PUE 5TH DAY
MADRID 2018

#PUEDAY18

WORKSHOP PYTHON

15:35h a 16:40h

PCAP: Programming Essentials in Python
Una introducción práctica al análisis de datos y machine learning con Python

Ponente: Jordi Ariño, Software Developer Manager at PUE

Networking Academy

PCAP: Programming Essentials in Python

Jordi Ariño

Software Developer Manager at PUE
Agile Developer

jordi.arino@pue.es
@jordiAS2K

#PUEDAY18

Networking Academy

PYTHON INSTITUTE
Open Education & Development Group

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Introducción

Introducción

PCAP: Programming Essentials in Python

Esta workshop se enmarca dentro de la iniciativa **Cisco Networking Academy** y tiene como objetivo trasladar los recursos docentes que el programa pone a disposición de sus centros e instructores para la **formación de alumnos en Python.**

Nuevo curso ***PCAP: Programming Essentials in Python*** disponible en la plataforma NetAcad, y desarrollado para formar alumnos en las habilidades necesarias para la programación con Python.

Networking Academy

PYTHON INSTITUTE
Open Education & Development Group

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

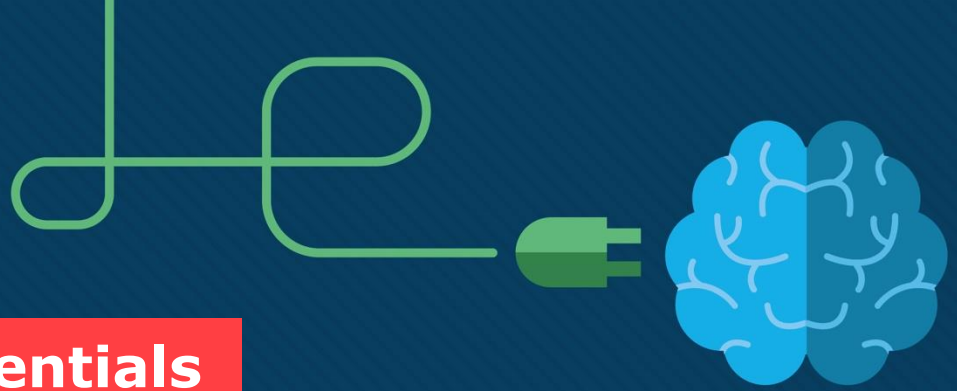
Introducción

PCAP: Programming Essentials in Python

Por otro lado, obtendrás una **visión práctica** de cómo utilizar **Python** para realizar análisis de datos mediante un rápido recorrido sobre las operaciones básicas, que van desde la carga de datos y la exploración gráfica básica, hasta el procesamiento de estos datos.

Para ello utilizaremos una serie de **librerías específicas** para el análisis, visualización y procesamiento de datos: *pandas*, *NumPy*, *seaborn, scikit-learn,*...

PCAP: Programming Essentials in Python



El portafolio educativo de Networking Academy

Colaborar para generar impacto

Introduction to Packet Tracer | Packet Tracer | Hackatones | Prototyping Lab | Prácticas laborales

	Exploratorio	Básico	Profesional
Redes		**Networking Essentials** **Mobility Fundamentals** **Talleres de tecnología emergente:** Programabilidad de redes con Cisco APIC-EM*	**CCNA R&S:** Introduction to Networks, R&S Essentials, Scaling Networks, Connecting Networks **CCNP R&S:** Switch, Route, TShoot
Seguridad	**Introduction to Cybersecurity**	**Cybersecurity Essentials**	**CCNA Cyber Ops** **CCNA Security**
IoT y análisis	**Introduction to IoT**	**IoT Fundamentals:** Connecting Things, Big Data & Analytics, IoT Security* Hackathon Playbook	
SO y TI	**NDG Linux Unhatched**	**NDG Linux Essentials** **IT Essentials**	**NDG Linux I** **NDG Linux II**
Programación		**CLA: Programming Essentials in C** **CPA: Programming Essentials in C++** **PCAP: Programming Essentials in Python** **Talleres de tecnología emergente:** Experimentando con API REST utilizando Cisco Spark*	**CLP: Advanced Programming in C*** **CPP: Advanced Programming in C++**
Laboral	**Be Your Own Boss**	**Entrepreneurship**	
Instrucción digital	**Get Connected**		

El portafolio educativo de Networking Academy

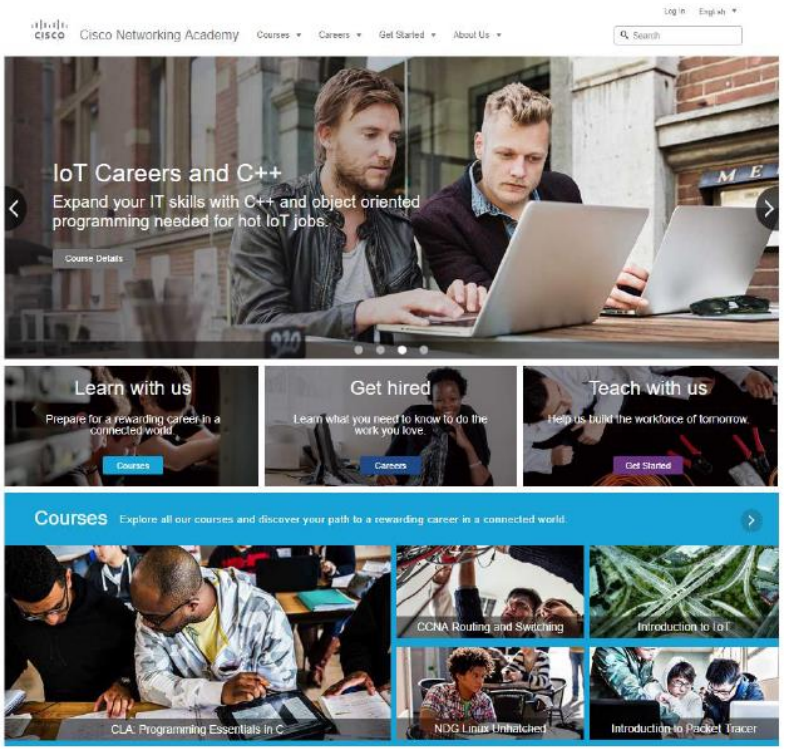
Colaborar para generar impacto

Introduction to Packet Tracer | Packet Tracer | Hackatones | Prototyping Lab | Prácticas laborales

	Exploratorio	Básico	Profesional
Redes		**Networking Essentials** **Mobility Fundamentals** **Talleres de tecnología emergente:** Programabilidad de redes con Cisco APIC-EM*	**CCNA R&S:** Introduction to Networks, R&S Essentials, Scaling Networks, Connecting Networks **CCNP R&S:** Switch, Route, TShoot
Seguridad	**Introduction to Cybersecurity**	**Cybersecurity Essentials**	**CCNA Cyber Ops** **CCNA Security**
IoT y análisis	**Introduction to IoT**	**IoT Fundamentals:** Connecting Things, Big Data & Analytics, IoT Security* Hackathon Playbook	
SO y TI	**NDG Linux Unhatched**	**NDG Linux Essentials** **IT Essentials**	**NDG Linux I** **NDG Linux II**
Programación		**CLA: Programming Essentials in C** **CPA: Programming Essentials in C++** **PCAP: Programming Essentials in Python** **Talleres de tecnología emergente:** Experimentando con API REST utilizando Cisco Spark*	**CLP: Advanced Programming in C*** **CPP: Advanced Programming in C++**
Laboral	**Be Your Own Boss**	**Entrepreneurship**	
Instrucción digital	**Get Connected**		

Programming Essentials in Python

PCAP: Programming Essentials in Python



Networking Academy

PYTHON INSTITUTE
Open Education & Development Group

www.netacad.com

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

PCAP: Programming Essentials in Python

Descripción general del curso

Diseñado para ser un curso fácil de entender y sencillo para principiantes, centrado en diversos tipos de recopilación de datos, herramientas de manipulación y operaciones de lógica y bit, y en la creación de API REST básicas.

Beneficios

Con PCAP: Programming Essentials in Python, aprenderá a diseñar, escribir, depurar y ejecutar programas codificados en el lenguaje Python. No se requiere ningún conocimiento previo de programación. El curso comienza con una orientación básica paso a paso hasta que se vuelva experto en resolver problemas más complejos.

Componentes educativos

- 5 módulos de contenido instructivo interactivo
- Más de 30 prácticas de laboratorio
- Herramienta en línea integrada para realizar prácticas de laboratorio y otras
- Exámenes por capítulo y exámenes finales

Coordinado con la certificación

Características

Público objetivo: Estudiantes de secundaria y en adelante

Requisitos previos: Ninguno

Requiere capacitación a cargo de instructor: No

Idiomas: Inglés

Presentación del curso: Con instructor

Tiempo estimado para completar el curso: 60 a 70 horas

Próximo curso recomendado: IoT Fundamentals, Networking Essentials, NDG Linux Essentials

www.netacad.com/es/courses/programming-python

Programming Essentials in Python

Edube SandBox

Herramienta en línea disponible a través de un navegador web que ofrece un interprete de Python.

https://edube.org/sandbox?language=python

Networking Academy

PYTHON INSTITUTE
Open Education & Development Group

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Sandbox

```python
import random as rnd

def max(op1, op2):
    return op1 if (op1 >= op2) else op2

def min(op1, op2):
    return op1 if (op1 <= op2) else op2


a = rnd.randint(1, 100)
b = rnd.randint(1, 100)
print("MAX({}, {}) = {}".format(a, b, max(a,b)))
print("MIN({}, {}) = {}".format(a, b, min(a,b)))
```

Console >_

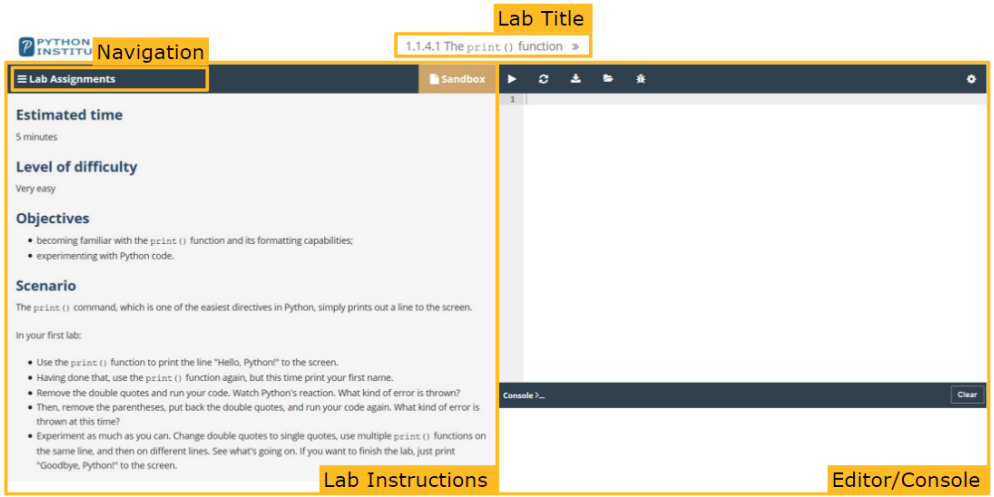
Clear

```
MAX(9, 43) = 43
MIN(9, 43) = 9
```

Programming Essentials in Python

Edube Labs

El curso contiene **más de 30 laboratorios** con soluciones incluidas.



Networking Academy

PYTHON INSTITUTE
Open Education & Development Group

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Estimated time

30-45 minutes

Level of difficulty

Hard

Objectives

- improving the student's skills in operating with strings and lists;
- converting strings into lists.

Scenario

As you probably know, Sudoku is a number-placing puzzle played on a 9x9 board. The player has to fill the board in a very specific way:

- each row of the board must contain all digits from 0 to 9 (the order doesn't matter)
- each column of the board must contain all digits from 0 to 9 (again, the order doesn't matter)
- each of the nine 3x3 "tiles" (we will name them "sub-squares") of the table must contain all digits from 0 to 9.

If you need more details, you can find them here.

Your task is to write a program which:

- reads 9 rows of the Sudoku, each containing 9 digits (check carefully if the entered data are valid)
- outputs "Yes" if the Sudoku is valid, and "No" otherwise.

Test your code using the data we've provided.

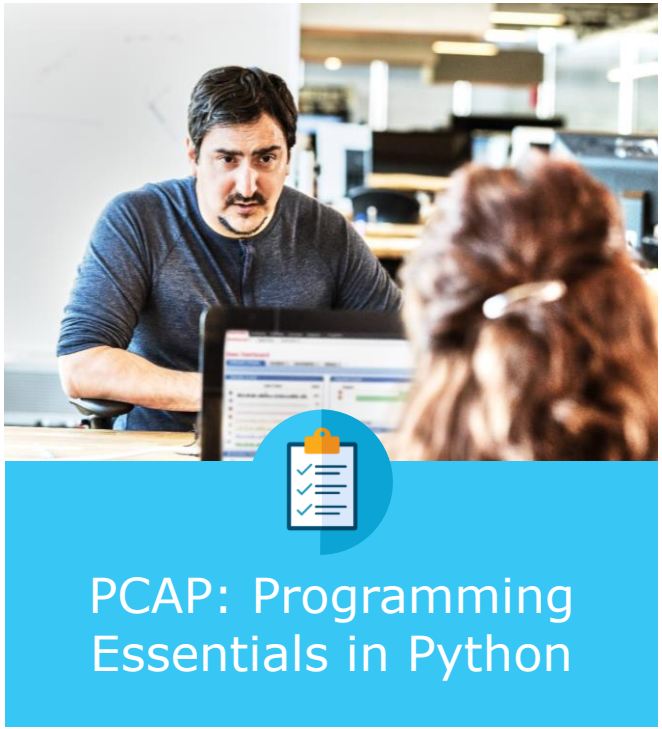
Example input
295743861 431865927 876192543 387459216 612387495 549216738 763524189 928671354 154938672

Example output
Yes

Example input
195743862 431865927 876192543 387459216 612387495 549216738 763524189 928671354 254938671

Example output
No

Programming Essentials in Python



Introduction

Module 0

0.1 Programming – absolute basics
- How does a computer program work?
- Natural languages vs. programming languages
- Compilations vs. interpretation

0.2 Python – a tool, not a reptile
- What is Python?
- Who is Python's creator?
- Why Python?
- Why not Python?

0.3 There is more than one Python
- Python 2 vs. Python 3
- Python aka Cpython
- Cython
- Jython
- PyPy and Rpython

Programming Essentials in Python

Introduction

Module 0

0.4 Begin your Python journey
- How to get Python and how to get to use it
- How to write and run your very first Python program
- How to spoil and fix your code

0.5 Edube Sandbox and Labs

Networking Academy
PYTHON INSTITUTE
Open Education & Development Group
pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Programming Essentials in Python

Part I: Basics

Module 1: Basics I

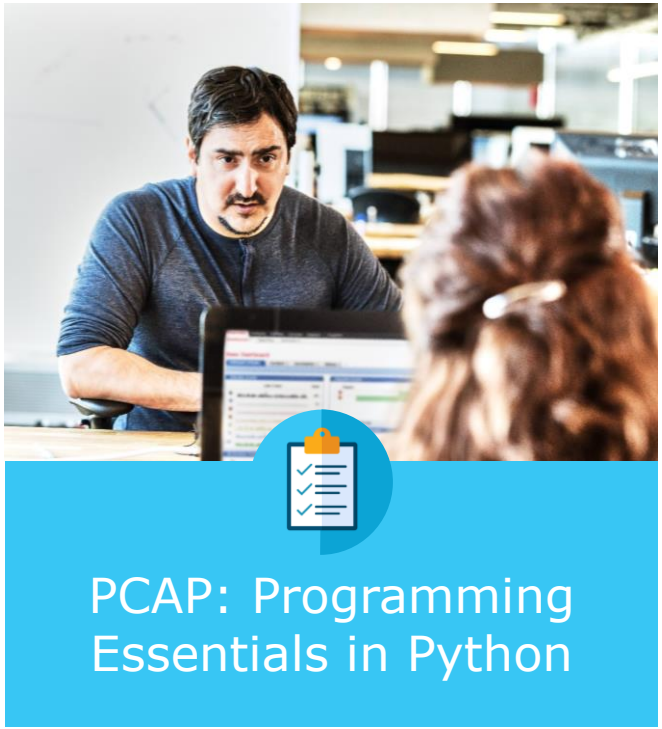
1.1 Your first program
- Your first program
- The print() function – how the computer talks to you
- The print() function – formatting the output

1.2 Python literals
- Literals – what they are anyway?
- Literals – integers
- Literals – floats
- Literals – strings
- Literals – Boolean values

1.3 Operators – data manipulation tools
- Operators and expressions
- Arithmetic operators
- Operators and their priorities
- Operators and their bindings

Networking Academy
PYTHON INSTITUTE
Open Education & Development Group
pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Programming Essentials in Python



Part I: Basics

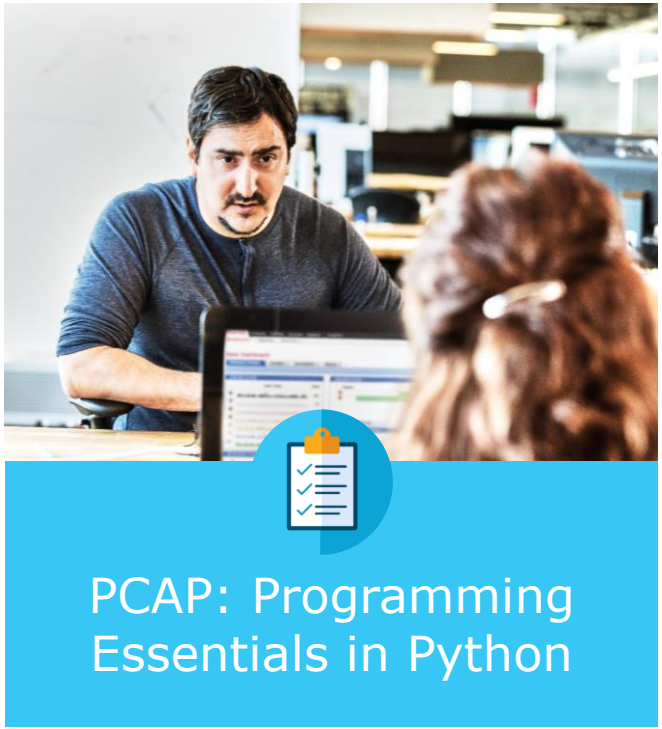
Module 1: Basics I

1.4 Variables – data-shaped boxes
- Variables – how to name them
- Variable names vs. Python keywords
- How to assign a variable
- How to comment your code
- Shortcut operators

1.5 How to talk to computer?
- Output vs. input
- How to input data with the input() function
- How to convert strings into numbers
- Some simple interactive programs
- String operators
- How to convert numbers into strings

Networking Academy
PYTHON INSTITUTE
Open Education & Development Group
pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Programming Essentials in Python



Part I: Basics

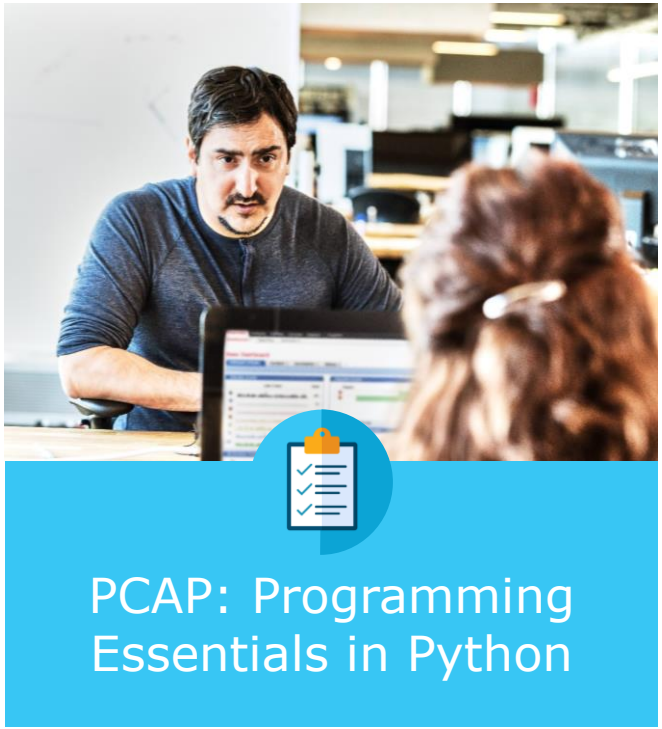
Module 2: Basics II

2.1 Making decisions in Python
- How to ask questions and how to get answers
- Relational operators
- Making use of the answers
- Conditions and conditional execution – the if statements
- How indentation makes the code
- The more conditional execution – if-else statements
- The elif clause
- Some simple examples

2.2 Python's loops
- Looping your code with while
- Looping your code with for
- Controlling your loops with break and continue

Networking Academy
PYTHON INSTITUTE
Open Education & Development Group
pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Programming Essentials in Python



Part I: Basics

Module 2: Basics II

2.3 Logic and bit operations in Python
- Computer logic and its operators
- Logical values vs. single bits
- Bitwise operators
- How to deal with single bits

2.4 Lists – collections of data
- Lists – why do we need them so much?
- How to create a list
- How to use a list
- Removing elements from a list
- How not to use a list
- List methods – methods vs. functions
- Adding elements to lists
- Making use of lists
- The second face of the for loop
- Lists in action

Networking Academy
PYTHON INSTITUTE
Open Education & Development Group
pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Programming Essentials in Python

Part I: Basics

Module 2: Basics II

2.5 Sorting simple lists – the bubble sort algorithm

2.6 Lists – some more details
- How lists are stored
- Slices – the powerful tools
- The in and not in operators

2.7 Lists in advanced application
- Lists in lists
- The list comprehension: why and how
- Lists in lists – matrices
- Lists in lists – 3rd dimension

Programming Essentials in Python

Part I: Basics

Module 3: Basics 3

3.1 Writing functions in Python
- Functions: why do we need them?
- Where do functions come from?
- Your first function

3.2 How functions communicate with their environment
- Parametrized functions
- How to define and use function parameters
- What is shadowing?
- Positional arguments
- Keyword arguments
- Mixed arguments
- Setting parameters' default values

Programming Essentials in Python

PCAP: Programming Essentials in Python

Part I: Basics

Module 3: Basics 3

3.3 Returning a result from a function
- A function's effects and results – the return statement
- Returning a value
- The None value
- Returning the non-None value
- Argument vs. parameter compatibility
- A list as a function's result

3.4 Scopes in Python
- Functions and scopes
- How do scopes work?
- How to make a variable global
- How the parameters interact with their arguments

Networking Academy

PYTHON INSTITUTE
Open Education & Development Group

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Programming Essentials in Python

Part I: Basics

Module 3: Basics 3

3.5 Creating functions
- Some exercises with designing and writing functions
- Recursion – how to make a function more powerful?

