

ESTADÍSTICA DESCRIPTIVA

IRONHACK

Estadística descriptiva

Aprendemos estadística porque podemos; observar la información adecuadamente, sacar la conclusión del gran volumen del conjunto de datos, hacer previsiones fiables sobre la actividad empresarial y mejorar el proceso empresarial. Para hacer todo este tipo de análisis se utiliza la estadística. Además, se clasifica en dos tipos: Estadística descriptiva y estadística inferencial.



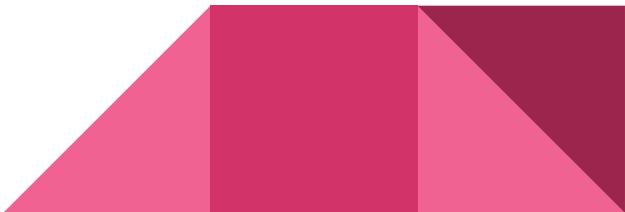
Estadística descriptiva

La estadística descriptiva resume los datos calculando la media, la mediana, la moda y la desviación estándar. Se distingue de la estadística inferencial por su objetivo de resumir la muestra en lugar de utilizar los datos para aprender más sobre la población.



Estadística descriptiva

En el análisis estadístico, hay tres conceptos fundamentales asociados a la descripción de los datos: localización o tendencia central, difusión o propagación y forma o distribución. Un conjunto de datos en bruto es difícil de describir; la estadística descriptiva describe el conjunto de datos de una manera más sencilla a través de:

- Medidas de tendencia central (Media, Mediana, Modo)
 - Medidas de dispersión (Rango, Cuartil, Percentiles, varianza y desviación estándar)
 - Medidas de simetría y curtosis
- 

Medidas de Tendencia Central

Medidas de Tendencia Central

El objetivo de la tendencia central es dar con el valor único que mejor describe las puntuaciones de la distribución. Se utilizan tres medidas básicas: la media (el valor promedio), la mediana (el valor situado en el medio) y la moda (el valor más frecuente).

Pandas

```
dataframe.mean()
```

```
dataframe.median()
```

```
dataframe.mode()
```

axis parameter



Media

La media aritmética de unos datos es la puntuación o valor promedio y se calcula simplemente sumando todas las puntuaciones y dividiendo por el número de puntuaciones.

Utiliza la información de cada una de las puntuaciones.

Pandas

```
dataframe.mean()
```

axis parameter



Mediana

Siempre que necesitemos encontrar un valor medio, recurriremos a la mediana para calcularla; debemos ordenar los valores en orden ascendente. La mediana también intenta definir un valor típico del conjunto de datos, pero a diferencia de la media, no requiere ser calculada. A tener en cuenta:

- Si hay un número impar de observaciones en el conjunto de datos, la mediana es el valor medio simple del orden ascendente de una columna concreta.
- Si el número de observaciones es par, la mediana es la media de los dos valores medios.

Pandas

```
dataframe.median()
```

axis parameter



Moda

La moda se utiliza como el valor que aparece con más frecuencia en nuestro conjunto de datos. El valor de la moda se suele calcular para las variables categóricas.

Pandas

```
dataframe.mode()
```

axis parameter



Ejemplo animals dataset

- Carga numpy y pandas en tu notebook o environment.
- Lee con pandas el dataset animals.csv
- Utiliza la función dtypes para ver los tipos de datos de las variables de tu dataset
- Obtén los valores de las medidas de tendencia central de cada una de las columnas e interprétalos.



#Las bibliotecas que vamos a utilizar y que son imprescindibles para trabajar con datos son numpy y pandas.

#numpy nos permite trabajar con vectores y matrices

```
import numpy as np
```

#pandas posee casi todo lo necesario para trabajar con análisis y manipulación de datos con Data Frames y Series a este nivel

```
import pandas as pd
```

#la función read de pandas nos permite abrir archivos de distintos formatos

```
animals = pd.read_csv('Data/animals.csv')
```

```
animals.head()
```

	brainwt	bodywt	animal
0	3.385	44.500	Arctic_fox
1	0.480	15.499	Owl_monkey
2	1.350	8.100	Beaver
3	464.983	423.012	Cow
4	36.328	119.498	Gray_wolf

