MAT-281 APLICACIONES DE LA MATEMÁTICA EN INGENIERÍA

Ayudantía 2: Gráfica en Python y Educated Guessing

Alberto Rubio S.

October 26, 2015

Universidad Técnica Federico Santa María

Objetivos

1. Pendientes ayudantía pasada: Funciones.
2. Gráfica en Python: Matplotlib. Ejemplo con base de datos.
3. Educated guessing.

FUNCIONES

Funciones en Python

Definición

Es una sección de un programa que realiza una determinada operación **de forma independiente al resto del código**. Se compone de:

- **Parámetros** que recibe la función como entrada.
- **Código** que corresponde a las operaciones que realiza.
- **Resultado** que es lo que entrega la función.

Observación

Las variables creadas para el uso interno de una función se denominan **variables locales** y existen sólo temporalmente.

EJERCICIO: Funciones en Python

Cree una función que determine la estimación OLS para un modelo de regresión lineal simple $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$. Posteriormente pruebe su modelo con la base de datos entregada.

Recuerdo: Recuerde los estimadores vienen dados por:

$$\min g(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2$$

SOLUCIÓN: FUNCIONES EN PYTHON

```python
def OLS(X,Y):
  import numpy as np
  CXY=0
  SXX=0
  Xmean=np.mean(X)
  Ymean=np.mean(Y)
  #Calculo numerador
  for i in range(len(X)) :
    CXY=CXY+(X[i]-Xmean)*(Y[i]-Ymean)
  #Calculo denominador
  for i in range(len(X)):
    SXX=SXX+(X[i]-Xmean)**2

  beta1=(CXY/SXX)
  beta0=(Ymean-beta1*Xmean)


  return beta1, beta0
```

