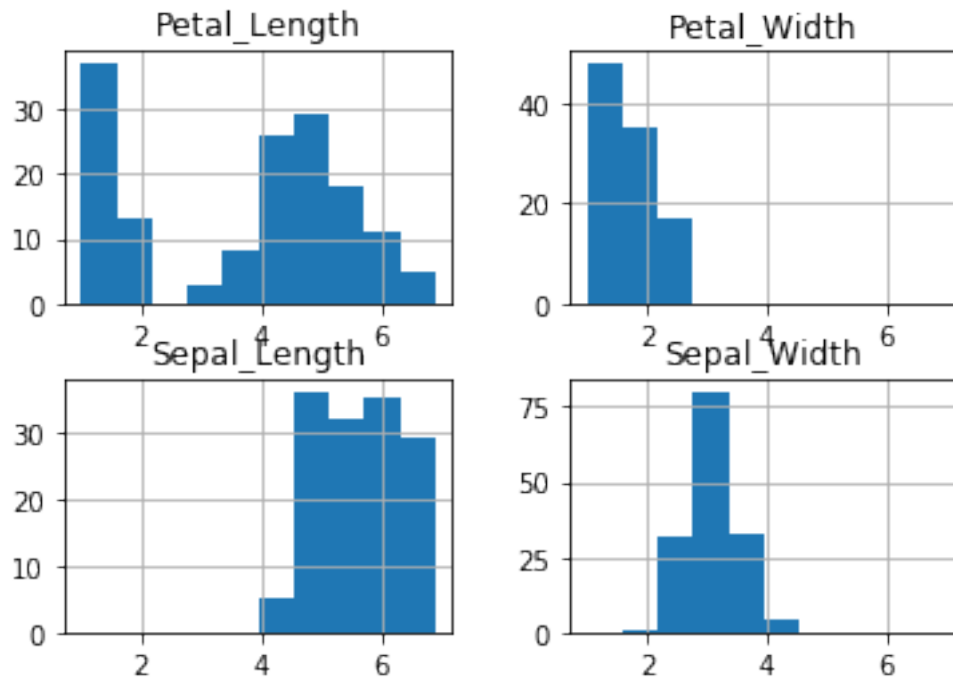
1.1 Veamos cómo hacer histogramas en Python. Un histograma reagrupa una variable continua en pequeñas “celdas” de tal forma que cada celda tiene una frecuencia distinta, y así, una probabilidad empírica distinta.

Usemos el conjunto de datos de iris:

- Pregunta: qué hace la coma (”,”) en medio de la línea 3?

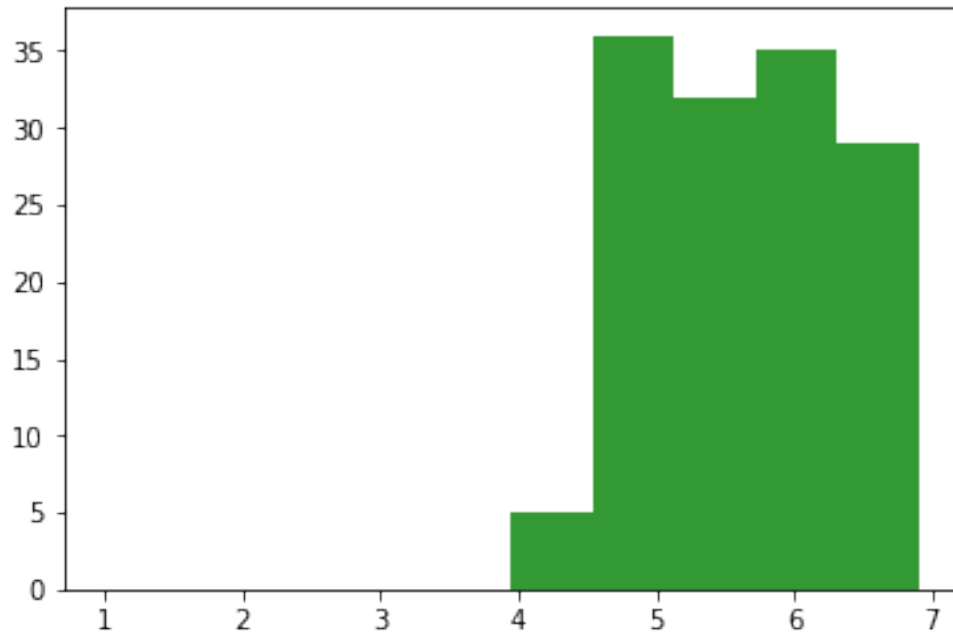
```
In [19]: import pandas as pd, numpy as np, matplotlib as plt
iris = pd.read_csv("iris.csv")
count, division = np.histogram(iris.Petal_Length)
iris.hist(bins=division)
```

```
Out[19]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f43fe8277b8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7f43fe7d07b8>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x7f43fe779e48>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7f43fe72a518>]],
              dtype=object)
```

