Session 1: Introduction

Module BUSN9690

Business Statistics with Python

Professor Shaomin Wu

House keeping: Learning and teaching

- What will happen
  - Lecture time: I introduce a topic
  - Lab time: You read questions, solve them, and then check the solutions
- In addition to listening, you learn by ***practicing***
- Help is always available!
  - Exercise time: you may ask for help
  - Office hours
- If more than two students have the same question, I will show the whole class how to do it
- If you have any questions, do ask: s.m.wu@kent.ac.uk

Teaching materials & Assessment

- Teaching materials: Available on moodle.kent.ac.uk
  - Use your Kent username and password to log in
  - Choose module BUSN9690, “Business Statistics with Python”
- Read the module guide
- Your attendance will be registered
- Lecture is recorded
- Assessment

Assessment	Time/Due date	Weighting
In-course test (VLN) 1	45 minutes	20%
In-course test 2	45 minutes	20%
Individual Assignment (up to 2500 words)	Submission due time: the first week in the spring term	60%

Readings and useful websites

- Readings:
  - Python
    - Heinold, B, 2012. *A practical introduction to Python programming*.
    - Verzani, J., 2016. *An Introduction to Statistics with Python -- With Applications in the Life Sciences*. Springer press.
  - Statistics
    - Weiss, N.A., 2012. Introductory statistics. Boston: Addison-Wesley.

- Useful websites
  - Python: https://www.w3schools.com/python/default.asp
  - Machine learning: https://www.kdnuggets.com
  - Datasets: https://archive.ics.uci.edu/ml/index.php

Two modules on statistical data analysis

- BUSN9690(CB969): Business Statistics with Python
- BUSN9040(CB9040): Machine Learning and Forecasting

- Why do we need to learn BUSN9690?
  - Need to know and be able to use a computing language
  - BUSN9690 paves the way for BUSN9040
    - Model checking
    - Coefficient interpretation

Indicative module content

1. Session 1. Introduction to the module
2. Session 2. Data Type and Basic Operators
3. Session 3. Python library: NumPy
4. Session 4. Python library: Pandas & SciPy
5. Session 5. Control statements
6. Session 6. Introduction to Statistics
7. Session 7. Reading Week
8. Session 8. Probability and Bayes Theorem
9. Session 9. Discrete probability distributions
10. Session 10. Continuous probability distributions
11. Session 11. Point estimates and confidence intervals
12. Session 12. Tests of Hypotheses