OLS.py

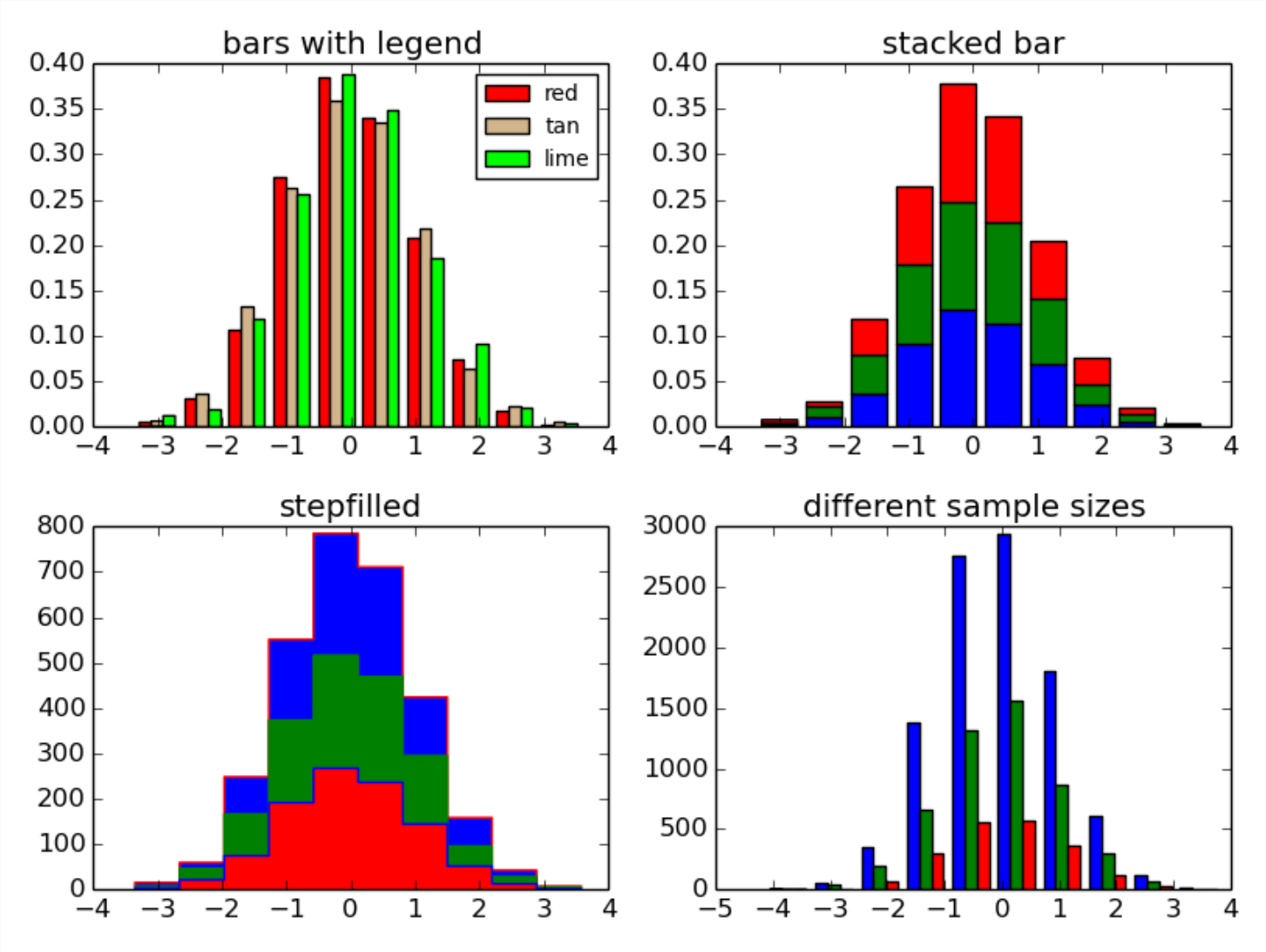
GRÁFICA EN PYTHON

¿Cómo graficar en Python?

Librería Matplotlib

- Librería gráfica de mucha utilidad en Python.
- Se pueden crear desde gráficos, histogramas, gráficos de barras, etc.
- Ver ejemplos, documentación y más información en ▸ Link
  A continuación algunos ejemplos:

Histograma


bars with legend
red
tan
lime
stacked bar
stepfilled
different sample sizes

Histograma

```python
#Librerias ocupadas
import numpy as np
import matplotlib.pyplot as plt

#Generacion datos aleatorios
n_bins = 10
x = np.random.randn(1000, 3)

fig, axes = plt.subplots(nrows=2, ncols=2)
ax0, ax1, ax2, ax3 = axes.flat

colors = ['red', 'tan', 'lime']
ax0.hist(x, n_bins, normed=1, histtype='bar', color=colors, label=colors)
ax0.legend(prop={'size': 10})
ax0.set_title('bars with legend')

ax1.hist(x, n_bins, normed=1, histtype='bar', stacked=True)
ax1.set_title('stacked bar')

ax2.hist(x, n_bins, histtype='step', stacked=True, fill=True)
ax2.set_title('stepfilled')
```

