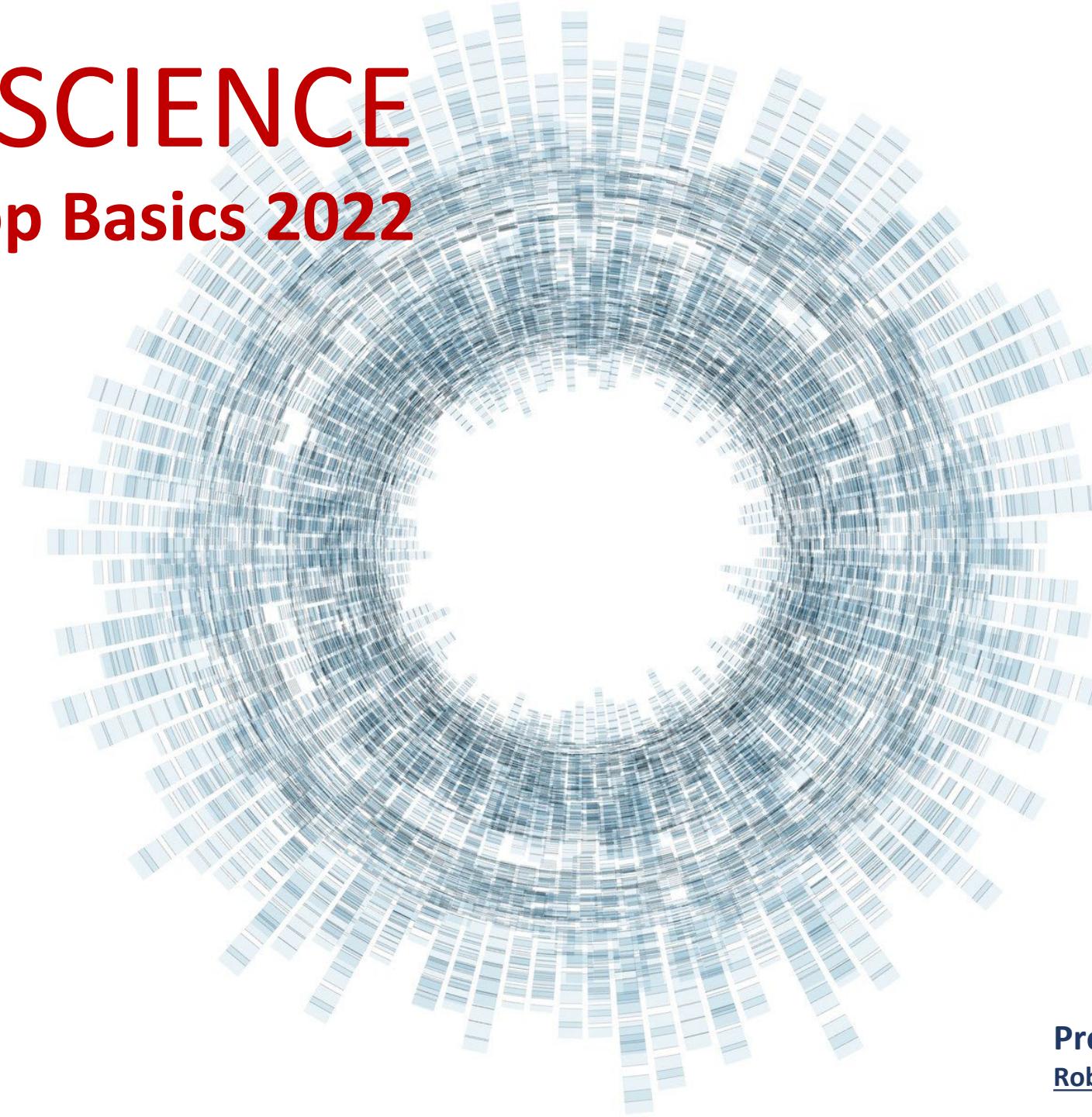


# DATA SCIENCE

## Workshop Basics 2022



# 01

Data Science

# 02

Data [Types]