3.6 Tuples and dictionaries
- Sequence types and mutability
- What is a tuple?
- How to create a tuple
- How to use a tuple
- What is a dictionary?
- How to make a dictionary
- How to use a dictionary
- How a dictionary and a tuple can work together

Networking Academy
PYTHON INSTITUTE
Open Education & Development Group
pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Programming Essentials in Python

Part II: Intermediate

Module 4: Intermediate I

4.1 Using modules
- Using modules
- What is a module?
- How to make use of a module?
- Importing a module

4.2 Some useful modules
- Working with standard modules
- Some functions from the math module
- Some functions from the random module
- Some functions from the platform module

4.3 What is a package?
- Modules and packages
- Your first module
- Your first package

Networking Academy
PYTHON INSTITUTE
Open Education & Development Group
pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Programming Essentials in Python

Part II: Intermediate

Module 4: Intermediate I

4.4 Errors – a programmer's daily bread
Errors, failures, and other plagues
Exceptions

4.5 The anatomy of an exception

4.6 Some of the most useful exceptions

4.7 Characters and strings vs. computers

4.8 The nature of Python's strings

4.9 String methods

Networking Academy

PYTHON INSTITUTE
Open Education & Development Group

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Programming Essentials in Python

Part II: Intermediate

Module 4: Intermediate I

4.10 Strings in action
- Comparing strings
- Sorting strings, and not only strings
- Strings vs. numbers

4.11 Four simple programs
- Caesar's cipher – the coder
- Caesar's cipher – the decoder
- Extracting numbers from a line of text
- Checking the IBAN

Networking Academy
PYTHON INSTITUTE
Open Education & Development Group
pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Programming Essentials in Python

Part II: Intermediate

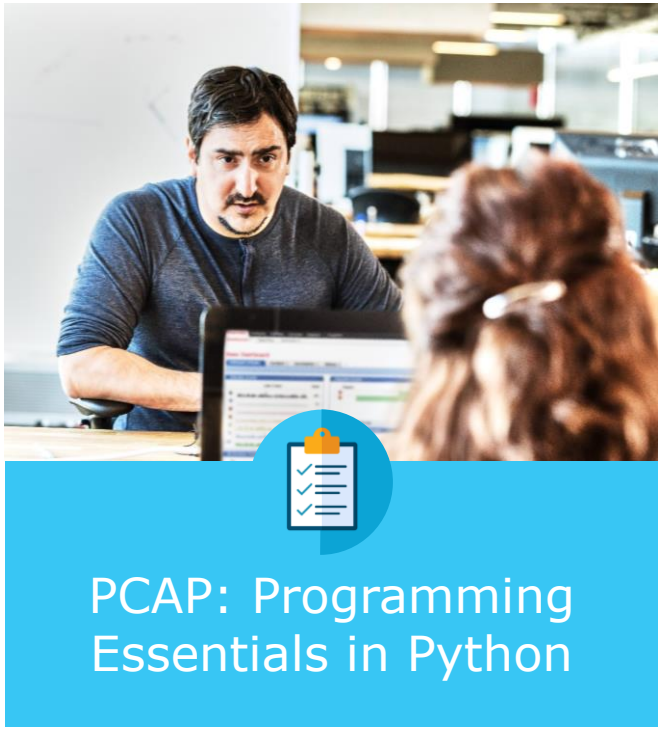
Module 5: Intermediate II

5.1 Basic concepts of object programming
- What is an object?
- The object – what is it again?
- What does an object have?
- Your first class

5.2 A short journey from the procedural to the object approach
- What is a stack?
- The stack – a procedural approach
- The stack from scratch

5.3 Properties
- Properties in detail
- Instance variables
- Class variables
- Checking an attribute's existence

Programming Essentials in Python



Part II: Intermediate

Module 5: Intermediate II

5.4 Methods
- Methods in detail
- The inner life of classes and objects
- Reflection and introspection – two names of the same phenomenon
- Investigating classes – what can we find out about them?

5.5 Inheritance – one of object programming foundations
- Inheritance – why and how
- How Python finds properties and methods
- How to build a hierarchy of classes
- Inheritance vs. composition
- Single inheritance vs. multiple inheritance
- Diamonds and why you don't want them
- Checking an attribute's existence

cisco Networking Academy
PYTHON INSTITUTE Open Education & Development Group
pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Programming Essentials in Python

Part II: Intermediate

Module 5: Intermediate II

5.6 Exceptions once again
- Exceptions are classes
- Detailed anatomy of an exception
- How to create your own exception
- How to use your own exception

5.7 Generators and closures
- Generators – where to find them
- The yield statement
- How to build your own generator
- More about list comprehensions
- The lambda function
- How and when to use lambdas

Networking Academy

PYTHON INSTITUTE
Open Education & Development Group

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Programming Essentials in Python

PCAP: Programming Essentials in Python

Part II: Intermediate

Module 5: Intermediate II

5.8 Processing files
- Accessing files from Python code
- File names
- File streams
- File handles
- Opening the streams
- Selecting text and binary modes
- Opening the stream for the first time
- Pre-opened streams
- Closing streams
- Diagnosing stream problems

5.9 Working with real files
- Dealing with text files
- How to work with binary files
- How to read bytes from the stream
- How to write bytes from the stream
- Copying files – a simple, but functional tool

Networking Academy

PYTHON INSTITUTE
Open Education & Development Group

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Programming Essentials in Python

Modelo de evaluación

El curso contiene una serie de *quizzes* y *tests de evaluación* para cada módulo y sección.

Además existe un **test de evaluación final**.

Module 1 Test 20 ptos. | 20 preguntas	Module 5 Test 30 ptos. | 30 preguntas
Module 2 Test 20 ptos. | 20 preguntas	Summary Test 1 30 ptos. | 30 preguntas
Module 3 Test 20 ptos. | 20 preguntas	Summary Test 2 30 ptos. | 30 preguntas
Module 4 Test 20 ptos. | 20 preguntas	Final Test 50 ptos. | 50 preguntas

Networking Academy

PYTHON INSTITUTE
Open Education & Development Group

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

PYTHON INSTITUTE

Module 4 Quiz

This quiz will help you prepare for the Module 4 Test. You will have 25 minutes to answer 10 questions. Click *Start Quiz* to begin. Good luck!

Start Quiz

Question 2 of 10 | Point Value: 1 | Total Points: 2 out of 10 | 24:40

If you want to import *pi* from *math*, you'd use:

- from pi import math
- import pi from math
- from math import pi

Correct

That's right! You answered correctly.

Continue

Module 4 Test

Esta es una vista previa de la versión publicada de la evaluación

Comenzado: 13 abr en 23:40

Instrucciones de la evaluación

Number of questions: **20**

Time limit: **30 minutes**

Allowed attempts: **3**

See Correct Answers: **Yes** (after the last attempt)

Points to score: **20** (each question is worth 1 point)

Passing score: **No**

Pregunta 1 **1 ptos.**

Knowing that a function named `fun()` resides in a module named `mod`, choose the proper way to import it:

- `from mod import fun`
- `import fun`
- `from fun import mod`
- `import fun from mod`

Siguiente ▸

Seguir editando esta evaluación

Preguntas

- Pregunta 1
- Pregunta 2
- Pregunta 3
- Pregunta 4
- Pregunta 5
- Pregunta 6
- Pregunta 7
- Pregunta 8

Tiempo de ejecución: Esconder
29 minutos, 49 segundos

Programming Essentials in Python

Certificación asociada

PCAP – Certified Associate in Python Programming

Exam Version	PCAP-31-01
Exam Length	65 minutes (exam) + 10 minutes (NDA/Tutorial)
Number of Questions	40
Format	Single-choice/Multiple-choice
Passing Score	80%
Exam Status	Available April 2018

https://pythoninstitute.org

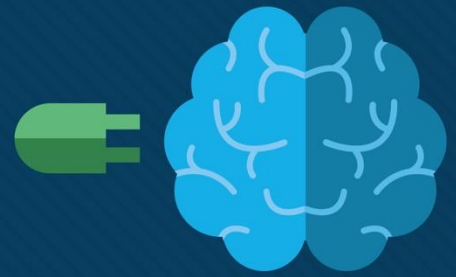
Es una credencial profesional que mide la capacidad de realizar tareas de codificación relacionadas con los **aspectos básicos de la programación en el lenguaje Python**, y las nociones fundamentales utilizadas en la **programación orientada a objetos**.

Networking Academy

PYTHON INSTITUTE
Open Education & Development Group

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

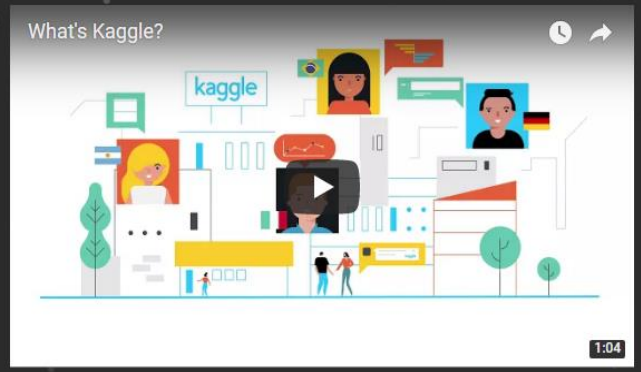
Hands-On Labs



kaggle Search kaggle Competitions Datasets Kernels Discussion Learn ••• Sign In

The Home of Data Science & Machine Learning

Kaggle helps you learn, work, and play



jobs board >

Create an account or Host a competition

Competitions ›

Climb the world's most elite machine learning

Datasets ›

Explore and analyze a collection of high quality

Kernels ›

Run code in the cloud and receive community feedback

Hands-On Labs

Titanic: Machine Learning from Disaster

kaggle

www.kaggle.com/c/titanic

Kaggle ofrece una serie de **competiciones** diseñadas para principiantes. La más popular de estas competiciones consiste en predecir qué pasajeros sobrevivieron al **hundimiento del Titanic**.

En esta competición, tenemos disponible un **conjunto de datos** con distinta información (*features*) sobre los pasajeros a bordo del Titanic.

Nuestro **objetivo** es, usando esta información de la que disponemos, **predecir si un pasajero sobrevivió o no** al hundimiento del Titanic.

Nuestra puntuación será el porcentaje de pasajeros que pronosticamos correctamente (*"accuracy" o precisión).*

Networking Academy

PYTHON INSTITUTE
Open Education & Development Group

Hands-On Labs

Titanic: Machine Learning from Disaster

kaggle

www.kaggle.com/c/titanic

Cada competición en *Kaggle* ofrece dos conjuntos de datos clave con los que se trabajará: un *conjunto de entrenamiento* (**training set**) y un *conjunto de pruebas* (**testing set**).

train.csv

test.csv

Gender Submission.csv

Networking Academy

PYTHON INSTITUTE
Open Education & Development Group

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Hands-On Labs

Training Set: Conjunto de entrenamiento

kaggle

www.kaggle.com/c/titanic

El **conjunto de entrenamiento** contiene los datos que podemos usar para construir nuestro **modelo de Machine Learning**.

Este conjunto está formado por una serie de *características* o *features* que ofrecen datos descriptivos sobre los pasajeros a bordo del Titanic.

Nuestro modelo estará basado en ***features*** como el género de los pasajeros o la clase en la que viajaban.

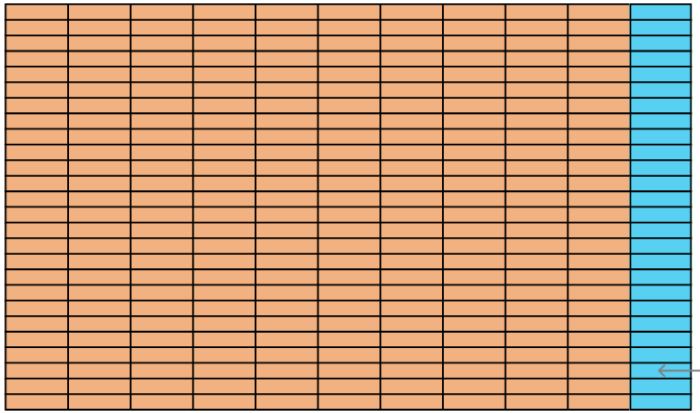
Finalmente ofrece una característica destinada al valor que queremos predecir: en este caso, si el pasajero sobrevivió o no al hundimiento del Titanic.

Hands-On Labs

Training Set: Conjunto de entrenamiento

kaggle

www.kaggle.com/c/titanic



Podemos usar ***feature engineering*** para crear nuevas características a partir de las existentes.

Hands-On Labs

Testing Set: Conjunto de pruebas

kaggle

www.kaggle.com/c/titanic

El **conjunto de pruebas** contiene la misma estructura de datos que el conjunto de entrenamiento, excepto la columna destinada al valor que queremos predecir.

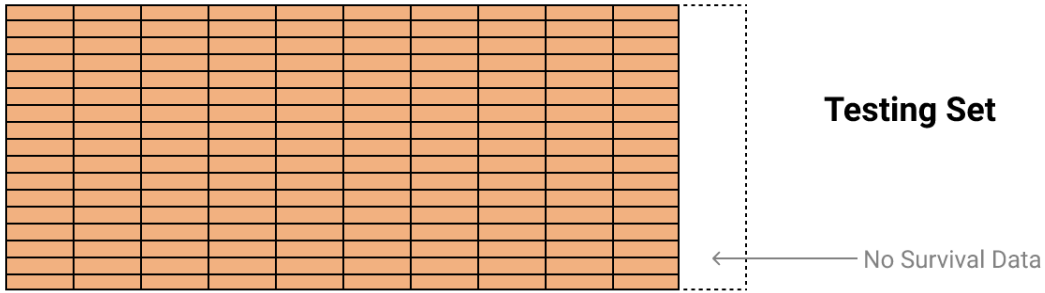
Además, generalmente, tiene menos **observaciones** (filas) que el conjunto de entrenamiento.

Será nuestra misión predecir ese valor, y saber si el pasajero sobrevivió o no al hundimiento del Titanic.

Para ello construiremos y entrenaremos un **modelo de Machine Learning** a partir del conjunto de entrenamiento, y usaremos ese modelo para hacer predicciones sobre los datos del conjunto de pruebas.

Hands-On Labs

Testing Set: Conjunto de pruebas



kaggle

www.kaggle.com/c/titanic

Para cada pasajero del **conjunto de pruebas**, usaremos el **modelo de Machine Learning** que hemos entrenado para predecir si sobrevivió o no al hundimiento del Titanic.

Hands-On Labs

Titanic: Machine Learning from Disaster

kaggle

www.kaggle.com/c/titanic

Es nuestro trabajo predecir si un pasajero sobrevivió o no al hundimiento del Titanic.

Para cada ID de pasajero en el **conjunto de pruebas**, debemos predecir un valor de 0 o 1 para la variable *"Survived"*.

Como resultado generaremos un archivo en formato csv con exactamente 418 filas –*una por pasajero*- más una fila de encabezado.

El archivo debe tener el siguiente formato:

```
PassengerId,Survived
 892,0
 893,1
 894,0
 Etc.
```

Networking Academy

PYTHON INSTITUTE
Open Education & Development Group

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Hands-On Labs

Titanic: Machine Learning from Disaster

kaggle

www.kaggle.com/c/titanic

Pasos a seguir

1. Acquire training and testing sets
2. Analyze and explore the data
3. Visualize the data and identify patterns
4. Prepare data for Machine Learning
5. Train a Machine Learning Model
6. Measure the accuracy of your model
7. Make your first Kaggle submission

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

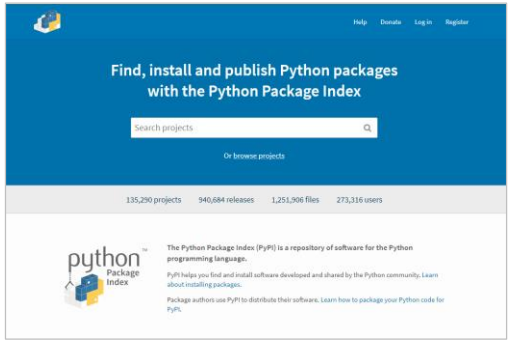
NumPy
- **Sitio web:** http://www.numpy.org
- **Install:** `pip install numpy`

pandas
- **Sitio web:** https://pandas.pydata.org
- **Install:** `pip install pandas`

matplotlib
- **Sitio web:** https://matplotlib.org/
- **Install:** `pip install matplotlib`

seaborn
- **Sitio web:** https://seaborn.pydata.org
- **Install:** `pip install seaborn`

scikit-learn
- **Sitio web:** http://scikit-learn.org
- **Install:** `pip install scikit-learn`



Python Package Index

https://pypi.org/

Networking Academy

PYTHON INSTITUTE
Open Education & Development Group

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load in

# Data Analysis and Manipulation
import random as rnd # pseudo-random number generators
import numpy as np   # linear algebra
import pandas as pd  # data processing, CSV file I/O (e.g. pd.read_csv)

# Data Visualization
import matplotlib.pyplot as plt # 2D plotting library
import seaborn as sns           # Statistical data visualization

# Machine Learning: Modelling Algorithms
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

# Machine Learning: Helpers
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import accuracy_score

# Others
import os # Operating system funcionalities

# Configure visualisations
sns.set_style("whitegrid")       # Default seaborn style sheet. Values: darkgrid, whitegrid, dark, white, ticks
sns.despine()                    # Remove the axis spines for the top and right axes

# Magic command/function to render the plots in the notebook itself
%matplotlib inline
```

```
Directory: input
['train.csv', 'test.csv', 'gender_submission.csv']
--------------------------------------------------------------------------------
```

Networking Academy

IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

```
# In this competition, the two files are named test.csv and train.csv.
# We'll start by using pandas.read_csv() library to read both files and then inspect their size.
train = pd.read_csv("../input/train.csv", sep=",")
test = pd.read_csv("../input/test.csv", sep=",")
titanic = [train, test]

print("TRAINING SET")
print("-"*50)
trainDimensions = train.shape;
print("Type: {}".format(type(test)))
print("Dimensions: {}".format(trainDimensions))
print("Number of rows: {}".format(trainDimensions[0]))
print("Number of columns: {}".format(trainDimensions[1]))
print()

print("TESTING SET")
print("-"*50)
testDimensions = test.shape;
print("Type: {}".format(type(train)))
print("Dimensions: {}".format(testDimensions))
print("Number of rows: {}".format(testDimensions[0]))
print("Number of columns: {}".format(testDimensions[1]))
```

```
TRAINING SET
--------------------------------------------------
Type: <class 'pandas.core.frame.DataFrame'>
Dimensions: (891, 12)
Number of rows: 891
Number of columns: 12