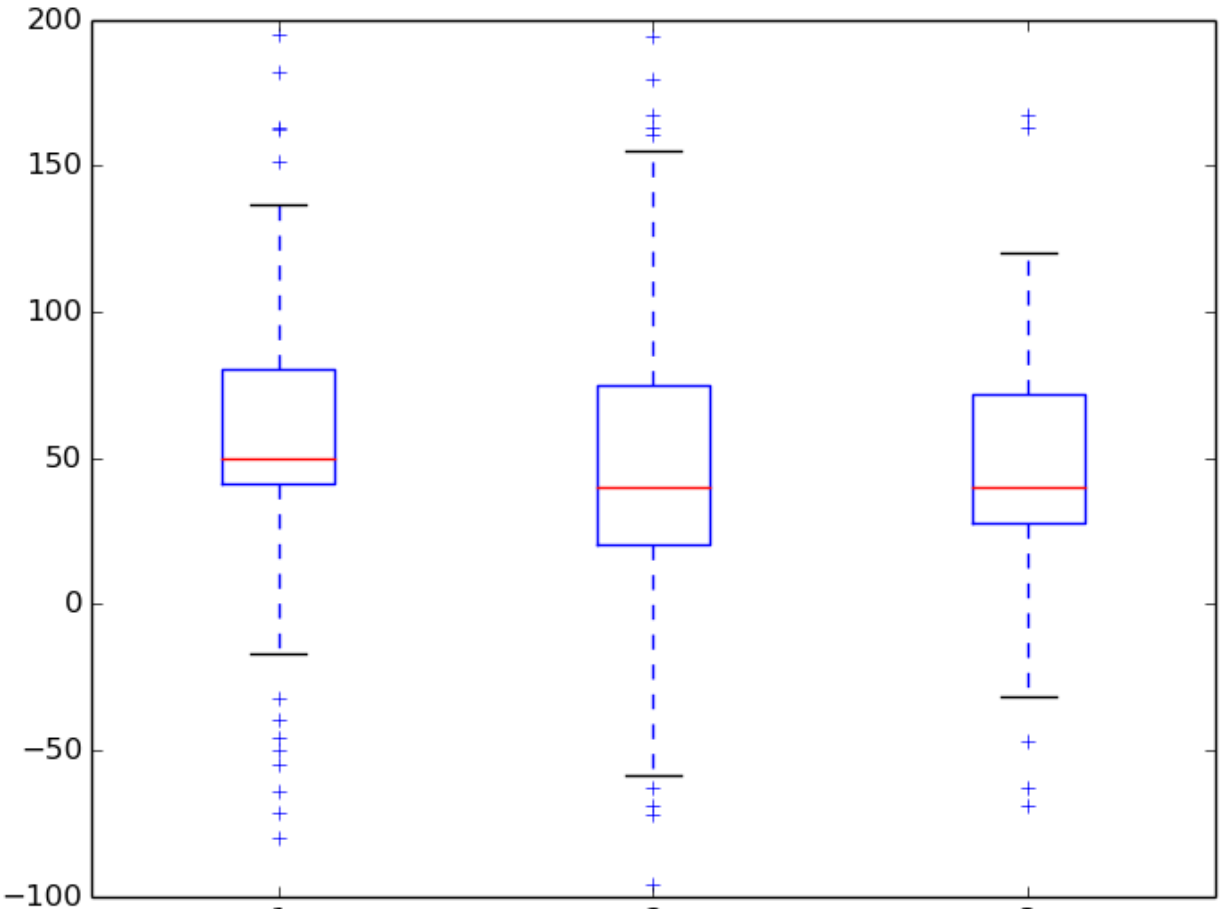
histograma.py

Histograma

```
1 ax2.set_title('stepfilled')
2
3 # Generacion histograma
4 x_multi = [np.random.randn(n) for n in [10000, 5000, 2000]]
5 ax3.hist(x_multi, n_bins, histtype='bar')
6 ax3.set_title('different sample sizes')
7
8 plt.tight_layout()
9 plt.show()
```

histograma.py

Box-plot



Box-plot

```
from pylab import *

# fake up some data
spread= rand(50) * 100
center = ones(25) * 50
flier_high = rand(10) * 100 + 100
flier_low = rand(10) * -100
data =concatenate((spread, center, flier_high, flier_low), 0)

# basic plot
boxplot(data)

# notched plot
figure()
boxplot(data,1)

# change outlier point symbols
figure()
boxplot(data,0,'gD')
```

boxplot.py

Box-plot

```
1  # don't show outlier points
2  figure()
3  boxplot(data,0,'')
4
5  # horizontal boxes
6  figure()
7  boxplot(data,0,'rs',0)
8
9  # change whisker length
10 figure()
```