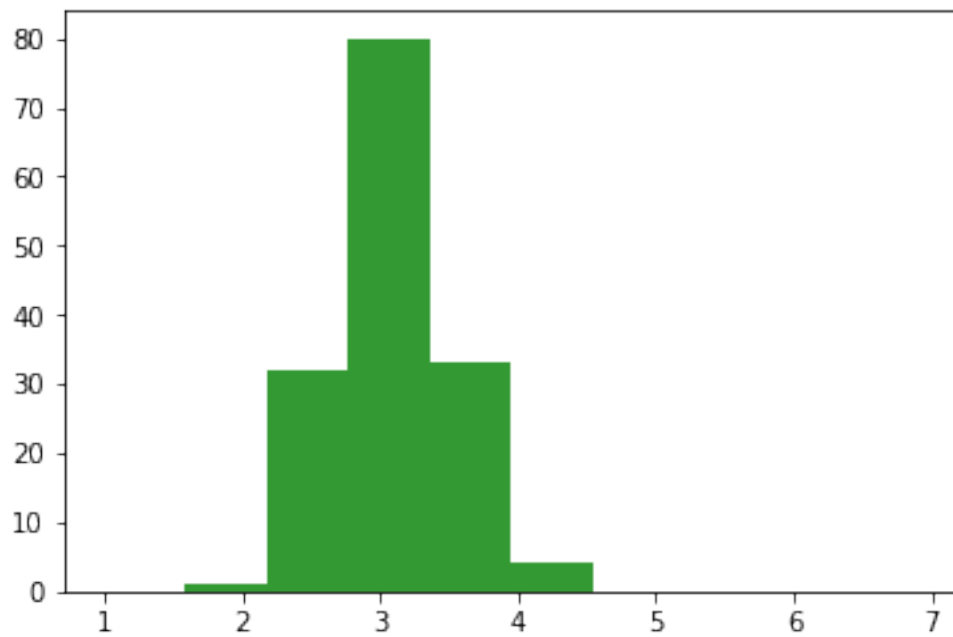


1.2 El código anterior mostró todos los histogramas, ahora ver sólo uno, debemos hacer un gráfico de tipo “subplot”: * Qué hace la tercera línea? Pruebe cambiando los números a ver qué sucede.

```
In [44]: import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_subplot(111)
x = iris.Sepal_Length
numBins = division
ax.hist(x,numBins,color='green',alpha=0.8)
plt.show()
```



```
In [32]: fig = plt.figure()
ax = fig.add_subplot(111)
x = iris.Sepal_Width
numBins = division
ax.hist(x,numBins,color='green',alpha=0.8)
plt.show()
```