TESTING SET
--------------------------------------------------
Type: <class 'pandas.core.frame.DataFrame'>
Dimensions: (418, 11)
Number of rows: 418
Number of columns: 11
```

Hands-On Labs

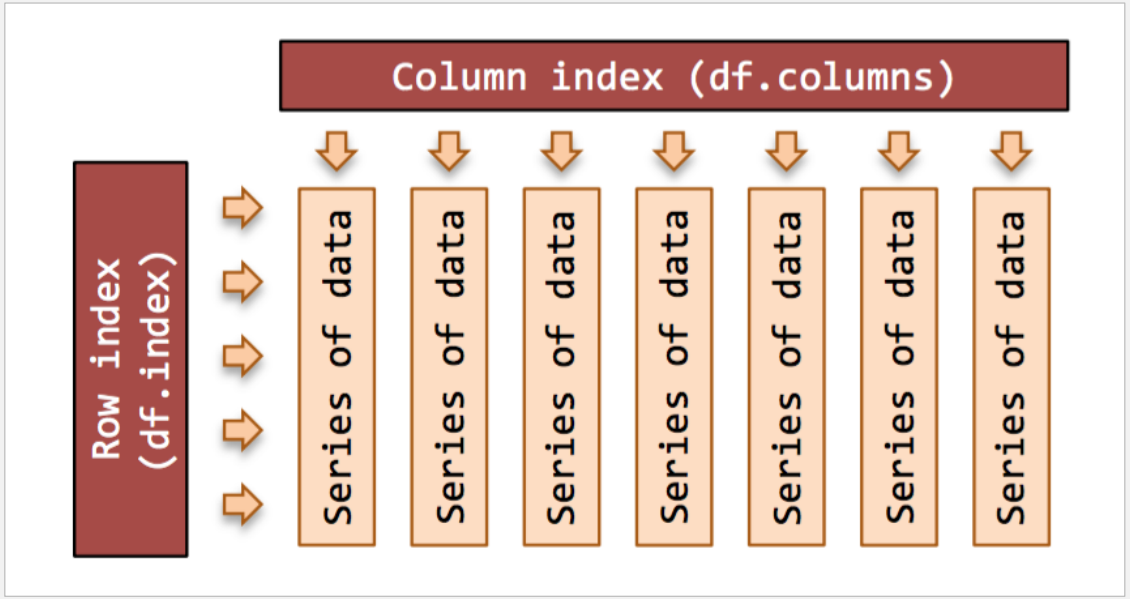
kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

pandas.core.frame.DataFrame



Para representar datos tabulares, **pandas** utiliza una estructura de datos propia llamada **dataframe**. Un dataframe es una estructura de datos bidimensional, altamente eficiente, que proporciona un conjunto de métodos y atributos para explorar, analizar y visualizar datos de forma fácil.

Cuando se carga un archivo dentro de un dataframe, **pandas** usa los valores de la primera fila para las etiquetas de columna y el número de fila para las etiquetas de fila. Colectivamente, las etiquetas se conocen como **índice**. Los dataframes contienen, por lo tanto, un índice de fila y un índice de columna.

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Training Set: Conjunto de entrenamiento

```
# pandas.DataFrame.sample(n=1): Returns a random sample of items of a dataframe.
# More info: https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.sample.html
train.sample(3)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
433	434	0	3	Kallio, Mr. Nikolai Erland	male	17.0	0	0	STON/O 2. 3101274	7.1250	NaN	S
275	276	1	1	Andrews, Miss. Kornelia Theodosia	female	63.0	1	0	13502	77.9583	D7	S
816	817	0	3	Heininen, Miss. Wendla Maria	female	23.0	0	0	STON/O2. 3101290	7.9250	NaN	S

```
# DataFrame.head(n=5): Return the first n rows.
# More info: https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.head.html
train.head(n=5)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
# DataFrame.tail(n=5): Return the last n rows.
# More info: https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.tail.html
train.tail(n=5)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Testing Set: Conjunto de pruebas

```
# pandas.DataFrame.sample(n=1): Returns a random sample of items of a dataframe.
# More info: https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.sample.html
test.sample(3)
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
380	1272	3	O'Connor, Mr. Patrick	male	NaN	0	0	366713	7.75	NaN	Q
317	1209	2	Rogers, Mr. Reginald Harry	male	19.0	0	0	28004	10.50	NaN	S
11	903	1	Jones, Mr. Charles Cresson	male	46.0	0	0	694	26.00	NaN	S

```
# DataFrame.head(n=5): Return the first n rows.
# More info: https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.head.html
test.head()
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

```
# DataFrame.tail(n=5): Return the last n rows.
# More info: https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.tail.html
test.tail()
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
413	1305	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Training Set: Conjunto de entrenamiento

```
# The iloc method allows us to retrieve rows and columns by position.
# In order to do that, we'll need to specify the positions of the rows that we want
train.iloc[0]
```

```
PassengerId                      1
Survived                         0
Pclass                           3
Name           Braund, Mr. Owen Harris
Sex                             male
Age                               22
SibSp                              1
Parch                              0
Ticket                     A/5 21171
Fare                            7.25
Cabin                            NaN
Embarked                           S
Name: 0, dtype: object
```

```
type(train.iloc[0])
```

```
pandas.core.series.Series
```

```
train.iloc[0:3]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

PUE IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Training Set: Conjunto de entrenamiento

```
train.iloc[[1,3,5]]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q

```
train.iloc[:]
```

26	27	0	3	Emir, Mr. Farred Chehab	male	NaN	0	0	2631	7.2250	NaN	C
27	28	0	1	Fortune, Mr. Charles Alexander	male	19.0	3	2	19950	263.0000	C23 C25 C27	S
28	29	1	3	O'Dwyer, Miss. Ellen "Nellie"	female	NaN	0	0	330959	7.8792	NaN	Q
29	30	0	3	Todoroff, Mr. Lalio	male	NaN	0	0	349216	7.8958	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
861	862	0	2	Giles, Mr. Frederick Edward	male	21.0	1	0	28134	11.5000	NaN	S
862	863	1	1	Swift, Mrs. Frederick Joel (Margaret Welles Ba...	female	48.0	0	0	17466	25.9292	D17	S
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.5500	NaN	S
864	865	0	2	Gill, Mr. John William	male	24.0	0	0	233866	13.0000	NaN	S

```
train.iloc[:5]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
train.iloc[888:]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Training Set: Conjunto de entrenamiento

```
train.iloc[:5,0:5]
```

	PassengerId	Survived	Pclass	Name	Sex
0	1	0	3	Braund, Mr. Owen Harris	male
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female
2	3	1	3	Heikkinen, Miss. Laina	female
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female
4	5	0	3	Allen, Mr. William Henry	male

```
train.iloc[:5, [0,3,4,5,2,1]]
```

	PassengerId	Name	Sex	Age	Pclass	Survived
0	1	Braund, Mr. Owen Harris	male	22.0	3	0
1	2	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	1
2	3	Heikkinen, Miss. Laina	female	26.0	3	1
3	4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	1
4	5	Allen, Mr. William Henry	male	35.0	3	0

```
train.loc[:, ["Name", "Sex", "Pclass", "Survived"]]
```

	Name	Sex	Pclass	Survived
0	Braund, Mr. Owen Harris	male	3	0
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	1	1
2	Heikkinen, Miss. Laina	female	3	1
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	1	1
4	Allen, Mr. William Henry	male	3	0

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Training Set: Conjunto de entrenamiento

```
train[["Name", "Sex", "Pclass", "Age", "Survived"]]
```

	Name	Sex	Pclass	Age	Survived
0	Braund, Mr. Owen Harris	male	3	22.0	0
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	1	38.0	1
2	Heikkinen, Miss. Laina	female	3	26.0	1
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	1	35.0	1
4	Allen, Mr. William Henry	male	3	35.0	0
5	Moran, Mr. James	male	3	NaN	0

```
train[train["Survived"] == 0][["Pclass", "Name", "Sex", "Age"]]
```

	Pclass	Name	Sex	Age
0	3	Braund, Mr. Owen Harris	male	22.0
4	3	Allen, Mr. William Henry	male	35.0
5	3	Moran, Mr. James	male	NaN
6	1	McCarthy, Mr. Timothy J	male	54.0
7	3	Palsson, Master. Gosta Leonard	male	2.0

```
train[train["Survived"] == 1][["Pclass", "Name", "Sex", "Age"]].sort_values(by=["Pclass", "Name"], ascending=[True, True])
```

	Pclass	Name	Sex	Age
730	1	Allen, Miss. Elisabeth Walton	female	29.00
305	1	Allison, Master. Hudson Trevor	male	0.92
460	1	Anderson, Mr. Harry	male	48.00
275	1	Andrews, Miss. Kornelia Theodosia	female	63.00
571	1	Appleton, Mrs. Edward Dale (Charlotte Lamson)	female	53.00
700	1	Astor, Mrs. John Jacob (Madeleine Talmadge Force)	female	18.00

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Training Set: Conjunto de entrenamiento

```
# pandas.DataFrame.dtypes: Return the dtypes in this object.
# https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.dtypes.html#pandas.DataFrame.dtypes
train.dtypes
```

```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

```
# pandas.DataFrame.count: Return Series with number of non-NA/null observations over requested axis
train.count()
```

```
PassengerId    891
Survived       891
Pclass         891
Name           891
Sex            891
Age            714
SibSp          891
Parch          891
Ticket         891
Fare           891
Cabin          204
Embarked       889
dtype: int64
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Training Set: Conjunto de entrenamiento

```
# pandas.DataFrame.info: Concise summary of a DataFrame.
# https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.info.html
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
PassengerId    891 non-null int64
Survived       891 non-null int64
Pclass         891 non-null int64
Name           891 non-null object
Sex            891 non-null object
Age            714 non-null float64
SibSp          891 non-null int64
Parch          891 non-null int64
Ticket         891 non-null object
Fare           891 non-null float64
Cabin          204 non-null object
Embarked       889 non-null object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.6+ KB
```

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Testing Set: Conjunto de pruebas

```
# pandas.DataFrame.dtypes: Return the dtypes in this object.
# https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.dtypes.html
test.dtypes
```

```
PassengerId      int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

```
# pandas.DataFrame.count: Return Series with number of non-NA/null observations over requested axis
test.count()
```

```
PassengerId    418
Pclass         418
Name           418
Sex            418
Age            332
SibSp          418
Parch          418
Ticket         418
Fare           417
Cabin           91
Embarked       418
dtype: int64
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Testing Set: Conjunto de pruebas

```
# pandas.DataFrame.info: Concise summary of a DataFrame.
# https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.info.html
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
PassengerId    418 non-null int64
Pclass         418 non-null int64
Name           418 non-null object
Sex            418 non-null object
Age            332 non-null float64
SibSp          418 non-null int64
Parch          418 non-null int64
Ticket         418 non-null object
Fare           417 non-null float64
Cabin          91 non-null object
Embarked       418 non-null object
dtypes: float64(2), int64(4), object(5)
memory usage: 36.0+ KB
```

Networking Academy

IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

¿Qué porcentaje de pasajeros sobrevivió?

```
train["Survived"].mean()
```

```
0.3838383838383838
```

```
train.mean()
```

```
PassengerId    446.000000
Survived         0.383838
Pclass           2.308642
Age             29.699118
SibSp            0.523008
Parch            0.381594
Fare            32.204208
dtype: float64
```

Del **conjunto de entrenamiento** sólo sobrevivieron al hundimiento del Titanic el 38.38% de los pasajeros.

Networking Academy

IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cúantos pasajeros sobrevivieron?

El tipo de aprendizaje automático al que nos enfrentamos es conocido como *clasificación* (**classification**), porque en nuestras predicciones debemos clasificar a cada pasajero como si sobrevivió o no al hundimiento.

Más específicamente, estamos ante un escenario de *clasificación binaria* (**binary classification**), ya que solo hay dos estados diferentes en los que se puede clasificar.

```
print("Survived: {}".format(train["Survived"].dtype))
print("Total: {}".format(train.shape[0]))
print("Values: {}".format(train["Survived"].count()))
print("Is Empty: {}".format(train["Survived"].isnull().any()))
print("Empty: {}".format(train["Survived"].isnull().sum()))
print("Mean: {}".format(train["Survived"].mean()))
```

```
Survived: int64
Total: 891
Values: 891
Is Empty: False
Empty: 0
Mean: 0.3838383838383838
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cúantos pasajeros sobrevivieron?

```
# absolute numbers
train["Survived"].value_counts()
#train["Survived"].value_counts().sort_index()
#train["Survived"].value_counts().sort_values(ascending=True)
```

```
0    549
1    342
Name: Survived, dtype: int64
```

```
# percentages
train["Survived"].value_counts(normalize = True).sort_index()
#train["Survived"].value_counts(normalize = True).sort_values(ascending=False)
```

```
0    0.616162
1    0.383838
Name: Survived, dtype: float64
```

```
train.groupby(["Survived"]).size()
```

```
0    549
1    342
Name: Survived, dtype: int64
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cúantos pasajeros sobrevivieron?

```
series = train["Survived"].value_counts()
series.plot.bar(width=0.9, figsize=(10,8))
plt.title("Pasajeros del Titanic por estado", fontsize=18)
plt.xlabel("Estado", fontsize=12)
plt.ylabel("Número de pasajeros", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Muertos", "Vivos"], fontsize=12, rotation=360)
plt.show()
```

```
series = train["Survived"].value_counts()
series.plot.barh(width=0.9, figsize=(10,8))
plt.title("Pasajeros del Titanic por estado", fontsize=18)
plt.xlabel("Número de pasajeros", fontsize=12)
plt.ylabel("Estado", fontsize=12)
plt.xticks(fontsize=12)
plt.yticks([0,1],  ["Muertos", "Vivos"], fontsize=12, rotation=360)
plt.show()
```

```
series = train["Survived"].value_counts(normalize = True)
series.plot.pie(labels = ["Muertos", "Vivos"],autopct='%.2f%%', shadow=False, figsize=(10,8))
plt.title("Pasajeros del Titanic por estado", fontsize=18)
plt.ylabel("")
plt.show()
```

Hands-On Labs

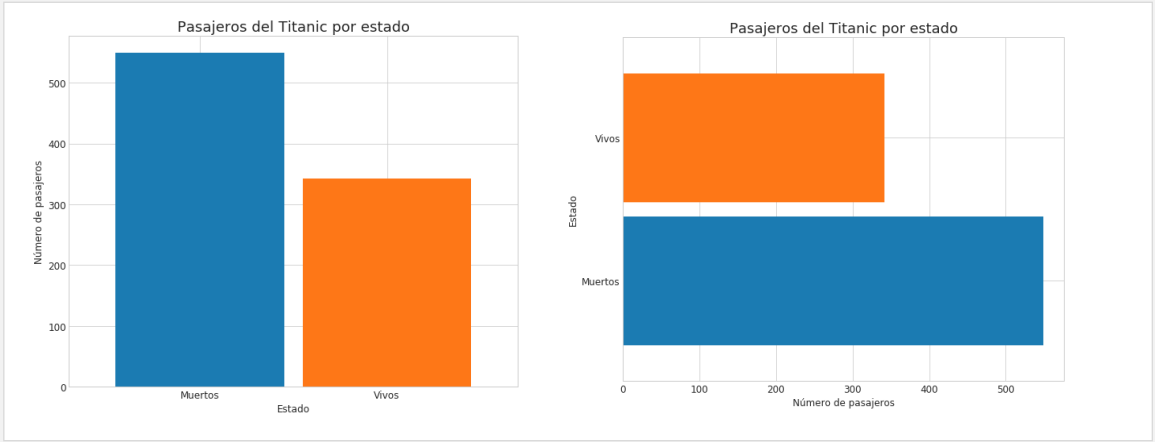
kaggle

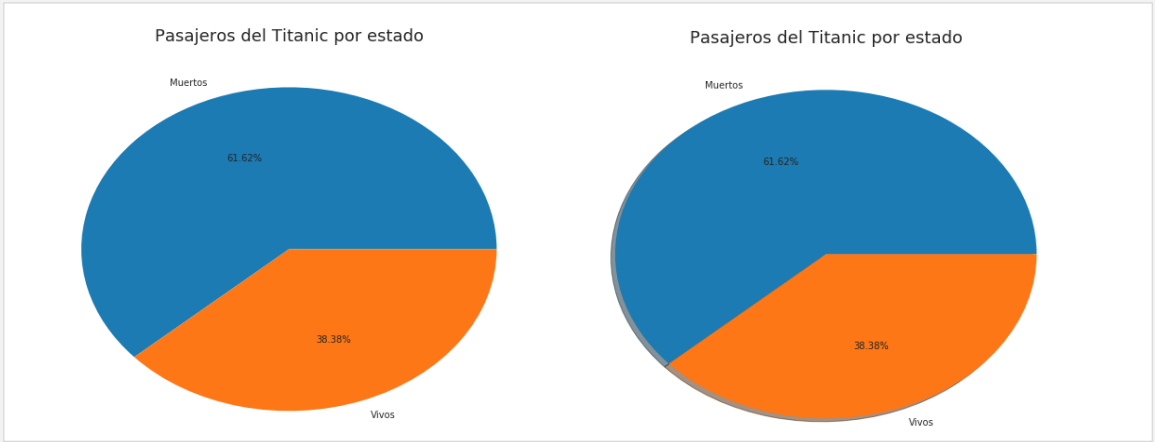
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cúantos pasajeros sobrevivieron?





Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Número de pasajeros según género

```
print("Sex: {}".format(train["Sex"].dtype))
print("Total: {}".format(train.shape[0]))
print("Values: {}".format(train["Sex"].count()))
print("Is Empty: {}".format(train["Sex"].isnull().any()))
print("Empty: {}".format(train["Sex"].isnull().sum()))
```

```
Sex: object
Total: 891
Values: 891
Is Empty: False
Empty: 0
```

```
train["Sex"].describe()
```

```
count      891
unique       2
top       male
freq       577
Name: Sex, dtype: object
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Número de pasajeros según género

```
train["Sex"].value_counts().sort_index()
```

```
female    314
male      577
Name: Sex, dtype: int64
```

```
train["Sex"].value_counts(normalize = True).sort_index()
```

```
female    0.352413
male      0.647587
Name: Sex, dtype: float64
```

```
train.groupby(["Sex"]).size()
```

```
female    314
male      577
Name: Sex, dtype: int64
```

```
train.groupby(["Sex"]).size() / train["Sex"].count()
```

```
female    0.352413
male      0.647587
Name: Sex, dtype: float64
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Número de pasajeros según género

```
series = train["Sex"].value_counts().sort_index()
series.plot.bar(width=0.9, figsize=(10,8))
plt.title("Pasajeros del Titanic por género", fontsize=18)
plt.xlabel("Género", fontsize=12)
plt.ylabel("Número de pasajeros", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Mujeres", "Hombres"], fontsize=12, rotation=360)
plt.show()
```

```
series = train["Sex"].value_counts().sort_index()
series.plot.barh(width=0.9, figsize=(10,8))
plt.title("Pasajeros del Titanic por género", fontsize=18)
plt.ylabel("Género", fontsize=12)
plt.xlabel("Número de pasajeros", fontsize=12)
plt.xticks(fontsize=12)
plt.yticks([0,1],  ["Mujeres", "Hombres"], fontsize=12, rotation=360)
plt.show()
```

```
series = train["Sex"].value_counts(normalize = True).sort_index()
series.plot.pie(labels = ["Mujeres", "Hombres"],autopct='%.2f%%', shadow=True, figsize=(10,8))
plt.title("Pasajeros del Titanic por género", fontsize=18)
plt.ylabel("")
plt.show()
```

Hands-On Labs

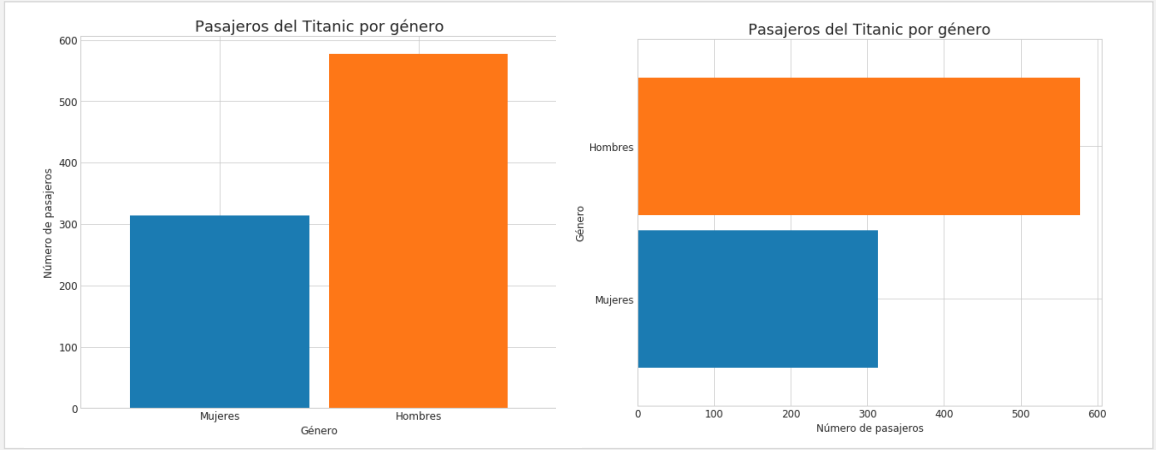
kaggle

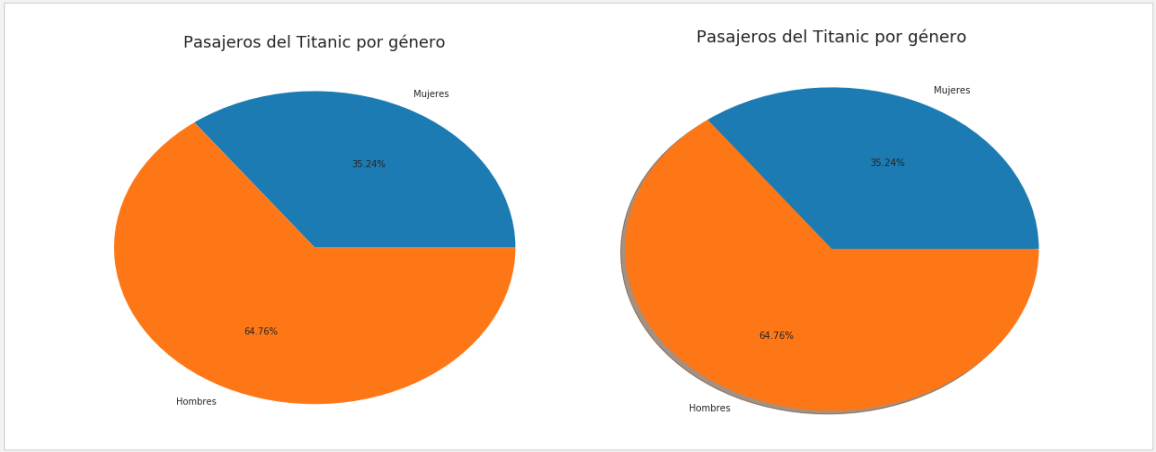
www.kaggle.com/c/titanic

Networking Academy

PUE IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Número de pasajeros según género





Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cuántos hombres sobrevivieron?