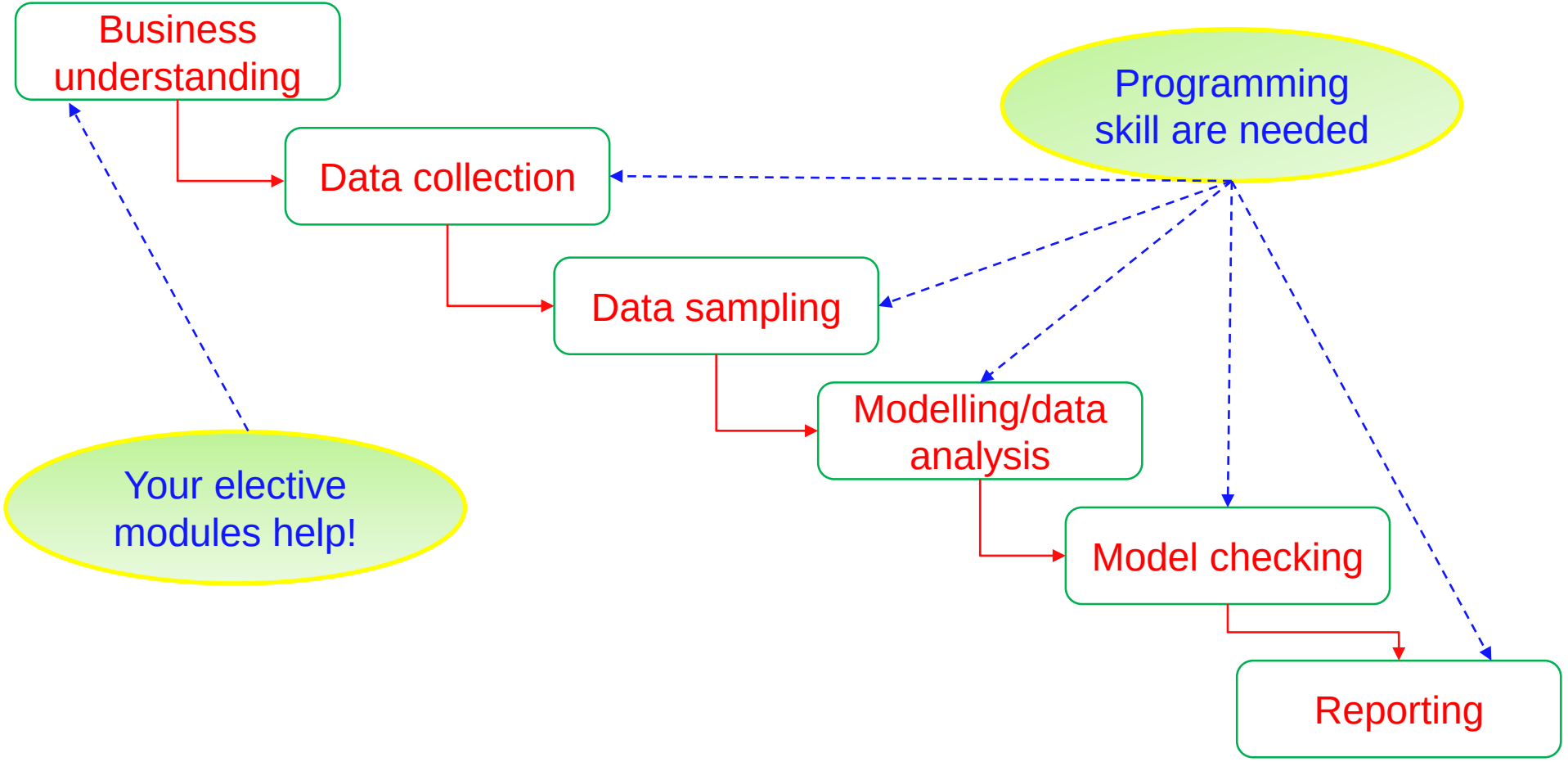
Some concepts

Philosophy of the module

- Statistical theory
  - can be very mathematical
- Practical statistics
  - from sample to population
  - help us understand the real world
- Choose the right model/method
  - then interpret the results
- Useful results must
  - be reported clearly
  - be helpful in decision making for business managers

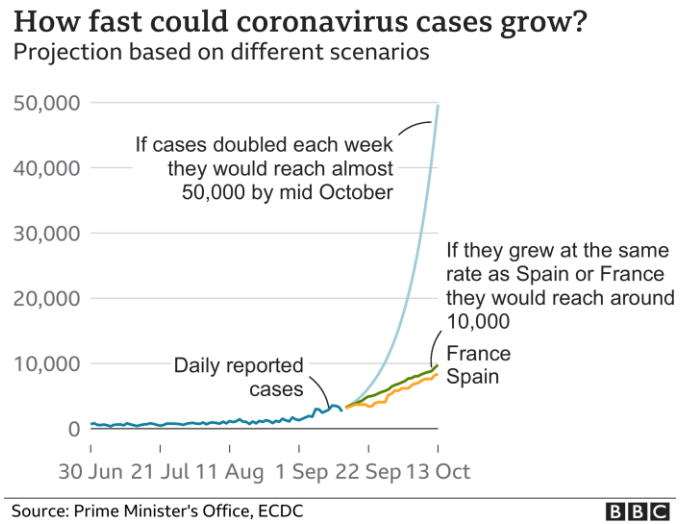
Model: the relationship between variables. Typically, it is a mathematical formula

A data analysis process



Example: coronavirus strategy in the UK

How can we curb COVID-19?	Test results	to sample data for different tasks	to build a model based on data collected	to check the model	to make suggestion
Business understanding	Data collection	Data sampling	Modelling	Model checking	Reporting



Basic terminology

- ***Data*** are the facts or measurements that are collected, analyzed, presented, and interpreted

- A ***variable*** is an attribute of an element that may assume different values; its synonyms: feature

- An ***observation*** is the set of values on the variables for a single element; its synonyms: instance

- ***Dataset*** refers to all the data, across all the observations and all variables, collected for any given study

Example: adult dataset

age	workclass	education	marital_status	occupation	gender	hours_per_week	income
39	State-gov	Bachelors	Never-married	Adm-clerical	Male	40	<=50K
50	Self-emp-not-inc	Bachelors	Married-civ-spouse	Exec-managerial	Male	13	<=50K
38	Private	HS-grad	Divorced	Handlers-cleaners	Male	40	<=50K
53	Private	11th	Married-civ-spouse	Handlers-cleaners	Male	40	<=50K
28	Private	Bachelors	Married-civ-spouse	Prof-specialty	Female	40	<=50K
37	Private	Masters	Married-civ-spouse	Exec-managerial	Female	40	<=50K
49	Private	9th	Married-spouse-absent	Other-service	Female	16	<=50K
52	Self-emp-not-inc	HS-grad	Married-civ-spouse	Exec-managerial	Male	45	>50K
31	Private	Masters	Never-married	Prof-specialty	Female	50	>50K
42	Private	Bachelors	Married-civ-spouse	Exec-managerial	Male	40	>50K
37	Private	Some-college	Married-civ-spouse	Exec-managerial	Male	80	>50K
30	State-gov	Bachelors	Married-civ-spouse	Prof-specialty	Male	40	>50K
23	Private	Bachelors	Never-married	Adm-clerical	Female	30	<=50K
32	Private	Assoc-acdm	Never-married	Sales	Male	50	<=50K
40	Private	Assoc-voc	Married-civ-spouse	Craft-repair	Male	40	>50K
34	Private	7th-8th	Married-civ-spouse	Transport-moving	Male	45	<=50K
25	Self-emp-not-inc	HS-grad	Never-married	Farming-fishing	Male	35	<=50K
32	Private	HS-grad	Never-married	Machine-op-inspct	Male	40	<=50K
38	Private	11th	Married-civ-spouse	Sales	Male	50	<=50K