1.3 Ahora grafiquemos un histograma con la densidad encima: * Qué hace la última línea del código?

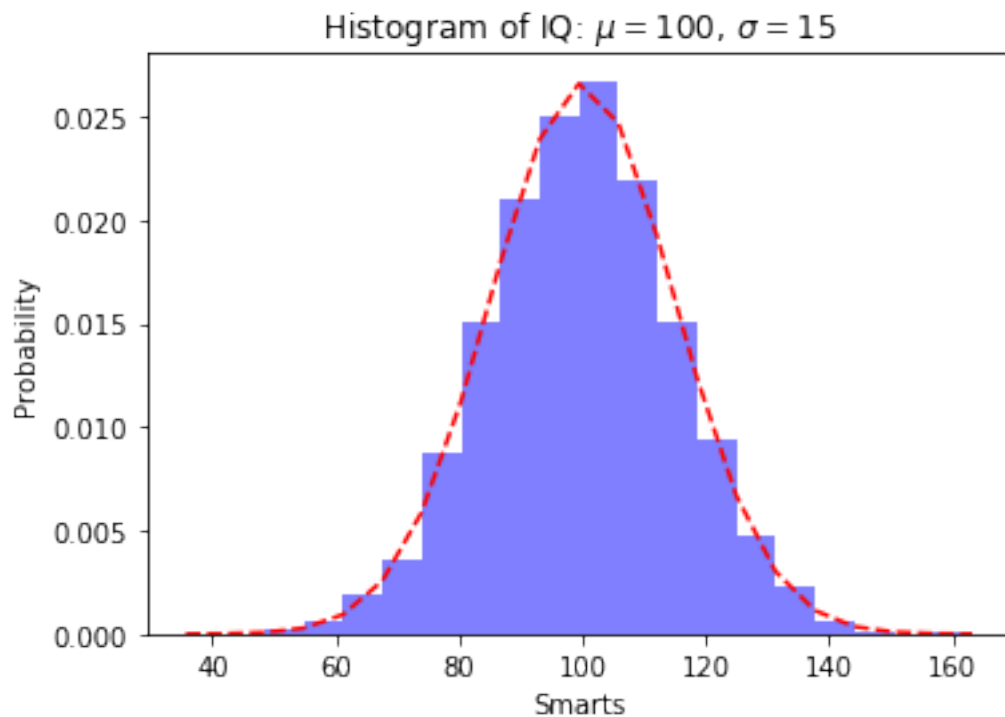
```
In [105]: import numpy as np
import matplotlib.pyplot as plt
import scipy.stats

mu = 100
sigma = 15
x = mu + sigma*np.random.randn(10000)

num_bins = 20
n, bins, patches = plt.hist(x, num_bins, density=1, facecolor='blue', alpha=0.5)

y = scipy.stats.norm.pdf(bins, mu, sigma)
plt.plot(bins, y, 'r--')
plt.xlabel('Smarts')
plt.ylabel('Probability')
plt.title(r'Histogram of IQ:  $\mu=100$ ,  $\sigma=15$ ')

plt.subplots_adjust(left=0.15) #Qué hace esta línea??
```



1.0.2 1.4 Actividad:

- Haga un histograma de la función de densidad $\text{Gamma}(2, a)$ para distintos valores de a , donde $a = 1, \dots, 5$ con $n = 1.000$. Guarde los valores generados de la distribución gamma en un data frame de pandas.

In []:

1.0.3 2. Scatterplots

2.1 Los gráficos scatter o dotplots, son útiles para contrastar dos variables, son frecuentemente usados para estudiar algún tipo de asociatividad.

