Machine Learning Workshop

Nicolas Känzig

Email: nkaenzig@gmail.com
Workshop Repository: https://github.com/nkaenzig/ml-workshop

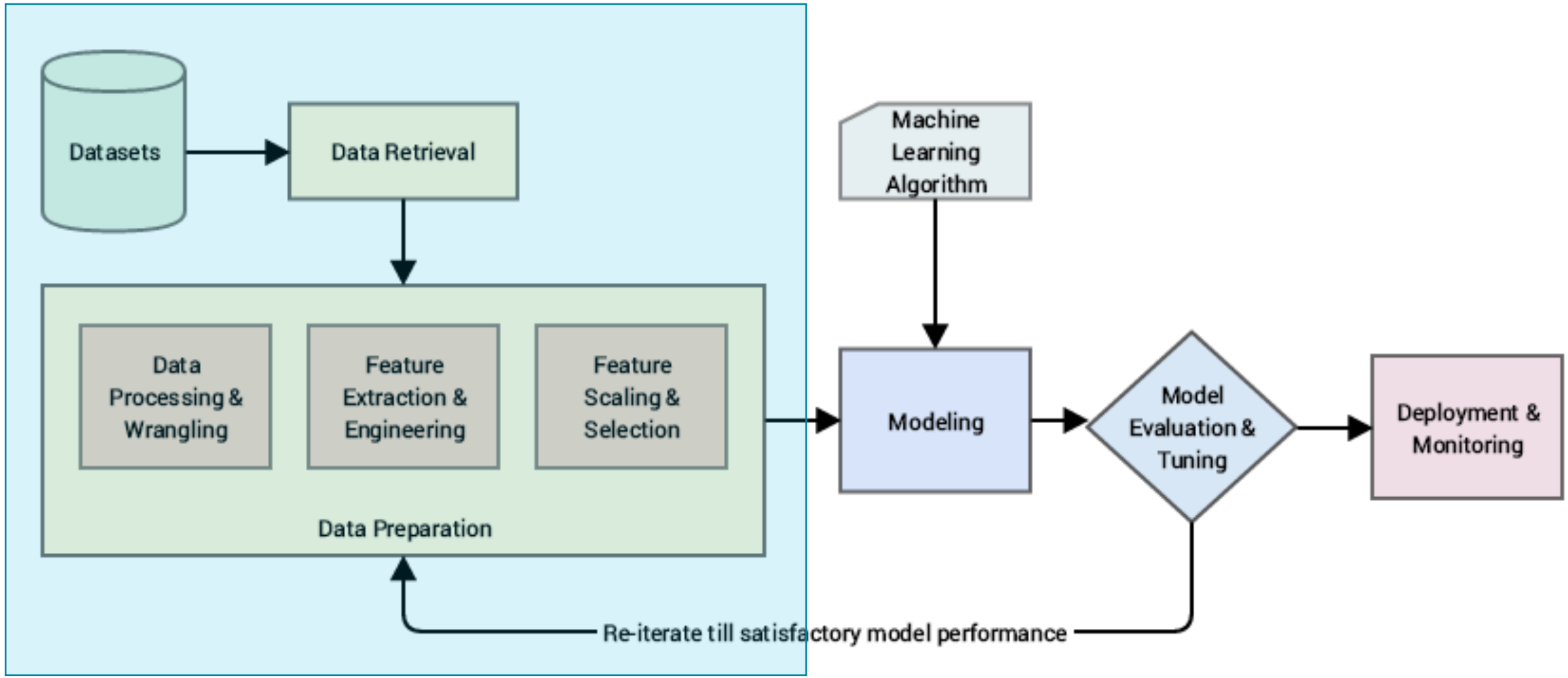
Contenido

Modulo 1
- Introducción ML
- Python

Modulo 2
- Análisis de datos
- Preprocesamiento de datos

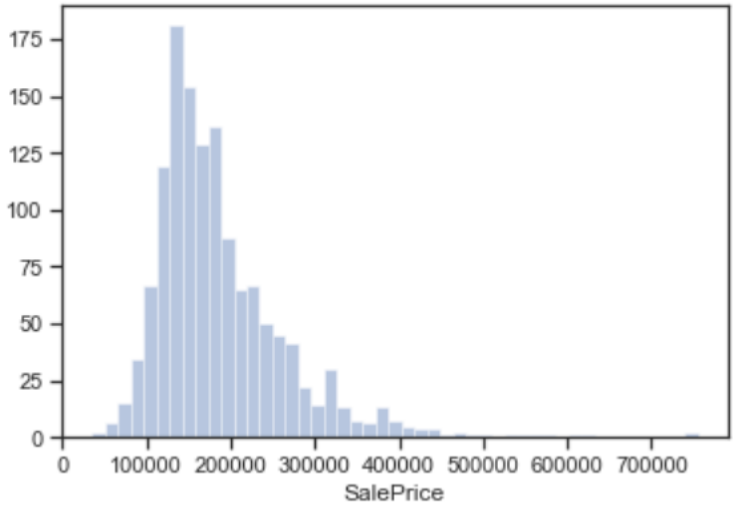
Modulo 3
- Modelos de ML
- Técnicas de evaluación