Variable: Age, workclass, education, marital_status, ….
Instance/Observation

Vector

- John, Anna, and Emma obtained their marks on four modules as shown below

	Business statistics	Simulation	Machine Learning	Big Data
John	65	60	70	80
Anna	72	65	55	65
Emma	56	64	67	53

- Let
  - $\mathrm{V_J} = (65 \quad 60 \quad 70 \quad 80)$,
  - $\mathrm{V_A} = (72 \quad 65 \quad 55 \quad 65)$, and
  - $\mathrm{V_E} = (56 \quad 64 \quad 67 \quad 53)$,

  Then V_J is called a vector, so is V_A (or V_E)

Matrix

	Business statistics	Simulation	Machine Learning	Big Data
John	65	60	70	80
Anna	72	65	55	65
Emma	56	64	67	53

$$M=\begin{pmatrix} 65 & 60 & 70 & 80 \\ 72 & 65 & 55 & 65 \\ 56 & 64 & 67 & 53 \end{pmatrix}$$

1st row
2nd row
3rd row

1st column
2nd column
3rd column
4th column

- M is called a matrix with 3 rows and 4 columns
- The order of a matrix is rows × columns
- Terms: entry/element, dimension, *n* by *m*, row, column

Questions

- $M_1 = \begin{pmatrix} 65 & 60 & 70 & 80 \\ 72 & 65 & 55 & 65 \end{pmatrix}$ is a matrix with __ rows and __ columns
- $M_2 = \begin{pmatrix} 65 \\ 72 \\ 56 \end{pmatrix}$ is a matrix with __ rows and __ columns
- $M_3 = (65 \quad 60 \quad 70 \quad 80)$ is a matrix with __ rows and __ columns

- M_2 and M_3 are also vectors

Matrix

- Let

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \ddots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}$$

then A is an n by m matrix, or an $n \times m$ matrix.

- If

$$B = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \cdots & \cdots & \ddots & \cdots \\ a_{1m} & a_{2m} & \cdots & a_{nm} \end{pmatrix}$$

Then B is an $m \times n$ matrix, or the transpose of the above matrix A, or $B = A^T$, of course, $A = B^T$

Simply put: put the k-th row of A to the k-th column of B

Matrix

- Denote

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \ddots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

where $n = m$, then A is a square matrix of order n

- Let

$$I_n = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

Then I_n is an $n \times n$ identity matrix. That is, the identity matrix I_n of size n is the $n \times n$ matrix in which all the elements on the main diagonal are equal to 1 and all other elements are equal to 0,

- Example
- Let $A = \begin{pmatrix} 65 & 60 & 70 & 80 \\ 72 & 65 & 55 & 65 \end{pmatrix}$ and $B = \begin{pmatrix} 65 & 72 \\ 60 & 65 \\ 70 & 55 \\ 80 & 65 \end{pmatrix}$, then $A = B^T$, or $B = A^T$
- $M_6 = \begin{pmatrix} 65 & 60 \\ 72 & 65 \end{pmatrix}$ is a square matrix of order 2
- $I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ is called a 3 x 3 identity matrix, or an identity matrix of order 3

Matrix addition/subtraction/multiplication/division

- If **A** is an *n* × *m* matrix and **B** is an *n* × *m* matrix,

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \ddots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}, \qquad B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \cdots & \cdots & \ddots & \cdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{pmatrix}$$

then ***matrix addition*** **C** = **A+B**: $C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1m} \\ c_{21} & c_{22} & \cdots & c_{2m} \\ \cdots & \cdots & \ddots & \cdots \\ c_{n1} & c_{n2} & \cdots & c_{nm} \end{pmatrix}$ is an *n* × m matrix, where $c_{ij} = a_{ij} + b_{ij}$

- Example

$$\begin{pmatrix} 1 & 2 & 0 \\ 0 & 7 & 6 \\ 4 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 5 & 2 & 1.2 \\ 2 & 0 & 9 \\ 2 & 5 & 12 \end{pmatrix} = \begin{pmatrix} 1+5 & 2+2 & 0+1.2 \\ 0+2 & 7+0 & 6+9 \\ 4+2 & 0+5 & 1+12 \end{pmatrix} = \begin{pmatrix} 6 & 4 & 1.2 \\ 2 & 7 & 15 \\ 6 & 5 & 13 \end{pmatrix}$$

- Similarly, subtraction $(A - B)$, multiplication $(A \circ B)$, and division $(A \oslash B)$ can be done, all of them are element-wise operators. That is, each element of A is subtracted/multiplied/divided by the corresponding element of B

Scalar Multiplication

- If **A** is an *n* × *m* matrix,

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \ddots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix},$$