```
# Males that survived vs males that passed away
train[train["Sex"] == 'male']["Survived"].value_counts().sort_index()
```

```
0    468
1    109
Name: Survived, dtype: int64
```

```
# Males that survived vs males that passed away (percentage)
train[train["Sex"] == "male"]["Survived"].value_counts(normalize = True).sort_index()
```

```
0    0.811092
1    0.188908
Name: Survived, dtype: float64
```

```
train[train["Sex"] == "male"].groupby("Survived").size().sort_index()
```

```
Survived
0    468
1    109
dtype: int64
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cuántos hombres sobrevivieron?

```
series = train[train["Sex"] == 'male']["Survived"].value_counts().sort_index()
series.plot.bar(width=0.9, figsize=(10,8))
plt.title("Hombres en el Titanic", fontsize=18)
plt.xlabel("Estado", fontsize=12)
plt.ylabel("Número de pasajeros", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Muertos", "Vivos"], fontsize=12, rotation=360)
plt.show()
```

```
series = train[train["Sex"] == 'male']["Survived"].value_counts().sort_index()
series.plot.barh(width=0.9, figsize=(10,8))
plt.title("Hombres en el Titanic", fontsize=18)
plt.xlabel("Número de pasajeros", fontsize=12)
plt.ylabel("Estado", fontsize=12)
plt.xticks(fontsize=12)
plt.yticks([0,1],  ["Muertos", "Vivos"], fontsize=12, rotation=360)
plt.show()
```

```
series = train[train["Sex"] == "male"]["Survived"].value_counts(normalize = True).sort_index()
series.plot.pie(labels = ["Muertos", "Vivos"],autopct='%.2f%%', shadow=True, figsize=(10,8))
plt.title("Hombres en el Titanic", fontsize=18)
plt.ylabel("")
plt.show()
```

Hands-On Labs

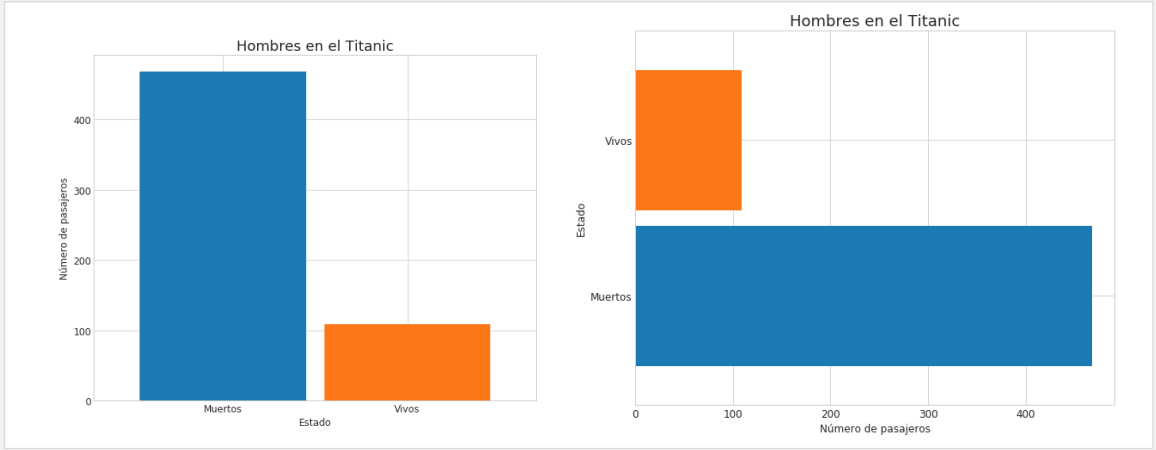
kaggle

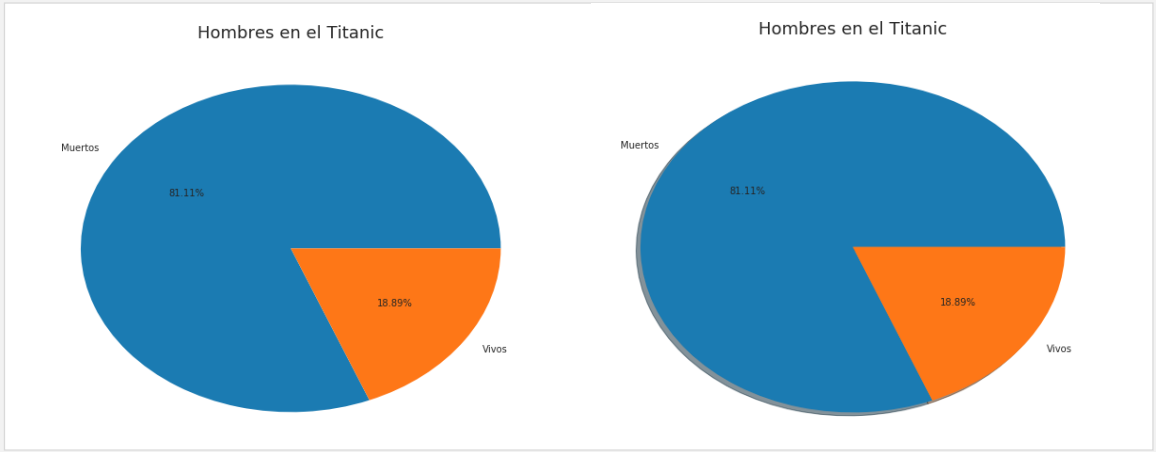
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cuántos hombres sobrevivieron?





Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cuántas mujeres sobrevivieron?

```
# Females that survived vs Females that passed away
train[train["Sex"] == "female"]["Survived"].value_counts().sort_index()
```

```
0     81
1    233
Name: Survived, dtype: int64
```

```
# Females that survived vs Females that passed away
train[train["Sex"] == "female"]["Survived"].value_counts(normalize = True).sort_index()
```

```
0    0.257962
1    0.742038
Name: Survived, dtype: float64
```

```
train[train["Sex"] == "female"].groupby("Survived").size().sort_index()
```

```
Survived
0     81
1    233
dtype: int64
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cuántas mujeres sobrevivieron?

```
series = train[train["Sex"] == 'female']["Survived"].value_counts().sort_index()
series.plot.bar(width=0.9, figsize=(10,8))
plt.title("Mujeres en el Titanic", fontsize=18)
plt.xlabel("Estado", fontsize=12)
plt.ylabel("Número de pasajeros", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Muertas", "Vivas"], fontsize=12, rotation=360)
plt.show()
```

```
series = train[train["Sex"] == 'female']["Survived"].value_counts().sort_index()
series.plot.barh(width=0.9, figsize=(10,8))
plt.title("Mujeres en el Titanic", fontsize=18)
plt.xlabel("Número de pasajeros", fontsize=12)
plt.ylabel("Estado", fontsize=12)
plt.xticks(fontsize=12)
plt.yticks([0,1],  ["Muertas", "Vivas"], fontsize=12, rotation=360)
plt.show()
```

```
series = train[train["Sex"] == "female"]["Survived"].value_counts(normalize = True).sort_index()
series.plot.pie(labels = ["Muertas", "Vivas"],autopct='%.2f%%', shadow=False, figsize=(10,8))
plt.title("Mujeres en el Titanic", fontsize=18)
plt.ylabel("")
plt.show()
```

Hands-On Labs

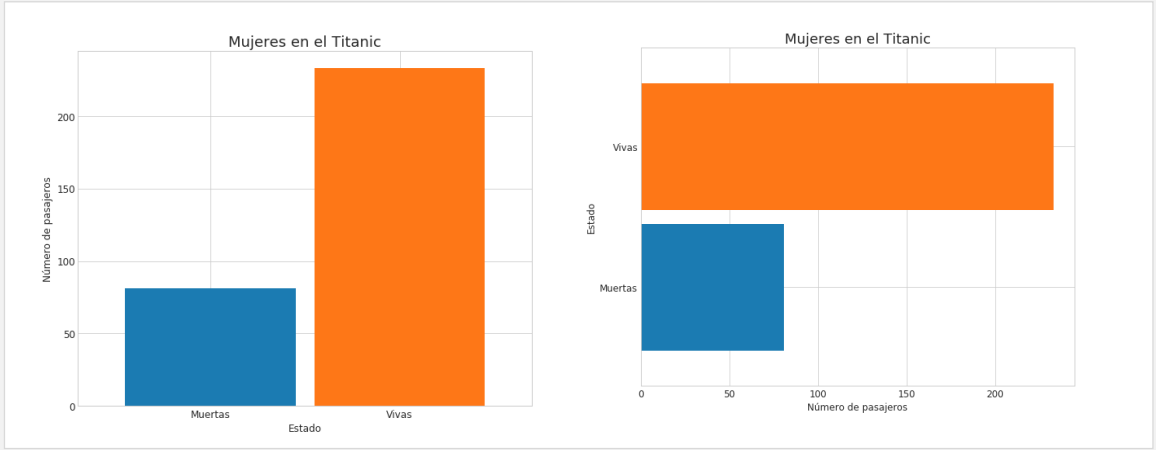
kaggle

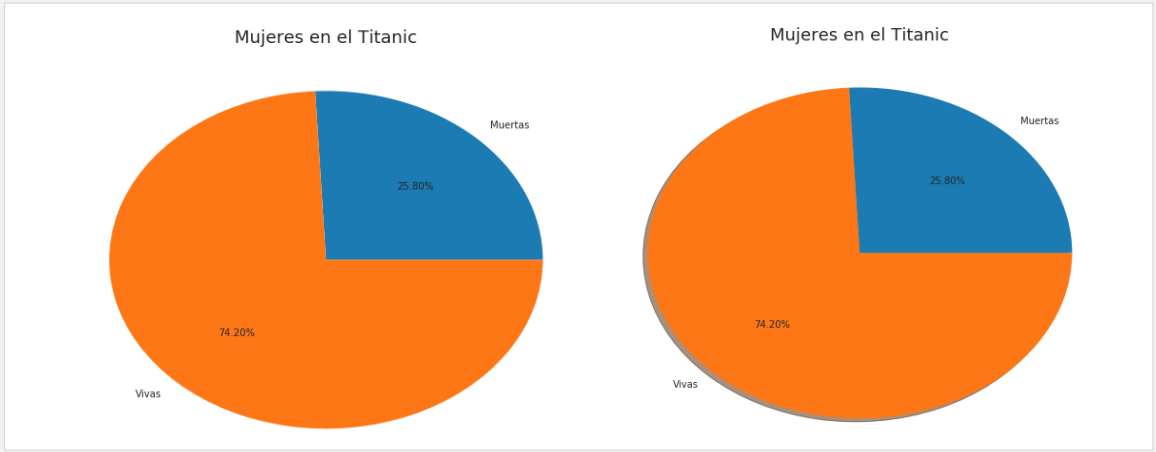
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cuántas mujeres sobrevivieron?





Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Rose o Jack?

```
train[["Sex", "Survived"]].groupby(['Sex'], as_index=True).mean().sort_values(by='Survived', ascending=False)
```

	Survived
Sex	
female	0.742038
male	0.188908

```
train.pivot_table(index = "Sex", values = "Survived")
```

	Survived
Sex	
female	0.742038
male	0.188908

```
series = train.pivot_table(index = "Sex", values = "Survived").sort_index()
series.plot.bar(width=0.9, figsize=(10,8))
plt.title("Mujeres vs Hombres", fontsize=18)
plt.xlabel("Género", fontsize=12)
plt.ylabel("Tasa de supervivencia", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Mujeres", "Hombres"], fontsize=12, rotation=360)
plt.show()
```

Hands-On Labs

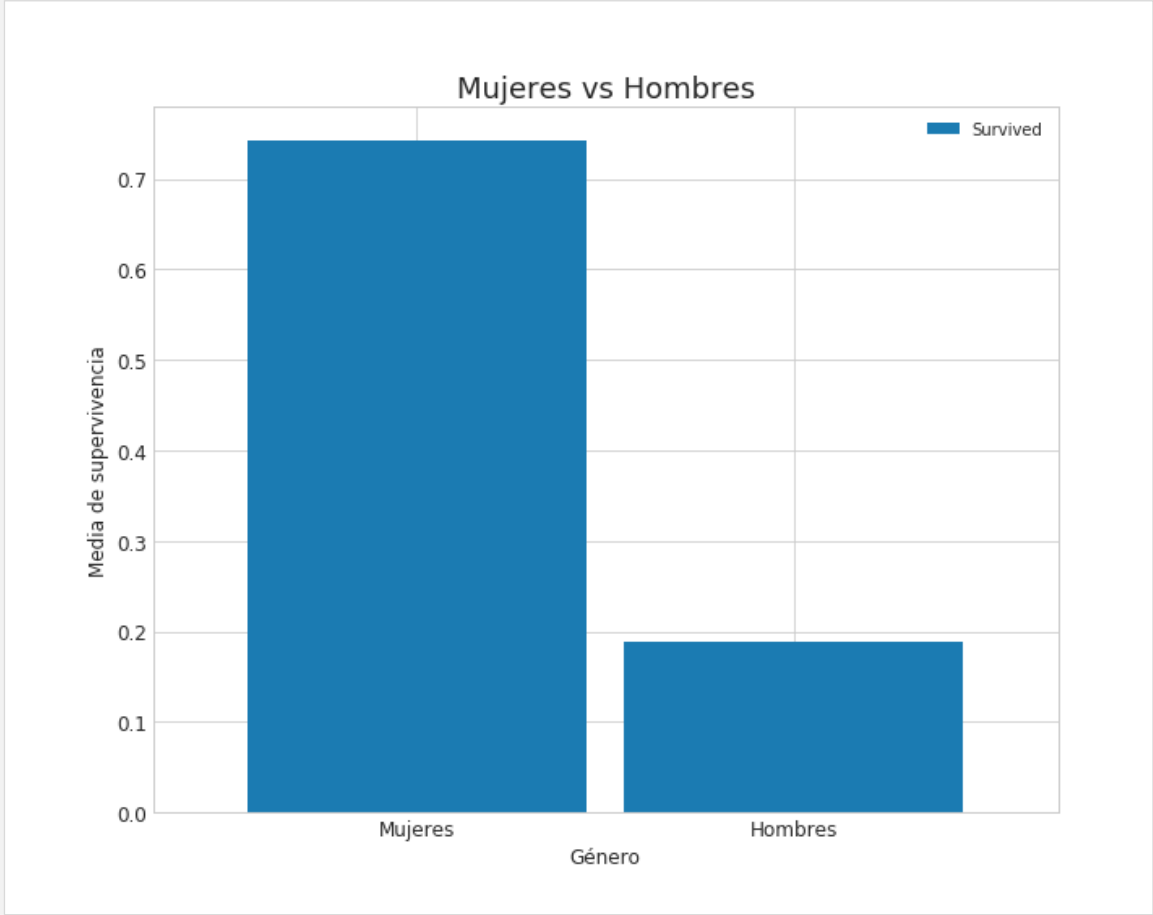
kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Rose o Jack?



Hands-On Labs

kaggle

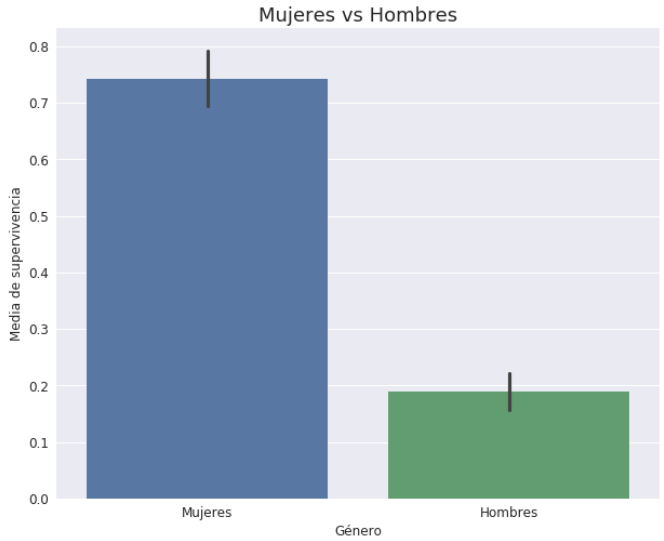
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Rose o Jack?

```
sns.set(rc={"figure.figsize":(10,8)})
sns.barplot(data=train, x="Sex", y="Survived", order= ["female", "male"])
plt.title("Mujeres vs Hombres", fontsize=18)
plt.xlabel("Género", fontsize=12)
plt.ylabel("Media de supervivencia", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Mujeres", "Hombres"], fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Número de pasajeros según clase

```
print("Pclass: {}".format(train["Pclass"].dtype))
print("Total: {}".format(train.shape[0]))
print("Values: {}".format(train["Pclass"].count()))
print("Empty: {}".format(train["Pclass"].isnull().sum()))
print("Mean: {}".format(train["Pclass"].mean()))
```

```
Pclass: int64
Total: 891
Values: 891
Empty: 0
Mean: 2.308641975308642
```

```
train["Pclass"].describe()
```

```
count    891.000000
mean       2.308642
std        0.836071
min        1.000000
25%        2.000000
50%        3.000000
75%        3.000000
max        3.000000
Name: Pclass, dtype: float64
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Número de pasajeros según clase

```
train["Pclass"].value_counts().sort_index()
```

```
1    216
2    184
3    491
Name: Pclass, dtype: int64
```

```
train["Pclass"].value_counts(normalize = True).sort_index()
```

```
1    0.242424
2    0.206510
3    0.551066
Name: Pclass, dtype: float64
```

```
train.groupby(["Pclass"]).size().sort_index()
```

```
1    216
2    184
3    491
Name: Pclass, dtype: int64
```

```
(train.groupby(["Pclass"]).size() / train["Pclass"].count()).sort_index()
```

```
1    0.242424
2    0.206510
3    0.551066
Name: Pclass, dtype: float64
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Número de pasajeros según clase

```
series = train["Pclass"].value_counts().sort_index()
series.plot.bar(width=0.9, figsize=(10,8))
plt.title("Pasajeros del Titanic por clase", fontsize=18)
plt.xlabel("Clase", fontsize=12)
plt.ylabel("Número de pasajeros", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1,2],  ["1ª clase", "2ª clase", "3ª clase"], fontsize=12, rotation=360)
plt.show()
```

```
series = train["Pclass"].value_counts().sort_index()
series.plot.barh(width=0.9, figsize=(10,8))
plt.title("Pasajeros del Titanic por clase", fontsize=18)
plt.xlabel("Número de pasajeros", fontsize=12)
plt.ylabel("Clase", fontsize=12)
plt.xticks(fontsize=12)
plt.yticks([0,1,2],  ["1ª clase", "2ª clase", "3ª clase"], fontsize=12, rotation=360)
plt.show()
```

```
series = train["Pclass"].value_counts(normalize = True).sort_index()
series.plot.pie(labels = ["1ª clase", "2ª clase", "3ª clase"],autopct='%.2f%%', shadow=True, figsize=(10,8))
plt.title("Pasajeros del Titanic por clase", fontsize=18)
plt.ylabel("")
plt.show()
```

Hands-On Labs

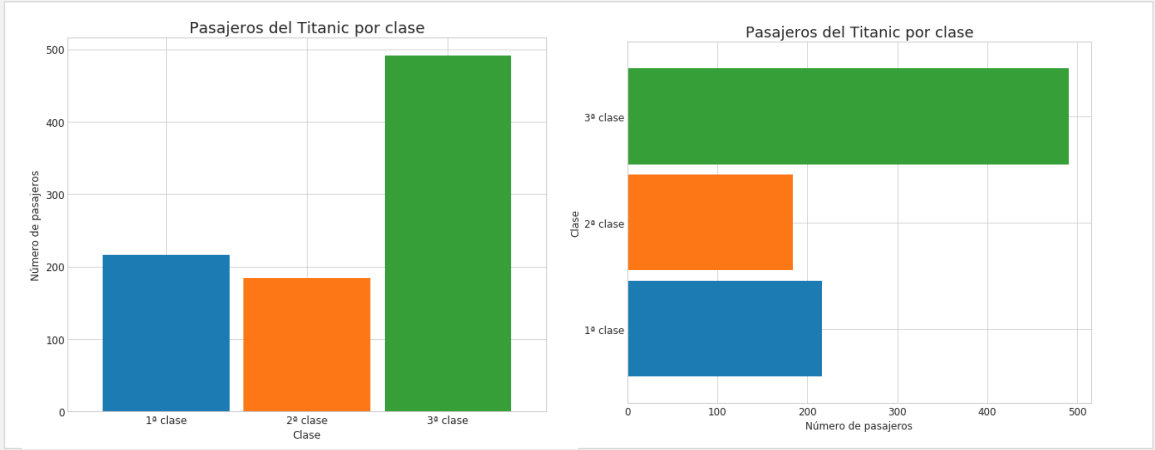
kaggle

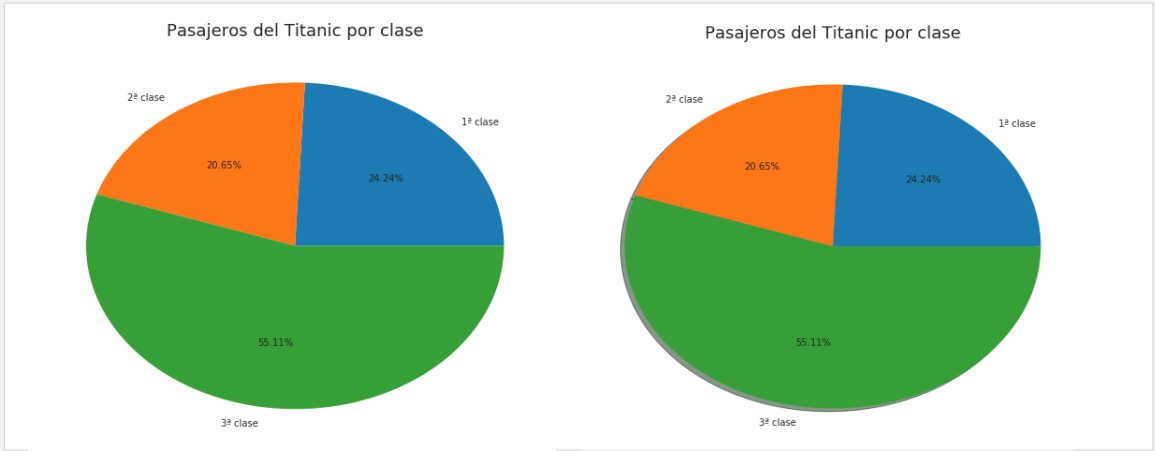
www.kaggle.com/c/titanic

Networking Academy

PUE IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Número de pasajeros según clase





Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cuántos pasajeros de 1ª clase sobrevivieron?

```
# First class passengers that survived vs that passed away
train[train["Pclass"] == 1]["Survived"].value_counts().sort_index()
```

```
0     80
1    136
Name: Survived, dtype: int64
```

```
# First class passengers that survived vs that passed away (percentage)
train[train["Pclass"] == 1]["Survived"].value_counts(normalize = True).sort_index()
```

```
0    0.37037
1    0.62963
Name: Survived, dtype: float64
```

```
train[train["Pclass"] == 1].groupby("Survived").size().sort_index()
```

```
Survived
0     80
1    136
dtype: int64
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cuántos pasajeros de 1ª clase sobrevivieron?

```
series = train[train["Pclass"] == 1]["Survived"].value_counts().sort_index()
series.plot.bar(width=0.9, figsize=(10,8))
plt.title("Pasajeros de 1ª clase", fontsize=18)
plt.xlabel("Estado", fontsize=12)
plt.ylabel("Número de pasajeros", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Muertos", "Vivos"], fontsize=12, rotation=360)
plt.show()
```

```
series = train[train["Pclass"] == 1]["Survived"].value_counts().sort_index()
series.plot.barh(width=0.9, figsize=(10,8))
plt.title("Pasajeros de 1ª clase", fontsize=18)
plt.ylabel("Estado", fontsize=12)
plt.xlabel("Número de pasajeros", fontsize=12)
plt.xticks(fontsize=12)
plt.yticks([0,1],  ["Muertos", "Vivos"], fontsize=12, rotation=360)
plt.show()
```

```
series = train[train["Pclass"] == 1]["Survived"].value_counts(normalize = True).sort_index()
series.plot.pie(labels = ["Muertos", "Vivos"],autopct='%.2f%%', shadow=True, figsize=(10,8))
plt.title("Pasajeros de 1ª clase", fontsize=18)
plt.ylabel("")
plt.show()
```

Hands-On Labs

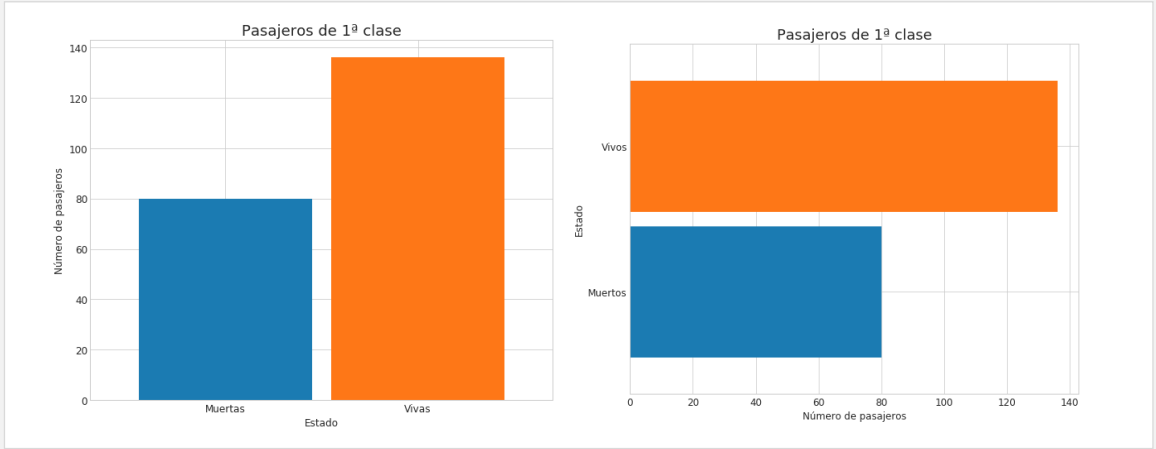
kaggle

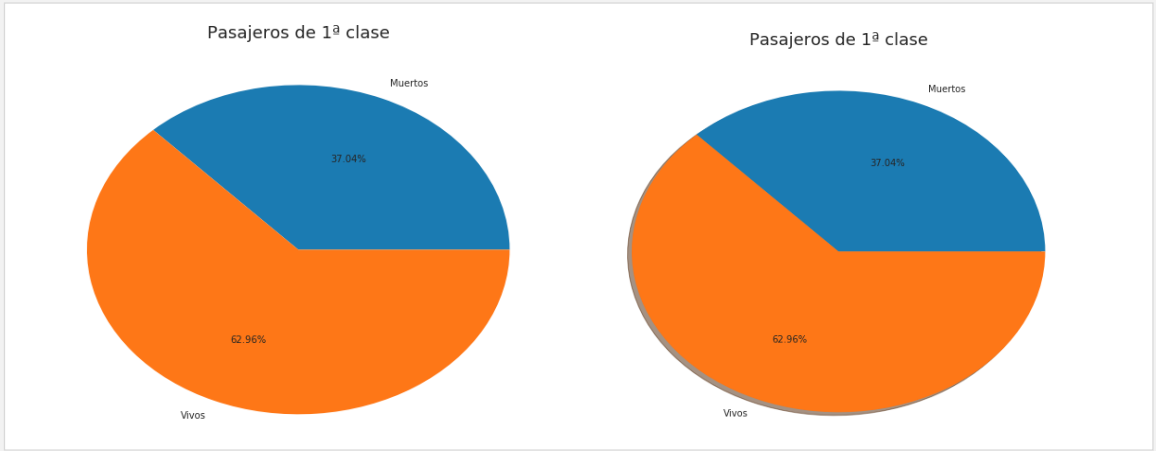
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cuántos pasajeros de 1ª clase sobrevivieron?





Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cuántos pasajeros de 2ª clase sobrevivieron?

```
# Second class passengers that survived vs that passed away
train[train["Pclass"] == 2]["Survived"].value_counts().sort_index()
```

```
0    97
1    87
Name: Survived, dtype: int64
```

```
# Second class passengers that survived vs that passed away
train[train["Pclass"] == 2]["Survived"].value_counts(normalize = True)
```

```
0    0.527174
1    0.472826
Name: Survived, dtype: float64
```

```
train[train["Pclass"] == 2].groupby("Survived").size().sort_index()
```

```
Survived
0    97
1    87
dtype: int64
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cuántos pasajeros de 2ª clase sobrevivieron?

```
series = train[train["Pclass"] == 2]["Survived"].value_counts().sort_index()
series.plot.bar(width=0.9, figsize=(10,8))
plt.title("Pasajeros de 2ª clase", fontsize=18)
plt.xlabel("Estado", fontsize=12)
plt.ylabel("Número de pasajeros", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Muertos", "Vivos"], fontsize=12, rotation=360)
plt.show()
```

```
series = train[train["Pclass"] == 2]["Survived"].value_counts().sort_index()
series.plot.barh(width=0.9, figsize=(10,8))
plt.title("Pasajeros de 2ª clase", fontsize=18)
plt.ylabel("Estado", fontsize=12)
plt.xlabel("Número de pasajeros", fontsize=12)
plt.xticks(fontsize=12)
plt.yticks([0,1],  ["Muertos", "Vivos"], fontsize=12, rotation=360)
plt.show()
```

```
series = train[train["Pclass"] == 2]["Survived"].value_counts(normalize = True).sort_index()
series.plot.pie(labels = ["Muertos", "Vivos"],autopct='%.2f%%', shadow=True, figsize=(10,8))
plt.title("Pasajeros de 2ª clase", fontsize=18)
plt.ylabel("")
plt.show()
```

Hands-On Labs

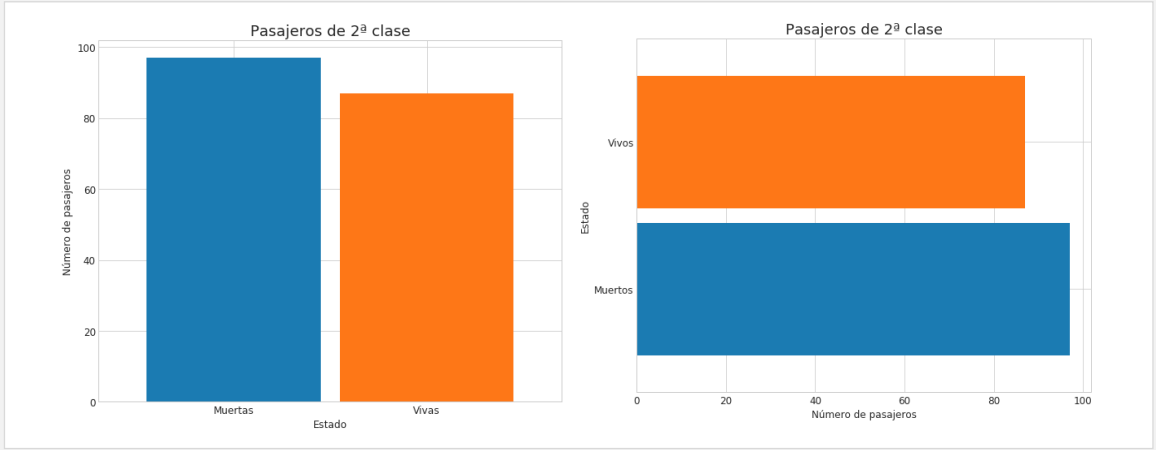
kaggle

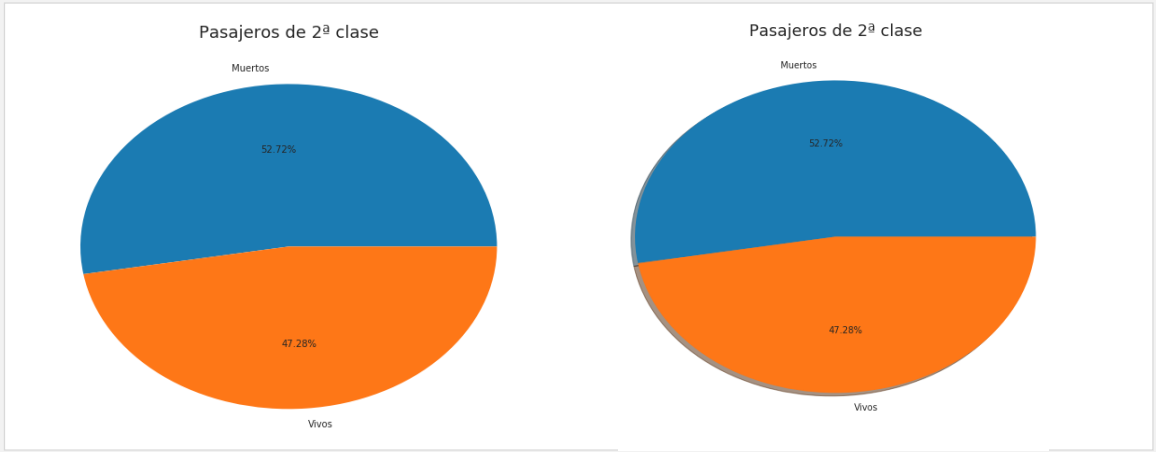
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cuántos pasajeros de 2ª clase sobrevivieron?





Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cuántos pasajeros de 3ª clase sobrevivieron?

```
# Third class passengers that survived vs that passed away
train[train["Pclass"] == 3]["Survived"].value_counts().sort_index()
```

```
0    372
1    119
Name: Survived, dtype: int64
```

```
# Third class passengers that survived vs that passed away
train[train["Pclass"] == 3]["Survived"].value_counts(normalize = True)
```

```
0    0.757637
1    0.242363
Name: Survived, dtype: float64
```

```
train[train["Pclass"] == 3].groupby("Survived").size().sort_index()
```

```
Survived
0    372
1    119
dtype: int64
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cuántos pasajeros de 3ª clase sobrevivieron?

```
series = train[train["Pclass"] == 3]["Survived"].value_counts().sort_index()
series.plot.bar(width=0.9, figsize=(10,8))
plt.title("Pasajeros de 3ª clase", fontsize=18)
plt.xlabel("Estado", fontsize=12)
plt.ylabel("Número de pasajeros", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Muertos", "Vivos"], fontsize=12, rotation=360)
plt.show()
```

```
series = train[train["Pclass"] == 3]["Survived"].value_counts().sort_index()
series.plot.barh(width=0.9, figsize=(10,8))
plt.title("Pasajeros de 3ª clase", fontsize=18)
plt.ylabel("Estado", fontsize=12)
plt.xlabel("Número de pasajeros", fontsize=12)
plt.xticks(fontsize=12)
plt.yticks([0,1],  ["Muertos", "Vivos"], fontsize=12, rotation=360)
plt.show()
```

```
series = train[train["Pclass"] == 3]["Survived"].value_counts(normalize = True).sort_index()
series.plot.pie(labels = ["Muertos", "Vivos"],autopct='%.2f%%', shadow=True, figsize=(10,8))
plt.title("Pasajeros de 3ª clase", fontsize=18)
plt.ylabel("")
plt.show()
```

Hands-On Labs

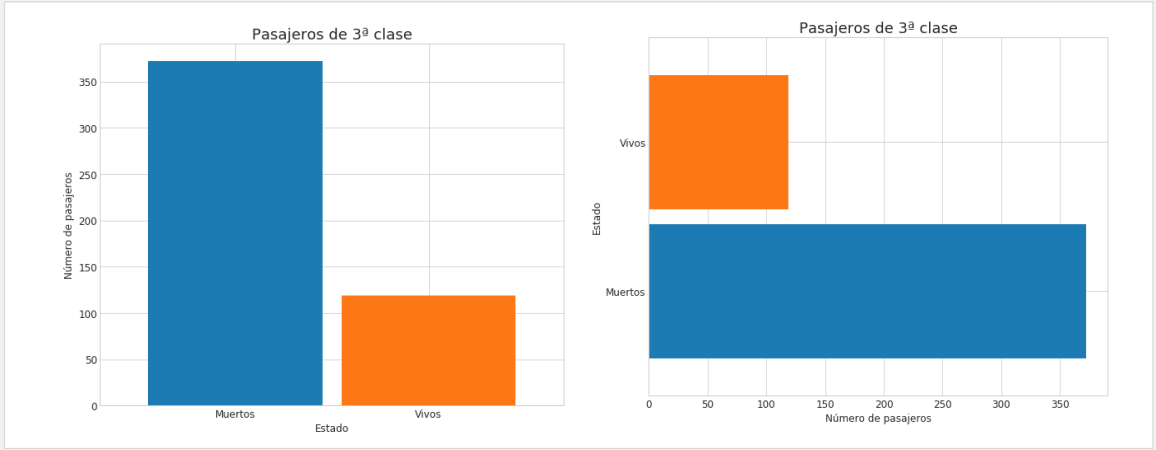
kaggle

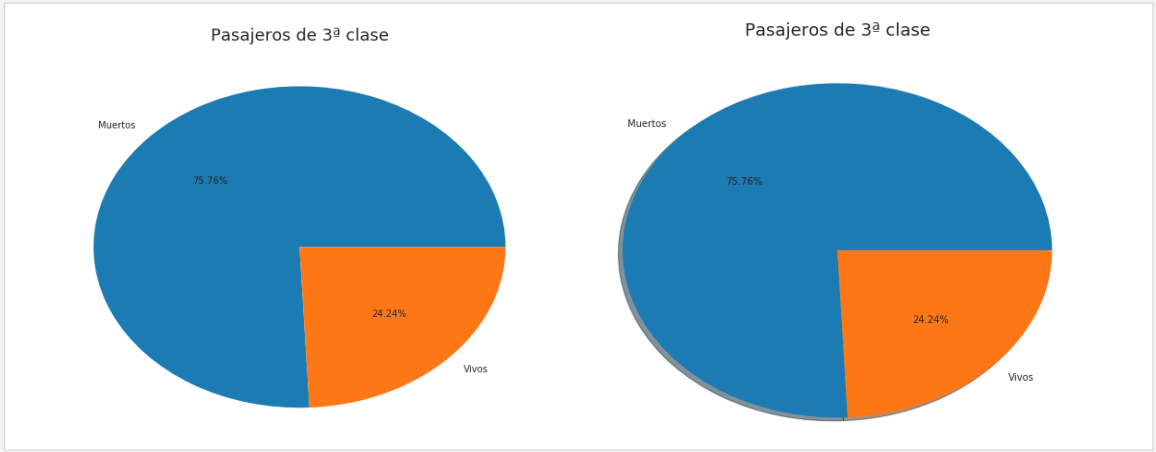
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cuántos pasajeros de 3ª clase sobrevivieron?





Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cal Hockley o Jack Dawson?

```
train[["Pclass", "Survived"]].groupby(['Pclass'], as_index=True).mean().sort_index()
```

Pclass	Survived
1	0.629630
2	0.472826
3	0.242363

```
train.pivot_table(index = "Pclass", values = "Survived").sort_index()
```

Pclass	Survived
1	0.629630
2	0.472826
3	0.242363

```
series = train.pivot_table(index = "Pclass", values = "Survived").sort_index()
series.plot.bar(width=0.9, figsize=(10,8))
plt.title("Supervivencia según clase", fontsize=18)
plt.xlabel("Clase", fontsize=12)
plt.ylabel("Tasa de supervivencia", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1,2],  ["1a clase", "2a clase", "3a clase"], fontsize=12, rotation=360)
plt.show()
```

Hands-On Labs

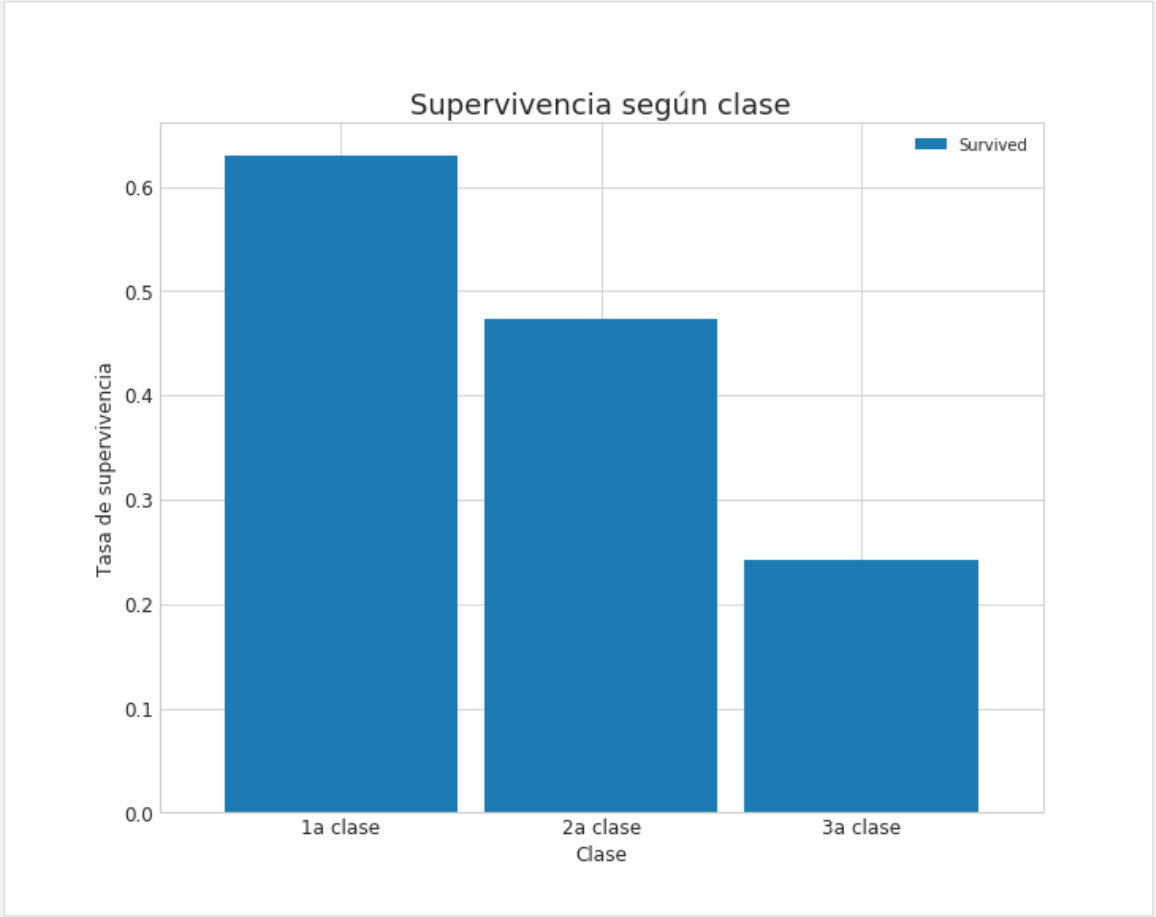
kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cal Hockley o Jack Dawson?



Hands-On Labs

kaggle

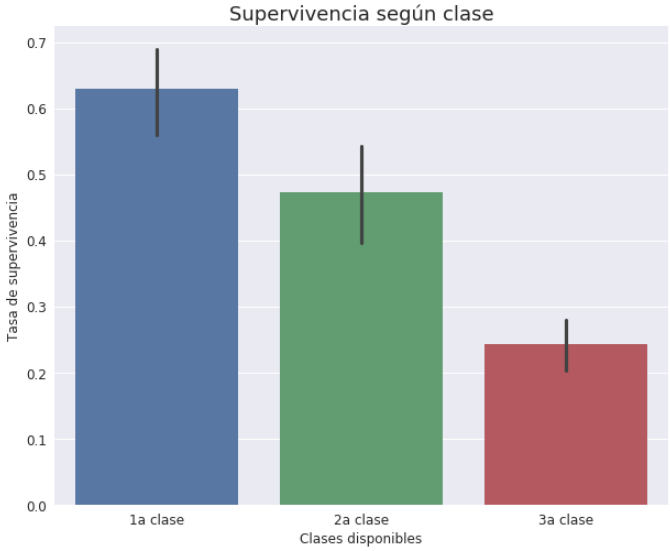
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Cal Hockley o Jack Dawson?

```
sns.set(rc={"figure.figsize":(10,8)})
sns.barplot(data=train, x="Pclass", y="Survived")
plt.title("Supervivencia según clase", fontsize=18)
plt.xlabel("Clases disponibles", fontsize=12)
plt.ylabel("Tasa de supervivencia", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1,2],  ["1a clase", "2a clase", "3a clase"], fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Pasajeros por género y clase

```
train[["Sex", "Pclass", "Survived"]].groupby(["Sex", "Pclass"]).count()
```

		Survived
Sex	Pclass	
female	1	94
	2	76
	3	144
male	1	122
	2	108
	3	347

```
train[["Sex", "Pclass", "Survived"]].groupby(["Pclass", "Sex"]).count()
```

		Survived
Pclass	Sex	
1	female	94
	male	122
2	female	76
	male	108
3	female	144
	male	347

Networking Academy

IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Hands-On Labs

kaggle

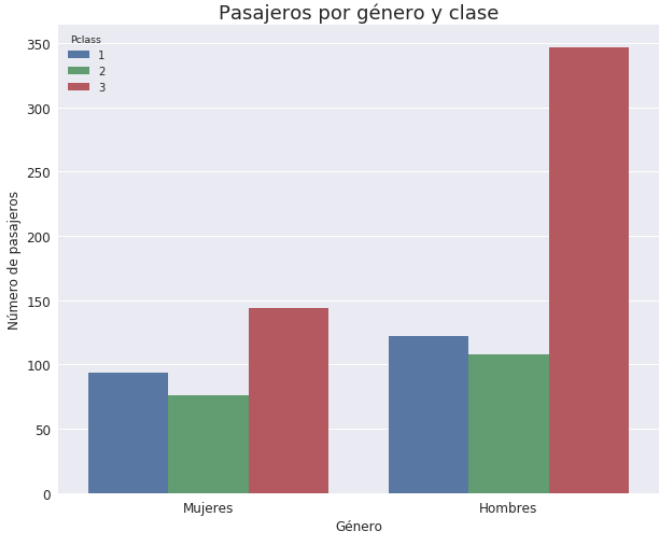
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Pasajeros por género y clase

```
sns.set(rc={"figure.figsize":(10,8)})
sns.countplot(data=train, x="Sex", hue="Pclass", order=["female", "male"])
plt.title("Pasajeros por género y clase", fontsize=18)
plt.xlabel("Género", fontsize=12)
plt.ylabel("Número de pasajeros", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Mujeres", "Hombres"], fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Pasajeros por género y clase

```
sns.set(rc={"figure.figsize":(10,8)})
sns.countplot(data=train, x="Pclass",  hue="Sex", order=[1,2,3])
plt.title("Pasajeros por clase y género", fontsize=18)
plt.xlabel("Clase", fontsize=12)
plt.ylabel("Número de pasajeros", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1,2],  ["1a clase", "2a clase", "3a clase"], fontsize=12, rotation=360)
plt.show()
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Supervivientes por género y clase

```
train[["Sex", "Pclass", "Survived"]].groupby(["Sex", "Pclass"]).mean()
```

		Survived
Sex	Pclass	
female	1	0.968085
	2	0.921053
	3	0.500000
male	1	0.368852
	2	0.157407
	3	0.135447

```
train[["Pclass", "Sex", "Survived"]].groupby(["Pclass", "Sex"]).mean()
```

		Survived
Pclass	Sex	
1	female	0.968085
	male	0.368852
2	female	0.921053
	male	0.157407
3	female	0.500000
	male	0.135447

```
train.pivot_table(index=["Sex", "Pclass"], values="Survived")
```

```
train.pivot_table(index=["Pclass", "Sex"], values="Survived")
```

Hands-On Labs

kaggle

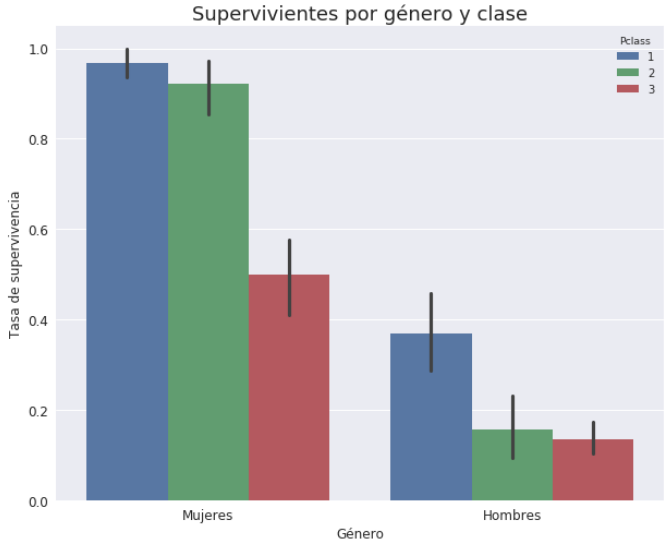
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Supervivientes por género y clase

```
sns.set(rc={"figure.figsize":(10,8)})
sns.barplot(data=train, x="Pclass", y="Survived")
plt.title("Supervivencia según clase", fontsize=18)
plt.xlabel("Clases disponibles", fontsize=12)
plt.ylabel("Tasa de supervivencia", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1,2],  ["1a clase", "2a clase", "3a clase"], fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

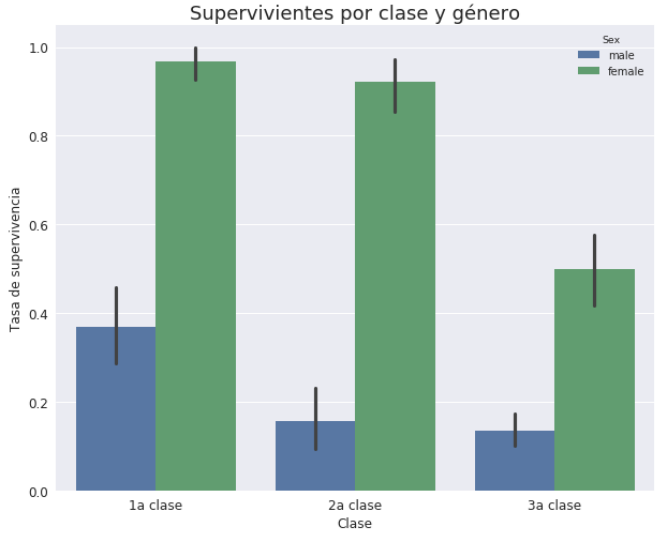
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Supervivientes por género y clase

```
sns.set(rc={"figure.figsize":(10,8)})
sns.barplot(data=train, x="Pclass", y="Survived", hue="Sex")
plt.title("Supervivientes por clase y género", fontsize=18)
plt.xlabel("Clase", fontsize=12)
plt.ylabel("Tasa de supervivencia", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1,2],  ["1a clase", "2a clase", "3a clase"], fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

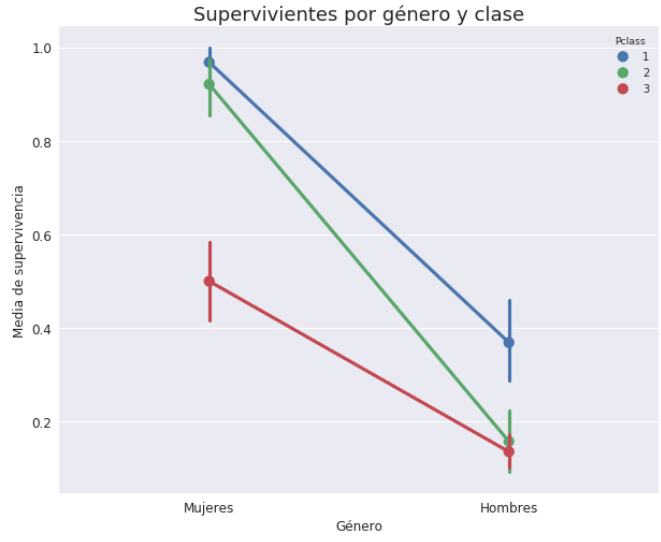
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Supervivientes por género y clase

```
sns.set(rc={"figure.figsize":(10,8)})
sns.pointplot(data=train, x="Sex", y="Survived", hue="Pclass", order=["female", "male"])
plt.title("Supervivientes por género y clase", fontsize=18)
plt.xlabel("Género", fontsize=12)
plt.ylabel("Media de supervivencia", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Mujeres", "Hombres"], fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

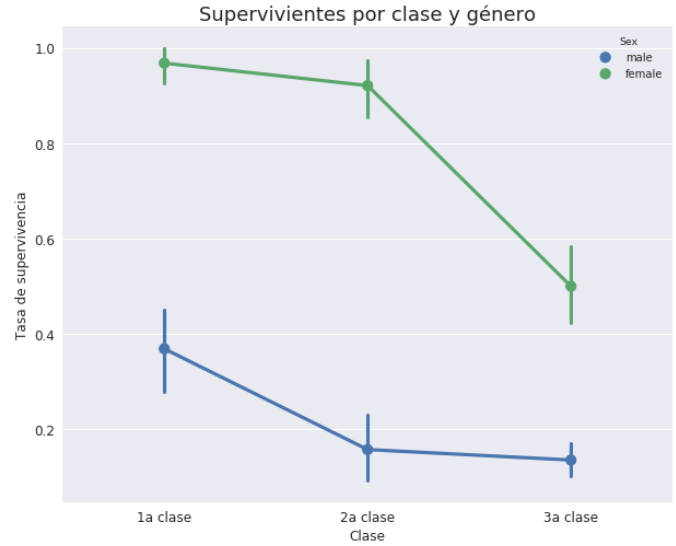
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Supervivientes por género y clase

```
sns.set(rc={"figure.figsize":(10,8)})
sns.pointplot(data=train, x="Pclass", y="Survived", hue="Sex")
plt.title("Supervivientes por clase y género", fontsize=18)
plt.xlabel("Clase", fontsize=12)
plt.ylabel("Tasa de supervivencia", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1,2],  ["1a clase", "2a clase", "3a clase"], fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Pasajeros según edad

```
print("Age: {}".format(train["Age"].dtype))
print("Total: {}".format(train.shape[0]))
print("Values: {}".format(train["Age"].count()))
print("Empty: {}".format(train["Age"].isnull().sum()))
print("Mean: {}".format(train["Age"].mean()))
```

```
Age: float64
Total: 891
Values: 714
Empty: 177
Mean: 29.69911764705882
```

```
train["Age"].describe()
```

```
count    714.000000
mean      29.699118
std       14.526497
min        0.420000
25%       20.125000
50%       28.000000
75%       38.000000
max       80.000000
Name: Age, dtype: float64
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

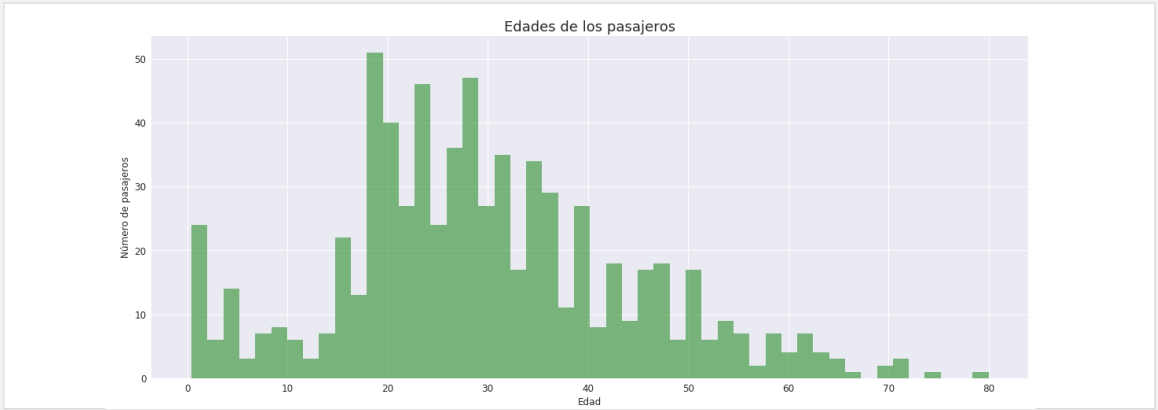
pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Pasajeros según edad

Las columnas **Sex** y **Pclass** son lo que llamamos características **categóricas**.

La columna **Age** debe tratarse de forma ligeramente diferente, ya que se trata de una columna numérica continua. Una forma de ver la distribución de valores en un conjunto numérico continuo es usar **histogramas**.

```
series = train["Age"]
series.plot.hist(alpha=0.5,color="green", bins=50, figsize=(20,8))
plt.title("Edades de los pasajeros", fontsize=18)
plt.xlabel("Edad", fontsize=12)
plt.ylabel("Número de pasajeros", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks(fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

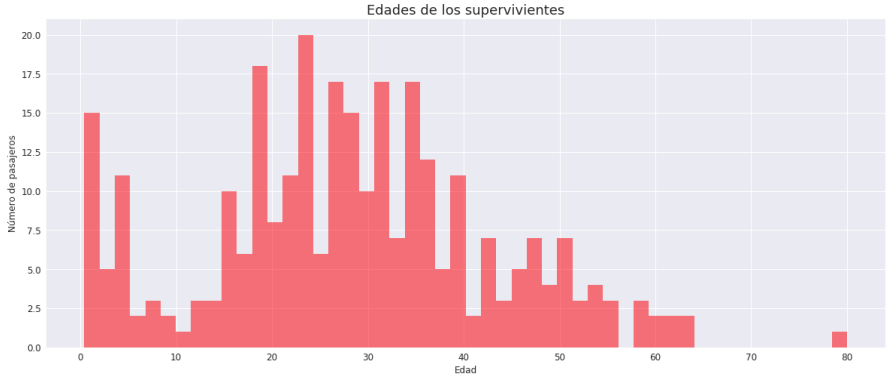
pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Supervivientes según edad

Las columnas **Sex** y **Pclass** son lo que llamamos características **categóricas**.

La columna **Age** debe tratarse de forma ligeramente diferente, ya que se trata de una columna numérica continua. Una forma de ver la distribución de valores en un conjunto numérico continuo es usar **histogramas**.

```
series = train[train["Survived"] == 1]["Age"]
series.plot.hist(alpha=0.5,color="red", bins=50, figsize=(20,8))
plt.title("Edades de los supervivientes", fontsize=18)
plt.xlabel("Edad", fontsize=12)
plt.ylabel("Número de pasajeros", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks(fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

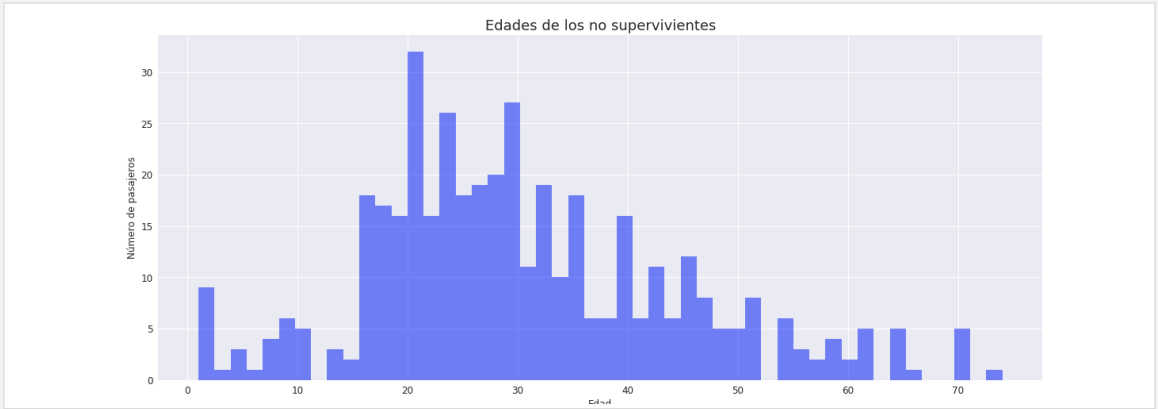
pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Supervivientes según edad

Las columnas **Sex** y **Pclass** son lo que llamamos características **categóricas**.

La columna **Age** debe tratarse de forma ligeramente diferente, ya que se trata de una columna numérica continua. Una forma de ver la distribución de valores en un conjunto numérico continuo es usar **histogramas**.

```
series = train[train["Survived"] == 0]["Age"]
series.plot.hist(alpha=0.5,color="blue", bins=50, figsize=(20,8))
plt.title("Edades de los no supervivientes", fontsize=18)
plt.xlabel("Edad", fontsize=12)
plt.ylabel("Número de pasajeros", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks(fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

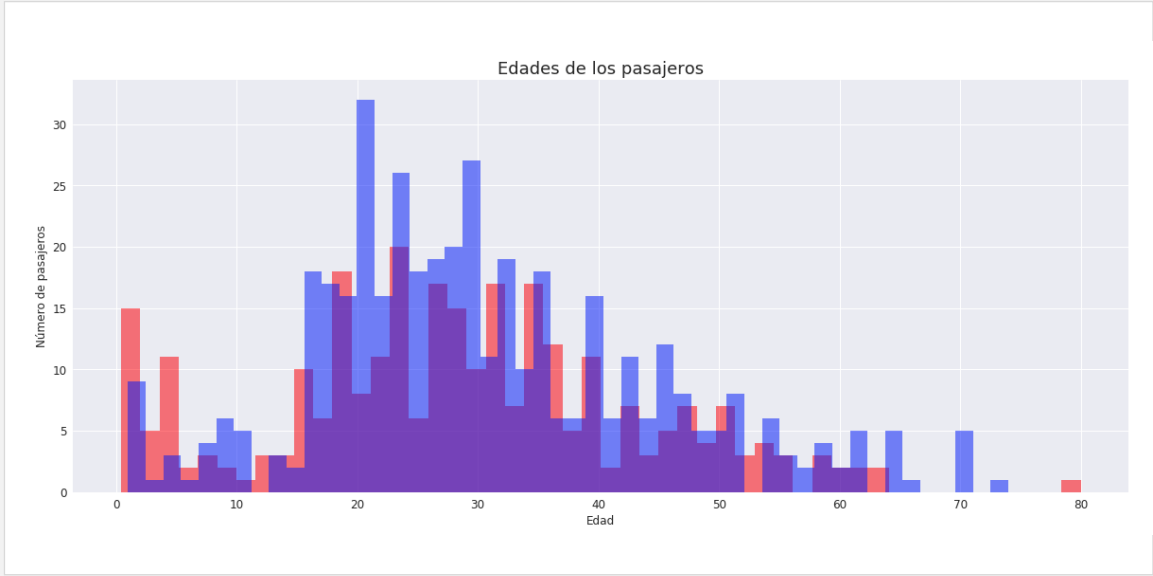
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Supervivientes según edad

```
survived = train[train["Survived"] == 1]["Age"]
died = train[train["Survived"] == 0]["Age"]
survived.plot.hist(alpha=0.5,color="red", bins=50, figsize=(20,8))
died.plot.hist(alpha=0.5,color="blue", bins=50, figsize=(20,8))
plt.title("Edades de los pasajeros", fontsize=18)
plt.xlabel("Edad", fontsize=12)
plt.ylabel("Número de pasajeros", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks(fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

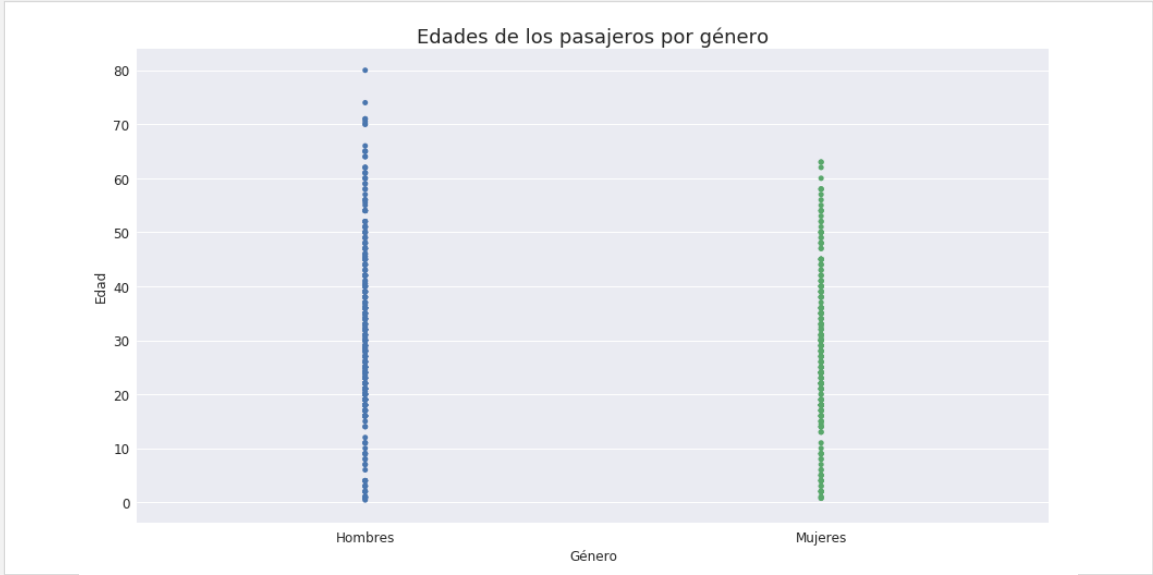
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Hombres y mujeres por edad

```
sns.set(rc={"figure.figsize":(15,8)})
sns.stripplot(data=train, x="Sex", y="Age", jitter = False)
plt.title("Edades de los pasajeros por género", fontsize=18)
plt.xlabel("Género", fontsize=12)
plt.ylabel("Edad", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Hombres", "Mujeres"], fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

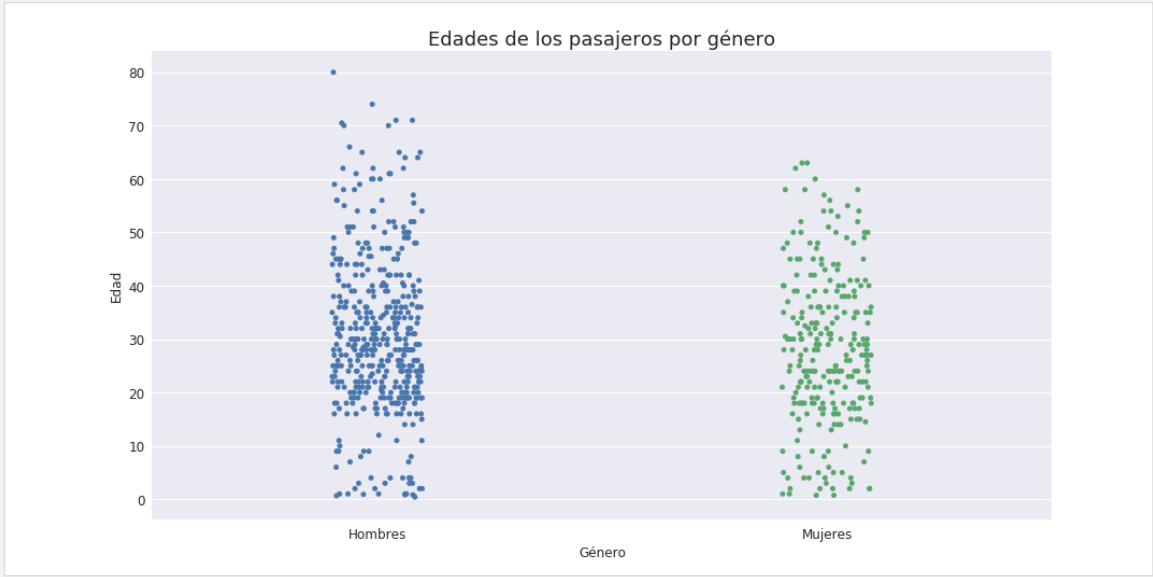
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Hombres y mujeres por edad

```
sns.set(rc={"figure.figsize":(15,8)})
sns.stripplot(data=train, x="Sex", y="Age", jitter = True)
plt.title("Edades de los pasajeros por género", fontsize=18)
plt.xlabel("Género", fontsize=12)
plt.ylabel("Edad", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Hombres", "Mujeres"], fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

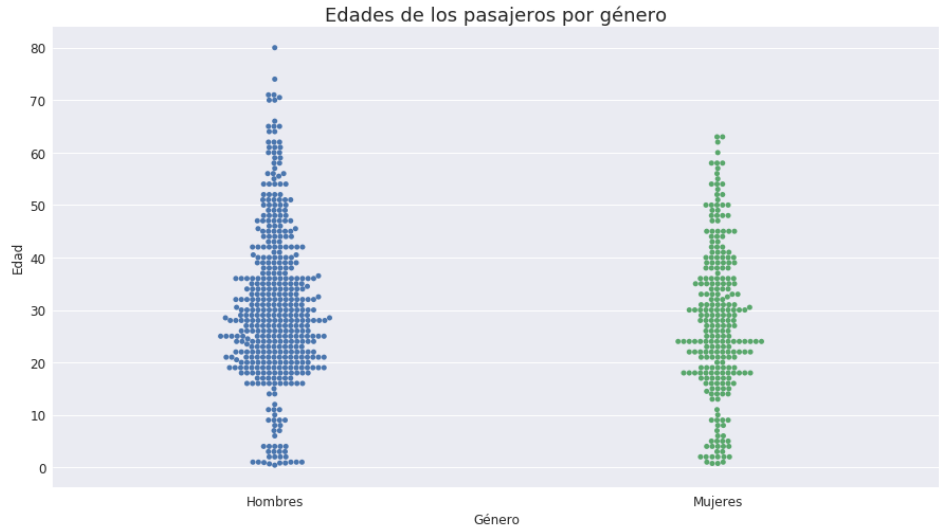
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Hombres y mujeres por edad

```
sns.set(rc={"figure.figsize":(15,8)})
sns.swarmplot(data=train, x="Sex", y="Age")
plt.title("Edades de los pasajeros por género", fontsize=18)
plt.xlabel("Género", fontsize=12)
plt.ylabel("Edad", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Hombres", "Mujeres"], fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

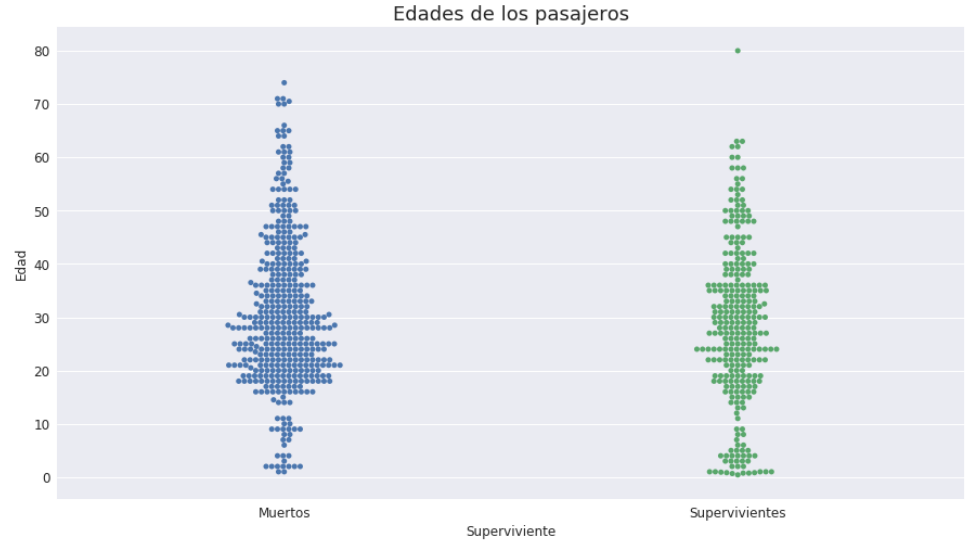
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Supervivientes por edad

```
sns.set(rc={"figure.figsize":(15,8)})
sns.swarmplot(data=train, x="Survived", y="Age")
plt.title("Edades de los pasajeros", fontsize=18)
plt.xlabel("Superviviente", fontsize=12)
plt.ylabel("Edad", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Muertos", "Supervivientes"], fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Supervivientes por edad y género

```
sns.set(rc={"figure.figsize":(15,8)})
sns.swarmplot(data=train, x="Survived", y="Age", hue="Sex")
plt.title("Edades de los pasajeros", fontsize=18)
plt.xlabel("Superviviente", fontsize=12)
plt.ylabel("Edad", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Muertos", "Supervivientes"], fontsize=12, rotation=360)
plt.legend(["Hombres", "Mujeres"])
plt.show()
```

Hands-On Labs

kaggle

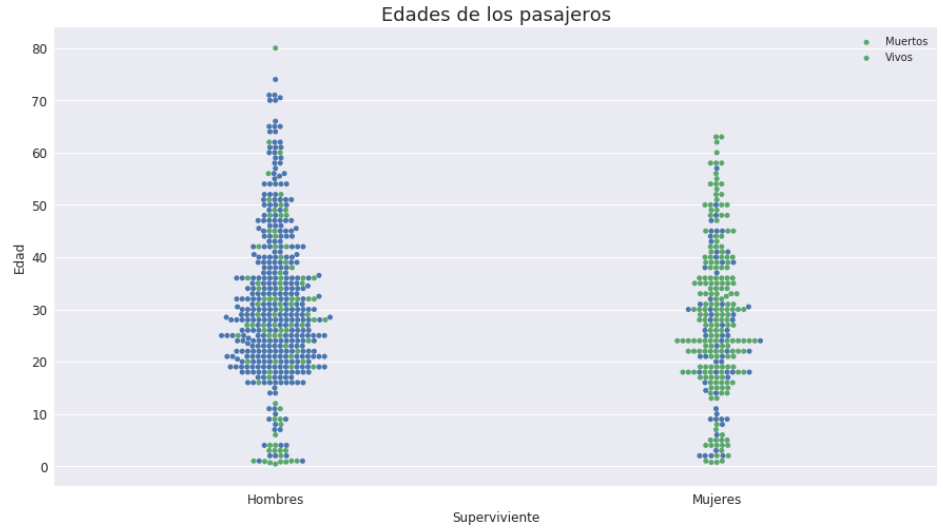
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Supervivientes por edad y género

```
sns.set(rc={"figure.figsize":(15,8)})
sns.swarmplot(data=train, x="Sex", y="Age", hue="Survived")
plt.title("Edades de los pasajeros", fontsize=18)
plt.xlabel("Superviviente", fontsize=12)
plt.ylabel("Edad", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Hombres", "Mujeres"], fontsize=12, rotation=360)
plt.legend(["Muertos", "Vivos"])
plt.show()
```



Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

PUE IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Qué hacemos con la edad?

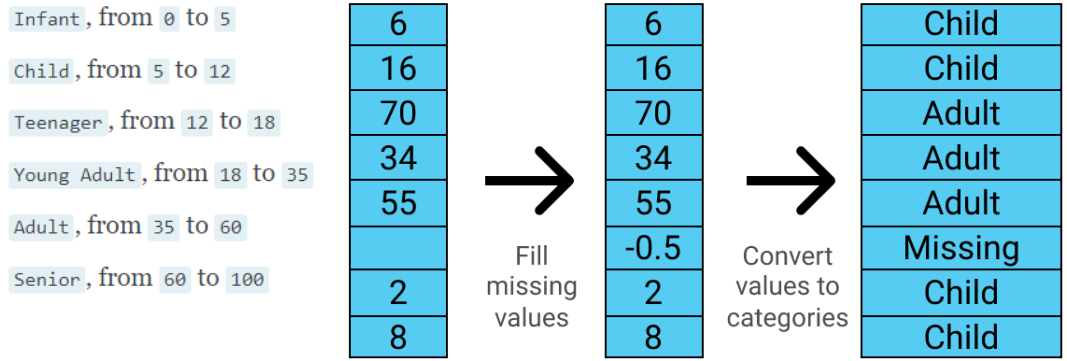
Para que el campo **Age** sea útil para nuestro modelo de aprendizaje automático, podemos separar esta característica **continua** en una característica **categórica** dividiéndola en **rangos**.

Usaremos la función *pandas.cut ()* para ayudarnos.

Antes de nada, debemos tener en cuenta dos cosas:

1. Cualquier cambio que hagamos en el conjunto de entrenamiento, también debemos realizarlo en nuestro conjunto de pruebas.

2. La mayoría de los algoritmos de Machine Learning requieren que todas las variables tengan valor, para poderlas usar en el entrenamiento del modelo.



Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Qué hacemos con la edad?

```
for dataframe in titanic:
    dataframe["Age"] = dataframe["Age"].fillna();
```

```
cut_points = [0,5,12,18,35,60,100]
label_names = ["Infant", "Child","Teenager", "YoungAdult", "Adult", "Senior"]
for dataframe in titanic:
    dataframe["Age"] = pd.cut(dataframe["Age"], cut_points, labels = label_names)
```

```
train["Age"].describe()
```

```
count           891
unique            6
top       YoungAdult
freq            535
Name: Age, dtype: object
```

```
train["Age"].value_counts().sort_index()
```

```
Infant         44
Child          25
Teenager       70
YoungAdult    535
Adult         195
Senior         22
Name: Age, dtype: int64
```

Hands-On Labs

kaggle

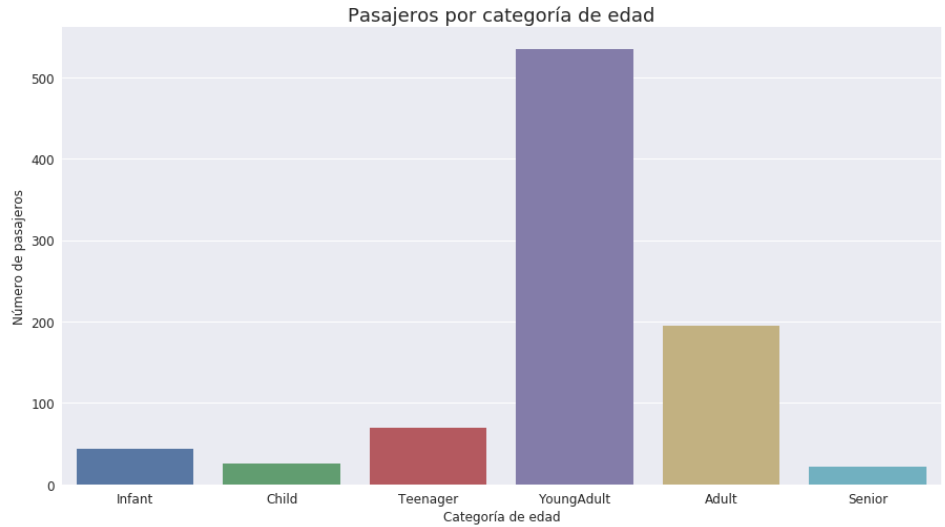
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Pasajeros por categoria de edad

```
sns.set(rc={"figure.figsize":(15,8)})
sns.countplot(data=train, x="Age")
plt.title("Pasajeros por categoría de edad", fontsize=18)
plt.xlabel("Categoría de edad", fontsize=12)
plt.ylabel("Número de pasajeros", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks(fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

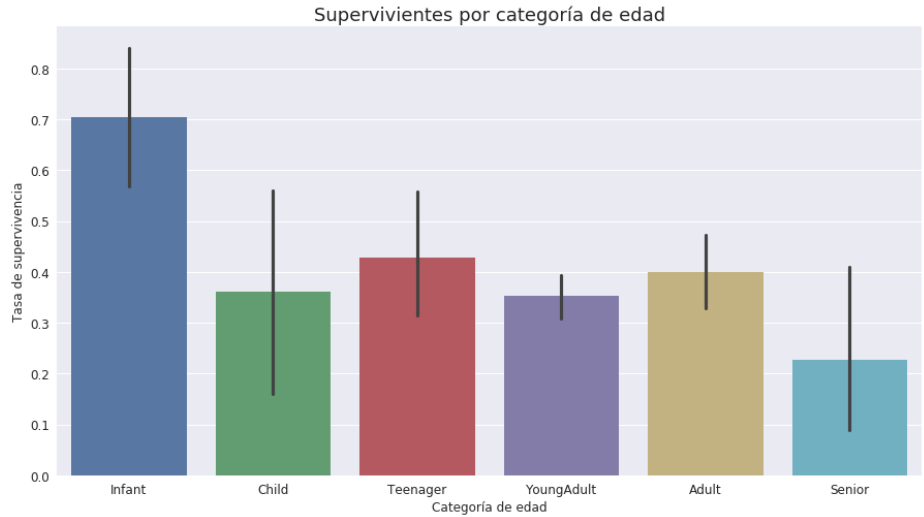
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Supervivientes por categoria de edad

```
sns.set(rc={"figure.figsize":(15,8)})
sns.barplot(data=train, x="Age", y="Survived")
plt.title("Supervivientes por categoría de edad", fontsize=18)
plt.xlabel("Categoría de edad", fontsize=12)
plt.ylabel("Tasa de supervivencia", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks(fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

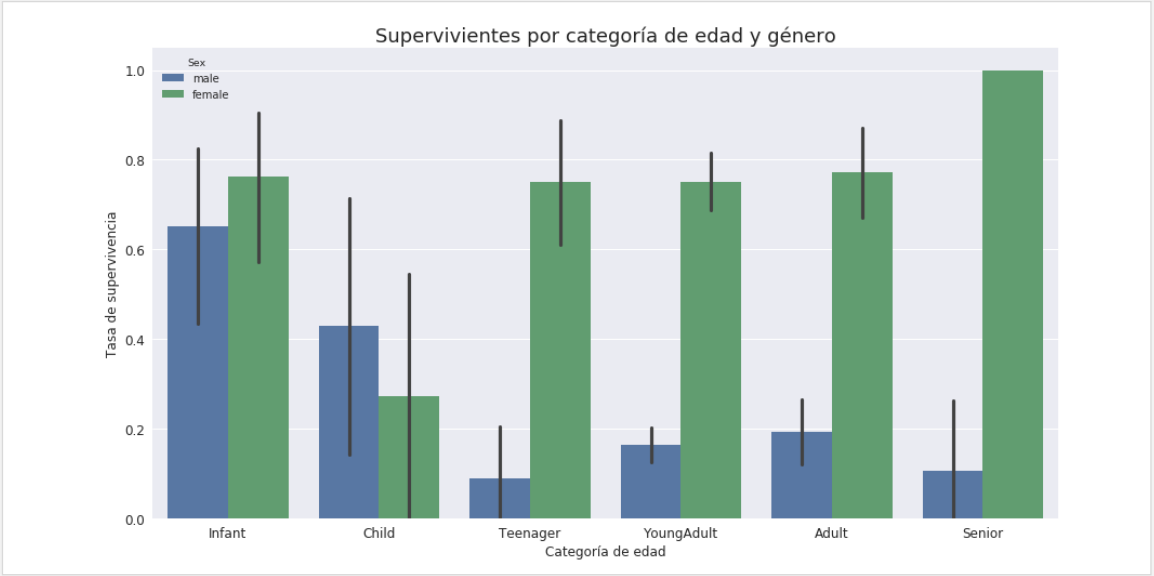
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Supervivientes por categoria de edad y género