# 03

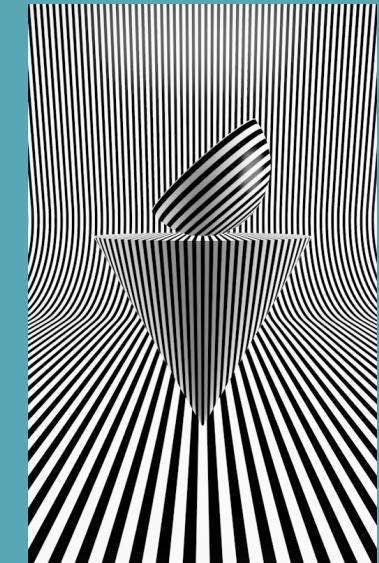
Tools

# 04

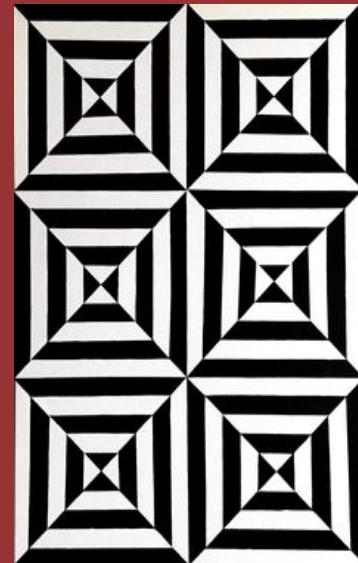
Do it Yourself

a JupyterHub  
Narrative  
for  
Data Science

# Fundamentals



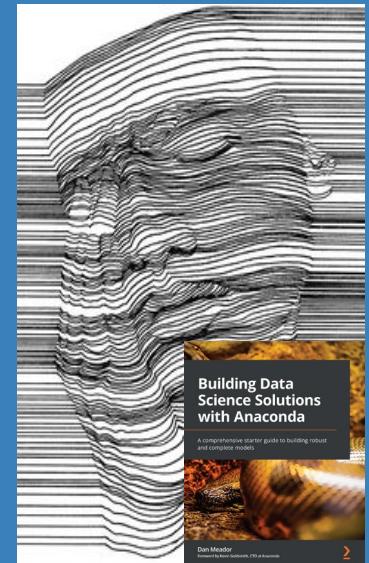
# Grammar of Data



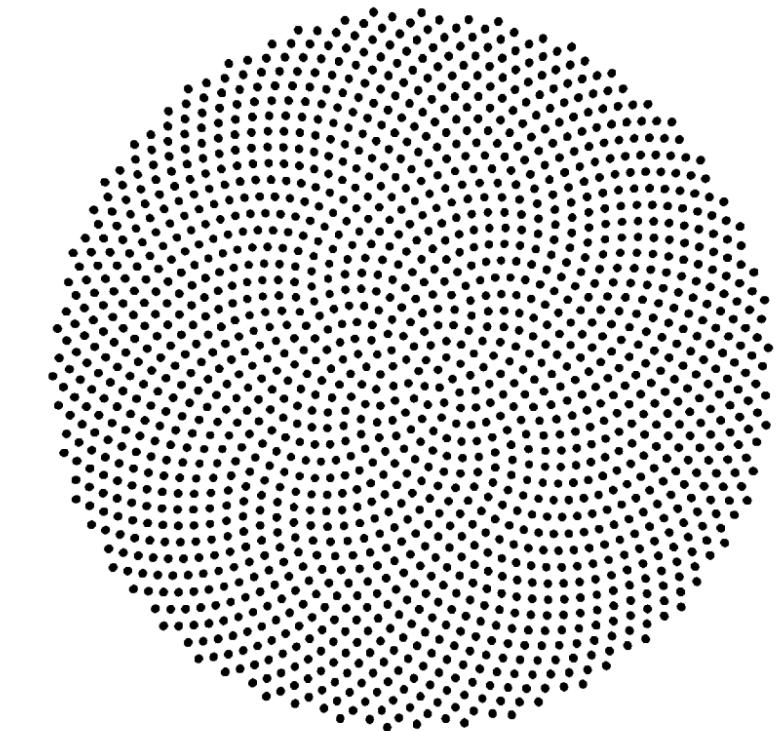
# Transforming data



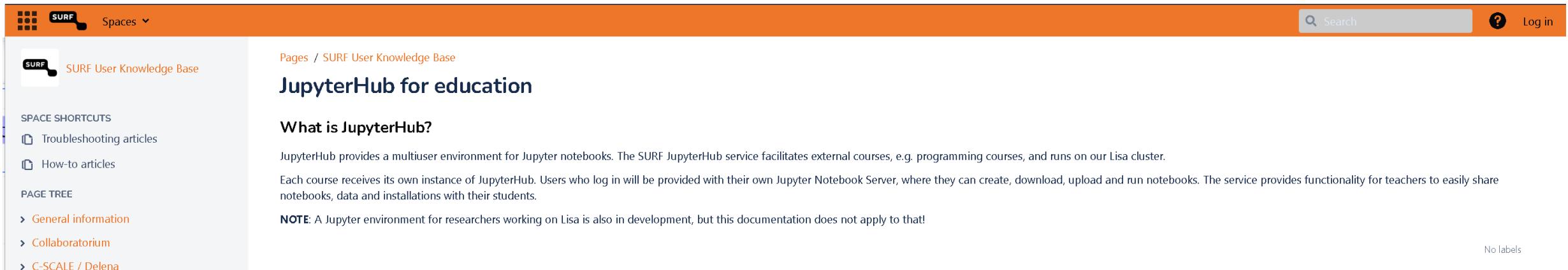
# Applying



# {JupyterHub login How to}



# {SURFsara: JupyterHub for Education}



The screenshot shows a web interface for the SURF User Knowledge Base. The top navigation bar is orange, featuring the SURF logo, a 'Spaces' dropdown, a search bar with a magnifying glass icon, and a 'Log in' button. The main content area has a white background. On the left, there's a sidebar with 'SPACE SHORTCUTS' (Troubleshooting articles, How-to articles) and a 'PAGE TREE' section with links to General information, Collaboratorium, and C-SCALE / Delena. The main content starts with a breadcrumb 'Pages / SURF User Knowledge Base' and the title 'JupyterHub for education'. Below the title is a section titled 'What is JupyterHub?' with a description: 'JupyterHub provides a multiuser environment for Jupyter notebooks. The SURF JupyterHub service facilitates external courses, e.g. programming courses, and runs on our Lisa cluster.' It also mentions that each course receives its own instance of JupyterHub, allowing users to create, download, upload, and run notebooks. A note at the bottom states: 'NOTE: A Jupyter environment for researchers working on Lisa is also in development, but this documentation does not apply to that!'. In the bottom right corner of the main content area, it says 'No labels'.

<https://servicedesk.surf.nl/wiki/display/WIKI/JupyterHub+for+education>

# {Login JupyterHub}



<https://jupyter.lisa.surfsara.nl/jhlhr003>

Minor: Data Driven Solutions

<https://jupyter.lisa.surfsara.nl/jhlhr004>

Minor: Digital Humans

Sign in

Username:

Password:

...

# {Start JupyterHub}

## Server Options

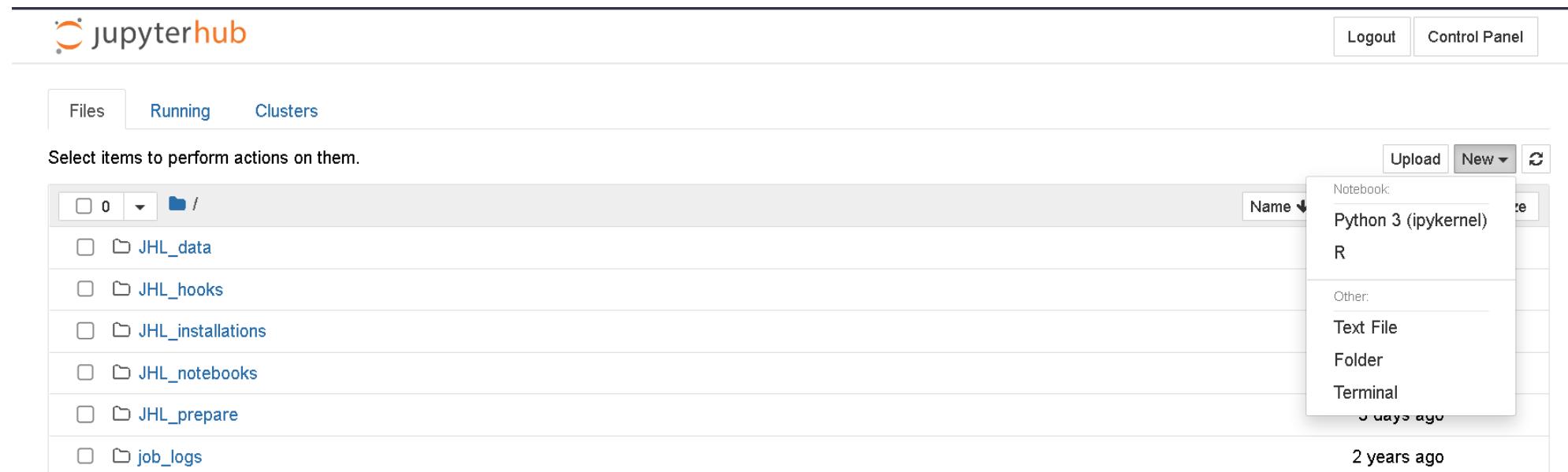
Select a job profile:

SURF jupyterhub jhlhr003 - outside course work

Start

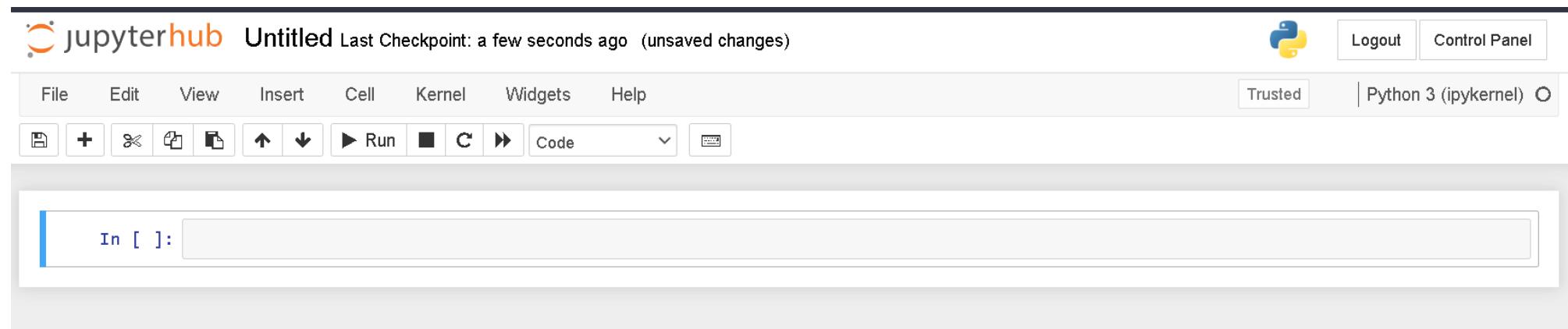
# {Create Jupyter Notebook}

→ Python3 (ipykernel)



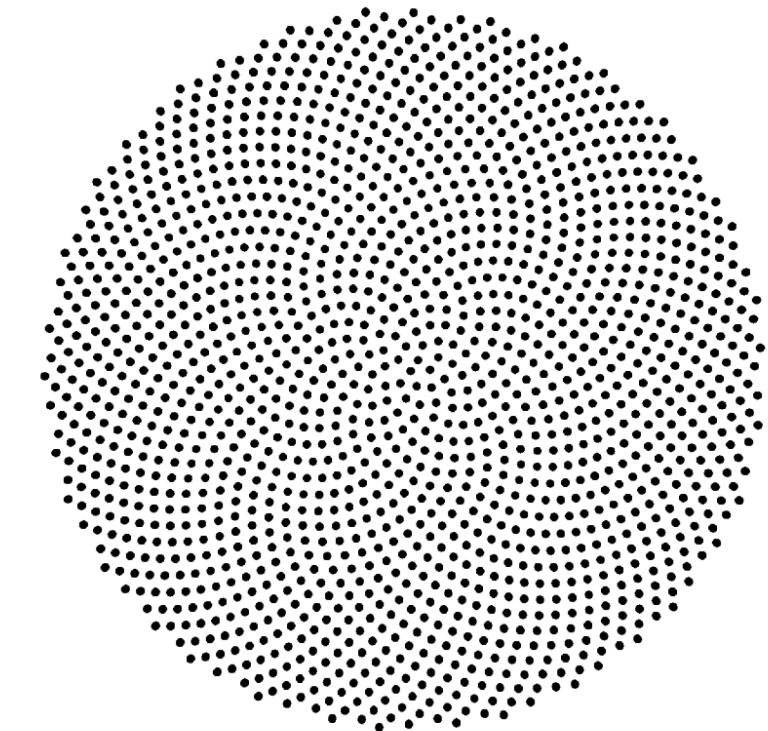
The screenshot shows the JupyterHub interface. At the top, there is a navigation bar with the JupyterHub logo, Logout, and Control Panel buttons. Below the navigation bar, there are three tabs: Files (selected), Running, and Clusters. A message "Select items to perform actions on them." is displayed above a file list. The file list includes a root directory and several sub-directories: JHL\_data, JHL\_hooks, JHL\_installations, JHL\_notebooks, JHL\_prepare, and job\_logs. On the right side, there is a sidebar with a "New" button and a dropdown menu. The dropdown menu shows options: Notebook (Python 3 (ipykernel) is selected), R, Other, Text File, Folder, and Terminal. It also displays a timestamp "3 days ago". At the bottom right of the sidebar, it says "2 years ago".

# {Create Jupyter Notebook}



# {DATA Science}

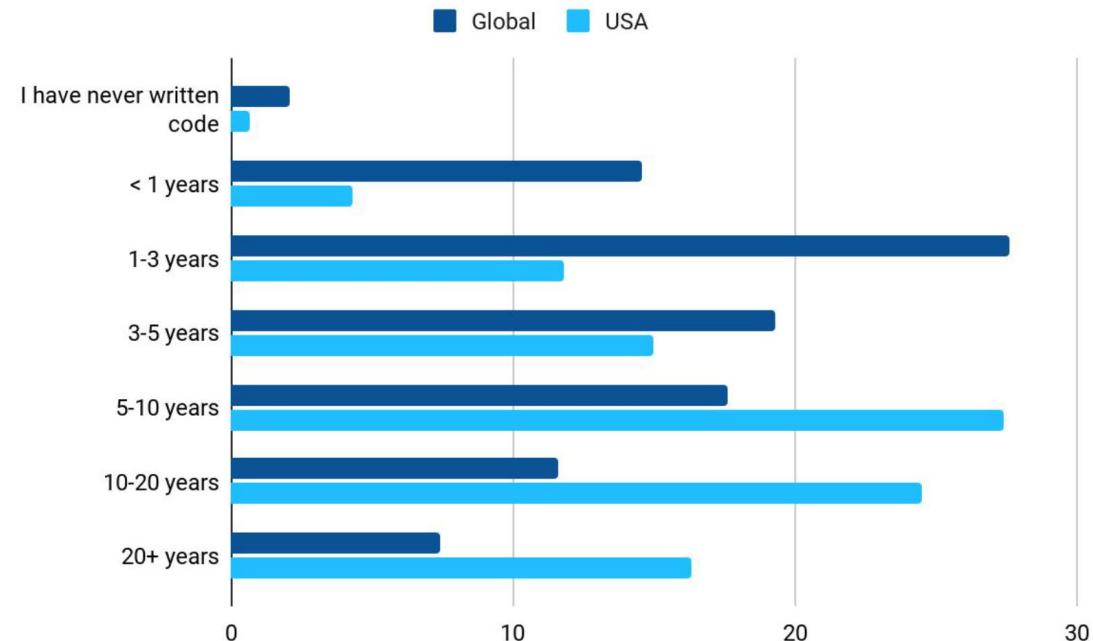
What is needed for  
**[DATA Science]**?



# Programming Experience

While most Kaggle data scientists have at least a few years of experience under their belt, a growing share have taken up programming within the last year (14.6% vs 9% in 2020).