De cara a un mejor entendimiento de nuestro data set podemos explorar el tipo de dato de cada columna con la función "dtypes" para Data Frames y "dtype" para Series

```
animals.dtypes
```

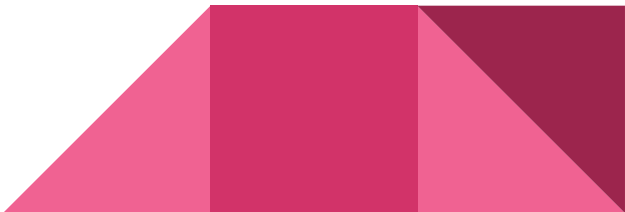
```
brainwt    float64
bodywt     float64
animal      object
dtype: object
```

En este caso tenemos dos columnas de "flotantes" (float64) y una de categóricos (pandas lo pone como "object"). [Otros tipos de datos](#) que encontraremos en pandas son los "enteros" (int64), "booleanos"(bool) y datos de tipo fecha (datetime64).

El número de columnas de cada tipo en un Data Frame puede obtenerse con la función get_dtype_counts()

```
animals.get_dtype_counts()
```

```
float64    2
object     1
dtype: int64
```



```
animals.columns
```

```
Index(['brainwt', 'bodywt', 'animal'], dtype='object')
```

```
animals.brainwt.astype(int)
```

0	3
1	0
2	1
3	464
4	36
5	27
6	14
7	1
8	4
9	0
10	0
11	0
12	1
13	0
14	0



```
animals.mean()
```

```
brainwt    198.794290  
bodywt     283.135355  
dtype: float64
```

```
animals['bodywt'].mean()
```

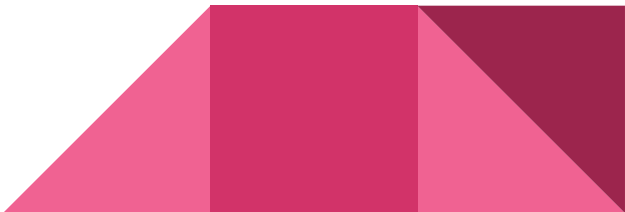
```
283.1353548387098
```

```
animals.median()
```

```
brainwt     3.3425  
bodywt     17.2500  
dtype: float64
```

```
animals.brainwt.mode()
```

```
0    0.023  
1    3.500  
dtype: float64
```



Medidas de Dispersión

Medidas de Dispersión

Entre ellas tenemos diferentes conceptos como Rango, Desviación Estándar, Varianza, Cuartil.

Pandas

```
dataframe.var()
```

```
dataframe.std()
```

```
dataframe.quantil(0.25)
```

```
dataframe.quantil(0.75)
```

```
dataframe.min()
```

```
dataframe.max()
```

axis parameter



Rango

El rango no es más que el mayor valor restado del menor. Ignora el efecto de los valores atípicos (outliers), sólo considera dos puntos en su estimación y no reconoce la distribución de los datos.

`numpy`

`dataframe.rank()`



Desviación estándar

La siguiente es la desviación; la desviación se calcula para saber cómo se han desviado los valores de la media. Al calcular la desviación, tenemos que ignorar los valores negativos y considerarlos como positivos.

numpy

`dataframe.std()`



Varianza

A continuación está la varianza, que se utiliza principalmente para encontrar la variación en el conjunto de datos. La varianza indica cuán cerca o lejos de la media están la mayoría de los valores de una determinada variable, y la desviación estándar (raíz cuadrada de la varianza) da la magnitud de la misma.

```
numpy  
dataframe.var()
```