The product $\alpha \boldsymbol{A}$ of a number α (also called a scalar) and a matrix A is computed by multiplying every entry of A by α: **C** = $\alpha \boldsymbol{A}$: $C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1m} \\ c_{21} & c_{22} & \cdots & c_{2m} \\ \cdots & \cdots & \ddots & \cdots \\ c_{n1} & c_{n2} & \cdots & c_{nm} \end{pmatrix}$ is an *n* × m matrix,

where $c_{ij} = \alpha a_{ij}$

- An example

$$3 \times \begin{pmatrix} 1 & 2 & 0 \\ 0 & 7 & 6 \\ 4 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 6 & 0 \\ 0 & 21 & 18 \\ 12 & 0 & 3 \end{pmatrix}$$

Dot product between two matrices

- If **A** is an $n \times m$ matrix and **B** is an $m \times p$ matrix,

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \ddots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}_{n \times m}, \qquad B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \cdots & \cdots & \ddots & \cdots \\ b_{m1} & b_{m2} & \cdots & b_{mp} \end{pmatrix}_{m \times p}$$

- then *matrix product* $C = A \cdot B$ (denoted $C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \cdots & \cdots & \ddots & \cdots \\ c_{n1} & c_{n2} & \cdots & c_{np} \end{pmatrix}$) is defined to be the $n \times p$ matrix

$$c_{ij} = a_{i1}b_{1j} + \cdots + a_{im}b_{mj} = \sum_{k=1}^{m} a_{ik}b_{kj} = \sum_{k=1}^{m} a_{ik}b_{kj}$$

- **Distinguish $A \cdot B$ from $A \circ B$. $A \cdot B$ is normally denoted by AB** (without multiplication sign or dot sign in-between)
- **Rules:**
  - **c_{ij} is the sum of the elements in the i-th row in A multiplying the elements in the j-th column in B**
  - **The dimensions of the resulting matrix are $n \times p$**

Dot product between two matrices---Example

- Let $A = \begin{pmatrix} 65 & 60 & 70 & 80 \\ 72 & 65 & 55 & 65 \end{pmatrix}$ and $B = \begin{pmatrix} 65 & 72 \\ 60 & 65 \\ 70 & 55 \\ 80 & 65 \end{pmatrix}$, then

$$C = AB = \begin{pmatrix} 65 & 60 & 70 & 80 \\ 72 & 65 & 55 & 65 \end{pmatrix}_{2\times 4} \begin{pmatrix} 65 & 72 \\ 60 & 65 \\ 70 & 55 \\ 80 & 65 \end{pmatrix}_{4\times 2} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$$

$$c_{11} = 65 \times 65 + 60 \times 60 + 70 \times 70 + 80 \times 80 = 19125$$
$$c_{12} = 65 \times 72 + 60 \times 65 + 70 \times 55 + 80 \times 65 = 17630$$
$$c_{21} = 72 \times 65 + 65 \times 60 + 55 \times 70 + 65 \times 80 = 17630$$
$$c_{22} = 72 \times 72 + 65 \times 65 + 55 \times 55 + 65 \times 65 = 16659$$

Properties, Inverse of a Matrix

- Let

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \ddots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

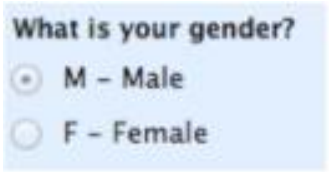
- If

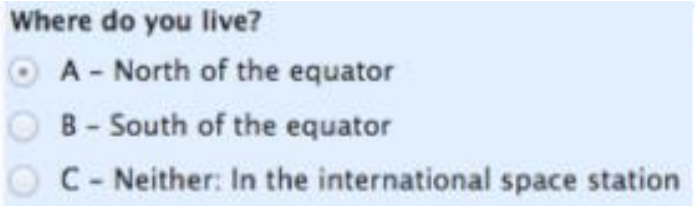
$$AB = BA = I_n$$

Then B is the inverse matrix of A

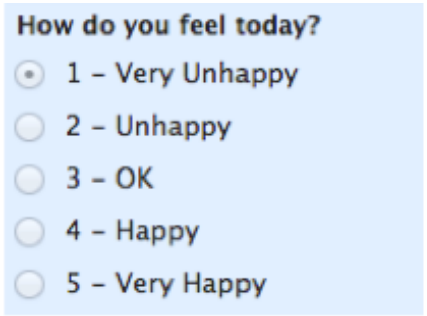
Level of measurement: Nominal & Ordinal

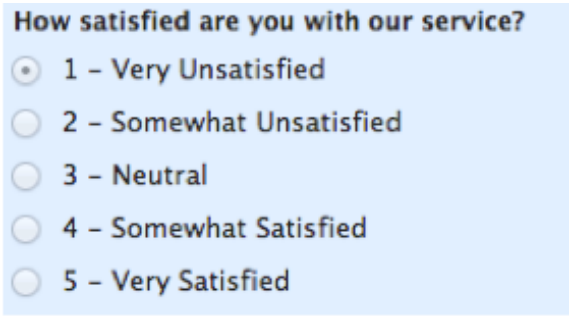
- *Nominal Data:* Numbers are used to represent an item or characteristic. For example, red=1, green=2, ... Note that such data should not be treated as numerical, since relative size has no meaning.