**Programming Experience for Data Scientists  
(Global vs USA)**

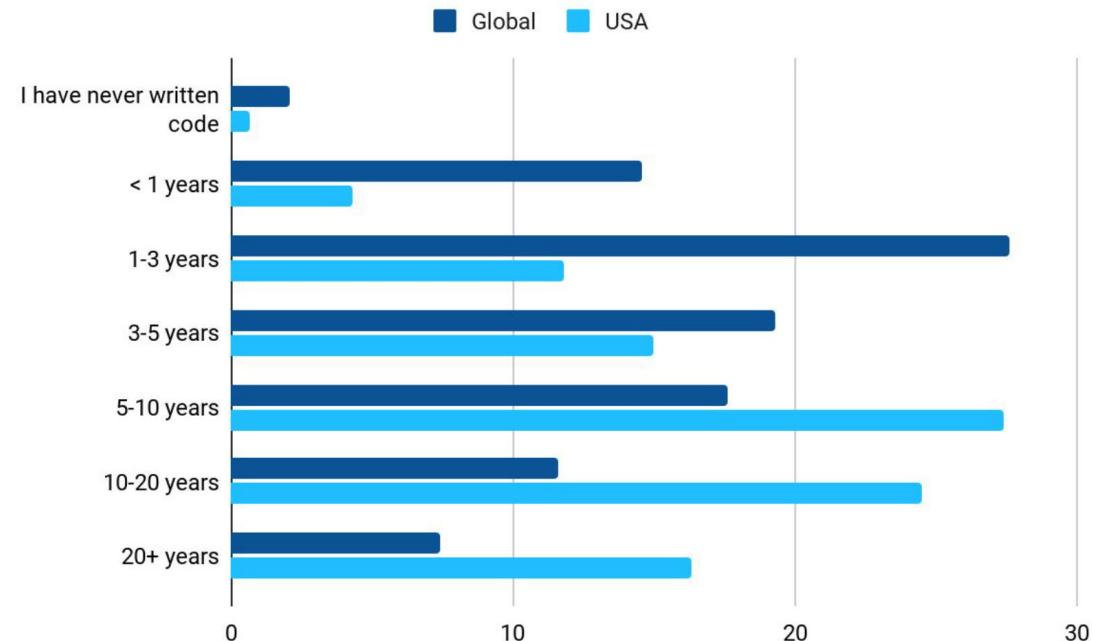


<https://www.kaggle.com/kaggle-survey-2021>

# Programming Experience

While most Kaggle data scientists have at least a few years of experience under their belt, a growing share have taken up programming within the last year (14.6% vs 9% in 2020).

**Programming Experience for Data Scientists  
(Global vs USA)**

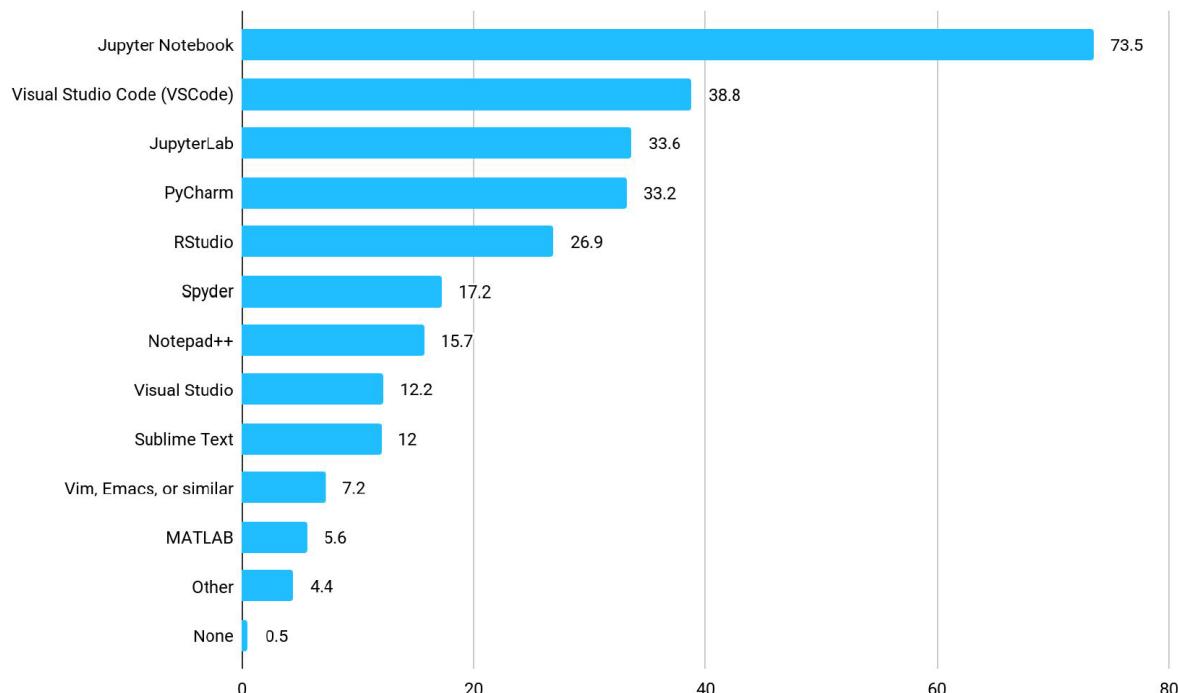


<https://www.kaggle.com/kaggle-survey-2021>

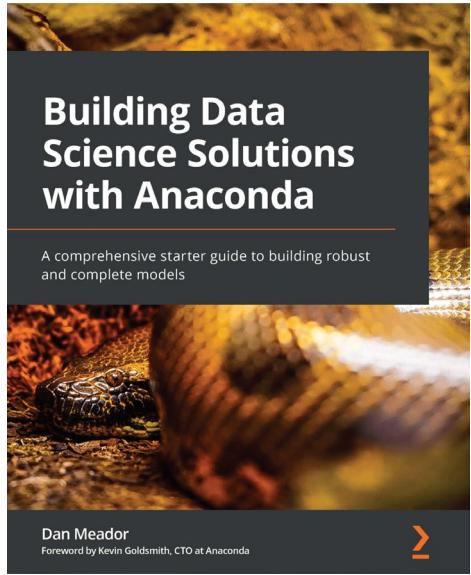
# Interactive Development Environments

Jupyter-based IDEs continue to be the go-to tool for data scientists, with around three-quarters of Kaggle data scientists using it. However, Visual Studio Code is in the second spot with 38%.

IDE Popularity



<https://www.kaggle.com/kaggle-survey-2021>



Dan Meador  
Foreword by Kevin Goldsmith, CTO at Anaconda



**ANACONDA DISTRIBUTION**

Most Trusted Distribution for Data Science

## ANACONDA NAVIGATOR

Desktop Portal to Data Science

## ANACONDA PROJECT

Portable Data Science Encapsulation

## DATA SCIENCE LIBRARIES

### Data Science IDEs



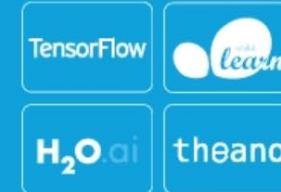
### Analytics & Scientific Computing



### Visualization



### Machine Learning



**CONDA<sup>®</sup>**

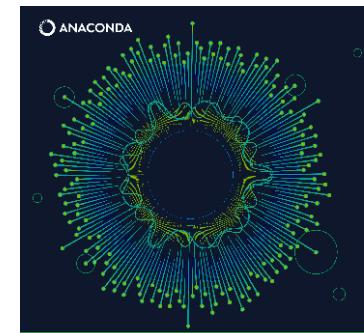
Data Science Package & Environment Manager

# ANACONDA

"Open source" refers to a community-driven model through which large, diverse groups of developers and users collaborate on projects via the Internet. The innovation and stewardship of the OS community has driven advances across all kinds of fields, including data science and machine learning. In a very real sense, open source is the richest pathway and the fastest engine of innovation in data science today. A few commonly used OS tools include SciPy, NumPy, Jupyter, Bokeh, R, and pandas.