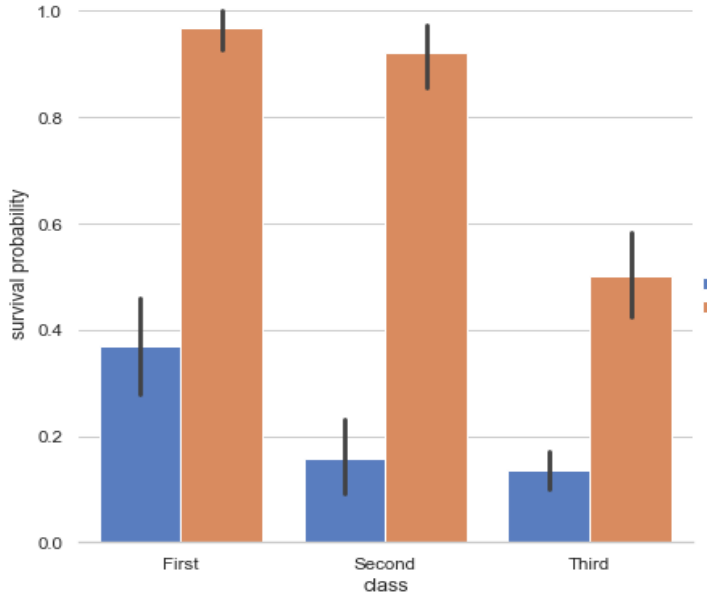
Análisis & Preprocesamiento de datos

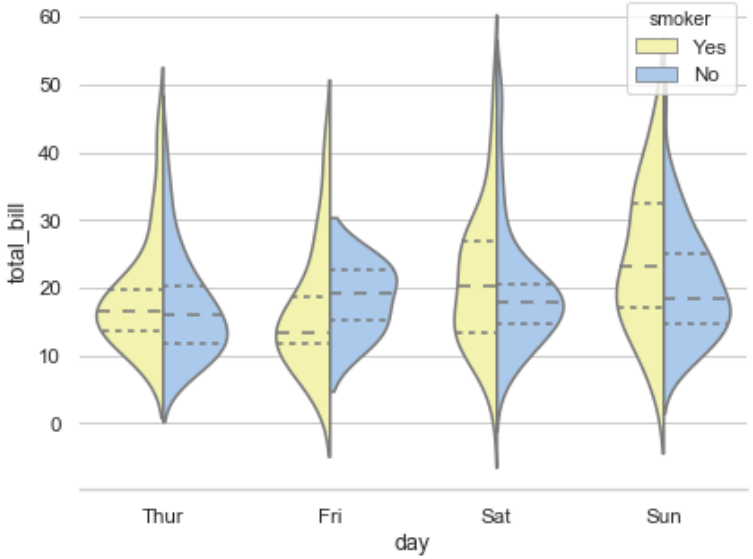

Datasets
Data Retrieval
Machine Learning Algorithm
Data Processing & Wrangling
Feature Extraction & Engineering
Feature Scaling & Selection
Data Preparation
Modeling
Model Evaluation & Tuning
Deployment & Monitoring
Re-iterate till satisfactory model performance

Análisis exploratorio de datos

- Crear visualizaciones y calcular medidas estadísticas para mejor entender los datos







Diferentes opiniones

A. *“El análisis exploratorio de datos en profundidad es primordial antes de entrenar un modelo”*

B. *“Hacer de antemano un amplio análisis de datos conduce a opiniones prejuiciadas sobre los datos y no es necesario para Machine Learning”*

Quien tiene razón?

Proceso

1. Limpieza de datos
   - Identificar valores faltantes
   - Filtrar datos que no son relevantes para la tarea / Filtrar errores
2. Análisis
   - Distribuciones & correlaciones
3. Preprocesamiento
   - Valores categóricos → numericos
   - Transformaciones (e.g. Standarización)

Proceso

4. Modelling
   - Empezar con un modelo sencillo, fácil para configurar & entrenar → **Baseline**
5. Machine Learning-Driven Data Analysis
   - Cuales son los features mas imporantes?
   - Analisar los errores que el modelo esta haciendo?

   → Usar estos insights para mejor entender los datos & mejorar el Dataset

Instrumentos

- Programación

- Liberarías

- IDEs

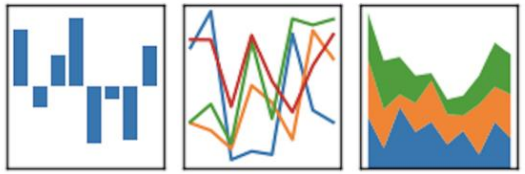
- Liberaría para computación científica
- Algebra Lineal:
  - Objetos para vectores/matrices (i.e. arrays)
- Estadistica:
  - Operaciones básicas: e.g. mean, median, std, percentiles, …

- Muchas de las operaciones son implementados en C

  → Mucho mas rápido que Python sin Numpy

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



	A	B	C	D
0	1.283449	0.405647	0.633235	-0.633953
1	-0.137045	-0.498740	-0.966406	-0.720781
2	-1.066049	0.458651	-1.384483	-0.174038
3	-0.823852	0.250134	0.973628	-0.174436
4	0.762657	-0.056813	1.097659	-0.449781
5	0.755400	-1.310918	0.146741	-0.315770
6	-0.523010	-0.438491	-1.010650	0.097777

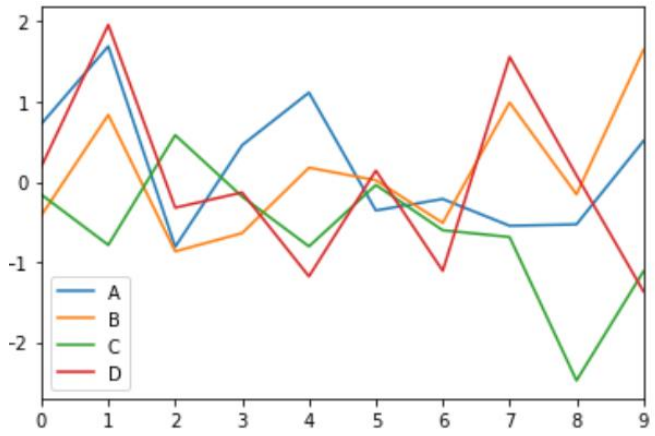
- **DataFrame** class
  - “Una Tabula con Index y Columnas”
  - Valores son un numpy.array

- Viene con muchas funciones útiles para análisis y preprocesamiento de datos
  - read_csv(), read_excel(), …
  - Valores faltantes: isna(), dropna(), fillna()
  - slicing, reshaping, sampling, shuffling, concatenating, …
  - Funciones de matplotlib para visualizaciones rapidas

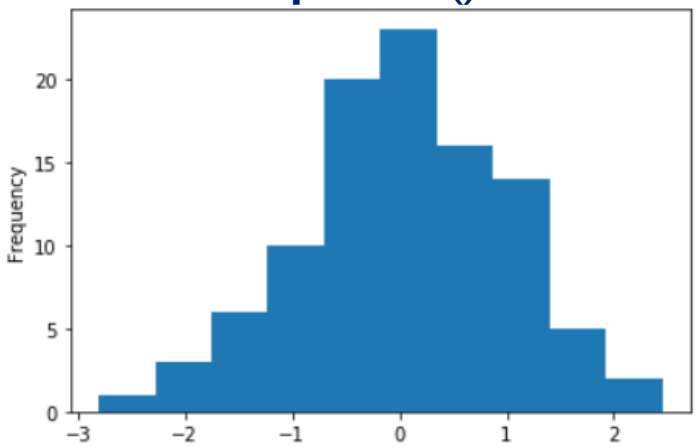
Pandas.DataFrame()

matplotlib

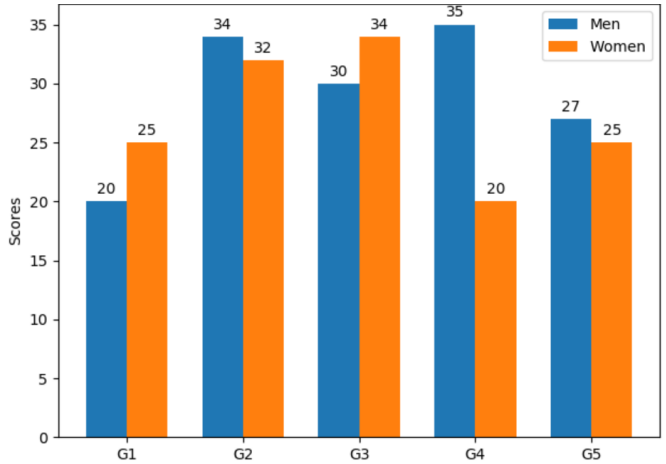
plt.plot()