- *Ordinal:* Numbers are used to rank. For example, the size of a cup of coffee can be small, median, and large. The dif*ference* between *ordinal data* and *nominal data* is that ordinal data contain both an equality (=) and a greater-than (>) relationship, whereas the nominal data contain only an equality (=) relationship.





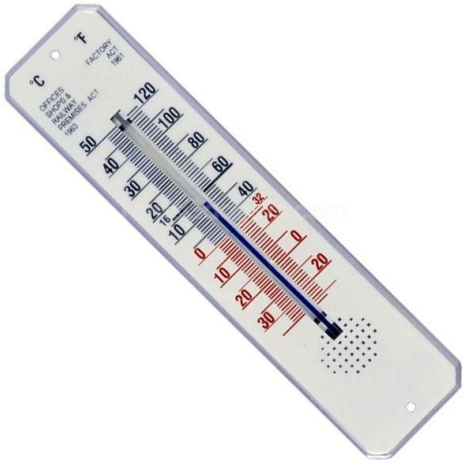
Level of measurement: Interval & Ratio

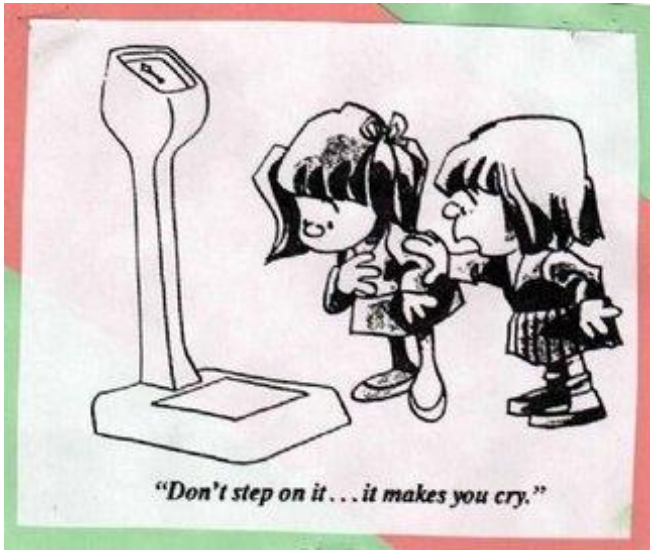
- *Interval Data:* If we have data with ordinal properties (> & =) and can also measure the distance between two data items, we have an interval measurement. For example
  - temperature with the Celsius scale.

Remarks: Ratios between numbers on the scale are not meaningful, so operations such as multiplication and division cannot be carried out directly.

- *Ratio Data:* Is the highest level of measurement and allows for all basic arithmetic operations, including division and multiplication. Data measured on a ratio scale have a fixed or nonarbitrary zero point. For example, cost, revenue and profit.

Remarks: In computer programming, both *nominal data* and *ordinal data* are treated as **string**


°C
°F
FACTORY ACT 1961
OFFICES SHOPS & RAILWAY PREMISES ACT 1961


"Don't step on it . . . it makes you cry."


km/h
MPH
VDO

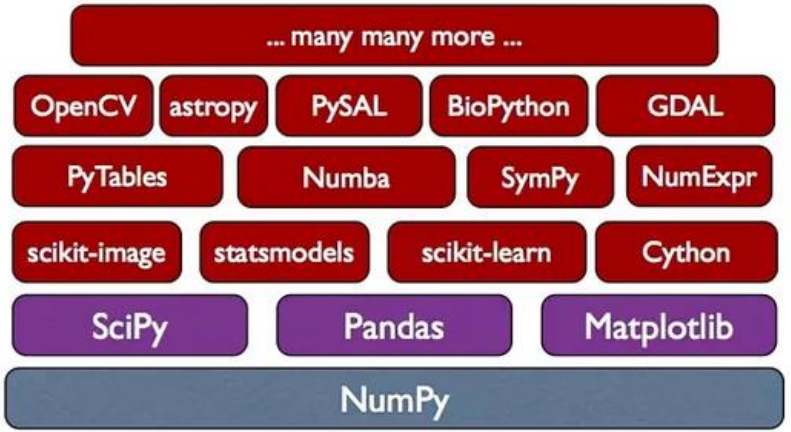
Introduction to Python

Why Python?