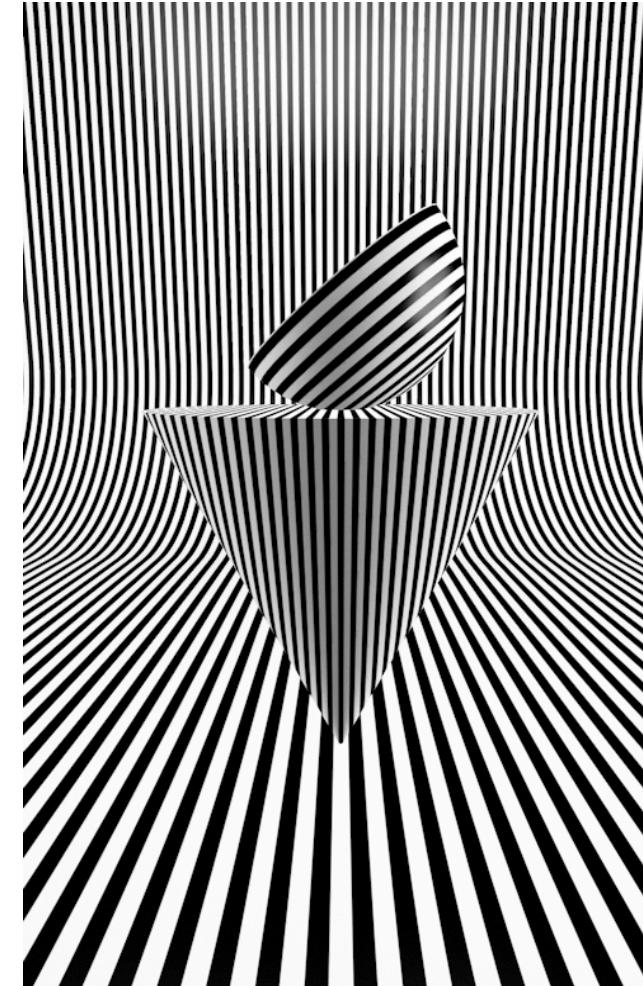


<https://www.anaconda.com/state-of-data-science-2021>



An Enterprise Guide to a  
Secure Data Science  
Pipeline

# {01} Data Science Fundamentals

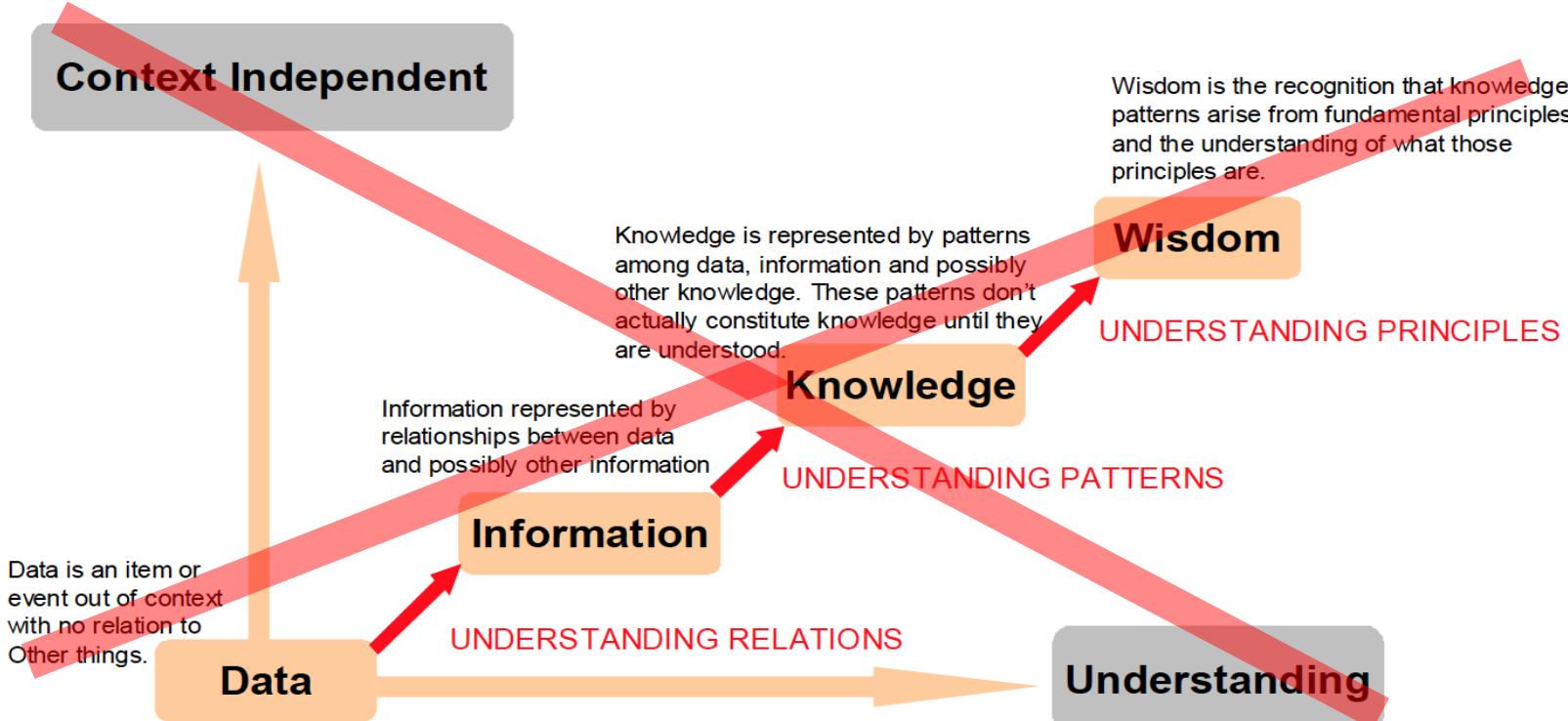


# Data / Informatie / kennis

Volgt niet noodzakelijk een (DIKW) hierarchie

## The Knowledge Pyramid: A Critique of the DIKW Hierarchy

This paper has been accepted for publication in *Journal of Information Science* and the final (edited, revised and typeset) version of this paper will be published in *Journal of Information Science* Vol/Issue/Year? By SAGE Publications Ltd. All rights reserved. © CILIP. For more information please visit: <http://jis.sagepub.co.uk>



## Big Data, the Internet of Things, and the Revised Knowledge Pyramid

Murray E. Jennex  
Fowler College of Business  
San Diego State University

**Abstract**  
The knowledge pyramid has been used for several years to illustrate the hierarchical relationships between data, information, knowledge, and wisdom. An earlier version of this paper presented a revised knowledge pyramid that included the Internet of Things as filtering and sense making, reversed the pyramid by posting there was more knowledge than data, and showed that the Internet of Things was the top level of the pyramid. This paper expands the revised knowledge pyramid to include the Internet of Things and sensors and readings as the top level of the data aspect of the knowledge pyramid. Previous thought was of data as reflections of reality as recorded by sensors and readings. This paper includes sensors and sensors and readings to create two layers of data. The top layer of data is the traditional transaction / operational data. The bottom layer is an expanded set of data reflecting massive data sets and sensors that are near mirrors of reality. The result is a knowledge pyramid that appears as an hourglass.

