

LA-CoNGA Physics

Curso: Introduction to Data Science

Profesores: Juan C. Basto-Pineda y Arturo Sánchez Pineda

PRACTICA FINAL: Análisis de datos con Python y ROOT C++

Nicolás Fernández Cinquepalmi

1 Enunciado

Se pide elegir una fuente de datos propia o brindada por los profesores del curso, con el fin de analizarla y extraer resultados.

2 Desarrollo

La fuente de datos utilizada fue extraída del siguiente link,

['Link de enlace a fuente de datos utilizada' \(Clic here\)](#)

El primer paso realizado fue cambiar los ';' por ',' en el archivo '.csv'. Esto se realizó en Microsoft Excel, presionando 'Ctrl + B', pestaña 'Reemplazar', 'Buscar: ;', 'Reemplazar con: ,' y haciendo clic en 'Reemplazar todos'. Así obtenemos un archivo '.csv' correctamente configurado para proceder a abrirlo en nuestro notebook.

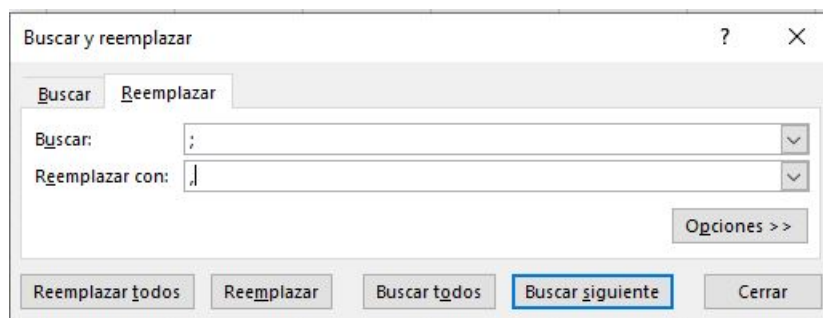


Figura 1: Reemplazo de ';' por ',' en Microsoft Excel

Se instala apropiadamente ATLAS-Open-Data-ubuntu-2020-v4 y se lo inicializa con la máquina virtual ('Oracle VM VirtualBox Manger').

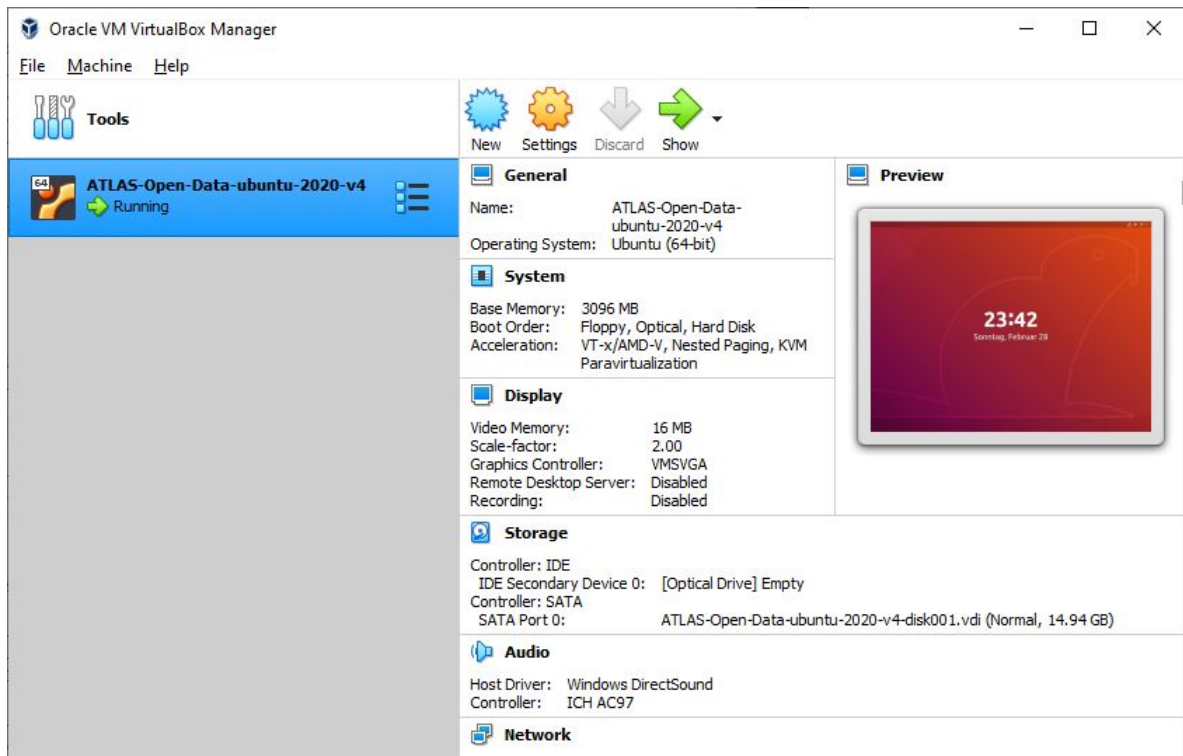


Figura 2: Virtual Machine

A continuación ingresamos en nuestro navegador e ingresamos la siguiente URL: localhost:8888 y colocamos el correspondiente password. Una vez allí procedemos a clonar nuestro repositorio de Gitmilab en un terminal,

```
student@ATLAS-opendata:~$ pwd
/home/student
student@ATLAS-opendata:~$ git clone https://gitmilab.redclara.net/fernandezn/ejercicios-clase-08-datos.git
```

Figura 3: 'Clone' del repositorio en Gitmilab

Configurado nuestro espacio de trabajo dentro de la máquina virtual, creamos un notebook Python 3 e importamos los módulos necesarios para realizar la actividad solicitada.

```
import pandas as pd
import math
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm
from collections import OrderedDict
from scipy.optimize import curve_fit
import math
import matplotlib.mlab as mlab
import scipy
from scipy.stats import norm
```

Figura 4: Módulos necesarios a importar

Importamos el archivo '.csv' con los datos y lo procesamos como Data Frame. Colocamos 'header = 0'

ya que debemos ignorar los encabezados o títulos de las columnas como datos.

```
file = '/home/student/ejercicios-clase-08-datos/data-used/data_analisis.csv'
datos = pd.read_csv(file,header=0)
```

Figura 5: Importación de archivo '.csv'

	individuo	origen	peso	tamano		individuo	origen	peso	tamano
0	cam01	Alemania	18	26.0	16	pet02	Italia	8	12.0
1	cam02	Alemania	38	62.0	17	pet03	Italia	28	40.0
2	cam03	Alemania	39	59.0	18	pet04	Italia	10	15.0
3	cam04	Alemania	28	41.0	19	pet05	Italia	19	28.5
4	cam05	Alemania	7	13.5	20	pet06	Italia	13	17.5
5	cam06	Alemania	29	46.5	21	pet07	Italia	16	29.0
6	cam07	Alemania	19	25.5	22	pet08	Italia	15	20.5
7	cam08	Alemania	25	40.5	23	pet09	Italia	36	54.0
8	cam09	Alemania	40	59.0	24	pet10	Francia	39	53.5
9	cam10	Alemania	16	19.0	25	tur06	Francia	33	49.5
10	tur01	Inglaterra	26	39.0	26	tur07	Francia	31	43.5
11	tur02	Inglaterra	17	21.5	27	tur08	Francia	9	15.5
12	tur03	Inglaterra	32	49.0	28	tur09	Francia	10	17.0
13	tur04	Inglaterra	21	31.5	29	tur10	Francia	5	11.5
14	tur05	Inglaterra	21	27.5	30	tur11	Francia	20	26.0
15	pet01	Inglaterra	6	11.0	31	tur12	Francia	27	43.5

Figura 6: Datos Utilizados

Comenzamos con el procesamiento de los datos. En primera instancia deseamos obtener los valores de 'peso' y 'tamano' filtrado por 'origen'. Para esto, creamos variables independientes que almacenen los datos respectivos a cada país, y además realizamos la suma de los valores para obtener los subtotales.

```
pesoalem = 0
pesoingla = 0
pesoita = 0
pesofra = 0

pesolistalem = []
pesolistingla = []
pesolistita = []
pesolistfra = []
```

Figura 7: Variables independientes como 'array'

Para el llenado de los arrays, utilizamos un ciclo 'for' que corra por todos los individuos y filtre con distintos condicionales 'if' por país y lo adhiera al array correspondiente.

```

for i in range(len(datos.individuo)):
    if datos.origen[i]=='Alemania':
        pesoalem += datos.peso[i]
        pesolistalem.append(datos.peso[i])
    elif datos.origen[i]=='Inglaterra':
        pesoingla += datos.peso[i]
        pesolistingla.append(datos.peso[i])
    elif datos.origen[i]=='Italia':
        pesoita += datos.peso[i]
        pesolistita.append(datos.peso[i])
    elif datos.origen[i]=='Francia':
        pesosfra += datos.peso[i]
        pesolistfra.append(datos.peso[i])

print('El peso total por país de origen es: ')
print('')
print('Ale, Ingla, Ita, Fra')
pesoalem, pesoingla, pesoita, pesosfra
print('')
print('Alemania: ', pesolistalem)
print('Inglaterra: ', pesolistingla)
print('Italia: ', pesolistita)
print('Francia: ', pesolistfra)

```

El peso total por país de origen es:

Ale, Ingla, Ita, Fra

Alemania: [18, 38, 39, 28, 7, 29, 19, 25, 40, 16]
 Inglaterra: [26, 17, 32, 21, 21, 6]
 Italia: [8, 28, 10, 19, 13, 16, 15, 36]
 Francia: [39, 33, 31, 9, 10, 5, 20, 27]

Figura 8: Ciclo 'for' y filtrado condicional 'if' para 'peso'

Realizamos un procedimiento similar, pero ahora analizando el 'tamano' y luego la cantidad de individuos por país de origen.

```

tamalem = 0
tamingla = 0
tamita = 0
tamfra = 0

tamlistalem = []
tamlistingla = []
tamlistita = []
tamlistfra = []

for i in range(len(datos.individuo)):
    if datos.origen[i]=='Alemania':
        tamalem += datos.tamano[i]
        tamlistalem.append(datos.tamano[i])
    elif datos.origen[i]=='Inglaterra':
        tamingla += datos.tamano[i]
        tamlistingla.append(datos.tamano[i])
    elif datos.origen[i]=='Italia':
        tamita += datos.tamano[i]
        tamlistita.append(datos.tamano[i])
    elif datos.origen[i]=='Francia':
        tamfra += datos.tamano[i]
        tamlistfra.append(datos.tamano[i])

print('El tamano total por país de origen es: ')
print('')
print('Ale, Ingla, Ita, Fra')
tamalem, tamingla, tamita, tamfra
print('')
print('Alemania: ', tamlistalem)
print('Inglaterra: ', tamlistingla)
print('Italia: ', tamlistita)
print('Francia: ', tamlistfra)

```

El tamano total por país de origen es:

Ale, Ingla, Ita, Fra

Alemania: [26.0, 62.0, 59.0, 41.0, 13.5, 46.5, 25.5, 40.5, 59.0, 19.0]
 Inglaterra: [39.0, 21.5, 49.0, 31.5, 27.5, 11.0]
 Italia: [12.0, 40.0, 15.0, 28.5, 17.5, 29.0, 20.5, 54.0]
 Francia: [53.5, 49.5, 43.5, 15.5, 17.0, 11.5, 26.0, 43.5]

Figura 9: Ciclo 'for' y filtrado condicional 'if' para 'tamano'


```

cantalem = 0
catingla = 0
cantita = 0
cantfra = 0

for i in range(len(datos.individuo)):
    if datos.origen[i]=='Alemania':
        cantalem += 1
    elif datos.origen[i]=='Inglaterra':
        catingla += 1
    elif datos.origen[i]=='Italia':
        cantita += 1
    elif datos.origen[i]=='Francia':
        cantfra += 1

print('El peso total por país de origen es: ')
print('')
print('Ale, Ingla, Ita, Fra')
cantalem, catingla, cantita, cantfra

El peso total por país de origen es:

Ale, Ingla, Ita, Fra

(10, 6, 8, 8)

```

Figura 10: Ciclo 'for' y filtrado condicional 'if' para cantidad de individuos

Con los valores obtenidos, realizamos el promedio por país en 'peso' y 'tamano'.

```

print('El peso promedio en Alemania es: ', "{:.2f}".format(pesoalem/cantalem))
print('El peso promedio en Inglaterra es: ', "{:.2f}".format(pesoingla/catingla))
print('El peso promedio en Italia es: ', "{:.2f}".format(pesoita/cantita))
print('El peso promedio en Francia es: ', "{:.2f}".format(pesofra/cantfra))

print('')

print('El tamaño promedio en Alemania es: ', "{:.2f}".format(tamalem/cantalem))
print('El tamaño promedio en Inglaterra es: ', "{:.2f}".format(tamingla/catingla))
print('El tamaño promedio en Italia es: ', "{:.2f}".format(tamita/cantita))
print('El tamaño promedio en Francia es: ', "{:.2f}".format(tamfra/cantfra))

El peso promedio en Alemania es: 25.90
El peso promedio en Inglaterra es: 20.50
El peso promedio en Italia es: 18.12
El peso promedio en Francia es: 21.75

El tamaño promedio en Alemania es: 39.20
El tamaño promedio en Inglaterra es: 29.92
El tamaño promedio en Italia es: 27.06
El tamaño promedio en Francia es: 32.50

```

Figura 11: Valores promedio por país de origen

Se puede observar que la población de origen Alemán cuanta con el mayor peso y tamaño promedio, lo cual es razonable. A fin de comprender mejor la distribución de los individuos por país de origen se realizará un gráfico de torta.

```

data = [cantalem, catingla, cantita, cantfra]
pais = ['Alemania', 'Inglaterra', 'Italia', 'Francia']

fig, ax = plt.subplots()
g = ax.pie(data, labels = pais, colors = ['SteelBlue', 'Gold', 'LightCoral', 'LimeGreen'],
           shadow=True, explode = (0.1, 0.1, 0.1, 0.1), autopct = '%1.2f%%')
plt.show

```

Figura 12: Ciclo 'for' y filtrado condicional 'if' para 'tamano'

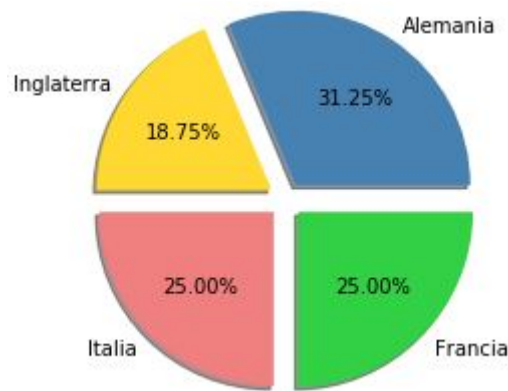


Figura 13: Gráfico de torta para distribución de individuos

Los valores promedios en 'peso' y 'tamano' por sí solos no proporcionar mucha información sobre la variable analizada. Es por esto que se realizan histogramas para ambos casos.

```
fig, axs = plt.subplots(1, 4, sharey=True, tight_layout=True)
bins = 8

axs[0].hist(pesolistalema, bins)
axs[0].set_title('Peso Alemania')
axs[0].set_xlabel("{:.2f}".format(np.mean(pesolistalema)))

axs[1].hist(pesolistingla, bins)
axs[1].set_title('Peso Inglaterra')
axs[1].set_xlabel("{:.2f}".format(np.mean(pesolistingla)))

axs[2].hist(pesolistita, bins)
axs[2].set_title('Peso Italia')
axs[2].set_xlabel("{:.2f}".format(np.mean(pesolistita)))

axs[3].hist(pesolistfra, bins)
axs[3].set_title('Peso Francia')
axs[3].set_xlabel("{:.2f}".format(np.mean(pesolistfra)))

Text(0.5, 0, '21.75')
```

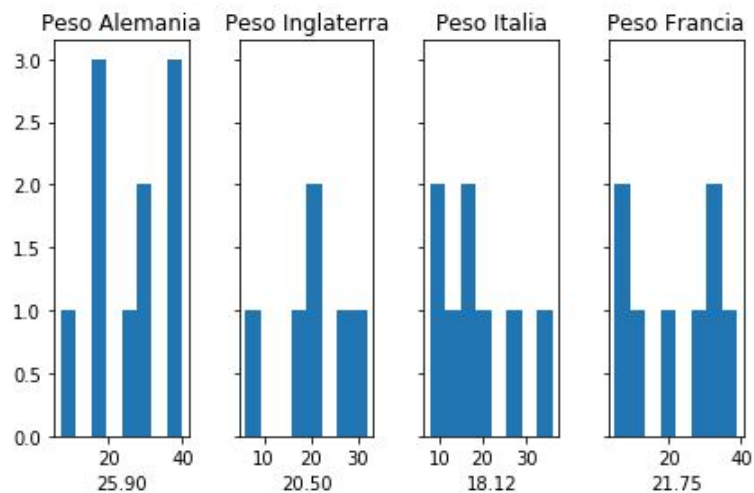


Figura 14: Histograma de 'peso' por país de origen

```

fig, axs = plt.subplots(1, 4, sharey=True, tight_layout=True)
bins = 8

axs[0].hist(tamlistalem, bins)
axs[0].set_title('Tam. Alemania')
axs[0].set_xlabel("{:.2f}".format(np.mean(tamlistalem)))

axs[1].hist(tamlistingla, bins)
axs[1].set_title('Tam. Inglaterra')
axs[1].set_xlabel("{:.2f}".format(np.mean(tamlistingla)))

axs[2].hist(tamlistita, bins)
axs[2].set_title('Tam. Italia')
axs[2].set_xlabel("{:.2f}".format(np.mean(tamlistita)))

axs[3].hist(tamlistfra, bins)
axs[3].set_title('Tam. Francia')
axs[3].set_xlabel("{:.2f}".format(np.mean(tamlistfra)))

Text(0.5, 0, '32.50')

```

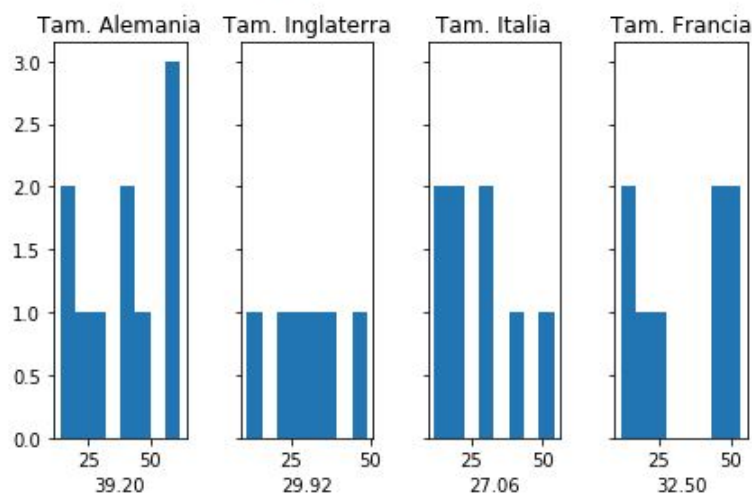


Figura 15: Histograma de 'tamano' por país de origen

Observando ambos histogramas podemos concluir que los individuos de origen inglés poseen los menores valores de desvío estándar para ambas variables, seguidos por los italianos. Para continuar analizando los datos utilizaremos ROOT importándolo en nuestro notebook de la siguiente manera,

```

import ROOT
import sys
from ROOT import TCanvas, TPad, TFile, TPaveLabel, TPaveText
from ROOT import gROOT
%jsroot on

Welcome to JupyROOT 6.18/04

!pip3 install root-pandas
import root_pandas

```

Figura 16: Importación de ROOT

Iniciamos de igual manera importando el archivo '.csv', almacenándolo en la variable 'df1'. Lo convertimos en '.root', lo guardamos y abrimos desde el directorio correspondiente y creamos una 'canvas' donde luego proyectaremos nuestros histogramas.

```

file = '/home/student/ejercicios-clase-08-datos/data-used/data_analisis.csv'
df1 = pd.read_csv(file,header=0)
df1.to_root('/home/student/ejercicios-clase-08-datos/data-used/data_analisis.root', key='mytree')
f = ROOT.TFile.Open('/home/student/ejercicios-clase-08-datos/data-used/data_analisis.root')
canvas = ROOT.TCanvas("Canvas","a first way to plot a variable",800,600)

```

Figura 17: Creación y utilización de archivo '.root'

A continuación analizaremos la distribución general de 'pesos' para todos los países, como así también la distribución general de 'tamano'.

```

hist = ROOT.TH1F("Variable: Peso","Distribucion de Pesos; Valor de Peso ; Events ",8,0,45)
for event in tree:
    hist.Fill(tree.peso)

print("Done!")

hist.SetFillColor(ROOT.kBlue-10)
canvas.Draw()
hist.Fit("gaus");

```

Done!

EXT		PARAMETER	VALUE	ERROR	STEP	SIZE	FIRST
NO.	NAME						DERIVATIVE
1	Constant	6.07339e+00	1.41538e+00	1.05736e-03	1.53433e-04		
2	Mean	2.18406e+01	2.38978e+00	2.24715e-03	-5.29036e-05		
3	Sigma	1.13300e+01	2.08095e+00	4.82966e-05	1.82003e-03		

Distribucion de Pesos

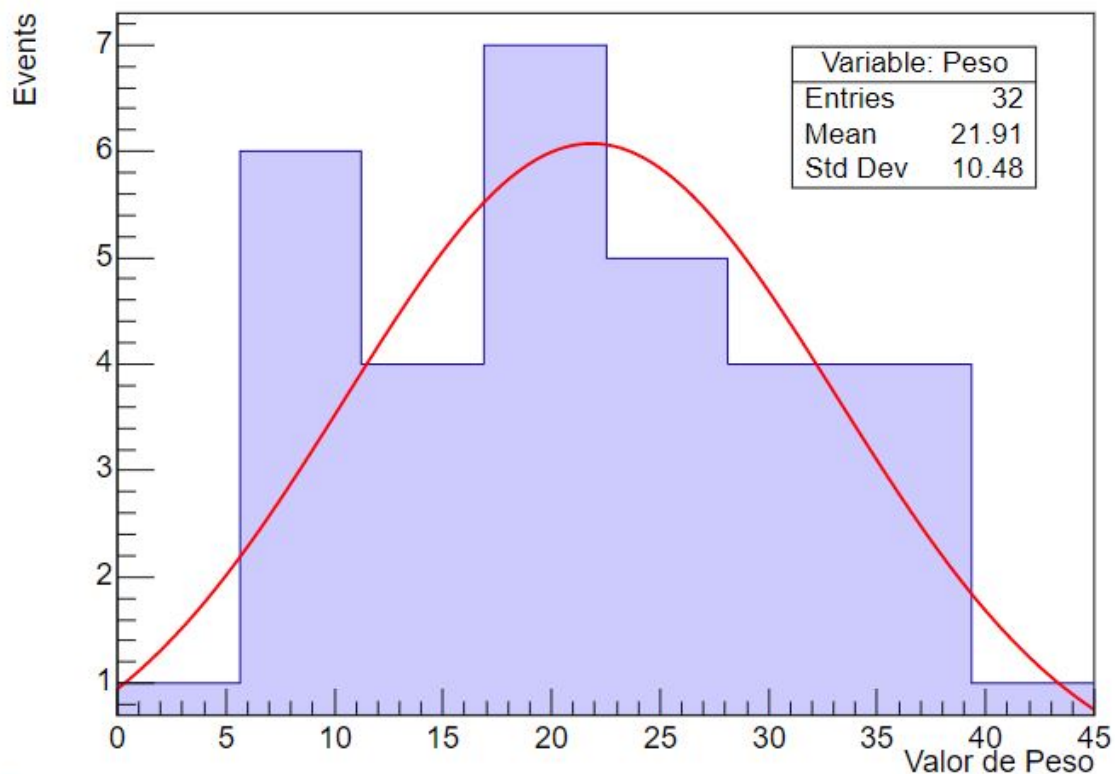


Figura 18: Histograma de 'peso' general


```

hist = ROOT.TH1F("Variable: Tamano","Distribucion de Tamanos; Valor de Tamano ; Events ",8,10,66)
for event in tree:
    hist.Fill(tree.tamano)

print("Done!")

hist.SetFillColor(ROOT.kBlue-10)
canvas.Draw()
hist.Fit("gaus");

```

Done!

```

FCN=8.33439 FROM HESSE      STATUS=NOT POSDEF      26 CALLS      1105 TOTAL
EDM=5.20343e-09 STRATEGY= 1 ERR MATRIX NOT POS-DEF

```

EXT NO.	PARAMETER NAME	VALUE	APPROXIMATE ERROR	STEP SIZE	FIRST DERIVATIVE
1	Constant	1.22835e+03	3.26288e+03	7.52706e-02	1.07954e-07
2	Mean	-5.13994e+02	1.26111e+02	2.80883e-03	3.33000e-06
3	Sigma	1.58227e+02	1.38177e+02	9.62716e-02	-8.22320e-05