Veamos un caso sencillo, utilizando el mismo conjunto de datos iris:

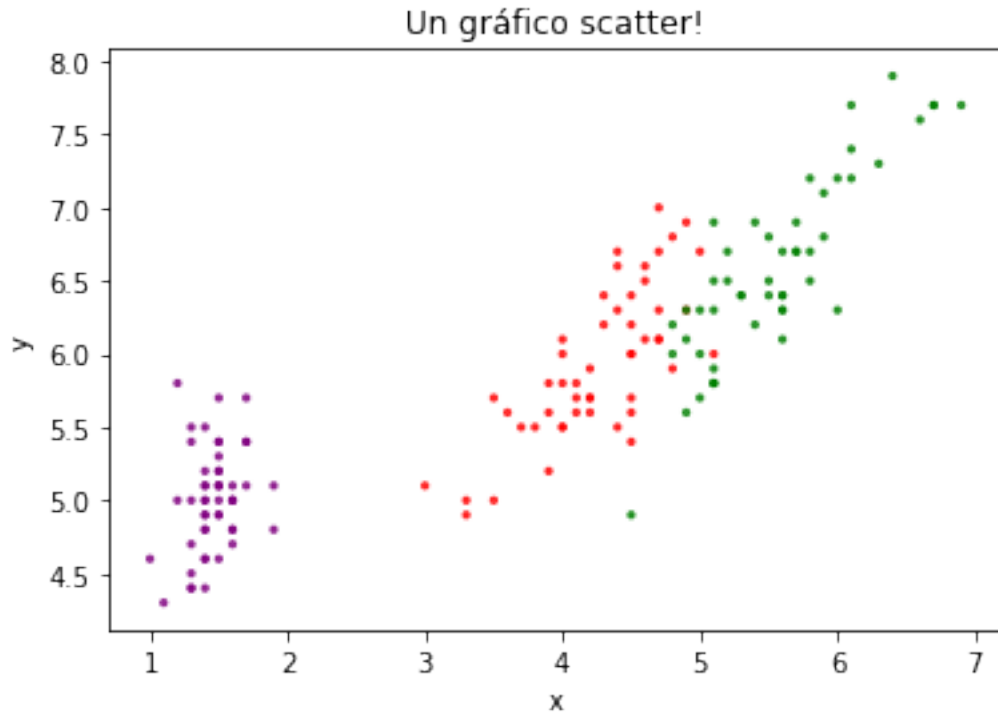
```
In [180]: import matplotlib.pyplot as plt
```

```
area = 5
```

```
color=[" " for x in range(len(iris.Sepal_Length))]
for i in range(1,len(iris.Sepal_Length)+1):
    if iris.Species[i]=='setosa':
        color[i-1]='purple'
    if iris.Species[i]=='versicolor':
        color[i-1]='red'
    if iris.Species[i]=='virginica':
        color[i-1]='green'
```

```
plt.scatter(iris.Petal_Length, iris.Sepal_Length, s=area, c=color, alpha=np.pi/4) #
plt.title('Un gráfico scatter!')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

```
Out[180]: Text(0,0.5,'y')
```



1.0.4 2.2 Actividad:

Replique el gráfico anterior, pero sólo mostrando aquellos puntos donde no esté 'virginica' como especie.

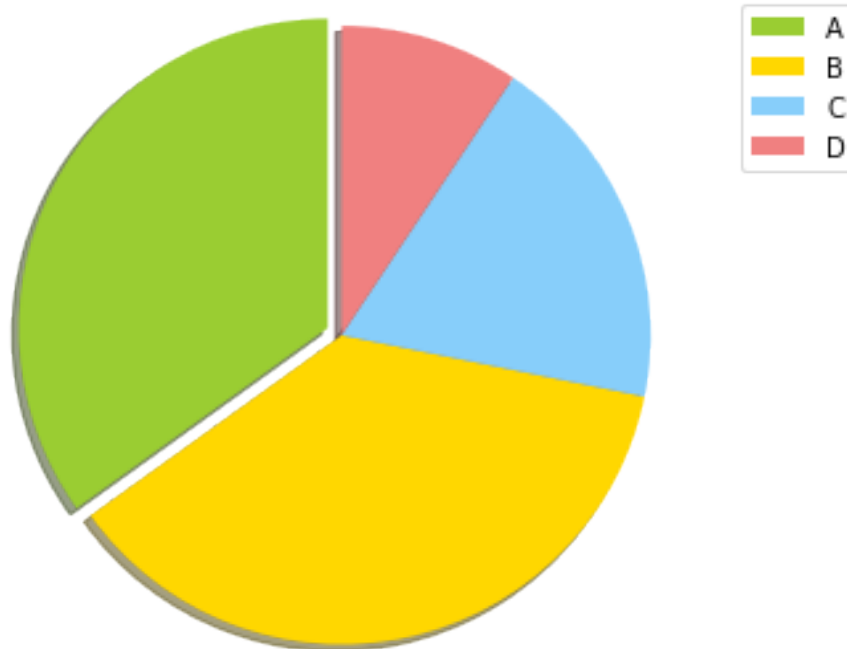
In []:

1.0.5 3. Gráficos de torta

3.1 Estos gráficos son útiles para mostrar proporciones, en el siguiente ejemplo muestra su uso normal.