**Keywords:** Knowledge Pyramid; Big Data; Internet of Things; Analytics; Knowledge Management

### Introduction

Jennex and Bursztaki (2013) revised the knowledge pyramid to incorporate learning, filtering, and transformation processes and technologies and to reflect their view of the hierarchy of knowledge between the knowledge management (KM) knowledge pyramid and the general knowledge pyramid. This paper reflects that the above generalizing knowledge base can identify others, processes, and technologies to accomplish this. This paper also models what is called data, Big Data, analytics, and the Internet of Things (IoT) into a further revised knowledge pyramid.

The importance and contribution of this paper is based on the rising importance of Big Data, analytics, and IoT. Large companies in the top 500 top performing organizations use analytics five times more often than lower performing organizations and double down on data and analytics.

On the other hand, the IoT is a rapidly approaching future with close to 5 billion devices expected to be connected to the Internet and ranging to 20 billion by 2020, promising that the IoT is promising to change the value proposition of the Internet (Barnaghi et al., 2012; Pomerleau, 2015).

### Acknowledgments

This paper is a progressive work with early versions being presented at the Human Interaction Conference, San Diego (HICSS-2009), the SIGPHIL meeting at the Americas Conference for Information Systems (AMCIS) (2011), and published in the *Internet Journal of Knowledge Management* (2013). Comments and discussions from these meetings and with readers of the papers have been used to expand and modify the model that is presented in this paper.

The DATA BASE for Advances in Information Systems

69

Volume 46, Number 4, November 2017

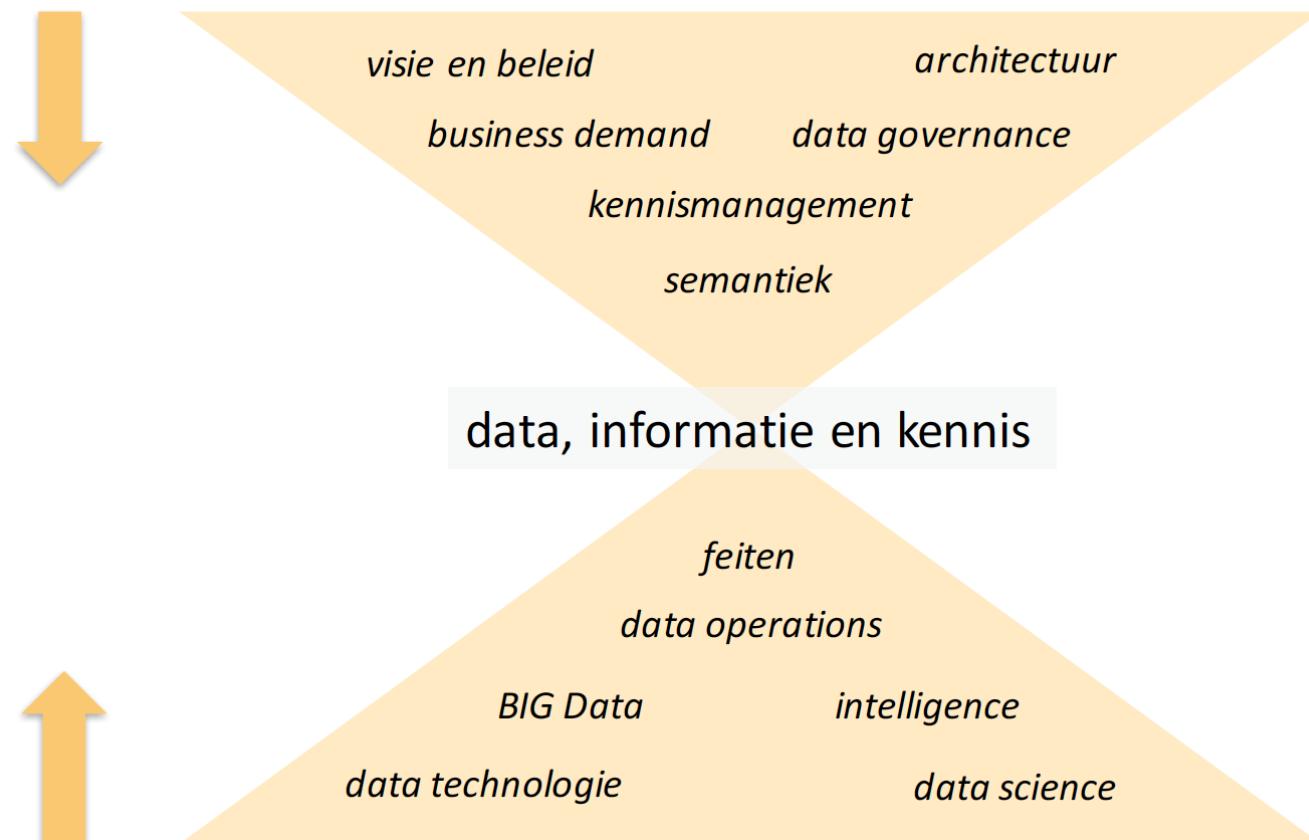
Journal of Information Science, XX (X) 2007, pp. 1-13 © CILIP, DOI: 10.1177/0165551507066001