- R is a language dedicated to statistics.
- Python is a general-purpose language with statistics modules.
- R has more statistical analysis features than Python, and specialized syntaxes.
- Python can build complex analysis pipelines that mix statistics with e.g. image analysis, text mining, or control of a physical experiment

An introduction to Python: Installation

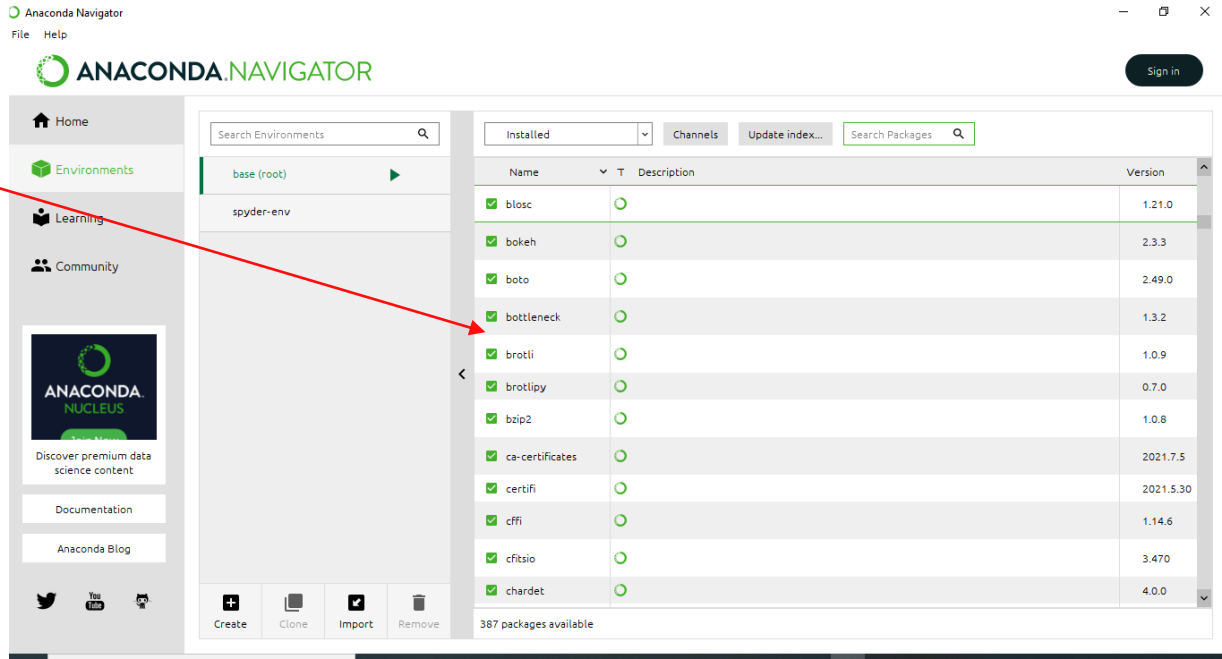
- Python is a programming language for professional data analysis and graphical display
- Python can be downloaded free of charge from https://www.python.org/
- https://www.anaconda.com/: a distribution of the Python and R programming languages
- CPython, Jython, IronPython, PyPy, and Cython
- Python libraries



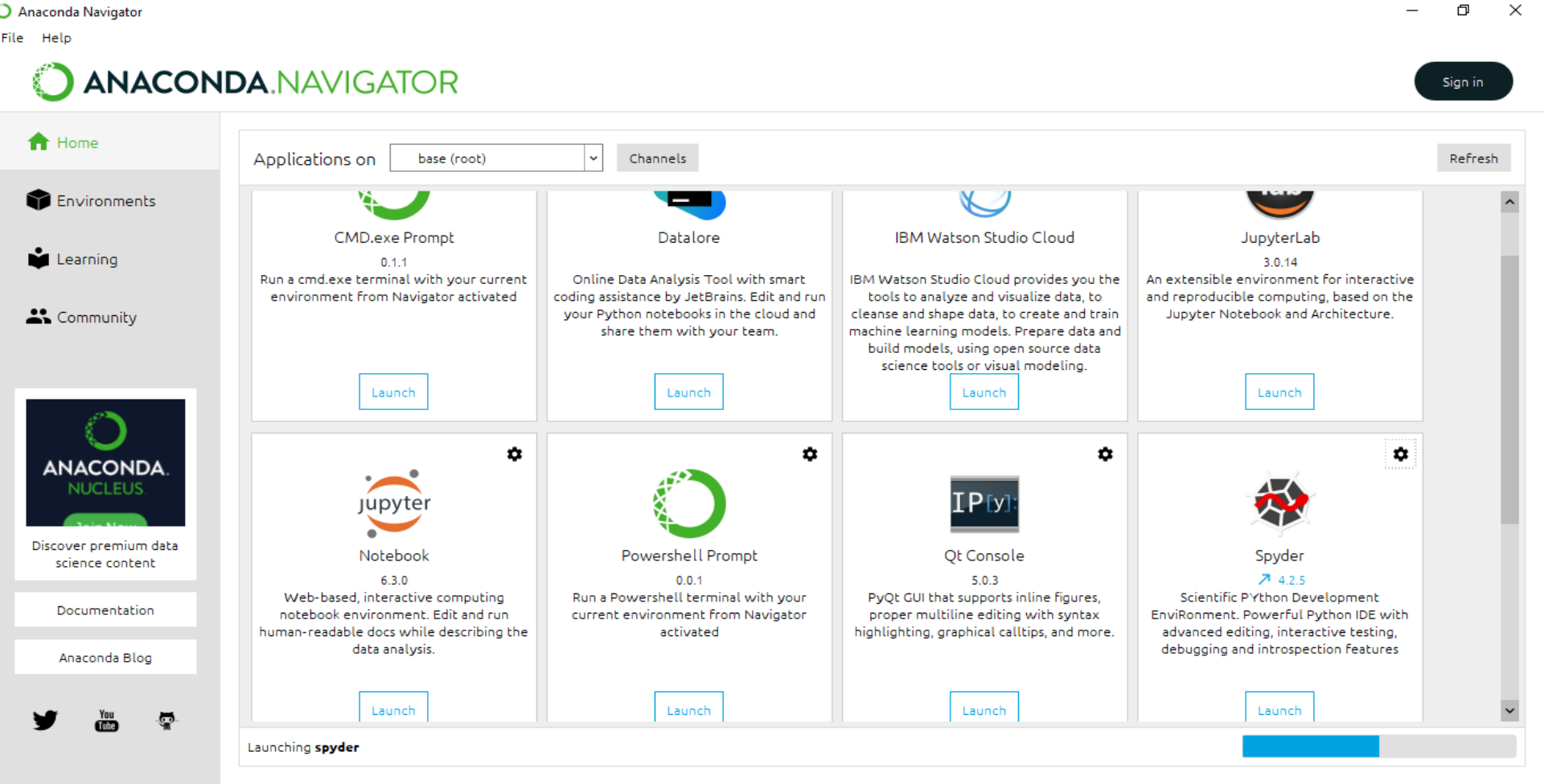
https://medium.com/data-science-everywhere/ml-series-day-6-pandas-for-beginners-part-1-4aacad767d1c