```
In [182]: import matplotlib.pyplot as plt
```

```
labels = ['A', 'B', 'C', 'D']
sizes = [38.4, 40.6, 20.7, 10.3]
colors = ['yellowgreen', 'gold', 'lightskyblue', 'lightcoral']
explode = (0.05, 0, 0, 0)
patches, texts = plt.pie(sizes, colors=colors, explode=explode, shadow=True, startangle=90)
plt.legend(patches, labels, loc="best")
plt.axis('equal')
plt.tight_layout()
```



1.0.6 3.2 Actividad

Utilizando el conjunto de datos de iris, grafique la proporción de especies en un gráfico de torta.

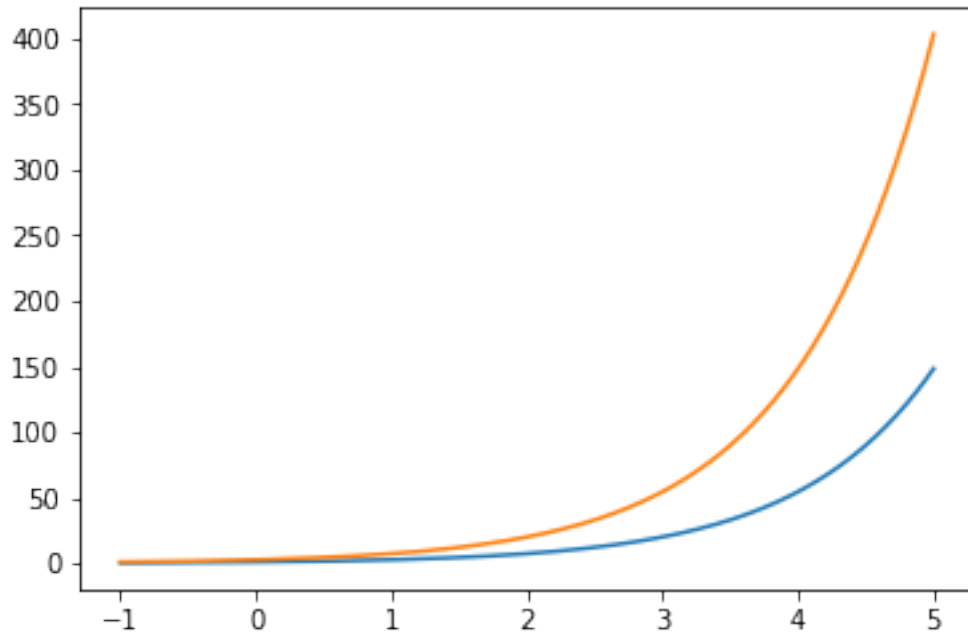
In []:

1.0.7 4 Gráficos de línea

4.1 Los gráficos de línea sirven para mostrar una variable respecto a otro eje. Es común en series de tiempo o procesos estocásticos. Un ejemplo con la función exponencial es el siguiente, donde se grafica e^x y e^{x+1} :