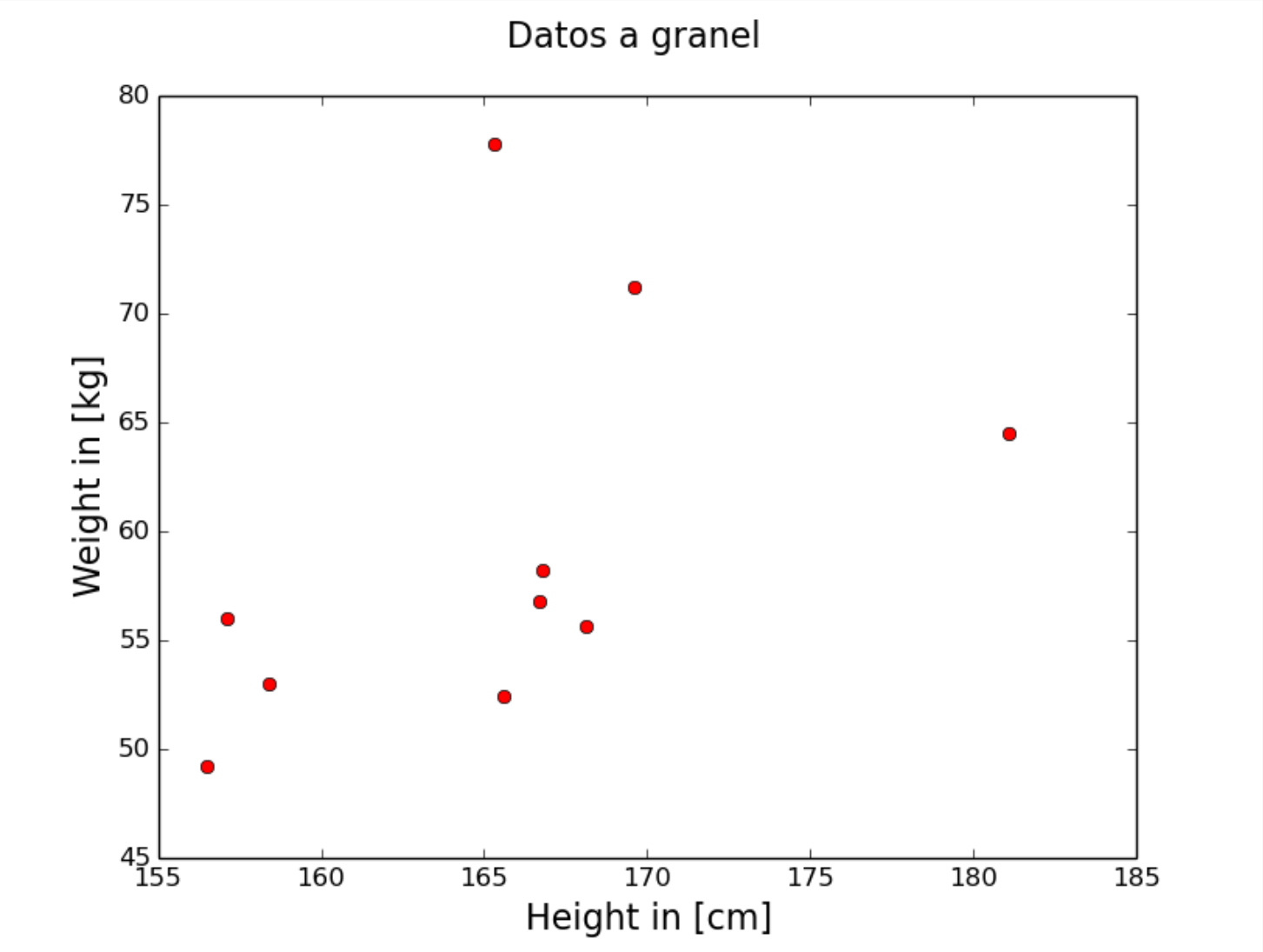
boxplot.py

Nota: Observar que éste código genera 4 tipos de box-plot.