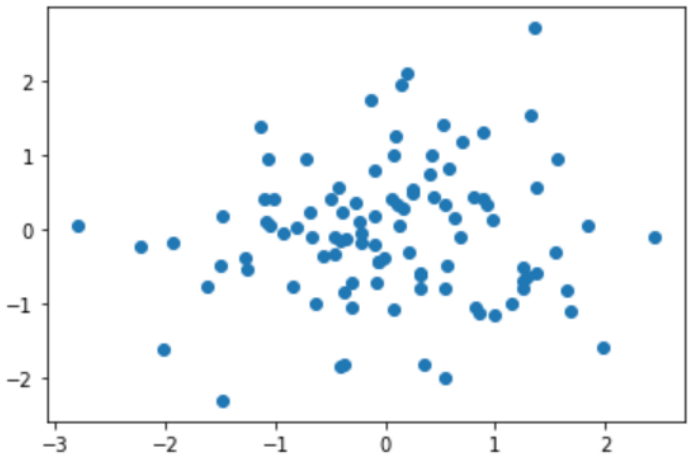
plt.hist()



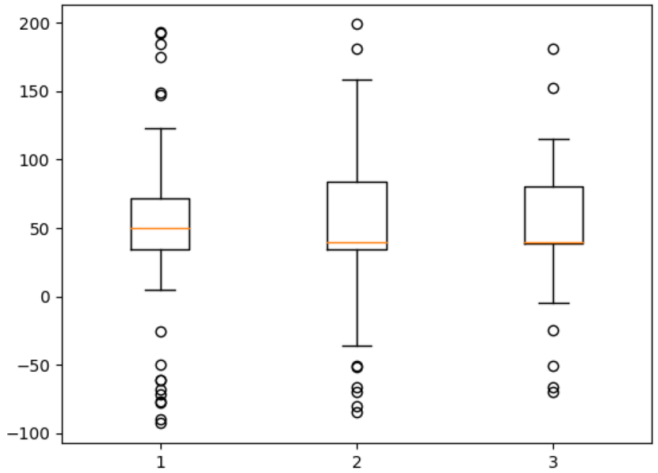
plt.bar()



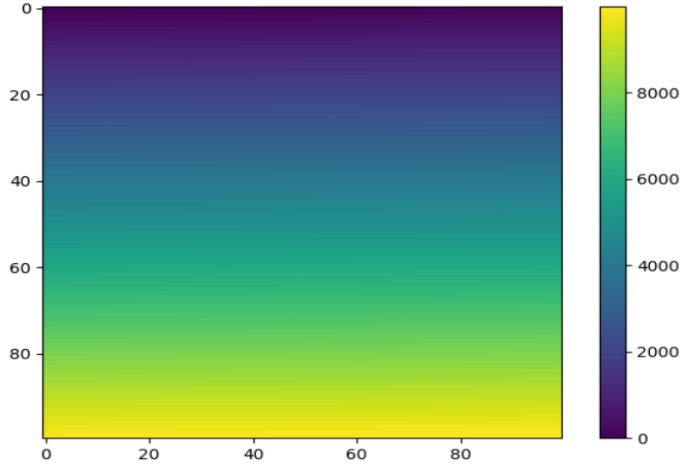
plt.scatter()



plt.boxplot()



plt.imshow()



Numerical vs. Categorical Features

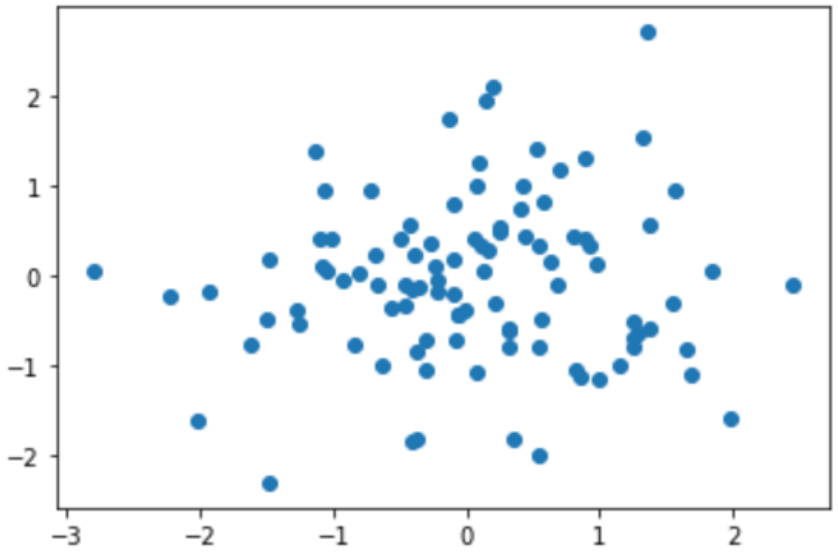
- Features (variables) numéricas requieren otra análisis que Features categóricas
  - E.g. calcular Pearson correlación entre 2 variables categóricas no es posible

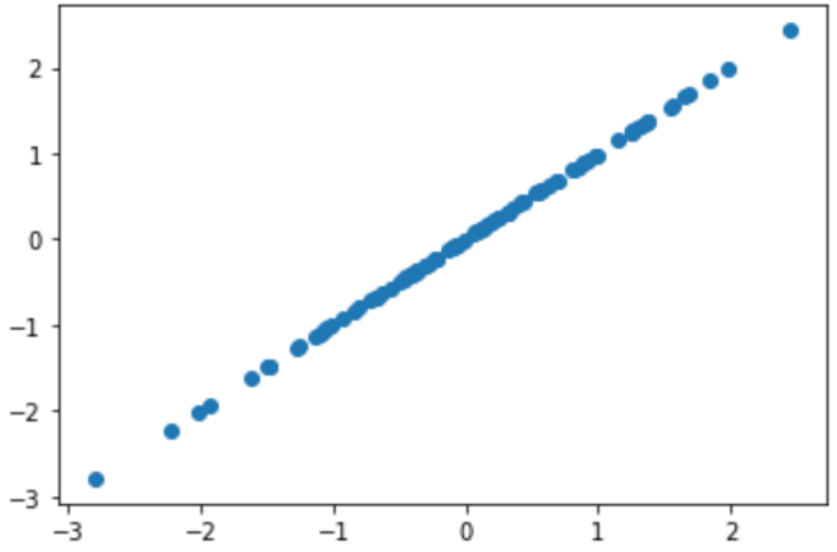
	age	education
0	42	HS-grad
1	33	HS-grad
2	32	Bachelors
3	31	Bachelors
4	30	Some-college
5	31	Preschool
6	18	11th

Correlación entre 2 variables numéricas

- **Medidas estadísticas:**
  - Pearson, Spearman, Mutual Information, …

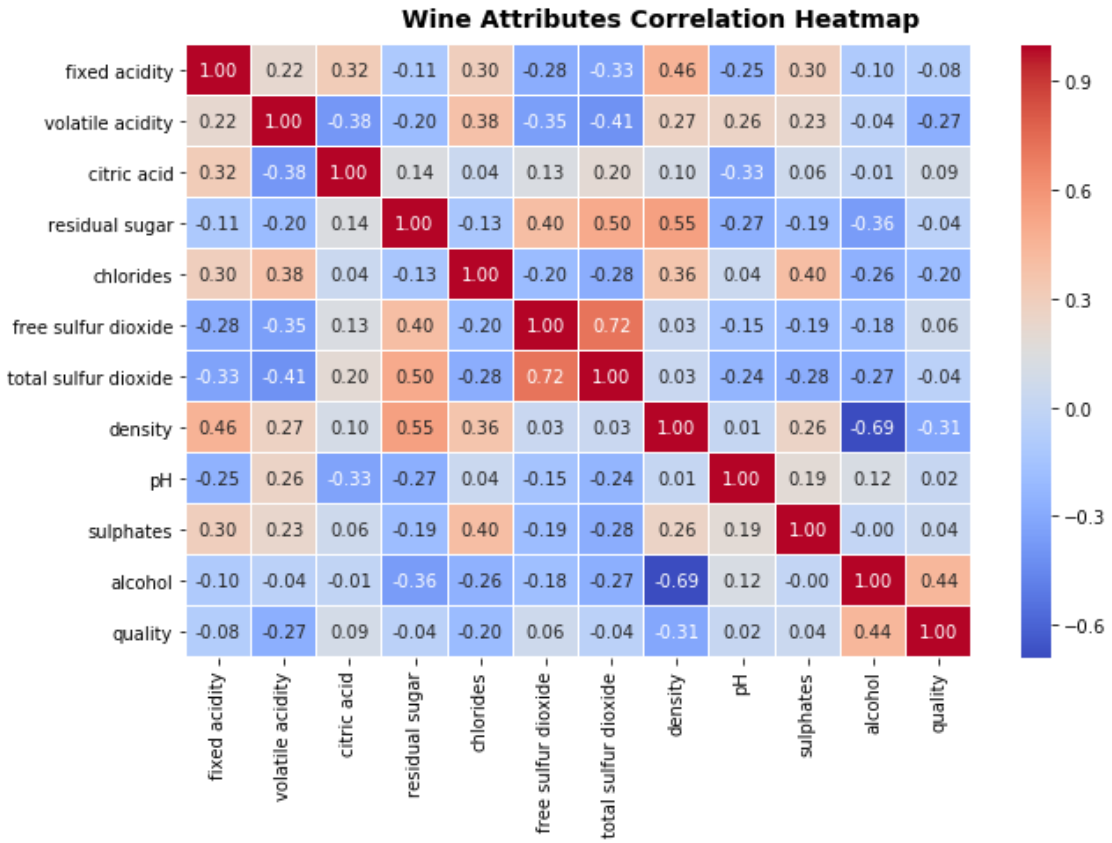
- **Visualizaciones**
  - Scatter-plots (plt.scatter())

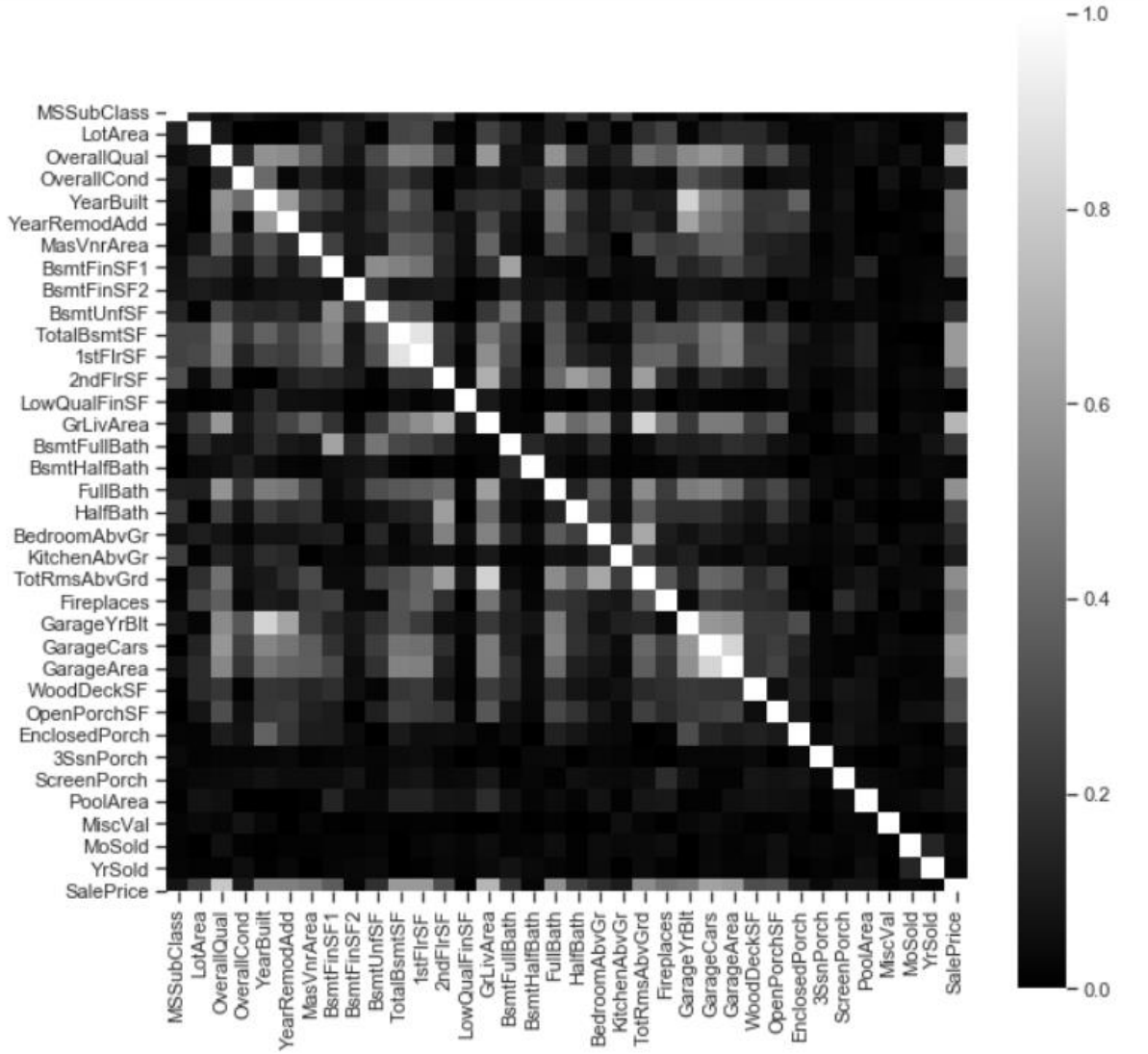




Correlación entre variables numéricas

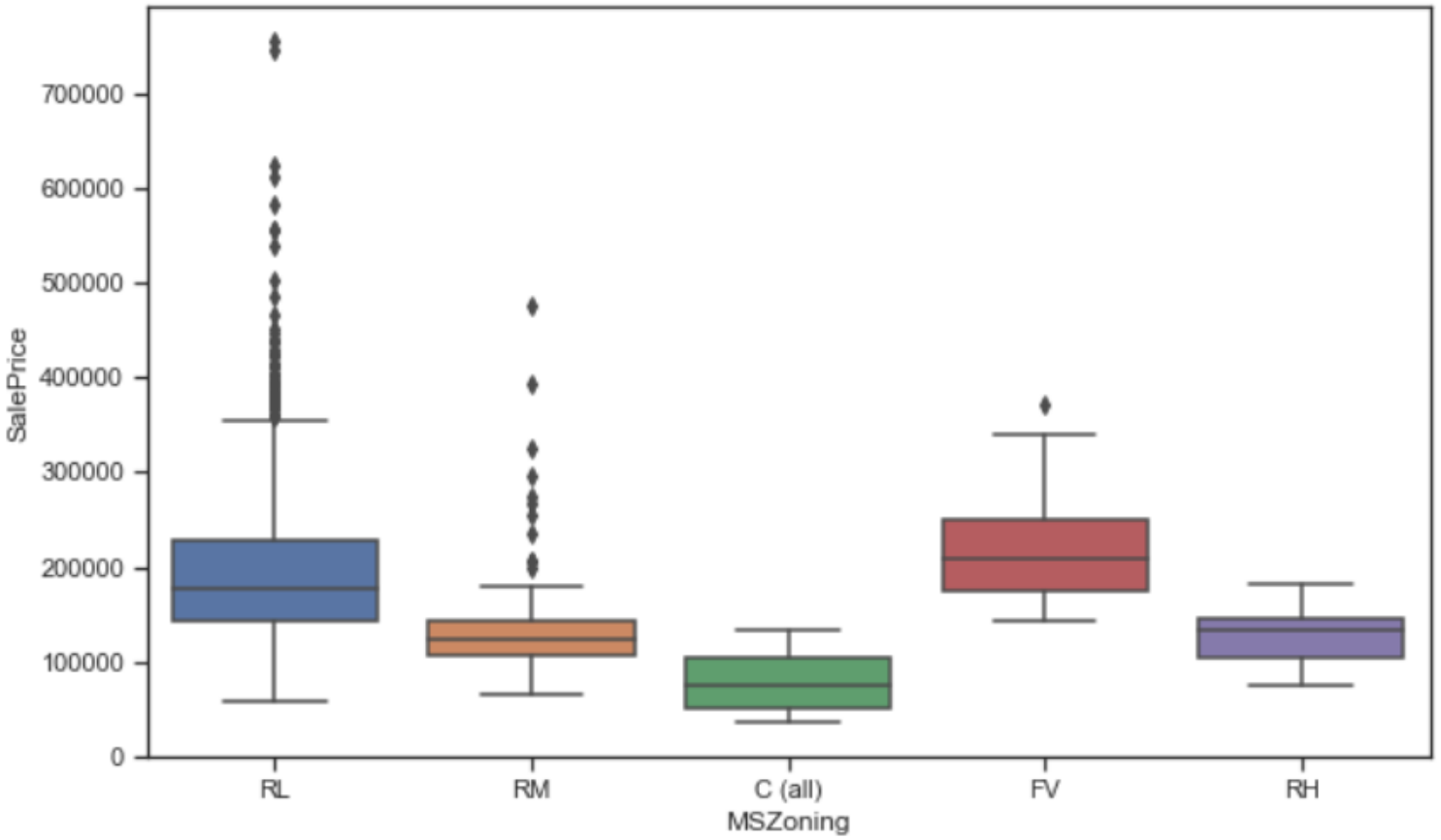
- Heatmap
  - **sns.heatmap**(df.corr().abs())

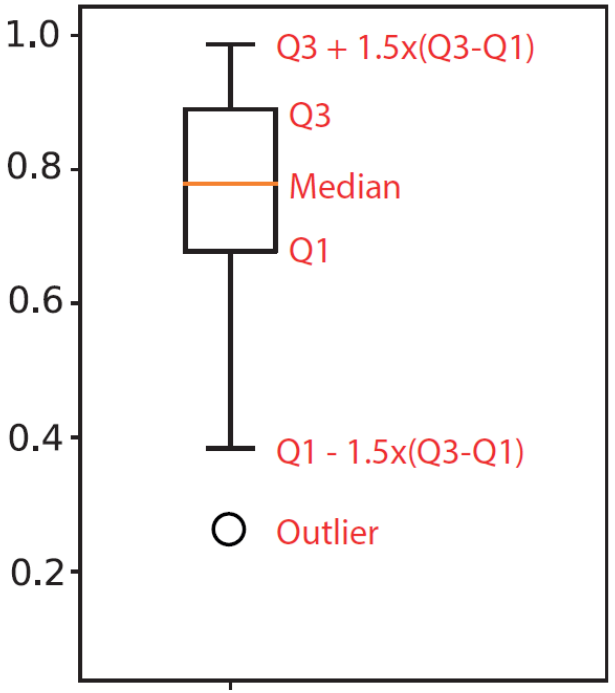




Correlación entre variables numéricas y categóricas

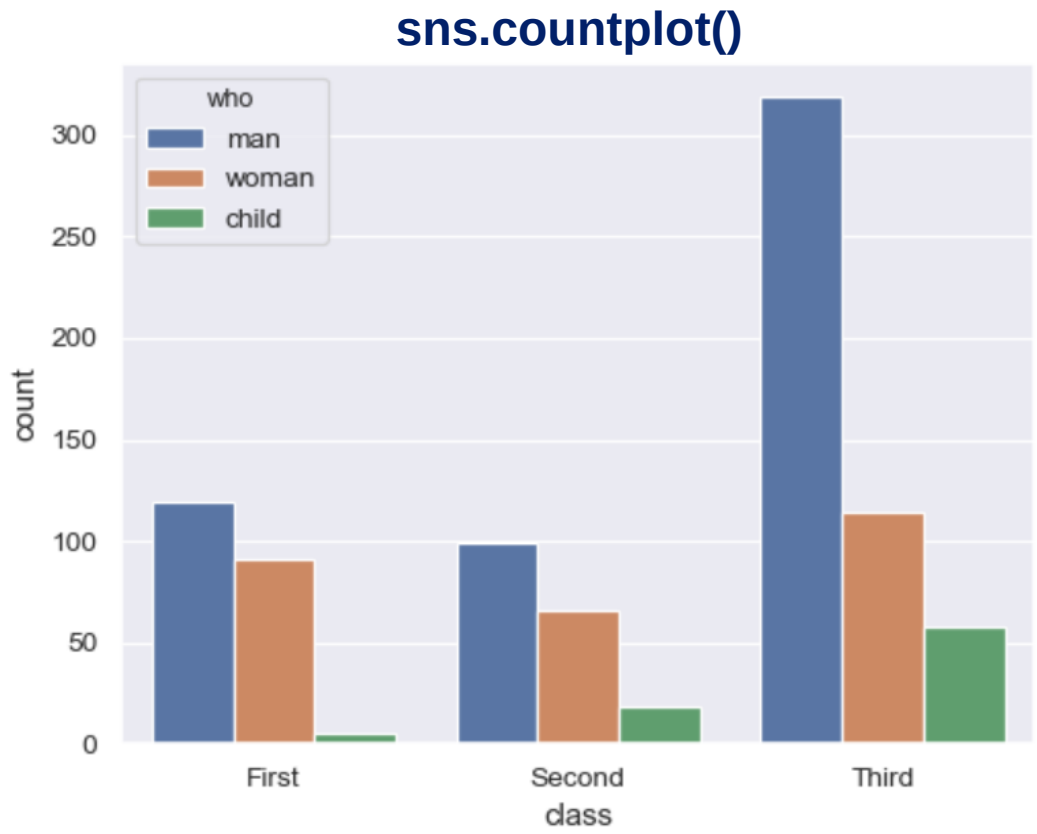
- **Visualizaciones**
  - Box-plots (plt.boxplot())





Correlación entre 2 variables categóricas

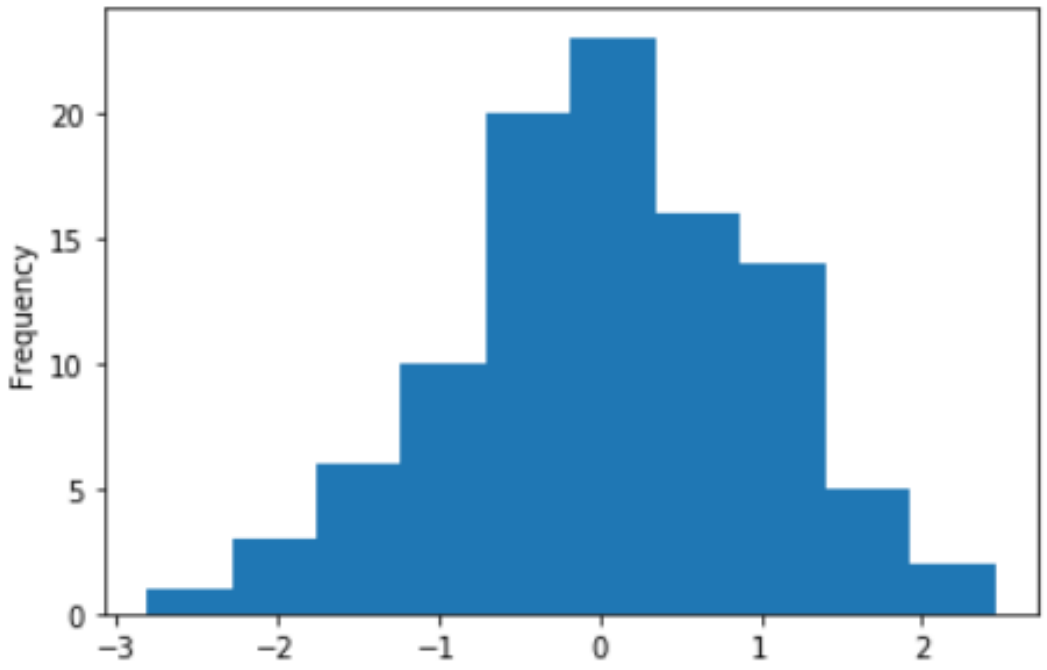
- **Métodos estadísticos**
  - Chi-Squared Test, Cramer's V
  - One-Hot-Encoding → Binary Correlation (Phi coefficient, Jaccard, Dice)
- **Visualizaciones**



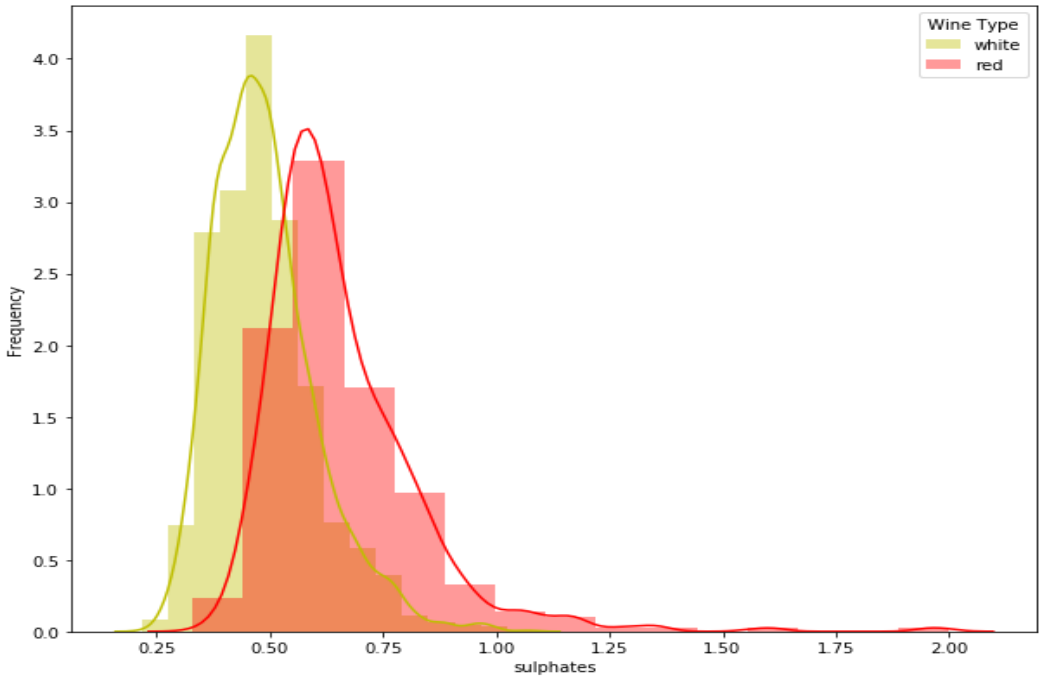
Visualizar distribuciones de variables numéricas

- **Visualizaciones**
  - Histogramas

plt.hist()



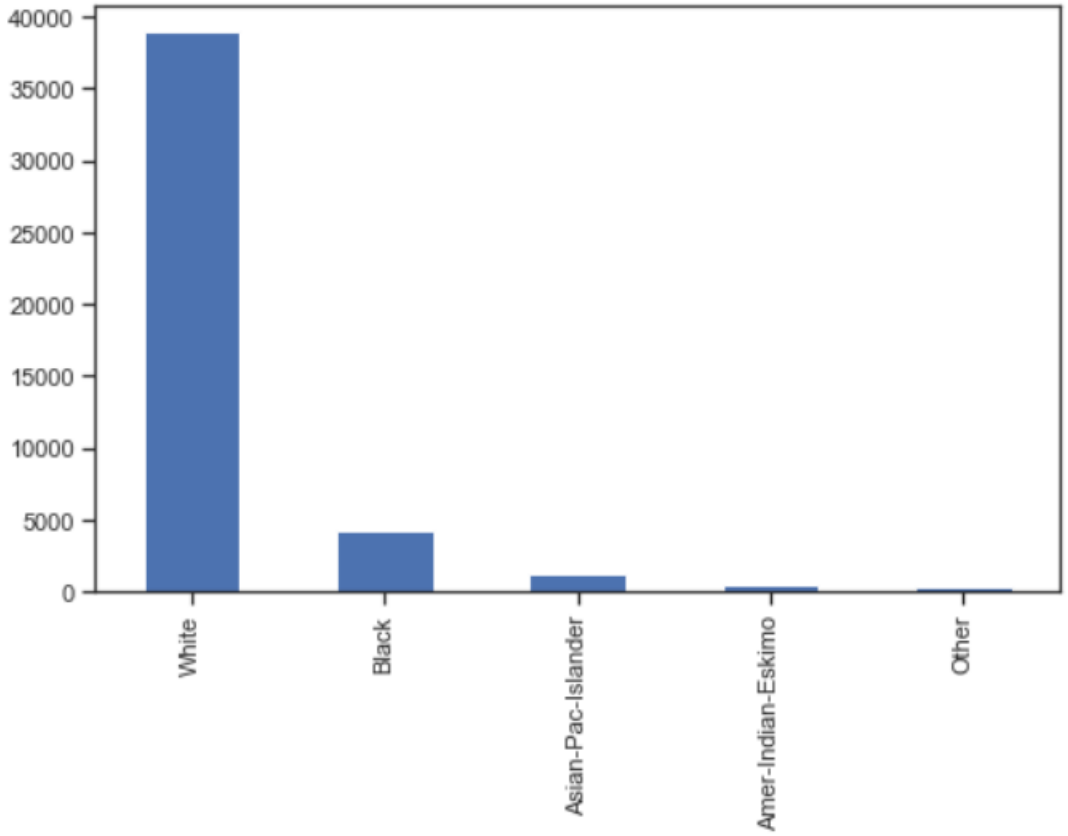
sns.distplot()



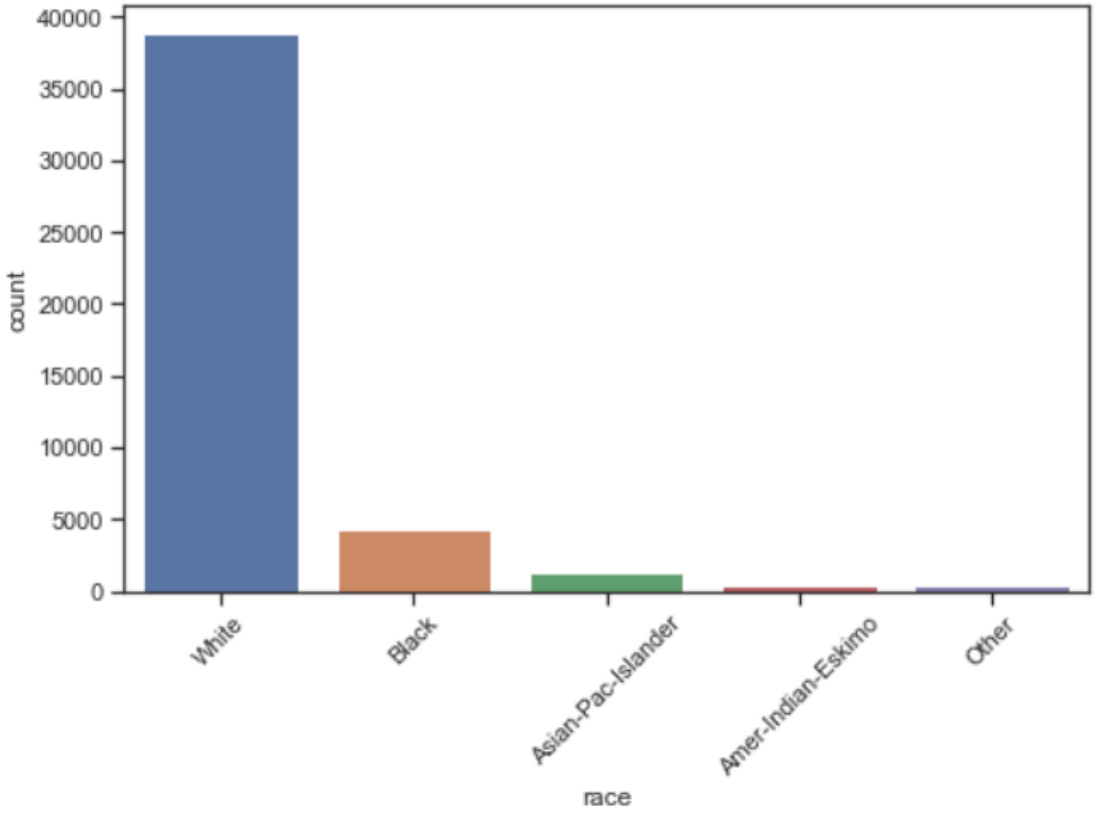
Visualizar distribuciones de variables categóricas

- Contados → Bar-plot

df['race'].value_counts().plot.bar()

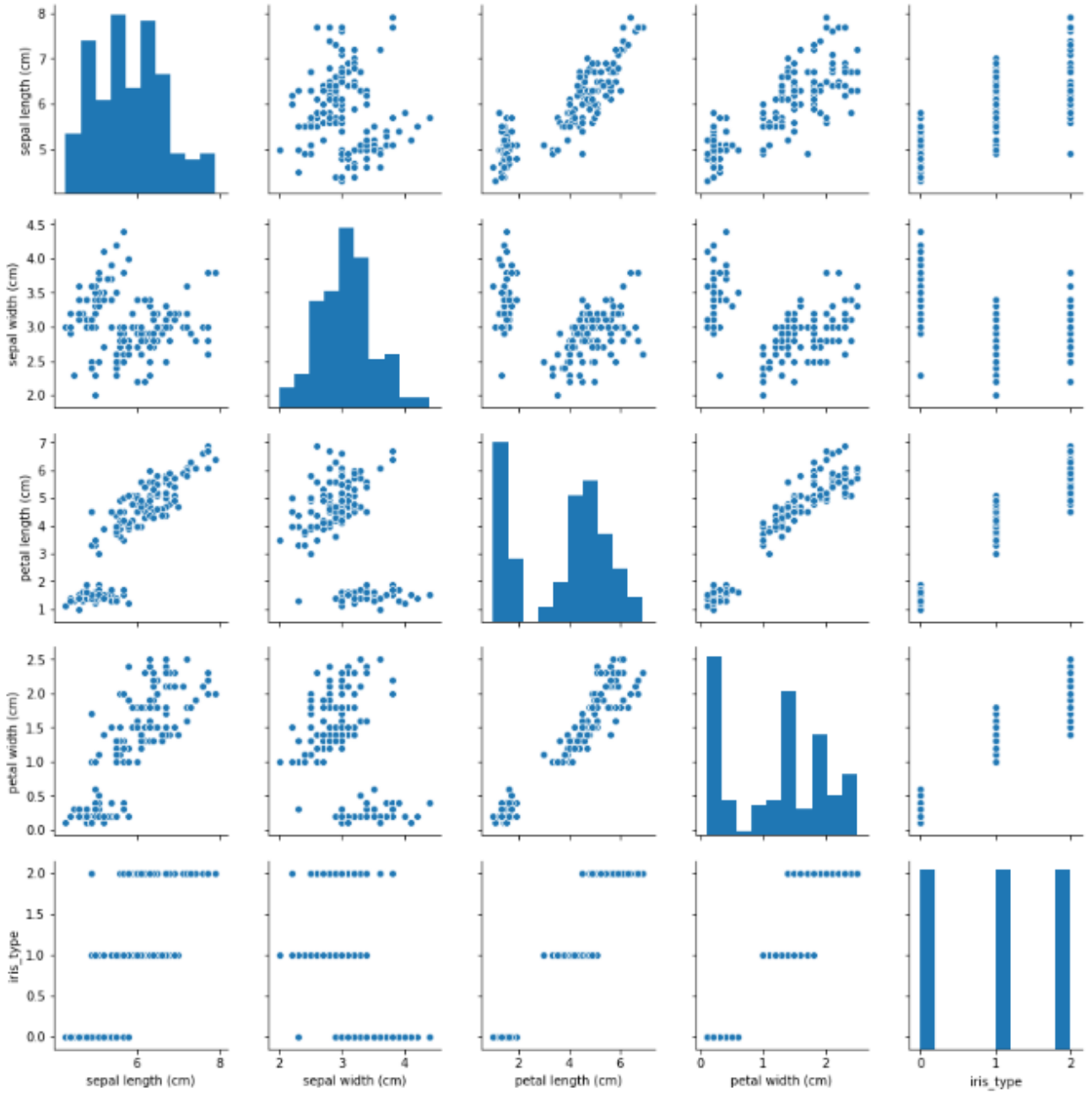


sns.countplot(x="race", data=df)



Seaborn

sns.pairplot(df)



Medidas estadísticas

- Moda
  - Valor con mayor frecuencia en los datos
- Mediana

  1, 3, 3, **6**, 7, 8, 9
- Media

$$\mu = \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

- Varianza

$$\operatorname{Var}(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

- Desviación estándar

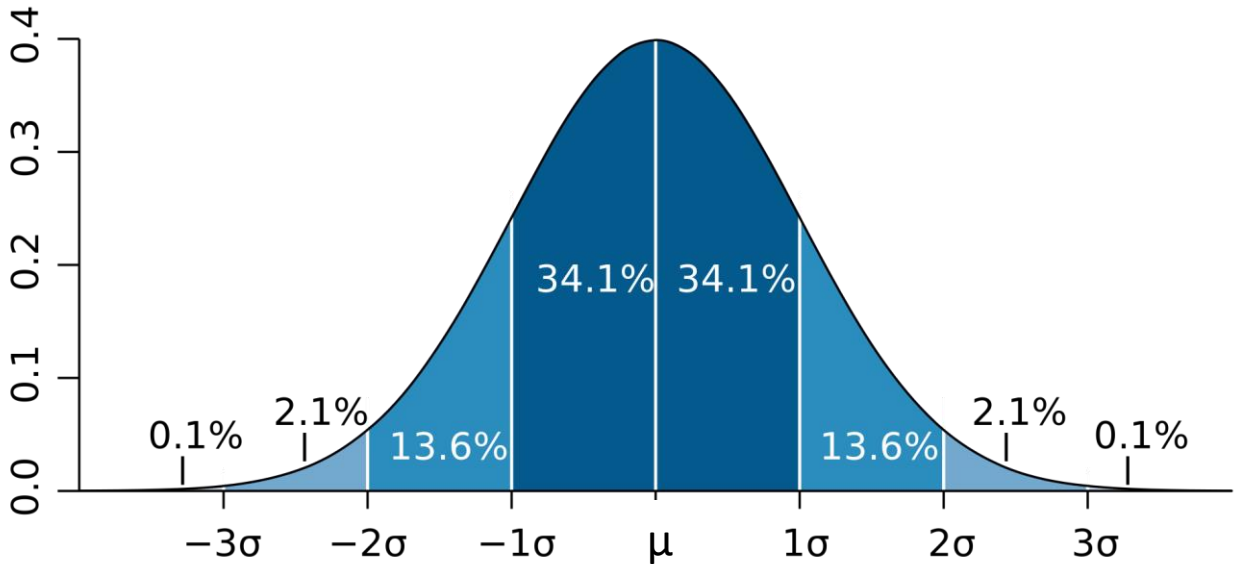
$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Estadística

- Distribución normal

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

  - μ : *La media (valor promedio)*
  - σ : *Desviación estándar*



Estadística v.s. Data Science

- Estadística
  - Pocos datos
    - Difícil tomar conclusiones sobre la distribución original

    → Pruebas de hipótesis, Intervalos de confianza, resultados significantes, ...

- Data Science / Machine Learing
  - Muchos datos
    - Mucho mas fácil tener confianza que los resultados obtenidos son validos para la distribución original

Workshop Repository:
https://github.com/nkaenzig/ml-workshop