<https://doi.org/10.1145/3158421.3158427>

<https://doi.org/10.1177/0165551508094050>

# Data / Informatie / kennis

verbindt mensen en vormt zo onze samenleving



## Data Deluge: too much data

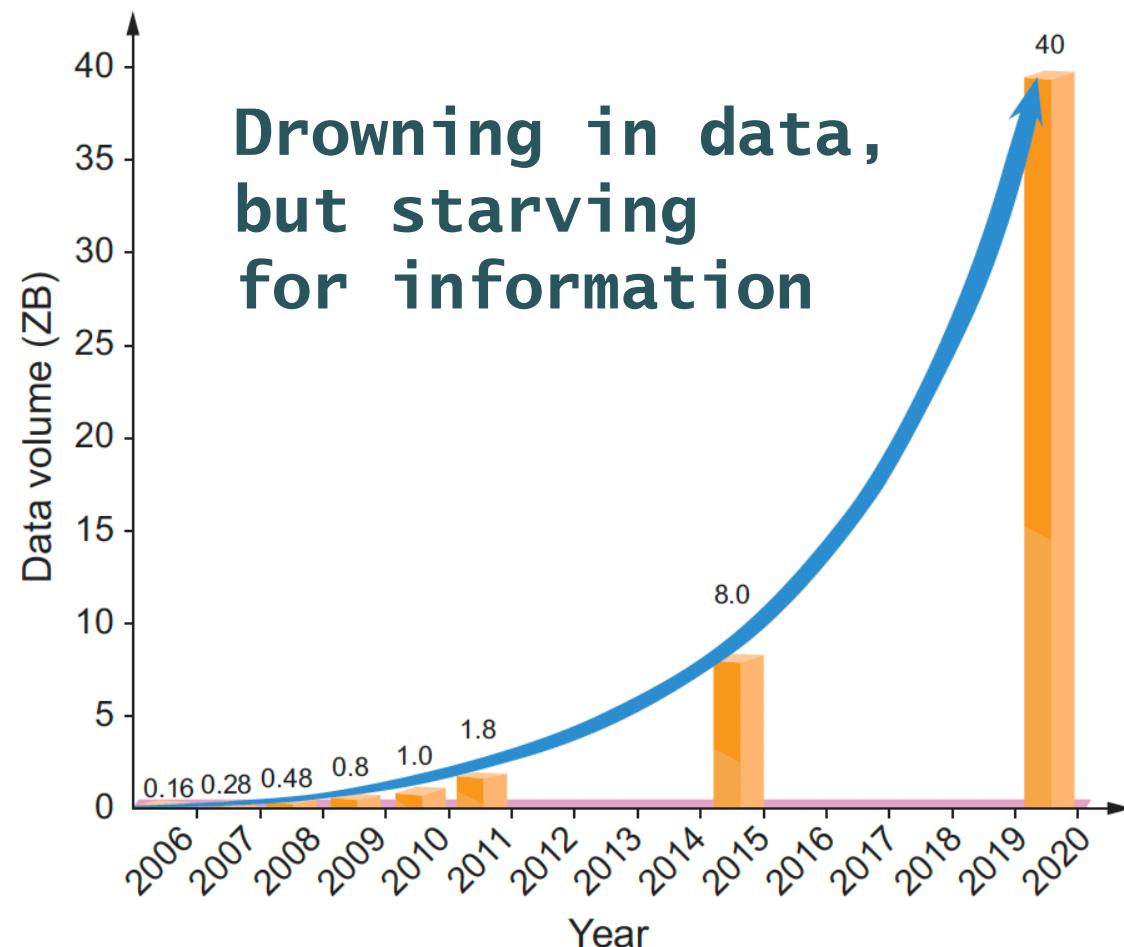
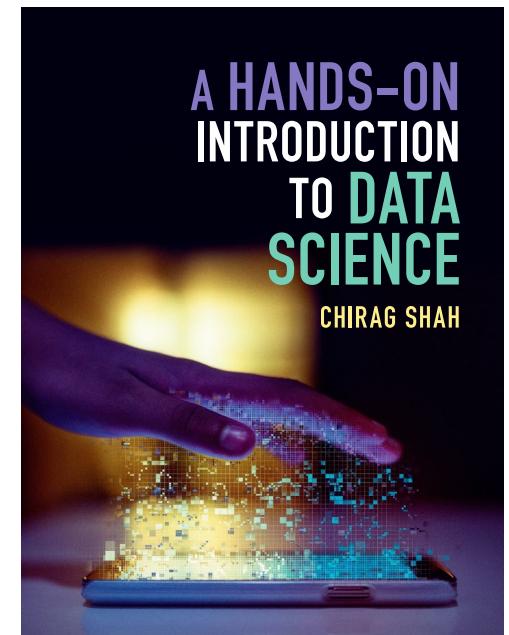
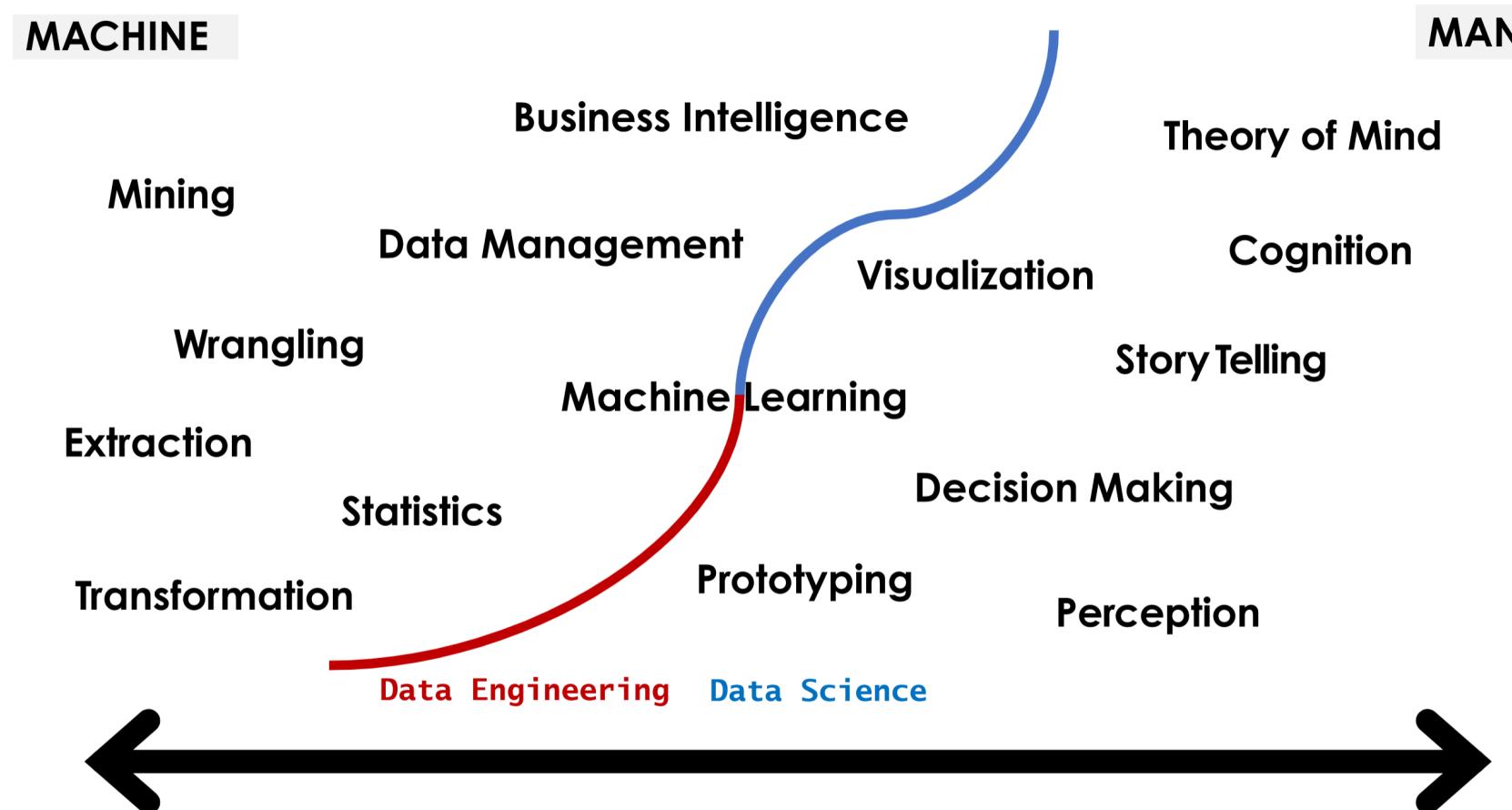