```
In [185]: import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(-1,5,1000)
y = np.exp(x)
plt.plot(x,y)
plt.plot(x,np.exp(x+1))
```

```
Out[185]: [<matplotlib.lines.Line2D at 0x7f43e107e390>]
```



1.0.8 4.2 Actividad

- Genere una muestra aleatoria binomial de tamaño 100, con probabilidad $p = \frac{1}{2}$, y de muestras de tamaño $n = 20$ y grafique las realizaciones de ésta en un gráfico semejante al anterior.
- Contraste este gráfico con uno de la misma distribución, pero con probabilidad $p = \frac{1}{4}$.

In []:

1.0.9 5 Gráficos de caja

5.1 Los gráficos de caja muestran la mediana, rangos intercuartílicos y datos posiblemente atípicos. En el siguiente ejemplo con simulación de normales puede verse fácilmente cómo hacer uno.

```
In [14]: import matplotlib.pyplot as plt
import numpy as np
```

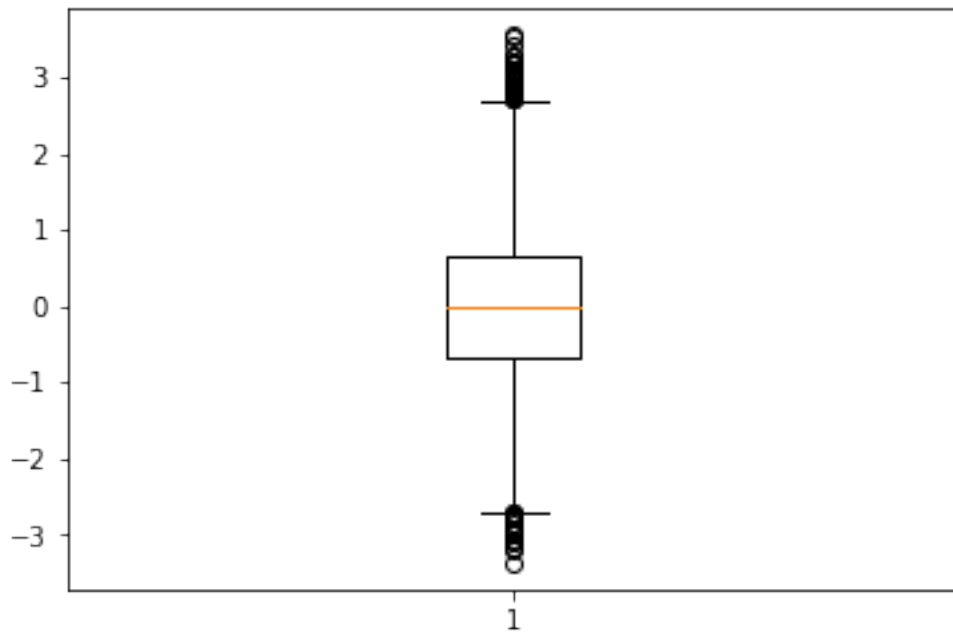
```
data=np.random.randn(10000)
```

```
# basic plot
plt.boxplot(data)
```

```
Out[14]: {'whiskers': [<matplotlib.lines.Line2D at 0x7f43fec8f048>,
<matplotlib.lines.Line2D at 0x7f43fec21eb8>],
'caps': [<matplotlib.lines.Line2D at 0x7f43fec21128>,
<matplotlib.lines.Line2D at 0x7f43fec21da0>],
'boxes': [<matplotlib.lines.Line2D at 0x7f43fec8f1d0>],
```



```
'medians': [<matplotlib.lines.Line2D at 0x7f43fec3aa20>],  
'fliers': [<matplotlib.lines.Line2D at 0x7f43fec3a6a0>],  
'means': []}
```



1.0.10 5.2 Actividad

- Haga distintos gráficos de caja (en un sólo gráfico) de cada columna para cada especie del conjunto de datos iris.
- Replique esto en R.

In []: