Day 1 Notes - Stats

# Creating a Lab environment

**To install git**

<https://github.com/git-guides/install-git>

Make sure 'git' is in your PATH

C:\Users\<user>\AppData\Local\GitHubDesktop\app-<version>\resources\app\git\cmd



**One time setup**

Download and install anaconda

<https://www.anaconda.com/products/individual>

$ conda env list

$ conda create --name pyds python=3.8

$ **conda activate pyds**

**# make sure your prompt says "pyds"**

$ git clone <https://github.com/fenago/machine-learning-essentials-module1>

$ cd machine-learning-essentials-module1

$ pip install -r req.txt

$ jupyter lab

**After restart**

$ cd machine-learning-essentials-module1

$ conda activate pyds

$ jupyter lab

# Stats

[Fantastic video series on Stats by Brandon Foltz](https://www.youtube.com/playlist?list=PLyuzqMwtJuvKwadONz4R9H2JFrOi8LFWt)

[Stats-1 worksheet](https://docs.google.com/spreadsheets/d/1p-OWIZhRZPUKCttjs30qus5IivWUgiBpMNwIQ-2qfgk/edit?usp=sharing) - work through some numbers

**Why are we dividing by (n-1) instead of n**

<https://en.wikipedia.org/wiki/Bessel%27s_correction>

https://stackoverflow.com/questions/41204400/what-is-the-difference-between-numpy-var-and-statistics-variance-in-python

**IQR**



**Boxplot**

https://mathworld.wolfram.com/Box-and-WhiskerPlot.html

**KDE / PDF**

* [KDE explained visually](https://mathisonian.github.io/kde/)
* [Gentle intro to PDF](https://machinelearningmastery.com/probability-density-estimation/)
* [Nice intro to pdf](https://online.stat.psu.edu/stat414/lesson/14/14.1)

**Examples of Left / Right distributions**

Left

* Age vs fitness
* Age vs wealth?
* India Olympic Medal Tally : Year vs # of Medals (left)
* Wealth accumulation and spending

Right

* Covid Infection and recovery rate

Uniform

* Exam scores
* Employee salary
* Height / weight
* Performance ratings

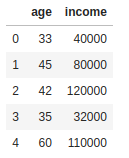
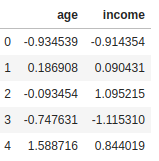
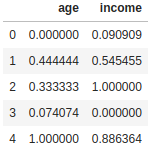
**Scaling**

Left - original data

Middle - z-scaled

Right - [0 - 1]

Z score (age) = (age - mean) / std

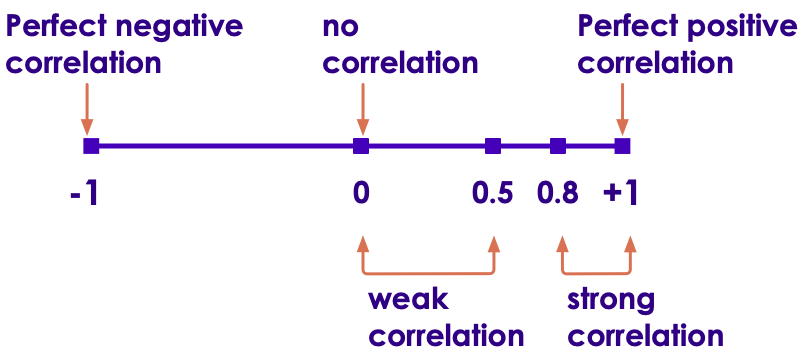
  

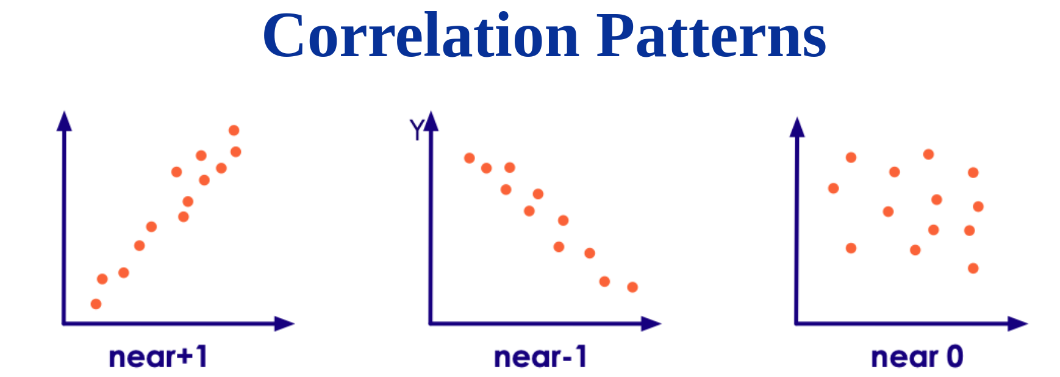


**Covariance**

[Nice explanation at 15:14 of this video](https://www.youtube.com/watch?v=xGbpuFNR1ME&list=PLyuzqMwtJuvKwadONz4R9H2JFrOi8LFWt&index=11&t=4s)

**Correlation**





**Anscombe Quartet**

<https://en.wikipedia.org/wiki/Anscombe%27s_quartet>

**Descriptive vs Inferential Statistics**

<https://www.scribbr.com/frequently-asked-questions/whats-the-difference-between-descriptive-and-inferential-statistics/>

**Standard deviation**

[Nice explainer video](https://www.youtube.com/watch?v=MRqtXL2WX2M)

**Lab-1 : Basic stats**

import numpy as np

np.mean(salaries)

**Stats bonus lab**

House sales data : <https://elephantscale-public.s3.amazonaws.com/data/house-prices/house-sales-simplified.csv>

import pandas as pd

house\_prices = pd.read\_csv('https://elephantscale-public.s3.amazonaws.com/data/house-prices/house-sales-simplified.csv')

house\_prices

# find sale by bedrooms

bedroom\_sales = house\_prices.groupby("Bedrooms").size()

bedroom\_sales

# extract 3br data

filtered = house\_prices.loc [house\_prices['Bedrooms'] == 3]

# extract salesprice for 3br

saleprice\_3br = filtered['SalePrice']

saleprice\_3br

# calculate all essential metrics

# range, mean, median, IQR

import numpy as np

print ("avg : ", np.mean (saleprice\_3br))

print ("min : ", np.min (saleprice\_3br))

# TODO : extract houses that were built after year 2000

# calculate price metrics

Compute the following:

* What is the most popular number of bedrooms?
* Extract sale prices for the most popular bedrooms
* Calculate range / mean / median / IQR for Saleprice
* I want to make an offer on a particular house. I want to make sure I offer a price that covers 90% sales. What is my offer price?
  + If Q3 (75%) is 600k -- this covers 75% of the houses sold

# Home work

* Setup your local anaconda env
* Complete labs 1 & 2