UNIVERSIDAD TECNOLOGICA DE LA MIXTECA

Labor et Sapientia Libertas

Chun quiaa Tiaha sihi Nillao

OAXACA

Maestría en Ciencia de Datos
Probabilidad y Estadistica para Ciencia de Datos

Comportamiento de las Medias

Profesora: Ana Delia Olvera Cervantes
Alumno: Melchor Nolasco Cosijoeza

7 de diciembre de 2024

leasson-23

December 7, 2024

1 Lección 23

[209]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import math
from itertools import combinations
```

1.1 Lectura y despliegue de información

[210]:

```
df = pd.read_csv("../data/datos-leccion-23.csv")
```

[211]:

```
df.head()
```

[211]:

```
                             nombre  peso-kg  talla-m  pesoNacimiento-kg  \
0      Sarabia López Carlos Alberto     55.0     1.56               3.00
1  Valencia Barrios Aracely Fabiola     60.0     1.62               4.50
2          Unda Lopez Jonathan Said     62.0     1.69               4.50
3       López González Luis Antonio     62.0     1.70               3.25
4          Diaz Yescas Laura Sarahi     63.0     1.61               3.50

   tallaNacimiento-cm   imc     helado fuma estadoCivil
0                50.0  22.6  Chocolate   No     Soltero
1                51.0  22.9      Limón  si      soltera
2                50.2  21.7 Chocolate    Si     Soltero
3                51.0  19.3   Vainilla   No     soltero
4                52.0  24.3       Nuez   No     Soltera
```

1.2 Peso (kg)

1.2.1 Selección de Clase

[212]:

```
weigth = df["peso-kg"]
weigth
```

```
[212]: 0      55.0
       1      60.0
       2      62.0
       3      62.0
       4      63.0
       5      65.0
       6      65.0
       7      65.9
       8      66.0
       9      68.0
       10     68.0
       11     68.2
       12     70.0
       13     70.0
       14     80.0
       15     83.0
       Name: peso-kg, dtype: float64
```

```
[213]: # Parse to numpy array
       weigthArray = weigth.to_numpy()
       weigthArray
```

```
[213]: array([55. , 60. , 62. , 62. , 63. , 65. , 65. , 65.9, 66. , 68. , 68. ,
              68.2, 70. , 70. , 80. , 83. ])
```

1.2.2 Generación de Combinaciones con 3 Muestras

Combinaciones

```
[214]: combinationsNumber = math.comb(16,3)
       combinationsNumber
```

```
[214]: 560
```

```
[215]: combinationList = list(combinations(weigthArray,3))
       len(combinationList)
```

```
[215]: 560
```

```
[216]: combinationList[:10]
```

```
[216]: [(55.0, 60.0, 62.0),
        (55.0, 60.0, 62.0),
        (55.0, 60.0, 63.0),
        (55.0, 60.0, 65.0),
        (55.0, 60.0, 65.0),
        (55.0, 60.0, 65.9),
        (55.0, 60.0, 66.0),
        (55.0, 60.0, 68.0),
        (55.0, 60.0, 68.0),
        (55.0, 60.0, 68.2)]
```

Promedios

[217]:

```
means = []
for t in combinationList:
    means.append(sum(t)/len(t))
len(means)
```

[217]:

```
560
```

[218]:

```
means[:10]
```

[218]:

```
[59.0,
 59.0,
 59.333333333333336,
 60.0,
 60.0,
 60.300000000000004,
 60.333333333333336,
 61.0,
 61.0,
 61.06666666666666]
```

1.2.3 Estadística y Comparación de los Datos Originales y los Promedios de las Muestras

Pesos originales

[219]:

```
weigth.describe()
```

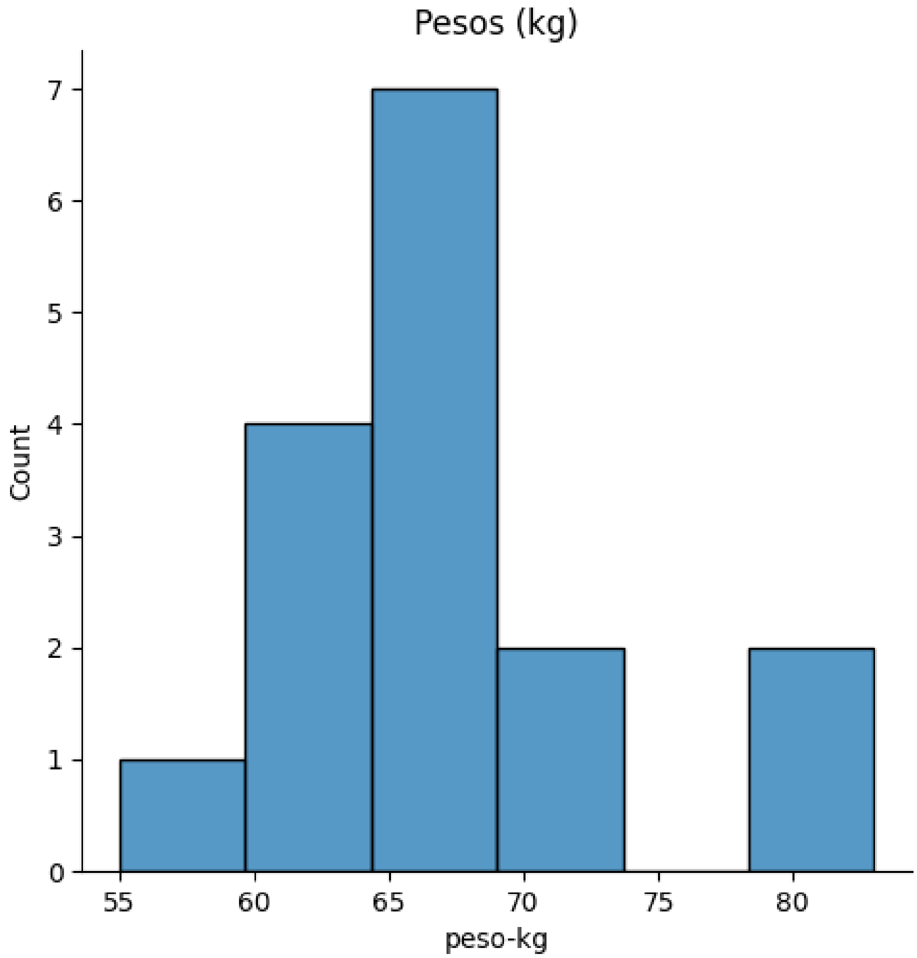
[219]:

```
count    16.000000
mean     66.943750
std       6.906998
min      55.000000
25%      62.750000
50%      65.950000
75%      68.650000
max      83.000000
Name: peso-kg, dtype: float64
```

[220]:

```
plt.figure(figsize=(10,4))
sns.displot(weigth)
plt.title("Pesos (kg)")
plt.show()
```

```
<Figure size 1000x400 with 0 Axes>
```



Promedio de las muestras

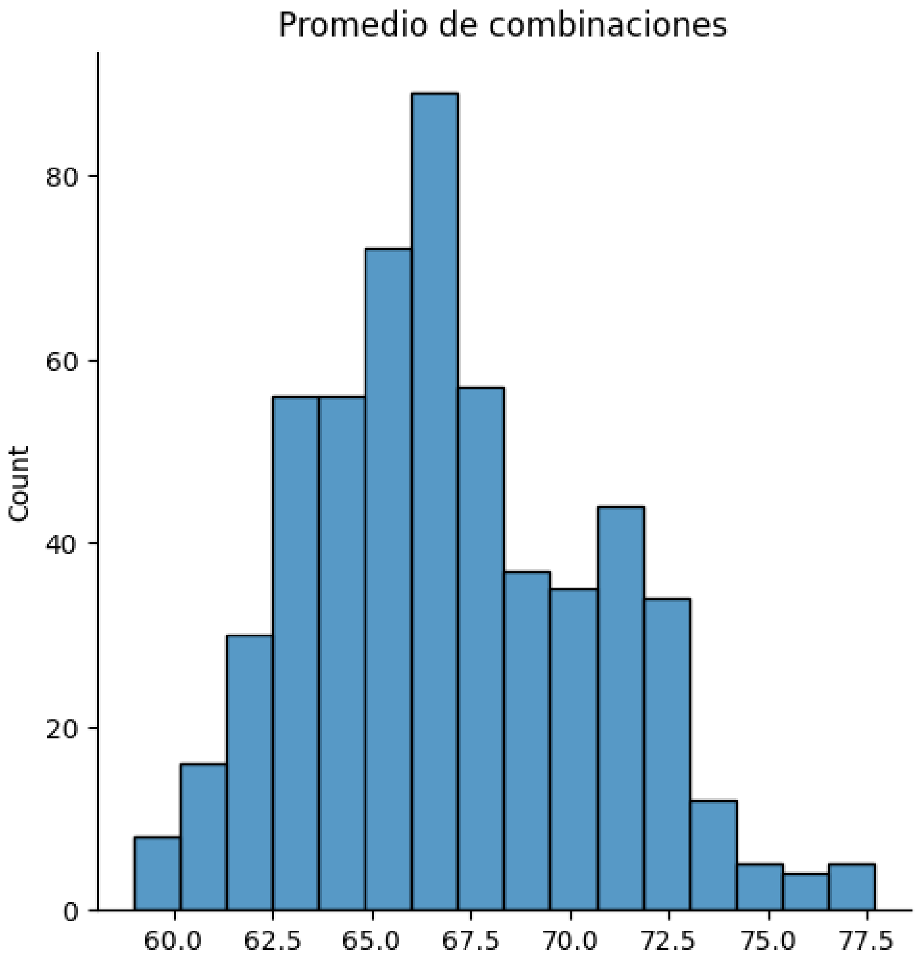
```
[221]: meansPd = pd.Series(means)
       meansPd.describe()
```

```
[221]: count    560.000000
       mean      66.943750
       std        3.597730
       min       59.000000
       25%       64.333333
       50%       66.400000
       75%       69.400000
       max       77.666667
       dtype: float64
```

```
[222]: plt.figure(figsize=(10,4))
       sns.displot(means)
```

```
plt.title("Promedio de combinaciones")
plt.show()
```

```
<Figure size 1000x400 with 0 Axes>
```



1.3 Talla Nacimiento (cm)

1.3.1 Selección de Clase

[223]:
```
birthSize = df["tallaNacimiento-cm"]
birthSize
```

[223]:
```
0      50.0
1      51.0
2      50.2
3      51.0
4      52.0
```

```
5      53.0
6      51.0
7      51.0
8      53.0
9      48.0
10     52.0
11     48.0
12     50.0
13     45.0
14     52.0
15     50.0
Name: tallaNacimiento-cm, dtype: float64
```

[224]:

```
# Parse to numpy array
birthSizeArray = birthSize.to_numpy()
birthSizeArray
```

[224]:

```
array([50. , 51. , 50.2, 51. , 52. , 53. , 51. , 51. , 53. , 48. , 52. ,
       48. , 50. , 45. , 52. , 50. ])
```

1.3.2 Generación de Combinaciones con 5 Muestras

Combinaciones

[225]:

```
combinationsNumber = math.comb(16,5)
combinationsNumber
```

[225]:

```
4368
```

[226]:

```
combinationList = list(combinations(birthSizeArray,5))
len(combinationList)
```

[226]:

```
4368
```

[227]:

```
combinationList[:10]
```

[227]:

```
[(50.0, 51.0, 50.2, 51.0, 52.0),
 (50.0, 51.0, 50.2, 51.0, 53.0),
 (50.0, 51.0, 50.2, 51.0, 51.0),
 (50.0, 51.0, 50.2, 51.0, 51.0),
 (50.0, 51.0, 50.2, 51.0, 53.0),
 (50.0, 51.0, 50.2, 51.0, 48.0),
 (50.0, 51.0, 50.2, 51.0, 52.0),
 (50.0, 51.0, 50.2, 51.0, 48.0),
 (50.0, 51.0, 50.2, 51.0, 50.0),
 (50.0, 51.0, 50.2, 51.0, 45.0)]
```

Promedios

[228]:
```
means = [sum(t)/len(t) for t in combinationList]
len(means)
```

[228]:
```
4368
```

[229]:
```
means[:10]
```

[229]:
```
[50.839999999999996,
 51.04,
 50.64,
 50.64,
 51.04,
 50.04,
 50.839999999999996,
 50.04,
 50.44,
 49.44]
```

1.3.3 Estadística y Comparación de los Datos Originales y los Promedios de las Muestras

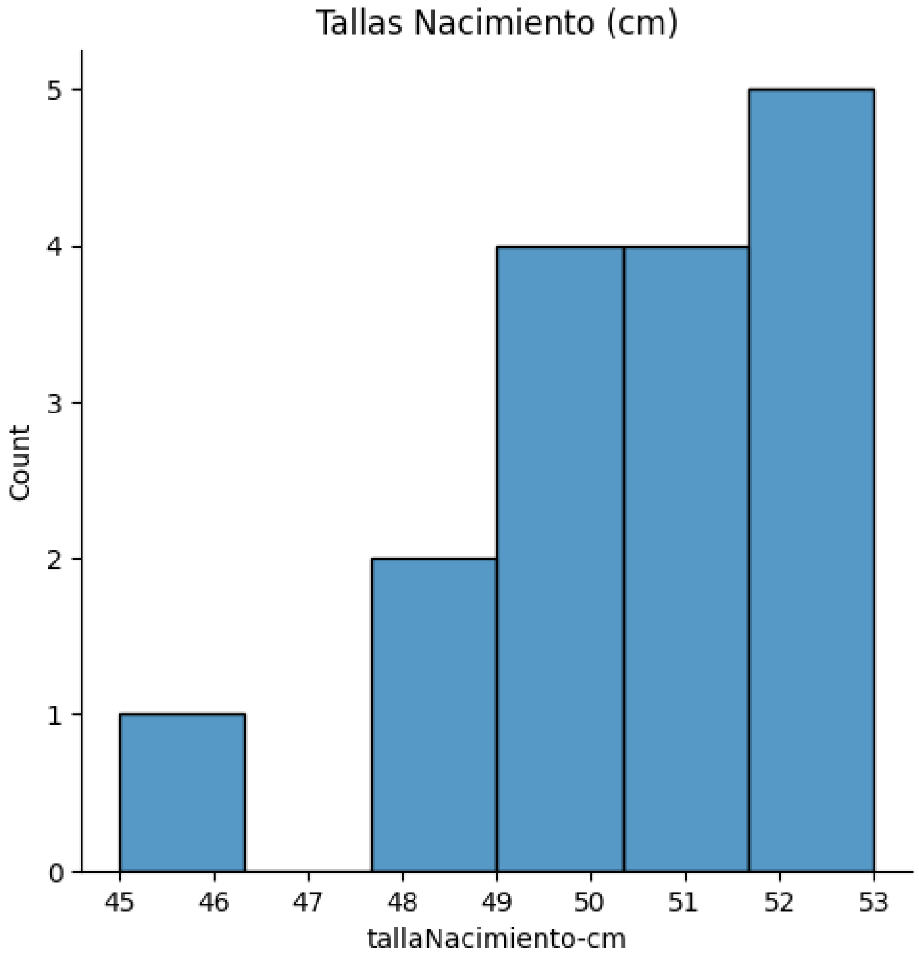
Pesos originales

[230]:
```
birthSize.describe()
```

[230]:
```
count    16.000000
mean     50.450000
std       2.062361
min      45.000000
25%      50.000000
50%      51.000000
75%      52.000000
max      53.000000
Name: tallaNacimiento-cm, dtype: float64
```

[231]:
```
plt.figure(figsize=(10,4))
sns.displot(birthSize)
plt.title("Tallas Nacimiento (cm)")
plt.show()
```

```
<Figure size 1000x400 with 0 Axes>
```



Promedio de las Muestras

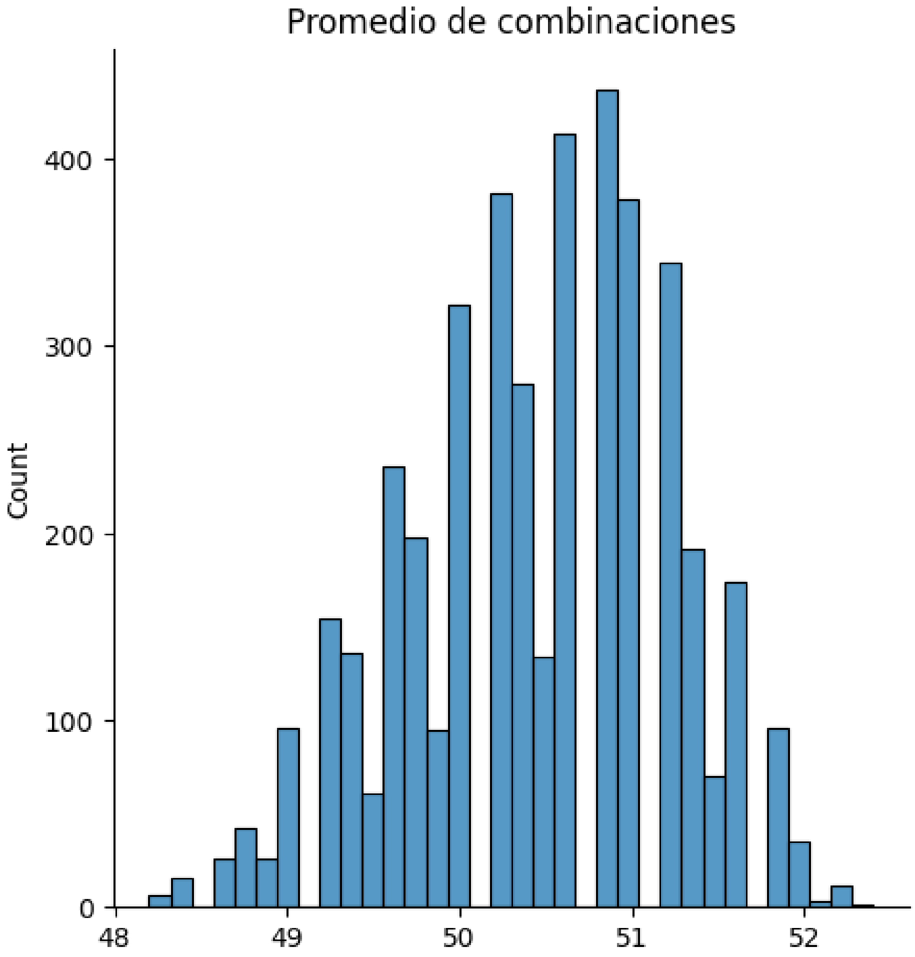
```
[232]: meansPd = pd.Series(means)
       meansPd.describe()
```

```
[232]: count    4368.000000
       mean       50.450000
       std         0.764832
       min        48.200000
       25%        49.960000
       50%        50.440000
       75%        51.000000
       max        52.400000
       dtype: float64
```

```
[233]: plt.figure(figsize=(10,4))
       sns.displot(meansPd)
```

```
plt.title("Promedio de combinaciones")
plt.show()
```

```
<Figure size 1000x400 with 0 Axes>
```


Promedio de combinaciones
Count