Install Anaconda

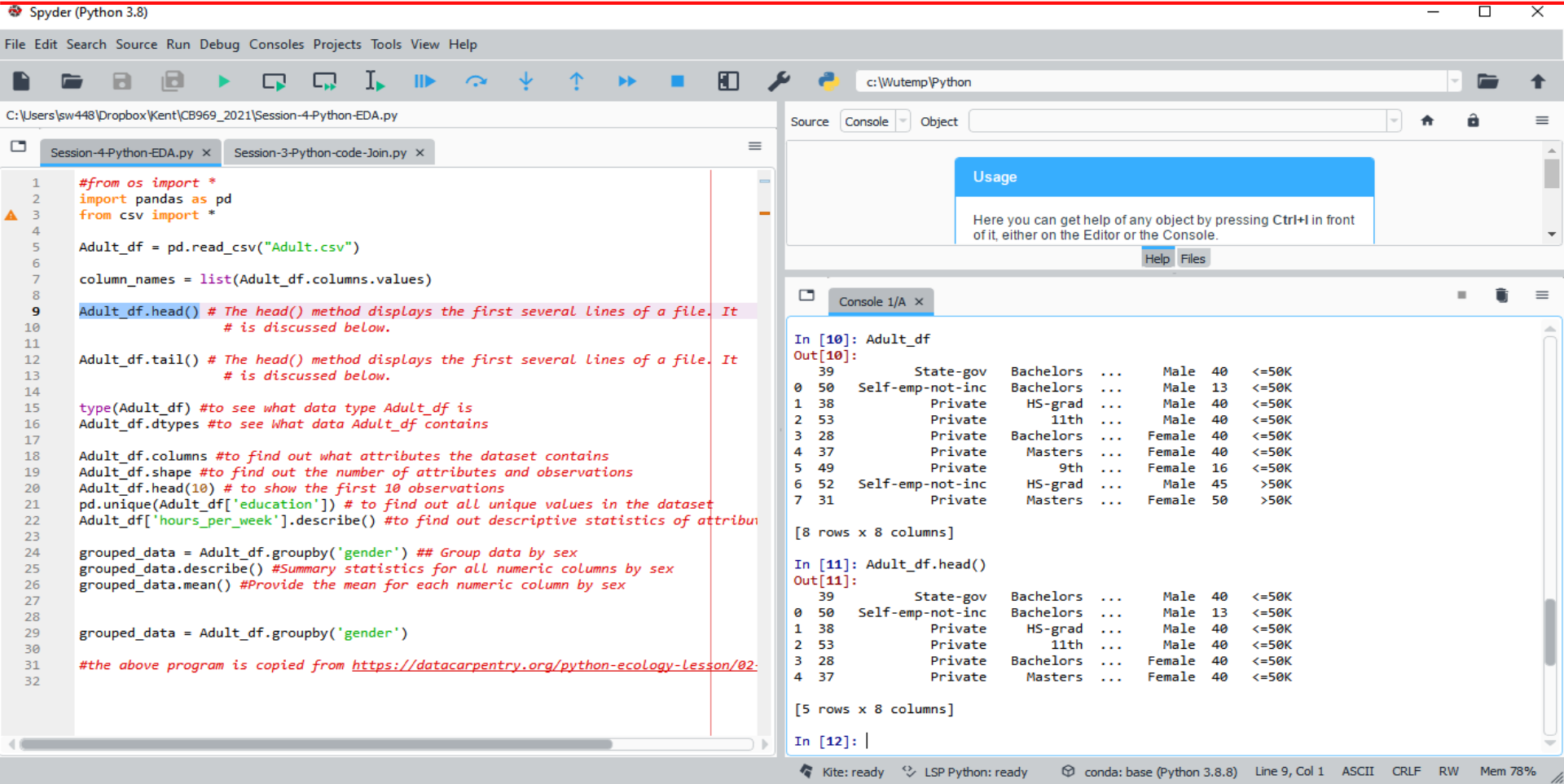
- You are encouraged to
  - install an individual edition of Anaconda;
  - then have a look at Getting started with Anaconda;
- You may check whether a library has been installed in anaconda
  - Start Navigator
  - Click the **Environments** tab
  - See the libraries installed



