

ANÁLISIS EXPLORATORIO DE DATOS

CLASE 14

ANÁLISIS EXPLORATORIO DE DATOS

Preface

This book is based on an important principle:

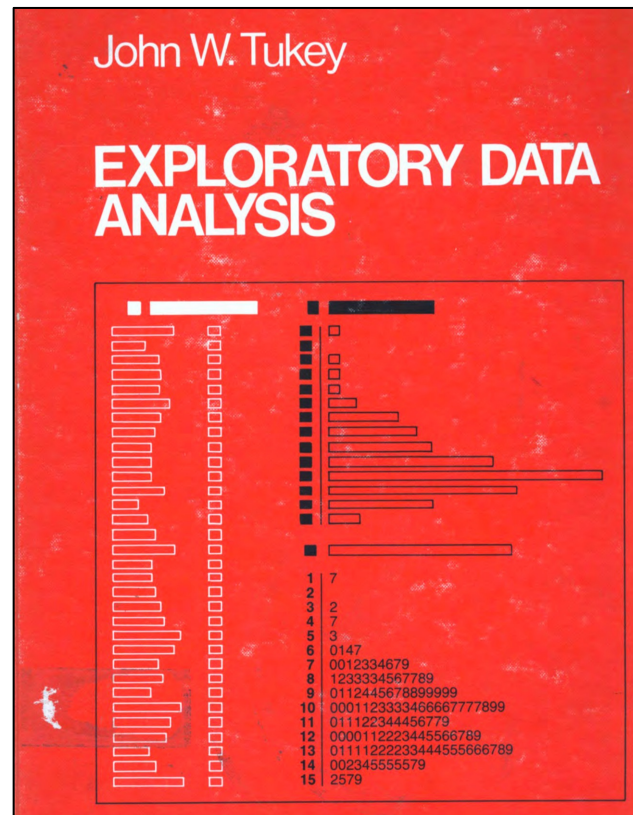
It is important to understand what you CAN DO before you learn to measure how WELL you seem to have DONE it.

Learning first what you can do will help you to work more easily and effectively.

This book is about exploratory data analysis, about looking at data to see what it seems to say. It concentrates on simple arithmetic and easy-to-draw pictures. It regards whatever appearances we have recognized as partial descriptions, and tries to look beneath them for new insights. Its concern is with appearance, not with confirmation.

Exploratory data analysis is an **attitude**, a state of flexibility, a willingness to **look for those things** that we believe are not there, as well as those we believe to be there.

John Tukey




ANÁLISIS EXPLORATORIO DE DATOS


No hay hipótesis ni modelos → los estamos buscando, la comprensión del problema cambia a medida que avanzamos.

Objetivos:


- Ganar intuición sobre la data, identificar patrones
- Confirmar si las preguntas son adecuadas
- Comparar distribuciones
- Chequear los datos (calidad, cantidad, sesgos, etc.)
- Identificar datos faltantes o outliers
- Resumir la data
- Comprender qué modelos o algoritmos son posibles de desarrollar



¿Tenemos la pregunta adecuada?
(no muy vaga, no tan específica, relevante)

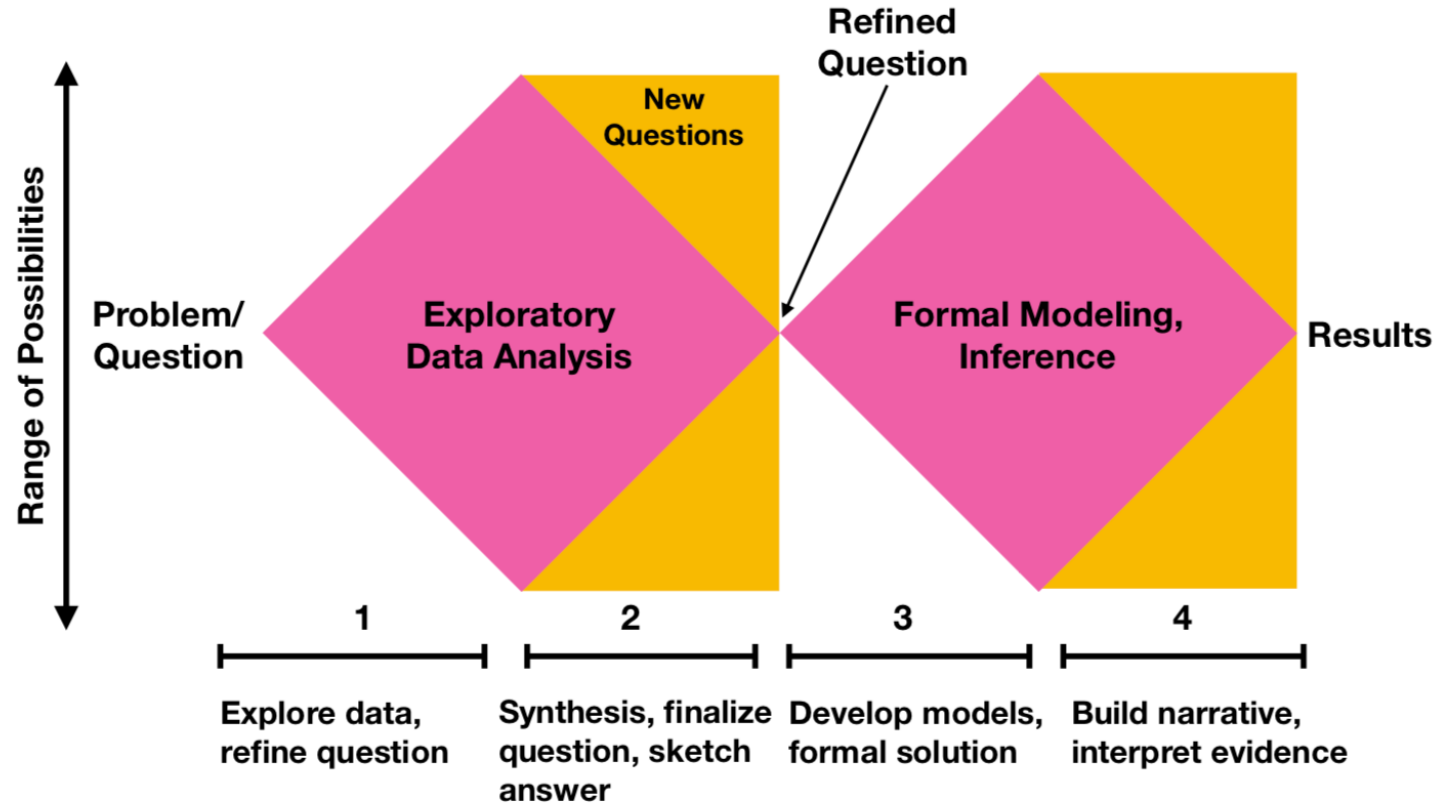


¿Tenemos la data adecuada?



¿Podemos bosquejar una solución?

ETAPAS DEL ANÁLISIS EXPLORATORIO



ANÁLISIS EXPLORATORIO DE DATOS

Algunos pasos que no pueden faltar:

1. Chequear metadata y estructura general de los datos: `df.columns`, `df.info()`
 - ¿Qué información tenemos de la data/variables antes de leer los archivos?
 - ¿Qué columnas / tipos de datos vemos al leer los datos en un DataFrame (`df`)?
2. Chequear primeras y últimas filas: `df.head()`, `df.tail()`
 - ¿Hay filas vacías/corruptas?
 - ¿Conviene ordenar los datos de alguna forma especial? Ej: fechas
3. Conteos generales: `df.info()`, `df.describe()`, `pd.unique()`, `pd.isnull()`, ...
 - N° de filas, N° columnas, N° valores nulos, N° categorías, N° valores únicos.
 - Estadísticas descriptivas: rangos de valores, totales, medias, dispersiones, correlaciones, etc.
4. Validar con fuentes externas.
 - ➔ ¿Hay valores extraños, faltan datos?

ANÁLISIS EXPLORATORIO DE DATOS

Algunos pasos que no pueden faltar:

5. Hacer gráficos:
 - Permiten resumir los datos
 - Permite verificar expectativas, y desviaciones de las expectativas
6. Plantear una solución “fácil” al problema, y testearla (con gráficos, tablas, etc).
7. Iterar, hasta tener convencimiento respecto a si:

¿Tenemos la pregunta adecuada?

¿Tenemos la data adecuada?

¿Podemos bosquejar una solución?

ANÁLISIS EXPLORATORIO DE DATOS

EDA	Estadístico	Gráfico
Una variable	Media Mediana Desviación estándar Varianza Percentiles Rango intercuartil (IQR=Q3-Q1) Distribución de probabilidad	Histograma FDA KDE Boxplot Choroplet
Numérica		
Categorica		Gráfico de torta (pie chart)
Multi-variable	Coef. Pearson Matriz de correlación Regresión Agrupación (groupby, pivot)	Scatterplot Jointplot Pairplots Histograma múltiple Serie de tiempo Stacked area Heatmap Pairplots... etc

pandas
numpy

scipy
sklearn

matplotlib, seaborn
geopandas

Estadísticas de Resumen

- **Media:** es la suma de todos los valores, dividida por el número de puntos. $\mu = \frac{1}{n} \sum_{i=1}^n x_i$
- **Mediana:** es el valor medio de un conjunto de datos. Es inmune a valores extremos o outliers. Para calcularla, se ordenan los datos y se elige el valor que queda en la mitad.
- **Percentiles:** el percentil p , corresponde al valor que es mayor al $p\%$ de los datos.
- **Varianza:** promedio de la distancia cuadrática de los datos a la media. Es una medida de la dispersión de los datos.

$$\text{Var} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

- **Desviación Estándar:** es la raíz cuadrada de la varianza. Está en la misma escala de unidades que los datos.

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$$

Estadísticas de Resumen

- **Covarianza:** es una medida de cómo dos cantidades varían juntas. Es la media del producto entre las diferencias de los valores respecto a la media.

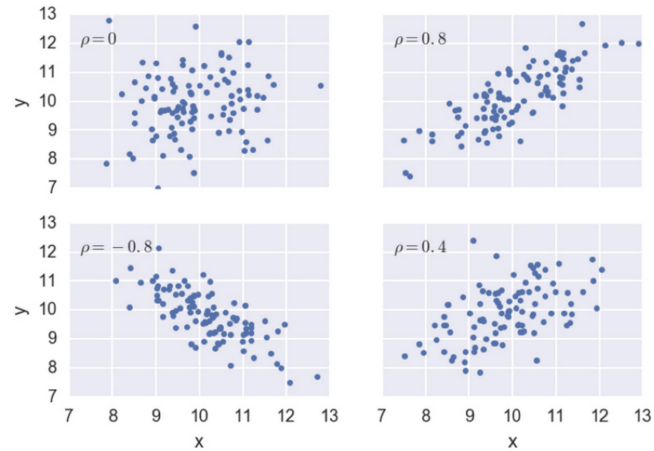
$$\text{Cov}(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

- Si x e y tienden a estar ambas arriba, o ambas abajo de la media, la covarianza es positiva.
 - Esto quiere decir que hay una correlación positiva: cuando x es alta, y es alta.
 - Por el contrario, si x es alta cuando y es baja, la covarianza es negativa y los datos están anticorrelacionados.
- **Coeficiente de Pearson (ρ):** para tener una medida más general y aplicable de la correlación entre dos variables, necesitamos que sea adimensional. Por lo tanto dividimos la covarianza por las desviaciones estándar de x e y .

$$\rho = \frac{\text{Cov}(x, y)}{\sigma_x \sigma_y}$$

- Es la comparación de la variabilidad en los datos debido a una codependencia (covarianza), con la variabilidad inherente de cada variable (sus desviaciones estándar).
 - Un valor **0** indica que **no hay correlación**, valor **1** indica **alta correlación**.

Estadísticas de Resumen



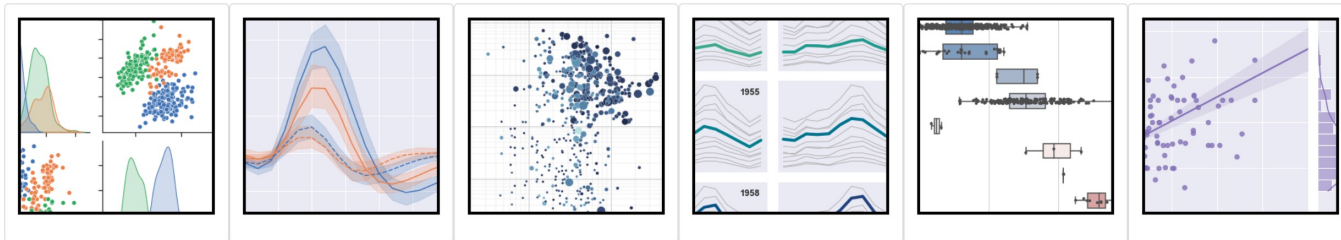
- **Coeficiente de Pearson (ρ):** para tener una medida más general y aplicable de la correlación entre dos variables, necesitamos que sea adimensional. Por lo tanto dividimos la covarianza por las desviaciones estándar de x e y .

$$\rho = \frac{\text{Cov}(x, y)}{\sigma_x \sigma_y}$$

- Es la comparación de la variabilidad en los datos debido a una codependencia (covarianza), con la variabilidad inherente de cada variable (sus desviaciones estándar).
- Un valor 0 indica que no hay correlación, valor 1 indica alta correlación.

Análisis Gráfico

seaborn: statistical data visualization



Seaborn is a Python data visualization library based on [matplotlib](#). It provides a high-level interface for drawing attractive and informative statistical graphics.

For a brief introduction to the ideas behind the library, you can read the [introductory notes](#). Visit the [installation page](#) to see how you can download the package and get started with it. You can browse the [example gallery](#) to see what you can do with seaborn, and then check out the [tutorial](#) and [API reference](#) to find out how.

To see the code or report a bug, please visit the [GitHub repository](#). General support questions are most at home on [stackoverflow](#) or [discourse](#), which have dedicated channels for seaborn.

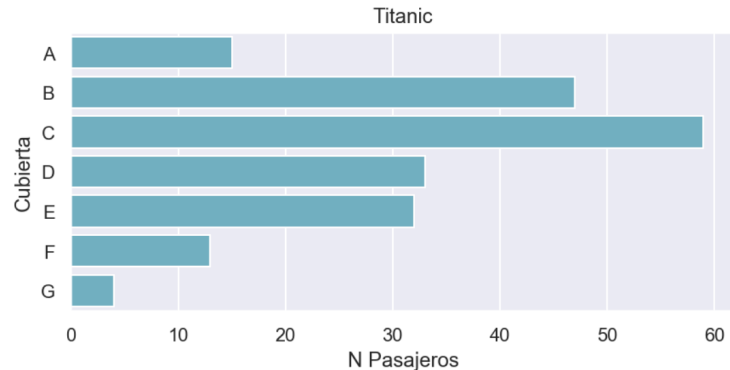
Dataset de Ejemplo: Pasajeros del Titanic

- El dataset **titanic.csv** contiene datos para 887 pasajeros del Titanic, cada uno de los cuales representa una fila.
- Las columnas indican atributos de la persona: si sobrevivió, edad, clase, sexo, tarifa pagada, etc.

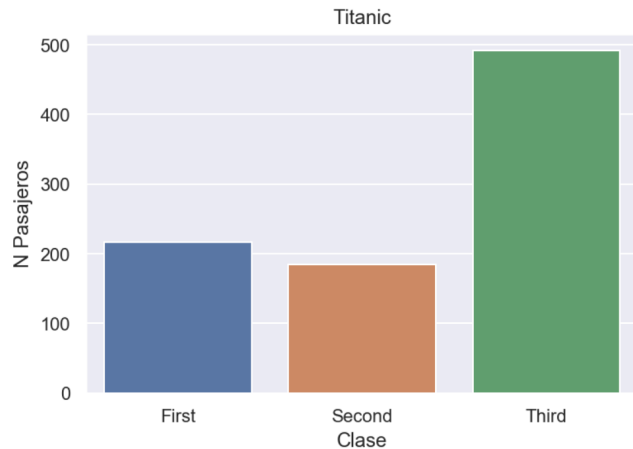
```
1 titanic = sns.load_dataset('titanic');  
2 titanic.info();
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 891 entries, 0 to 890  
Data columns (total 15 columns):  
survived      891 non-null int64  
pclass       891 non-null int64  
sex          891 non-null object  
age          714 non-null float64  
sibsp        891 non-null int64  
parch        891 non-null int64  
fare         891 non-null float64  
embarked     889 non-null object  
class        891 non-null category  
who          891 non-null object  
adult_male   891 non-null bool  
deck         203 non-null category  
embark_town   889 non-null object  
alive        891 non-null object  
alone        891 non-null bool  
dtypes: bool(2), category(2), float64(2), int64(4), object(5)  
memory usage: 80.6+ KB
```

Variables categóricas: countplot



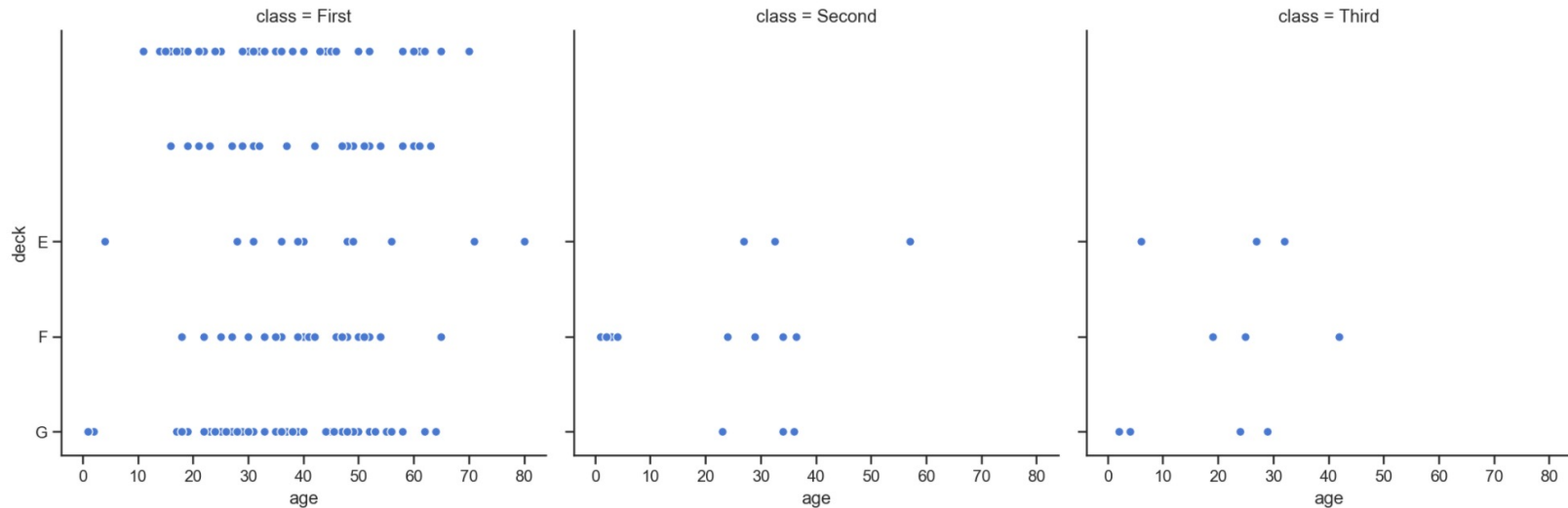
```
1 f, ax = plt.subplots(figsize=(7, 3));  
2 ax = sns.countplot(y="deck", data=titanic, color="c");  
3 ax.set_title('Titanic');  
4 ax.set_xlabel('N Pasajeros')  
5 ax.set_ylabel('Cubierta')
```



```
1 ax = sns.countplot(x="class", data=titanic)  
2 ax.set_title('Titanic');  
3 ax.set_ylabel('N Pasajeros');  
4 ax.set_xlabel('Clase');
```

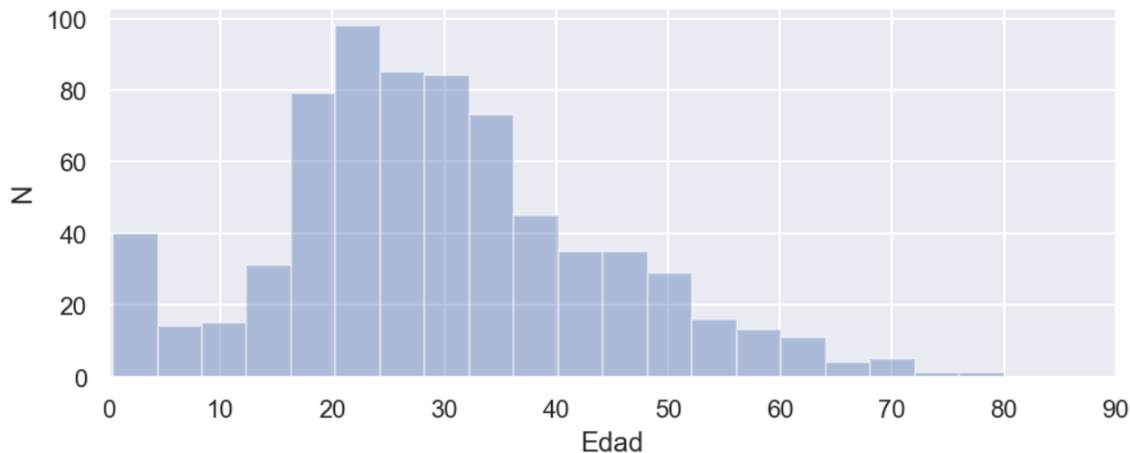
Variables categóricas: relplot

```
1 sns.set(style="ticks", palette="muted")  
2 sns.relplot(x="age", y="deck", col="class", data=titanic);
```



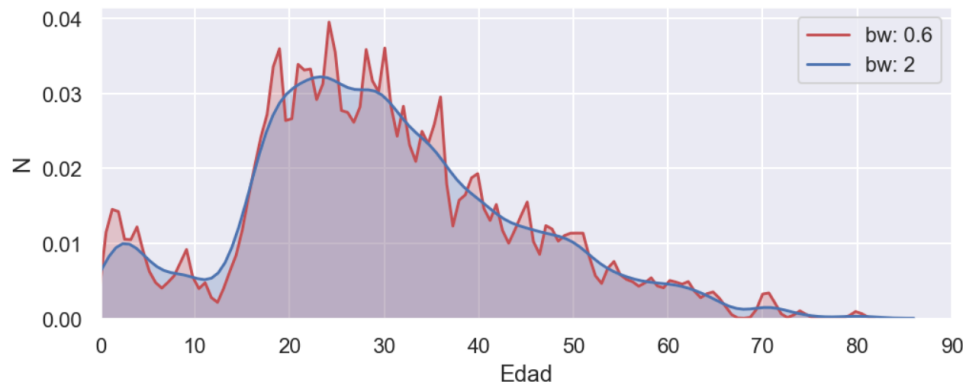
Histograma y Función de Densidad

```
1 # ¿Cuál es la distribución de edades de los pasajeros del Titanic?
2 import seaborn as sns
3 sns.set(color_codes=True)
4
5 f, ax = plt.subplots(1,1, figsize=(8, 3));
6 ax = sns.distplot(titanic.age, kde=False, bins=20)
7
8 ax.set(xlim=(0, 90));
9 ax.set_ylabel('N');
10 ax.set_xlabel('Edad');
```



Histograma y Función de Densidad

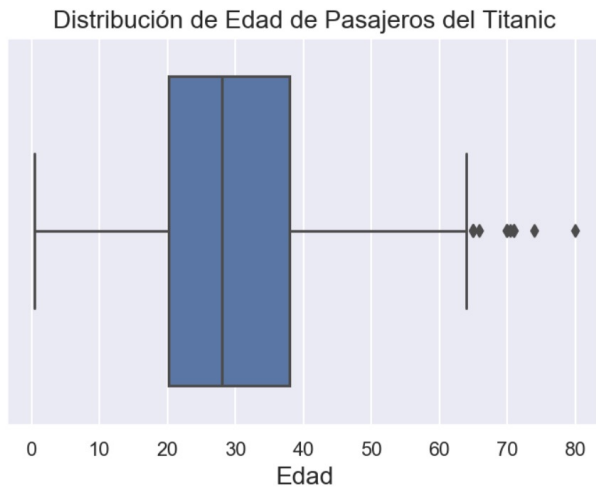
```
1 # ¿Cuál es la distribución de edades de los pasajeros del Titanic?
2 import seaborn as sns
3 sns.set(color_codes=True)
4
5 f, ax = plt.subplots(1,1, figsize=(8, 3));
6
7 sns.kdeplot(titanic.age, bw=0.6, label="bw: 0.6", shade=True, color="r",ax=ax);
8 sns.kdeplot(titanic.age, bw=2, label="bw: 2", shade=True,ax=ax);
9
10 ax.set(xlim=(0, 90));
11 ax.set_ylabel('N');
12 ax.set_xlabel('Edad');
```



Boxplot

Una Variable

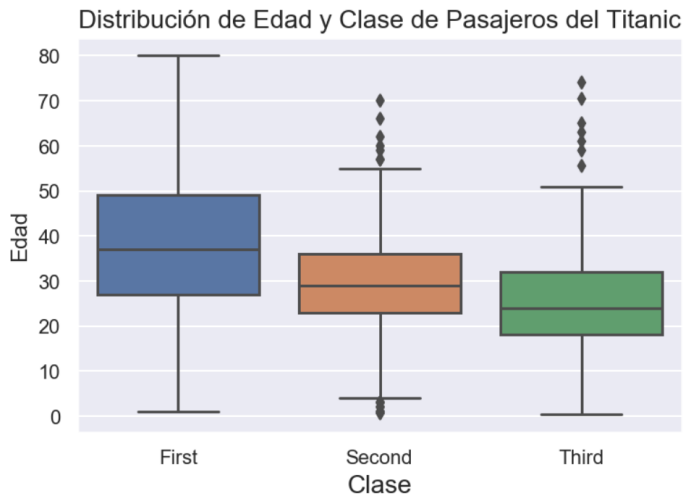
```
1 # seaborn
2 ax = sns.boxplot(x='age', data=titanic)
3 ax.set_ylabel(None);
4 ax.set_xlabel('Edad', fontsize=14);
5 ax.set_title('Distribución de Edad de Pasajeros del Titanic', fontsize=14);
```



Boxplot

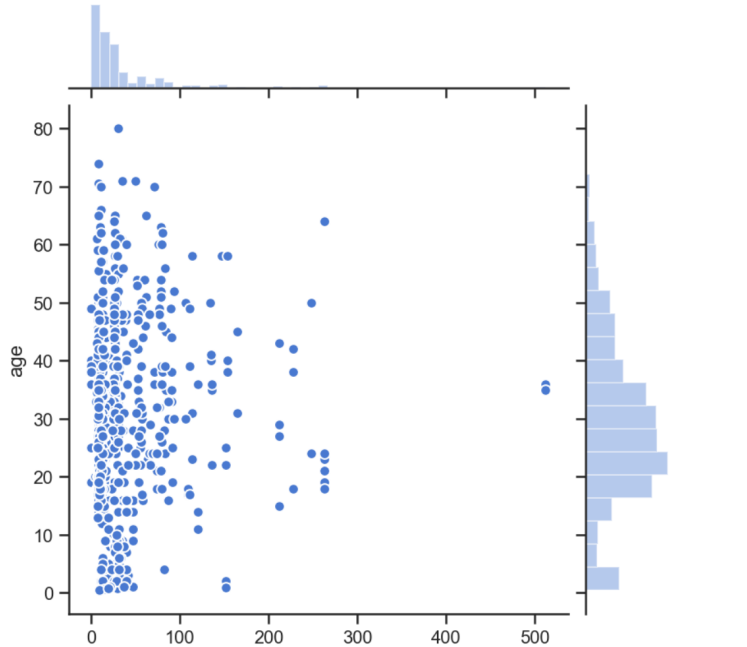
Dos variables: ¿la primera clase se componía de pasajeros más jóvenes o mayores?

```
: 1 ax = sns.boxplot(x='class', y='age', data=titanic)
  2 ax.set_ylabel('Edad');
  3 ax.set_xlabel('Clase', fontsize=14);
  4 ax.set_title('Distribución de Edad y Clase de Pasajeros del Titanic', fontsize=14);
```



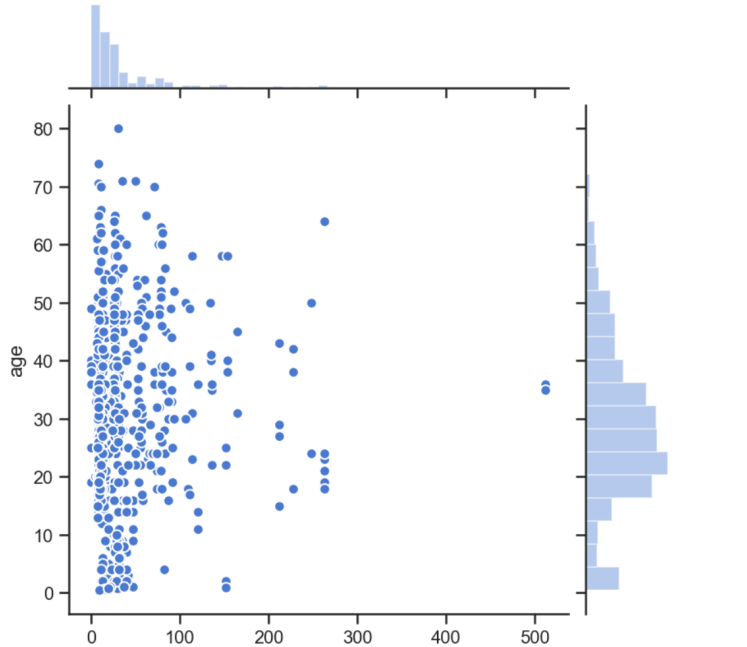
Scatterplot / Jointplot : Relación entre 2 variables

```
1 sns.jointplot("fare", "age", data=titanic, s=40, edgecolor="w", linewidth=1);  
2 ax.set_ylabel('Tarifa');  
3 ax.set_xlabel('Edad', fontsize=14);  
4 ax.set_title('Relación entre Edad y Tarifa Pagada por Pasajeros del Titanic', fontsize=14);
```



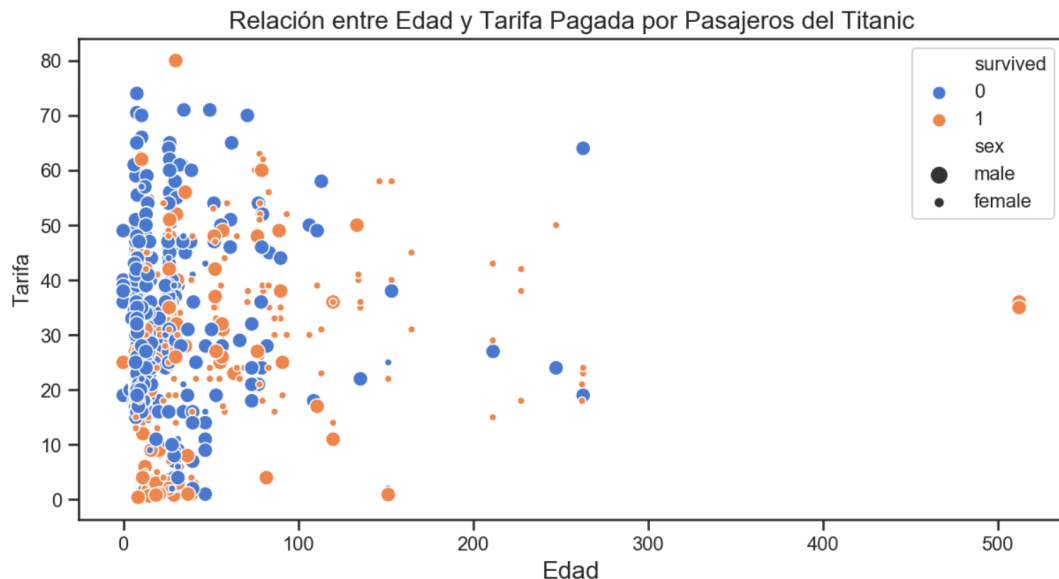
Scatterplot / Jointplot : Relación entre 2 variables

```
1 sns.jointplot("fare", "age", data=titanic, s=40, edgecolor="w", linewidth=1);  
2 ax.set_ylabel('Tarifa');  
3 ax.set_xlabel('Edad', fontsize=14);  
4 ax.set_title('Relación entre Edad y Tarifa Pagada por Pasajeros del Titanic', fontsize=14);
```



Scatterplot y Distplot: Relación entre 2 o más variables

```
1 f, ax = plt.subplots(1,1, figsize=(10, 5))
2 sns.scatterplot(x="fare", y="age", hue="survived", size="sex", data=titanic, ax=ax);
3 ax.set_ylabel('Tarifa');
4 ax.set_xlabel('Edad', fontsize=14);
5 ax.set_title('Relación entre Edad, Tarifa Pagada, Supervivencia y Sexo', fontsize=14);
6 plt.show();
```



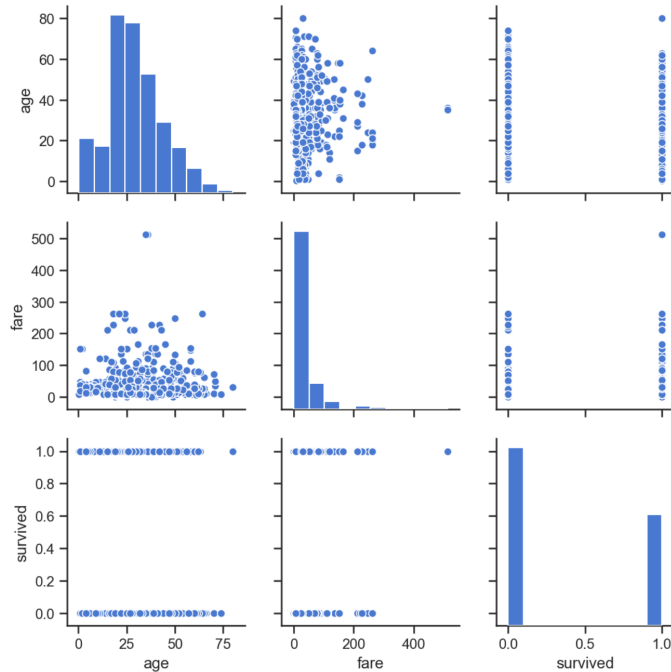
Pairplot: Análisis multivariable

```
1 titanic.columns
```

```
Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare', 'class', 'deck'], dtype='object')
```

```
1 graficar = ['age', 'fare', 'survived', 'deck']
```

```
1 titanic_plot = titanic.loc[:,graficar]  
2 sns.pairplot(titanic_plot);
```



Pairplot: Análisis multivariable

```
: 1 titanic.columns  
  
: Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare', 'class', 'deck'], dtype  
      ='object')  
  
: 1 graficar = ['age', 'fare', 'survived', 'deck']  
  
: 1 titanic_plot = titanic.loc[:,graficar]  
  2 sns.pairplot(titanic_plot,hue='deck');
```