Cuartiles

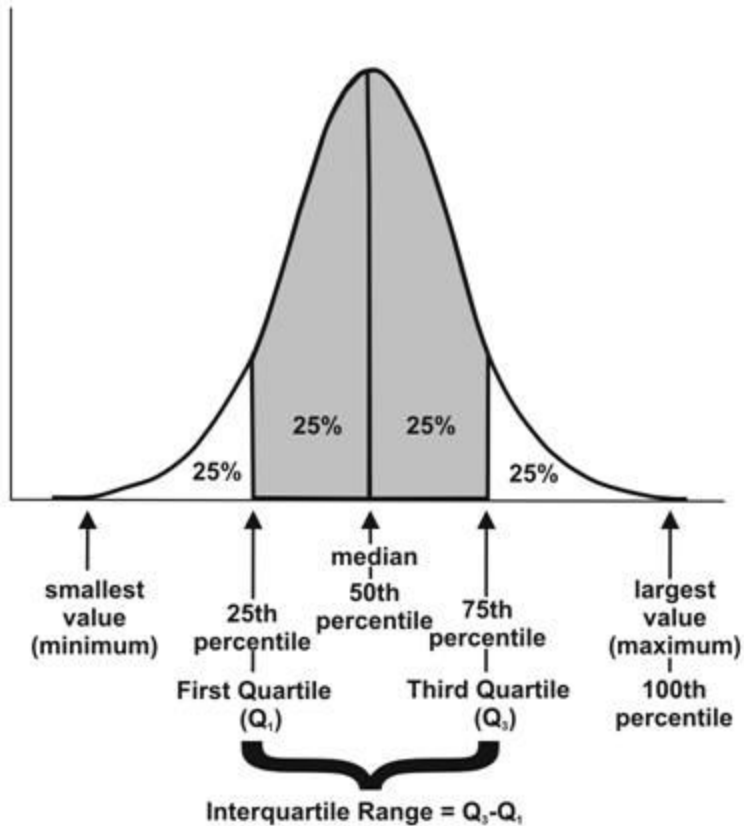
Los cuartiles de distribución son los tres valores que dividen los datos en cuatro partes iguales, como se indica a continuación, donde Q1 es el percentil 25, Q2 es el percentil 50 y Q3 es el percentil 75;

`numpy`

```
dataframe.quantil(0.25)
```

```
dataframe.quantil(0.75)
```





función describe()

Podemos utilizar la función describe de pandas sobre un dataframe (o una selección del dataframe) para obtener todas estas medidas.

pandas

dataframe.describe()



Ejemplo animals dataset

- Obtén los valores de las medidas de dispersión de cada una de las columnas e interprétalos.



```
animals.var()
```

```
brainwt      808528.832320  
bodywt       865418.787715  
dtype: float64
```

```
animals.std()
```


```
brainwt      899.182313  
bodywt       930.278876  
dtype: float64
```

```
animals.quantile(0.25)
```

```
brainwt      0.60  
bodywt       4.25  
Name: 0.25, dtype: float64
```

```
animals.quantile(0.75)
```

```
brainwt      48.20125  
bodywt      165.99825  
Name: 0.75, dtype: float64
```

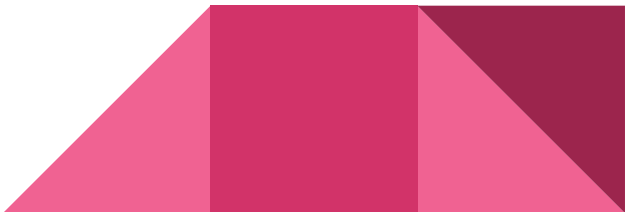



```
animals.min()
```

```
brainwt      0.005  
bodywt       0.14  
animal      African_elephant  
dtype: object
```

```
animals.max()
```

```
brainwt      6654.18  
bodywt       5711.86  
animal      Yellow-bellied_marmot  
dtype: object
```



```
animals.describe()
```

	brainwt	bodywt
count	62.000000	62.000000
mean	198.794290	283.135355
std	899.182313	930.278876
min	0.005000	0.140000
25%	0.600000	4.250000
50%	3.342500	17.250000
75%	48.201250	165.998250
max	6654.180000	5711.860000



Asimetría y Curtosis

Asimetría y Curtosis

La asimetría es la medida de la simetría o, más exactamente, de la falta de simetría. Por ejemplo, una distribución o conjunto de datos es simétrico si tiene el mismo aspecto a la izquierda y a la derecha de los datos del punto central.

Mientras que la curtosis es la medida de si los datos tienen colas pesadas o ligeras con respecto a la distribución normal.

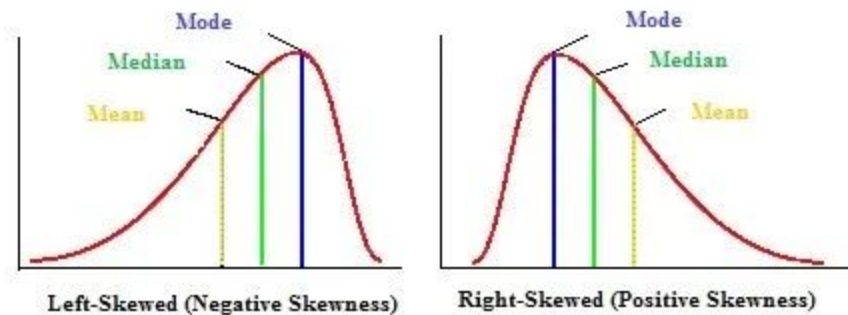
`numpy`

`dataframe.skew()`

`dataframe.kurtosis()`



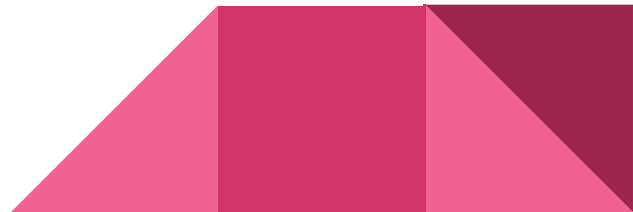
Asimetría



Curtosis

Leptocúrtica: Existe una gran concentración de los valores en torno a su media

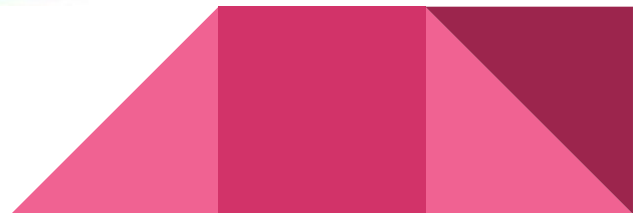
curtosis > 3



Curtosis

Mesocúrtica: Existe una concentración normal de los valores en torno a su media

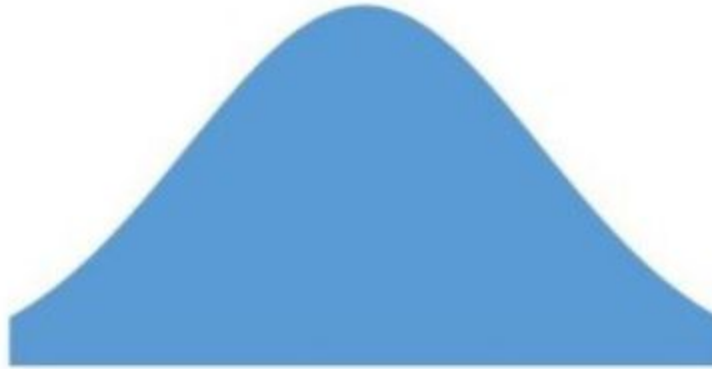
curtosis = 3



Curtosis

Platicúrtica: Existe una baja concentración de los valores en torno a su media

curtosis < 3



- Skewness. Asimetría de la distribución. ¿Es la media menor/mayor/igual que la mediana?
- Curtosis. ¿Hay mucha concentración de valores alrededor de la media?

```
animals.skew()
```

```
brainwt    6.563612  
bodywt     5.071528  
dtype: float64
```

```
animals.kurtosis()
```

```
brainwt    45.741060  
bodywt     26.269725  
dtype: float64
```

En nuestro caso la media es mayor que la mediana en ambas columnas, por lo que debe haber algunos outliers que provoquen esta asimetría. De la misma forma, la curtosis se presenta positiva en ambas columnas, indicando que los valores tienden a concentrarse alrededor de la media (distribución picuda)



Plots

Visualización matplotlib

A través de visualización también podemos entender en gran medida cómo se comporta la distribución de los datos muestrales.

Los plots más indicados para esto son:

- boxplots
- histogramas
- barplots



Boxplots

Los diagramas de caja son una visualización de una distribución de un conjunto de datos basado en un resumen de 5 números: mínimo, percentil 25, media, mediana, percentil 75 y máximo.

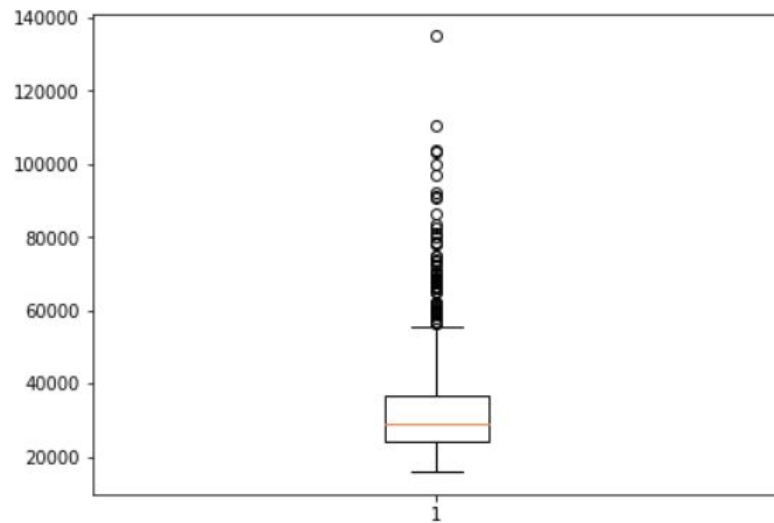
Los datos entre los percentiles 25 y 75 se dibujan dentro de la caja. Dibujamos un límite fuera de la caja llamada los bigotes. Los diagramas de caja también indican cuán extremos son los valores extremos al trazarlos como puntos individuales.

Esta gráfica nos da un resumen visual de los datos y muestra si los datos son simétricos o no, confirmando lo que hemos dicho antes.



Boxplots

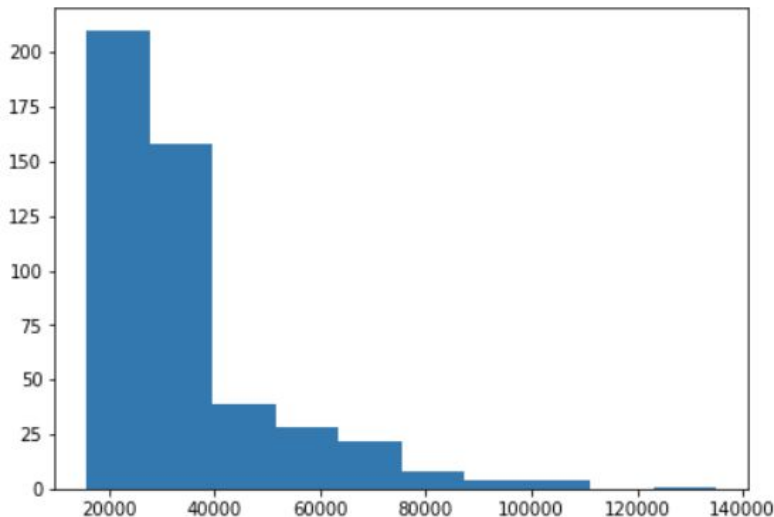
```
import matplotlib  
%matplotlib inline  
dataframe.boxplot();
```



Histogramas

Los histogramas son una buena manera de ver la distribución de frecuencia de nuestro conjunto de datos.

```
import matplotlib  
%matplotlib inline  
dataframe.hist();
```



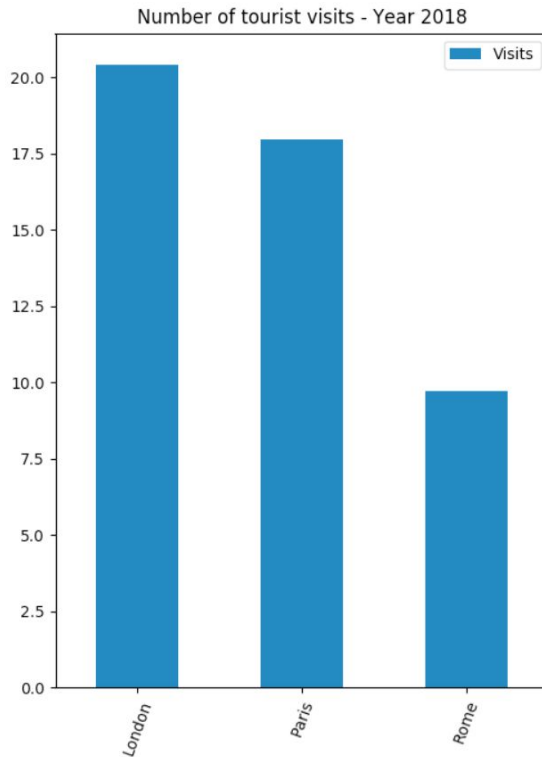
Barplots

Un diagrama de barras o gráfico de barras es un gráfico que representa la categoría de los datos con barras rectangulares con longitudes y alturas proporcionales a los valores que representan. Los gráficos de barras pueden trazarse horizontal o verticalmente. Un gráfico de barras describe las comparaciones entre las categorías discretas. Uno de los ejes del gráfico representa las categorías específicas que se comparan, mientras que el otro eje representa los valores medidos correspondientes a esas categorías.