Anaconda


Anaconda Navigator
File Help
ANACONDA.NAVIGATOR
Sign in
Home
Environments
Learning
Community
Applications on
base (root)
Channels
Refresh
CMD.exe Prompt
0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated
Launch
Datalore
Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.
Launch
IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.
Launch
JupyterLab
3.0.14
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch
ANACONDA. NUCLEUS
Discover premium data science content
Documentation
Anaconda Blog
Notebook
6.3.0
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch
Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated
Launch
Qt Console
5.0.3
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch
Spyder
4.2.5
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch
Launching spyder

Spyder


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
c:\Wutemp\Python
C:\Users\sw448\Dropbox\Kent\CB969_2021\Session-4-Python-EDA.py
Session-4-Python-EDA.py
Session-3-Python-code-Join.py
#from os import *
import pandas as pd
from csv import *
Adult_df = pd.read_csv("Adult.csv")
column_names = list(Adult_df.columns.values)
Adult_df.head() # The head() method displays the first several lines of a file. It
is discussed below.
Adult_df.tail() # The head() method displays the first several lines of a file. It
is discussed below.
type(Adult_df) #to see what data type Adult_df is
Adult_df.dtypes #to see What data Adult_df contains
Adult_df.columns #to find out what attributes the dataset contains
Adult_df.shape #to find out the number of attributes and observations
Adult_df.head(10) # to show the first 10 observations
pd.unique(Adult_df['education']) # to find out all unique values in the dataset
Adult_df['hours_per_week'].describe() #to find out descriptive statistics of attribu
grouped_data = Adult_df.groupby('gender') ## Group data by sex
grouped_data.describe() #Summary statistics for all numeric columns by sex
grouped_data.mean() #Provide the mean for each numeric column by sex
grouped_data = Adult_df.groupby('gender')
#the above program is copied from https://datacarpentry.org/python-ecology-lesson/02-
Source Console Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help Files
Console 1/A
In [10]: Adult_df
Out[10]:
39 State-gov Bachelors ... Male 40 <=50K
0 50 Self-emp-not-inc Bachelors ... Male 13 <=50K
1 38 Private HS-grad ... Male 40 <=50K
2 53 Private 11th ... Male 40 <=50K
3 28 Private Bachelors ... Female 40 <=50K
4 37 Private Masters ... Female 40 <=50K
5 49 Private 9th ... Female 16 <=50K
6 52 Self-emp-not-inc HS-grad ... Male 45 >50K
7 31 Private Masters ... Female 50 >50K
[8 rows x 8 columns]
In [11]: Adult_df.head()
Out[11]:
39 State-gov Bachelors ... Male 40 <=50K
0 50 Self-emp-not-inc Bachelors ... Male 13 <=50K
1 38 Private HS-grad ... Male 40 <=50K
2 53 Private 11th ... Male 40 <=50K
3 28 Private Bachelors ... Female 40 <=50K
4 37 Private Masters ... Female 40 <=50K
[5 rows x 8 columns]
In [12]:
Kite: ready
LSP Python: ready
conda: base (Python 3.8.8)
Line 9, Col 1
ASCII
CRLF
RW
Mem 78%

Basic operations

```
> 2+5
> 2   +        5
> 5-2
> 5*2
> 5/2
> 4+3*2
> (4+3)*2
> 2*2*2
> 2**3
```

Built functions

- int(1.12) #keep the integer part of 1.12, resulting in 1
- float(1) #convert the integer to a float number, resulting in 1.0
- pow(4,2) #power(4,2)=4^2
- print("Hello World") #print out "Hello World" in the Prompt window
- help("numpy") #asking what numpy is.
- max(3,4,2,5) #what is the maximum value among the four numbers
- min(3,4,2,5) #what is the minimum value among the four numbers
- help #enter the help environment
- q #quit from the help environment
- https://docs.python.org/3/library/functions.html for details
- https://www.w3schools.com/python/default.asp : a good website on Python Tutorial