```
sns.set(rc={"figure.figsize":(15,8)})
sns.barplot(data=train, x="Age", y="Survived", hue="Sex")
plt.title("Supervivientes por categoría de edad y género", fontsize=18)
plt.xlabel("Categoría de edad", fontsize=12)
plt.ylabel("Tasa de supervivencia", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks(fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

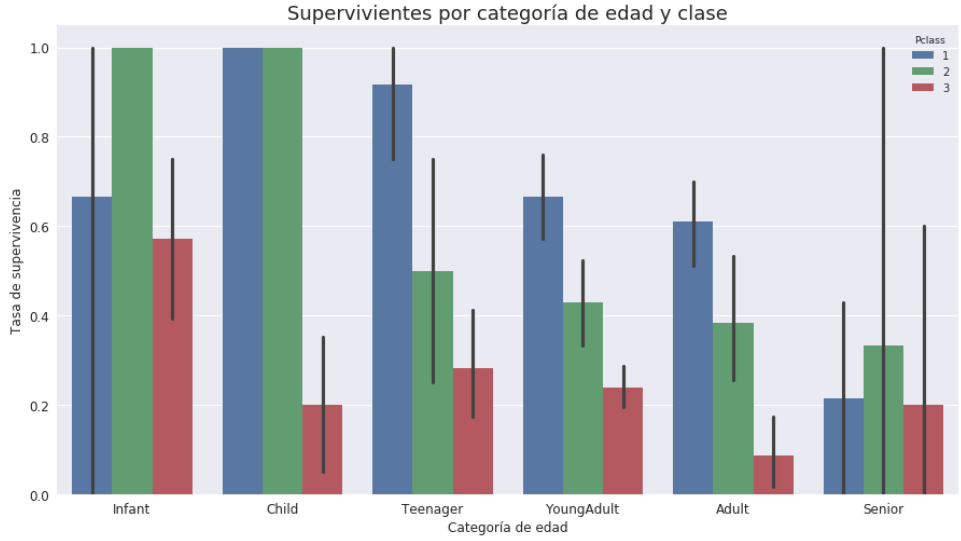
www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Supervivientes por categoria de edad y clase

```
sns.set(rc={"figure.figsize":(15,8)})
sns.barplot(data=train, x="Age", y="Survived", hue="Pclass")
plt.title("Supervivientes por categoría de edad y clase", fontsize=18)
plt.xlabel("Categoría de edad", fontsize=12)
plt.ylabel("Tasa de supervivencia", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks(fontsize=12, rotation=360)
plt.show()
```



Hands-On Labs

kaggle

www.kaggle.com/c/titanic

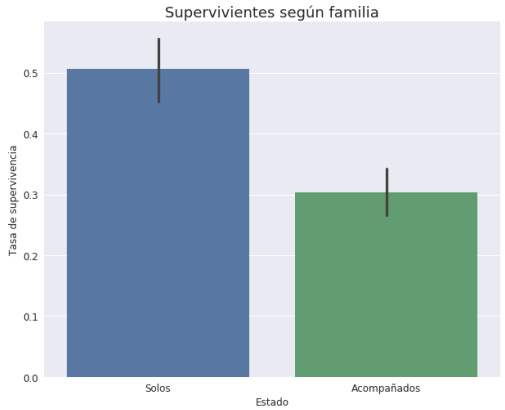
Networking Academy

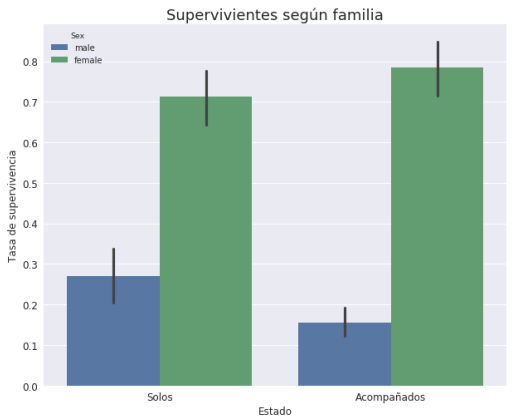
pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Feature Engineering

```
for dataframe in titanic:
    dataframe["IsAlone"] = ((dataframe["SibSp"] + dataframe["Parch"]) == 0)
```

```
sns.set(rc={"figure.figsize":(10,8)})
sns.barplot(data=train, x="IsAlone", y="Survived")
plt.title("Supervivientes según familia", fontsize=18)
plt.xlabel("Estado", fontsize=12)
plt.ylabel("Tasa de supervivencia", fontsize=12)
plt.yticks(fontsize=12)
plt.xticks([0,1],  ["Solos", "Acompañados"], fontsize=12, rotation=360)
plt.show()
```





Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Preparando los datos para Machine Learning

Hemos identificado tres columnas o características que pueden ser útiles en nuestro modelo **para predecir la supervivencia o no** de los pasajeros del Titanic:

- *Sex*
- *Pclass*
- *Age*

Antes de **construir nuestro modelo**, debemos preparar estas columnas para el aprendizaje automático. La mayoría de los algoritmos de *Machine Learning* no pueden entender las etiquetas de texto, por lo que **debemos convertir nuestros valores en números**.

Además, debemos tener cuidado de **no implicar ninguna relación numérica** en casos **donde no haya ninguna**.

`pandas.get_dummies()`

Pclass
3
1
3
1
3
3
1
3
3
2

Pclass_1	Pclass_2	Pclass_3
0	0	1
1	0	0
0	0	1
1	0	0
0	0	1
0	0	1
1	0	0
0	0	1
0	0	1
0	1	0

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Preparando los datos para Machine Learning

```
pd.get_dummies(train["Age"], prefix="Age")
```

	Age_Infant	Age_Child	Age_Teenager	Age_YoungAdult	Age_Adult	Age_Senior
0	0	0	0	1	0	0
1	0	0	0	0	1	0
2	0	0	0	1	0	0
3	0	0	0	1	0	0
4	0	0	0	1	0	0
5	0	0	0	1	0	0
6	0	0	0	0	1	0
7	1	0	0	0	0	0
8	0	0	0	1	0	0

```
for column in ["Pclass", "Sex", "Age", "Embarked", "IsAlone"]:
    train = pd.concat([train, pd.get_dummies(train[column], prefix=column)], axis = 1)
    test = pd.concat([test, pd.get_dummies(test[column], prefix=column)], axis = 1)
```

```
train.sample(2)
```

```
test.sample(2)
```

```
train = train.drop(["Pclass", "Name", "Sex", "Age", "SibSp", "Parch", "Ticket", "Fare", "Cabin", "Embarked", "IsAlone"], axis = 1)
test = test.drop(["Pclass", "Name", "Sex", "Age", "SibSp", "Parch", "Ticket", "Fare", "Cabin", "Embarked", "IsAlone"], axis = 1)
titanic = [train,test]
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Construyendo nuestro modelo de Machine Learning

Ahora que nuestros datos están preparados, estamos listos para **construir y entrenar a nuestro primer modelo** de Machine Learning. Como ejemplo usaremos un modelo de tipo **Logistic Regression.**

Para ello, nos ayudaremos de la biblioteca **scikit-learn,** que nos ofrece muchas herramientas que facilitan el aprendizaje automático.

El flujo de trabajo scikit-learn consta de cuatro pasos principales:

1. Construir el modelo de Machine Learning que deseamos usar
2. Entrenar el modelo con los datos de entrenamiento (**Fit**)
3. Usa el modelo para hacer predicciones (**Predict**)
4. Evaluar la precisión de las predicciones (**Accuracy**)

```
from sklearn.linear_model import LogisticRegression
```

A entrenar nuestro modelo ☺

```
model = LogisticRegression()
model.fit(train.iloc[:, 2:], train["Survived"])
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

Networking Academy

IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Entrenando a nuestro modelo de Machine Learning

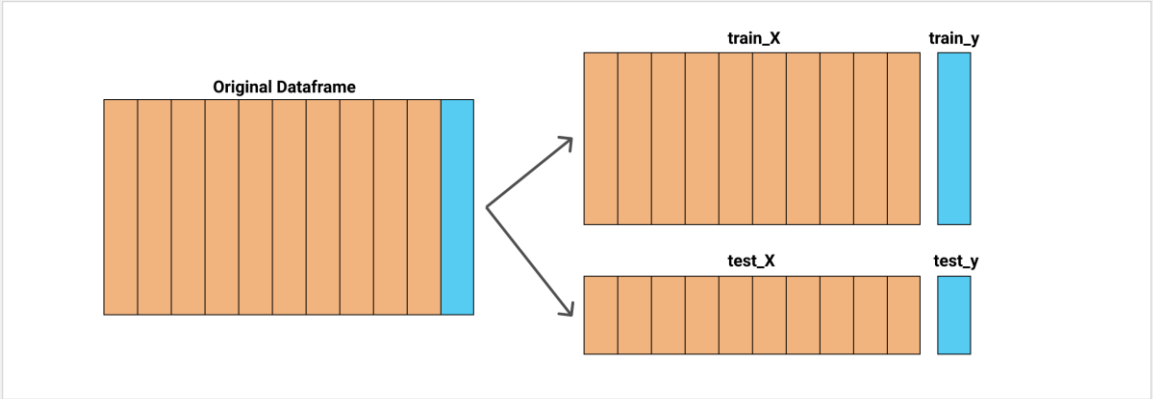
¡Felicidades, acabas de entrenar a tu primer modelo de aprendizaje automático! Nuestro siguiente paso es averiguar cómo de preciso es nuestro modelo, y para hacer eso, tendremos que hacer algunas predicciones.

Recuerda que tenemos un conjunto de datos de prueba, que podríamos usar para hacer predicciones. Si hiciéramos predicciones sobre ese conjunto de datos, al no tener la columna "Survived", tendríamos que enviarlo a Kaggle para averiguar nuestra precisión.

Podríamos entrenar y predecir en nuestro conjunto de datos de entrenamiento. Sin embargo, si hacemos esto, hay una alta probabilidad de que nuestro modelo se considere **overfit**, lo que significa que funcionará bien porque estamos probando los mismos datos en los que hemos entrenado, pero luego funcionar mucho peor en datos nuevos, no vistos.

Dividiremos nuestro cojunto de entrenamiento en dos:

1. Una parte para **entrenar** nuestro modelo (80% de las observaciones)
2. Una parte para hacer **predicciones** y probar nuestro modelo (20% de las observaciones)



Hands-On Labs

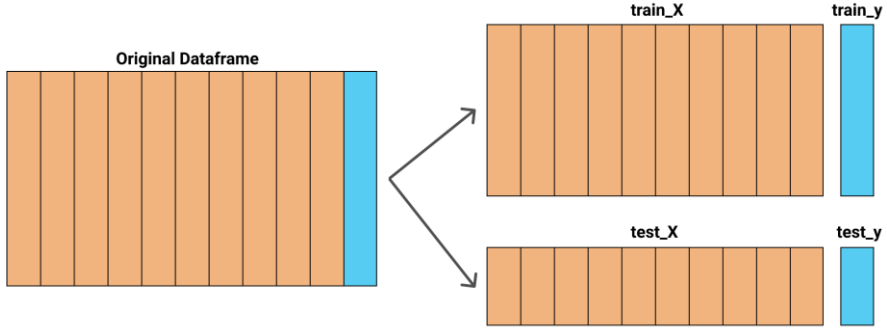
kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Entrenando a nuestro modelo de Machine Learning




```
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import accuracy_score
```

```
train_X, test_X, train_y, test_y =
    train_test_split(train.iloc[:, 2:], train["Survived"], test_size=0.20,random_state=0)
```

```
model = LogisticRegression()
model.fit(train_X, train_Y)
predictions = model.predict(test_X)
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

PUE IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Entrenando a nuestro modelo de Machine Learning

Our model's prediction	The actual value	Correct
0	0	Yes
1	0	No
0	1	No
1	1	Yes
1	1	Yes

```
accuracy = accuracy_score(test_Y, predictions)
print("Accuracy: {}".format(accuracy))
```

```
Accuracy: 0.8100558659217877
```

Nuestro modelo tiene una precisión del 81.0%.

Debido a que este conjunto de datos es bastante pequeño, existe una gran probabilidad de que nuestro modelo esté sobreajustado, y que por lo tanto, no funcione tan bien en datos totalmente nuevos.

Hands-On Labs

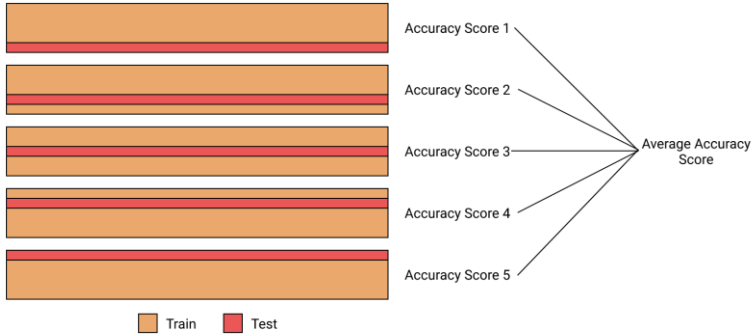
kaggle

www.kaggle.com/c/titanic

Networking Academy

PUE IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Entrenando a nuestro modelo de Machine Learning



Cross-validation: *k-fold cross validation*

```
model = LogisticRegression()
scores = cross_val_score(model, train.iloc[:, 2:], train["Survived"], cv=10)
scores.sort()
accuracy = scores.mean()

print("Puntuaciones: {}".format(scores))
print("Precision: {}".format(accuracy))
```

```
Puntuaciones: [0.76404494 0.7752809  0.78651685 0.78651685 0.78651685 0.8
 0.80681818 0.82022472 0.83333333 0.87640449]
Precision: 0.8035657133129043
```

Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Prediciendo con nuestro modelo de Machine Learning

```
model = LogisticRegression()
model.fit(train.iloc[:, 2:], train["Survived"])
kaggle_predictions = model.predict(test.iloc[:,1:])
```

```
test_ids = test["PassengerId"]
submission_df = {"PassengerId": test_ids, "Survived": kaggle_predictions}
submission = pd.DataFrame(submission_df)
```

```
submission.to_csv("submission.csv",index=False)
```

Hands-On Labs

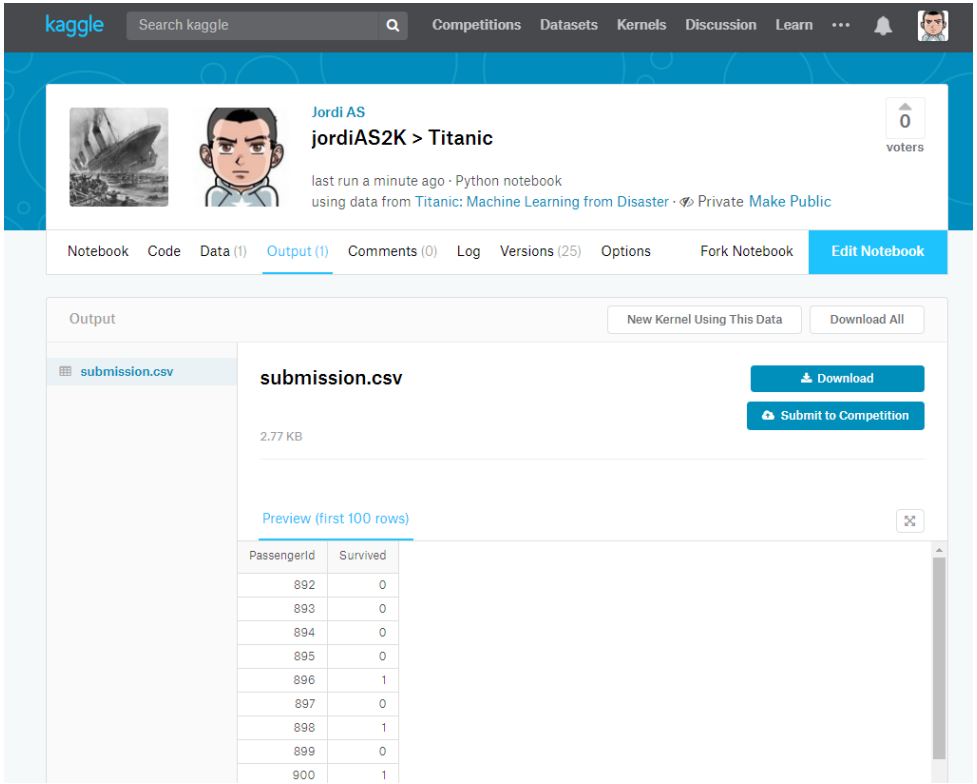
kaggle

www.kaggle.com/c/titanic

Networking Academy

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Subiendo nuestras predicciones a Kaggle

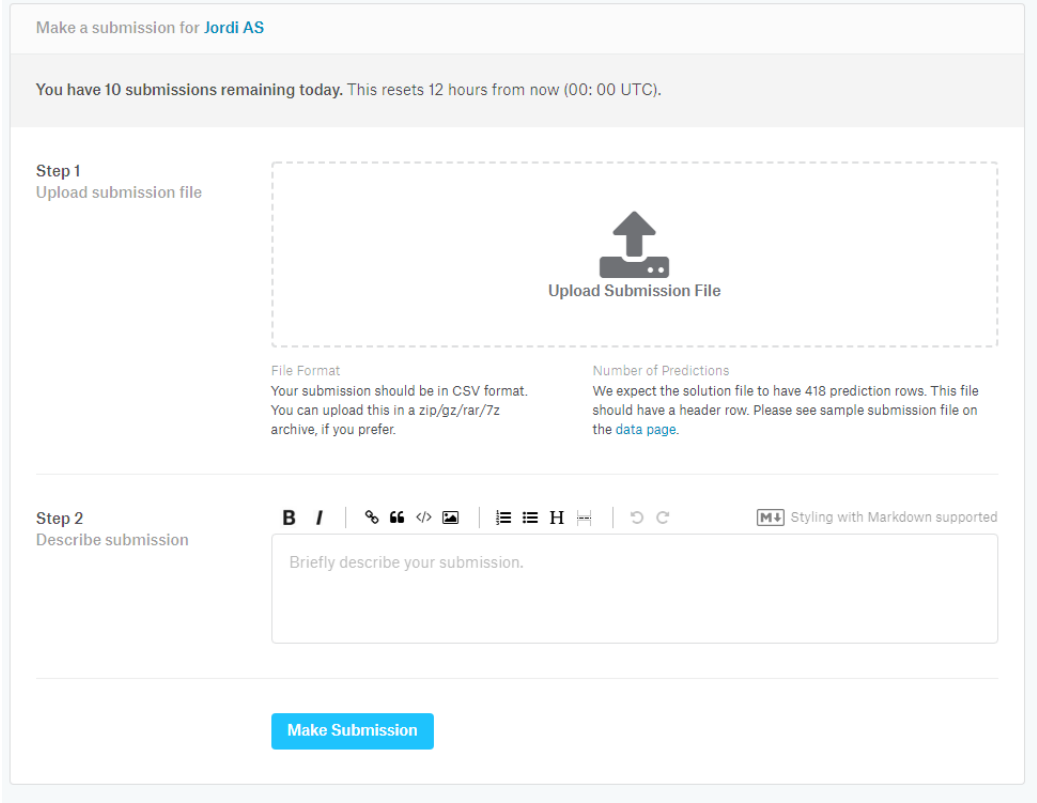


Hands-On Labs

kaggle

www.kaggle.com/c/titanic

Subiendo nuestras predicciones a Kaggle



Networking Academy

PUE IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

Hands-On Labs

Subiendo nuestras predicciones a Kaggle

kaggle

www.kaggle.com/c/titanic

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
submission.csv	just now	0 seconds	0 seconds	0.75598

Complete

Jump to your position on the leaderboard

Public Leaderboard | Private Leaderboard

This leaderboard is calculated with approximately 50% of the test data.
The final results will be based on the other 50%, so the final standings may be different.

Raw Data | Refresh

#	△1w	Team Name	Kernel	Team Members	Score	Entries	Last
1	–	souravstat			1.00000	1	2mo
2	–	giim			1.00000	1	2mo
3	–	toto250990			1.00000	32	2mo
4	–	BIG_CHENG			1.00000	2	2mo
5	–	Martin Høy			1.00000	4	1mo
6	–	ddaraujo			1.00000	1	1mo
7	–	ryemitan			1.00000	1	18d
8	–	Douglas 3			1.00000	17	9d

Networking Academy

IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¡Muchas gracias!

jordi.arino@pue.es
@jordiAS2K

#PUEDAY18

Networking Academy

PYTHON INSTITUTE
Open Education & Development Group

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

¿Alguna pregunta?

www.pue.es/cisco
educacion@pue.es

#PUEDAY18

Networking Academy

PYTHON INSTITUTE
Open Education & Development Group

pue IMPULSANDO EL CONOCIMIENTO TIC CUALIFICADO

BARCELONA – MADRID

www.pue.es

¡Gracias!

#PUEDAY18

educacion@pue.es

93 206 02 49

PUE 5TH DAY MADRID 2018

ORACLE ACADEMY

cloudera ACADEMIC PARTNER

Microsoft Imagine Academy

vmware IT ACADEMY

Linux Professional Institute

RED HAT ACADEMY redhat.

Networking Academy cisco