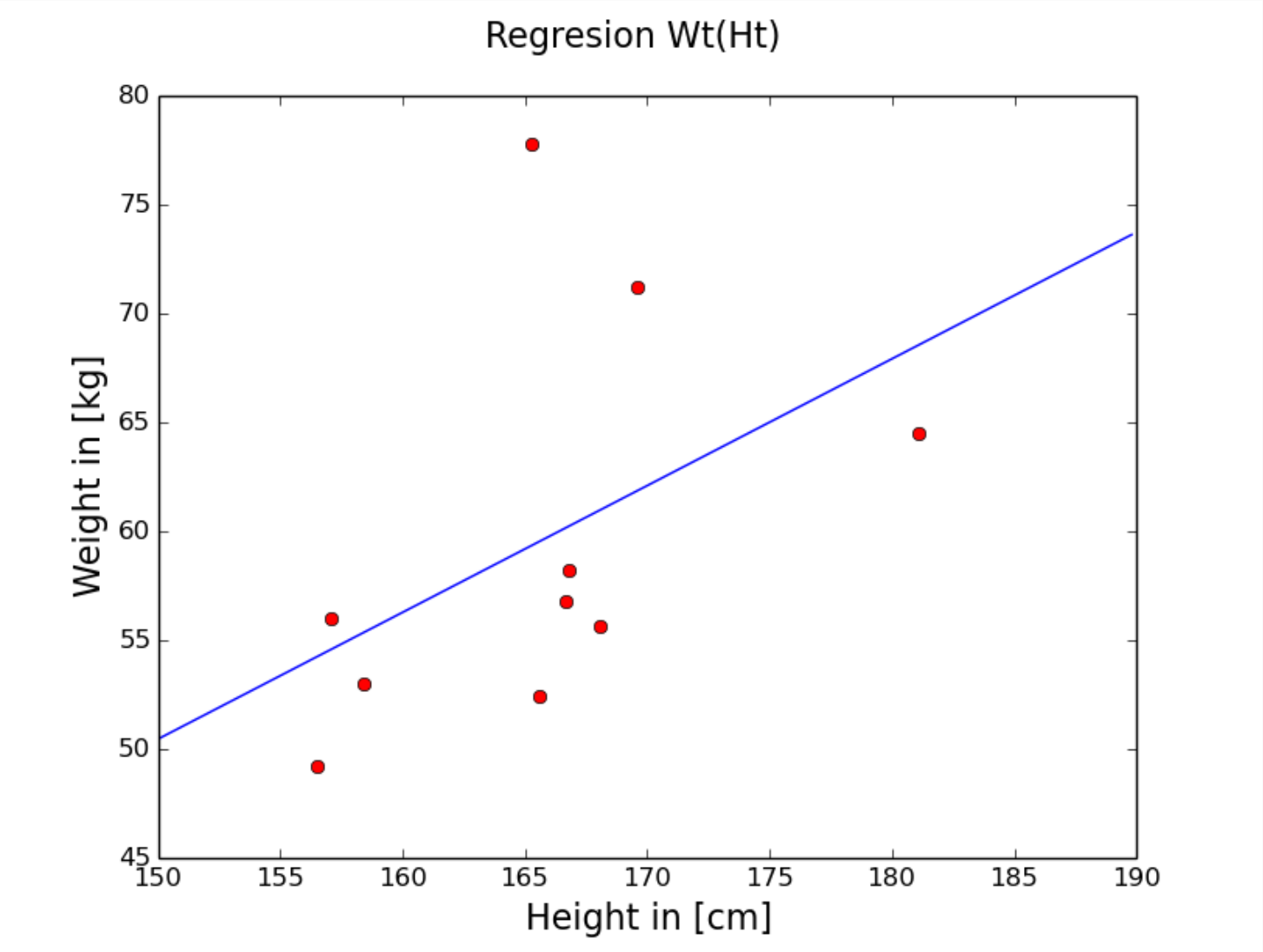
EJERCICIO: Base de datos regresión

Para la base de datos anterior grafique la nube de puntos inicial y el ajuste de la regresión.

SOLUCIÓN: BASE DE DATOS REGRESIÓN



SOLUCIÓN: BASE DE DATOS REGRESIÓN



SOLUCIÓN: BASE DE DATOS REGRESIÓN

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  import OLS
4
5  Ht=np.array
       ([169.6,166.8,157.1,181.1,158.4,165.6,166.7,156.5,168.1,165.3])
6  Wt=np.array([71.2,58.2,56.0,64.5,53.0,52.4,56.8,49.2,55.6,77.8])
7
8
9  print OLS.OLS(Ht,Wt)
10
11
12 ## Segunda Parte. Grafico base de datos junto a la regresion
13 #Datos puros
14 plt.plot(Ht, Wt, 'ro')
15 plt.suptitle("Datos a granel", fontsize=16)
16 plt.xlabel("Height in [cm]", fontsize=16)
17 plt.ylabel("Weight in [kg]", fontsize=16)
18 plt.show()
```

Regresion.py

SOLUCIÓN: Base de datos regresión

```
#Ajuste de la regresion.
#Recordar que queremos graficar y=beta0.est+beta1.est*x
t= np.arange(150, 190, 0.2)
plt.plot(Ht, Wt, 'ro')
plt.plot(t,0.5821*t-36.876)
plt.suptitle("Regresion Wt(Ht)", fontsize=16)
plt.xlabel("Height in [cm]", fontsize=16)
plt.ylabel("Weight in [kg]", fontsize=16)
plt.show()
```

Regresion.py

Algunas observaciones

- Lo más complicado de graficar en Python puede ser leer la base de datos.
- Para bases de datos en formato **Excel** ocupar librería **xlrd**. Ver más en ▸ Link
- Para bases de datos en formato **CSV** ocupar librería **csv**. Ver más en ▸ Link

EDUCATED GUESSING

Educated Guessing

Estime la cantidad de plumones consumidos en la universidad en 1 año.

- Un año consta de 2 semestres académicos de 17 semanas cada uno.
- **Supuesto**: Cada semana hay clases sólo los días de semana.
- Cada día consta de 1 bloques de clase.
- **Supuesto**: Cada bloque horario tiene distribución de clases homogéneo.
- **Supuesto**: En cada bloque horario están ocupadas aproximadamente el 85% de las salas.
- **Aproximación**: Hay salas en el Edificio C, 15 salas por piso en el P, 10 salas en el B, 4 salas en el edificio M.
- **Supuesto**: Suponer que un pofesor gasta 1 plumón por clase.

Educated Guessing

$$\begin{aligned}
\text{Total salas} &= 111 \\
\text{N Plumones diarios} &= 0.85 \times 111 \times 7 = 660.45 \approx 660 \\
\text{N Plumones semestrales} &= \text{N Plumones diarios} \times 17 \times 5 = 56100 \\
\text{N Plumones anuales} &= \text{N Plumones semestrales} \times 2 = 112200
\end{aligned}$$

Preguntas?