Distribucion de Tamanos

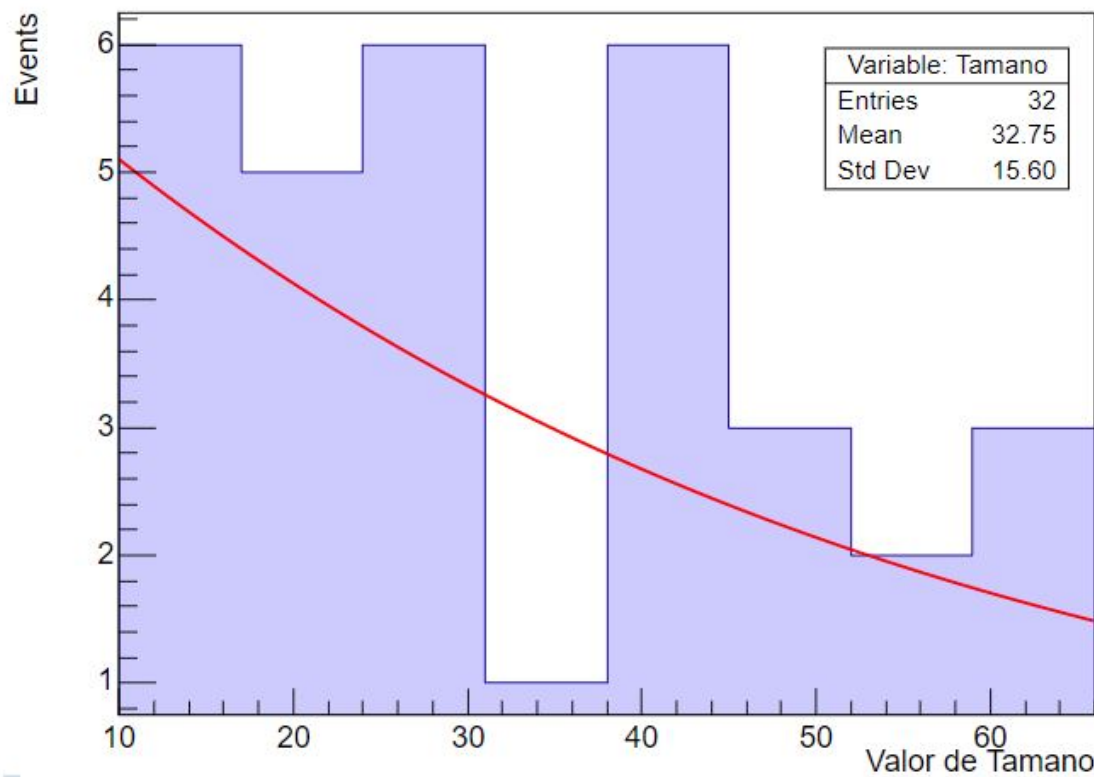


Figura 19: Histograma de 'tamano' general

Podemos inspeccionar que el histograma de 'peso' posee una distribución con una moderada aproximación a una curva gaussiana con media de 21.91 y desvío estándar de 10.48, mientras que en el histograma de 'tamano' no ocurre lo mismo. Se puede decir entonces, que la distribución de pesos posee mayor uniformidad que la distribución de tamaños.

Se realizará un último análisis correspondiente a la población alemana, con el objetivo de investigar si sus histogramas permiten un ajuste gaussiano consistente.