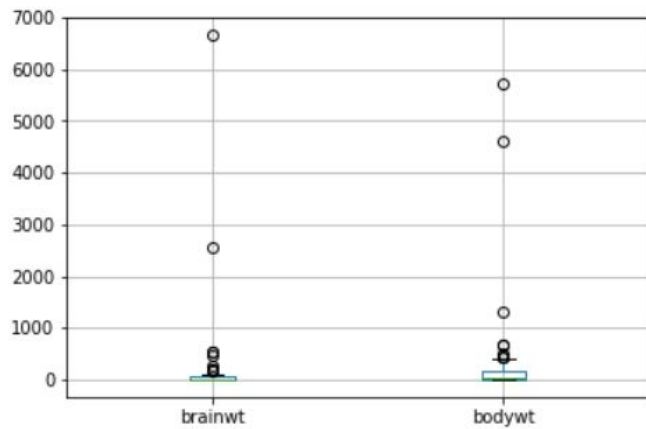
Barplots

```
import matplotlib  
%matplotlib inline  
dataframe.bar();
```

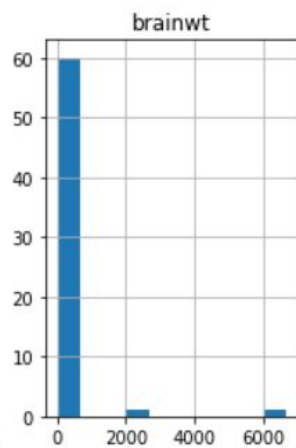
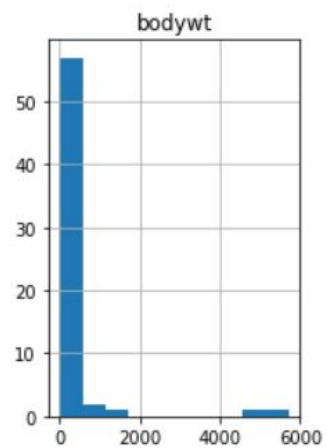



```
import matplotlib
%matplotlib inline

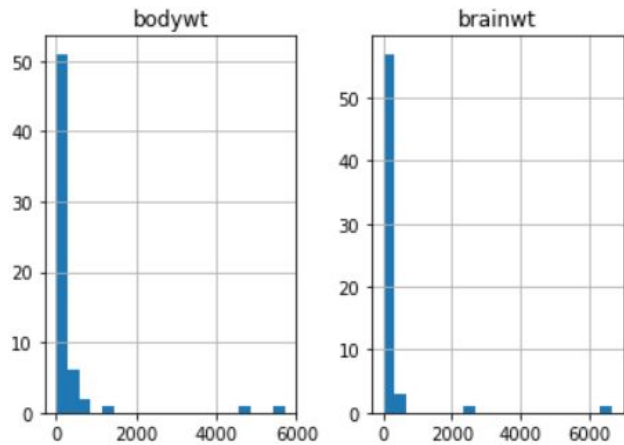
animals.boxplot();
```



```
animals.hist();
```



```
animals.hist(bins = 20);
```



Estos dos histogramas nos dicen que la mayor parte de los datos se encuentran en el extremo izquierdo y que tenemos unos cuantos outliers.



Vuestro turno

House Prices Dataset

Analiza las variables Saleprice, RoofStyle, SaleTyep y Beedrooms.

1. Lee la descripción de los datos del dataset y toda la información de estas variables (qué significan, cómo se miden, etc), aquí: [kaggle](#)
2. Descarga el train.csv, léelo con pandas y obtén los descriptivos de cada una de las variables (por separado). Interprétalos.
3. Haz las visualizaciones pertinentes para cada una de las variables. Interprétalas.