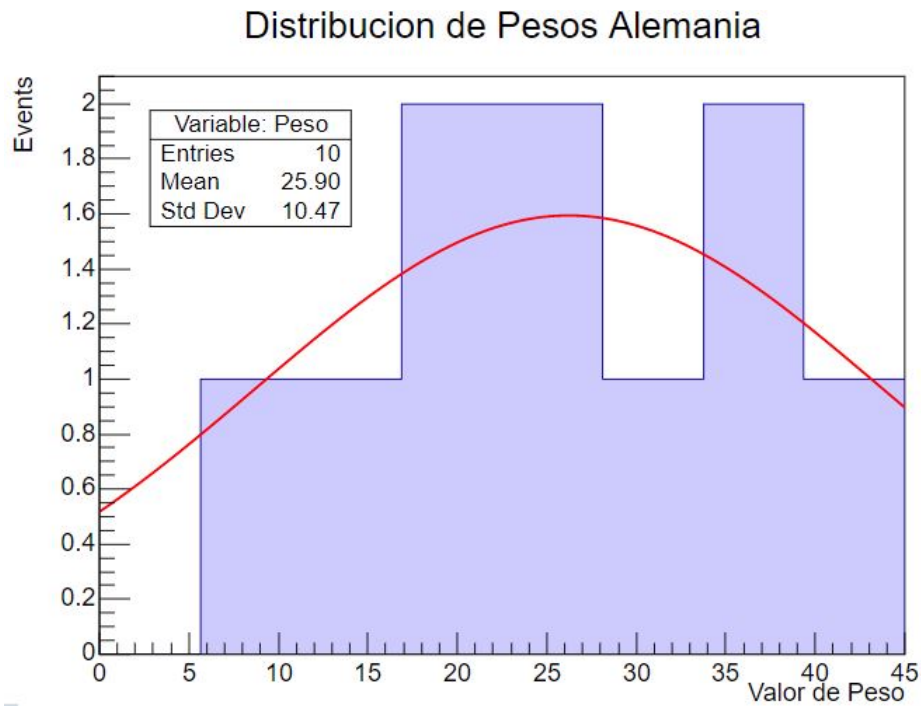


Figura 20: Histograma de 'peso' para Alemania

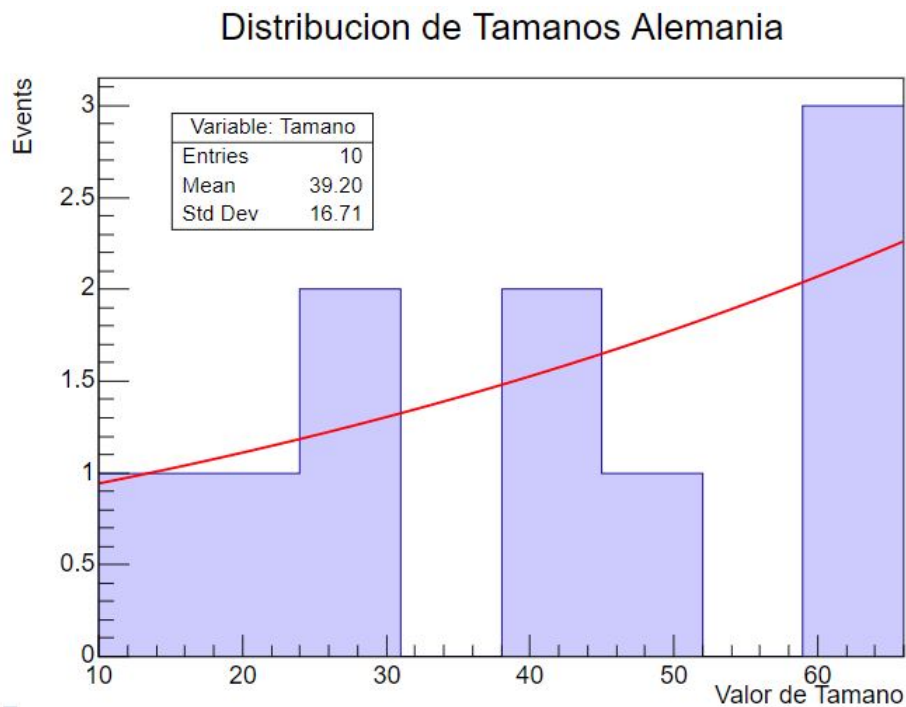


Figura 21: Histograma de 'tamano' para Alemania

Finalmente, podemos decir que la distribución de 'peso' se ajusta moderadamente, mientras que la distribución de 'tamano' posee una gran dispersión. Esto nos sugiere que si definimos un parámetro 'densidad' como peso / tamaño, la población de Alemania contaría con el individuo de mayor 'densidad'.

3 Observaciones Finales

Los datos analizados conducen a las interpretaciones brindadas en la sección anterior, sin embargo, se requiere de un mayor muestreo de individuos para concluir fehacientemente con los resultados obtenidos en este trabajo.

A pesar de que los datos no necesariamente reflejen valores reales, es un ejercicio útil para aprender a utilizar herramientas sumamente necesarias para la investigación científica.

4 Bibliografía

- 1 ['Opendata.Atlas.CERN-VM' \(Clic here\)](#)
- 2 ['Swan-Gallery-CERN-basic' \(Clic here\)](#)
- 3 ['Swan-Gallery-CERN-root-primer' \(Clic here\)](#)
- 4 ['Notebooks-Collection-Opendata-Atlas-CERN' \(Clic here\)](#)