Figure 1.1

Increase of data volume in last 15 years. (Source: IDC's Digital Universe Study, December 2012.<sup>5</sup>)



# {Data Engineering versus Data Science}



Inspired by Daniel Keim, "Visual Analytics: Definition, Process, and Challenges"

# DATA SCIENCE focuses on **Analytics**

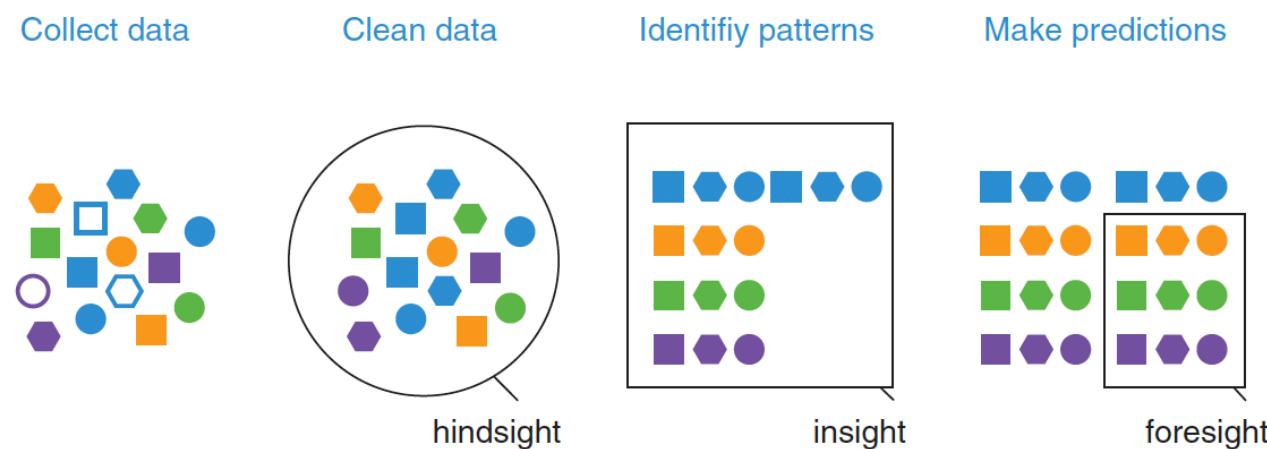
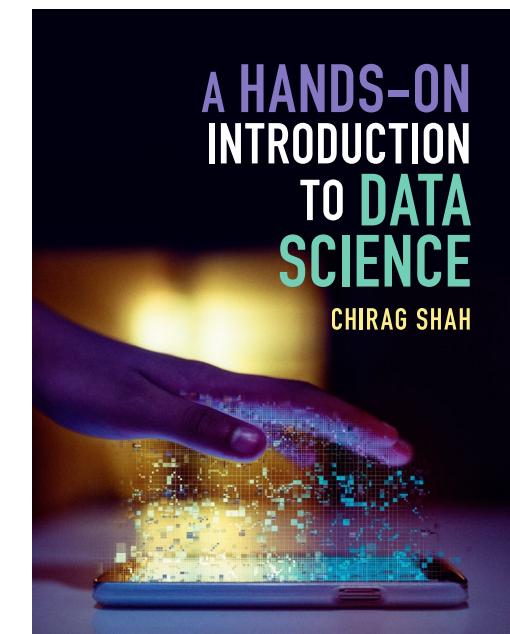


Figure 3.11 Process of predictive analytics.<sup>7</sup>



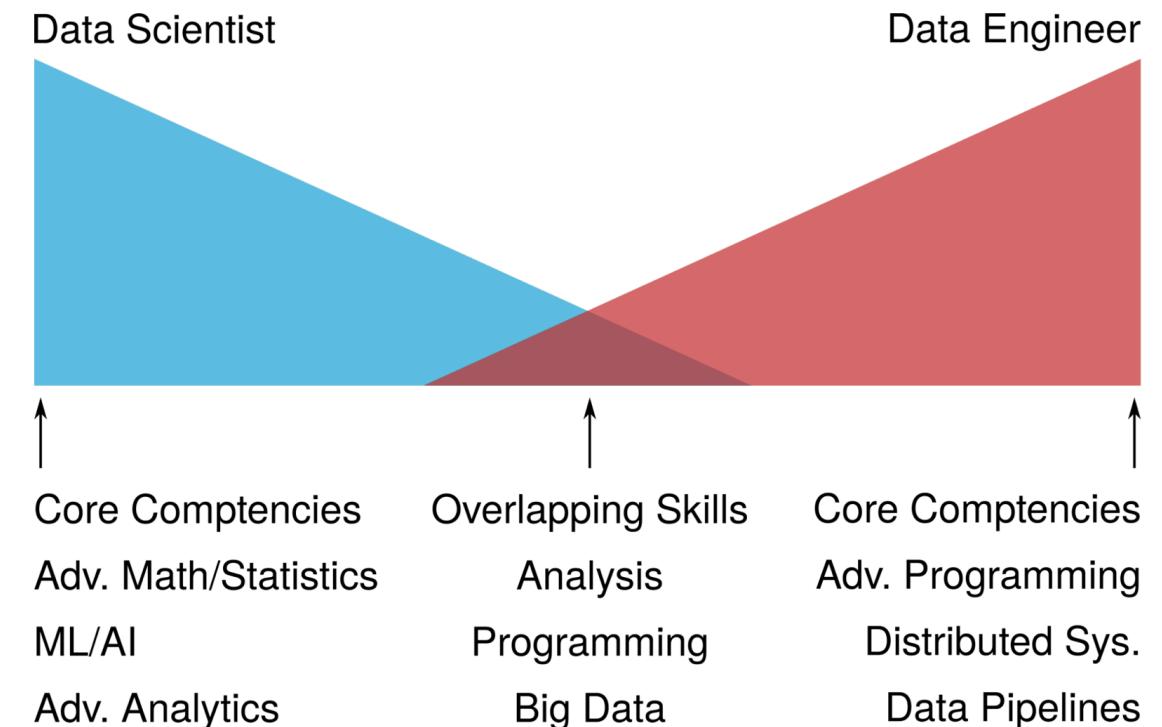
# DATA ENGINEERING

## focuses on **Problem Solving**

Data engineers build and maintain **data pipelines**

**Data pipelines** encompass the journey and processes that data undergoes within a company.

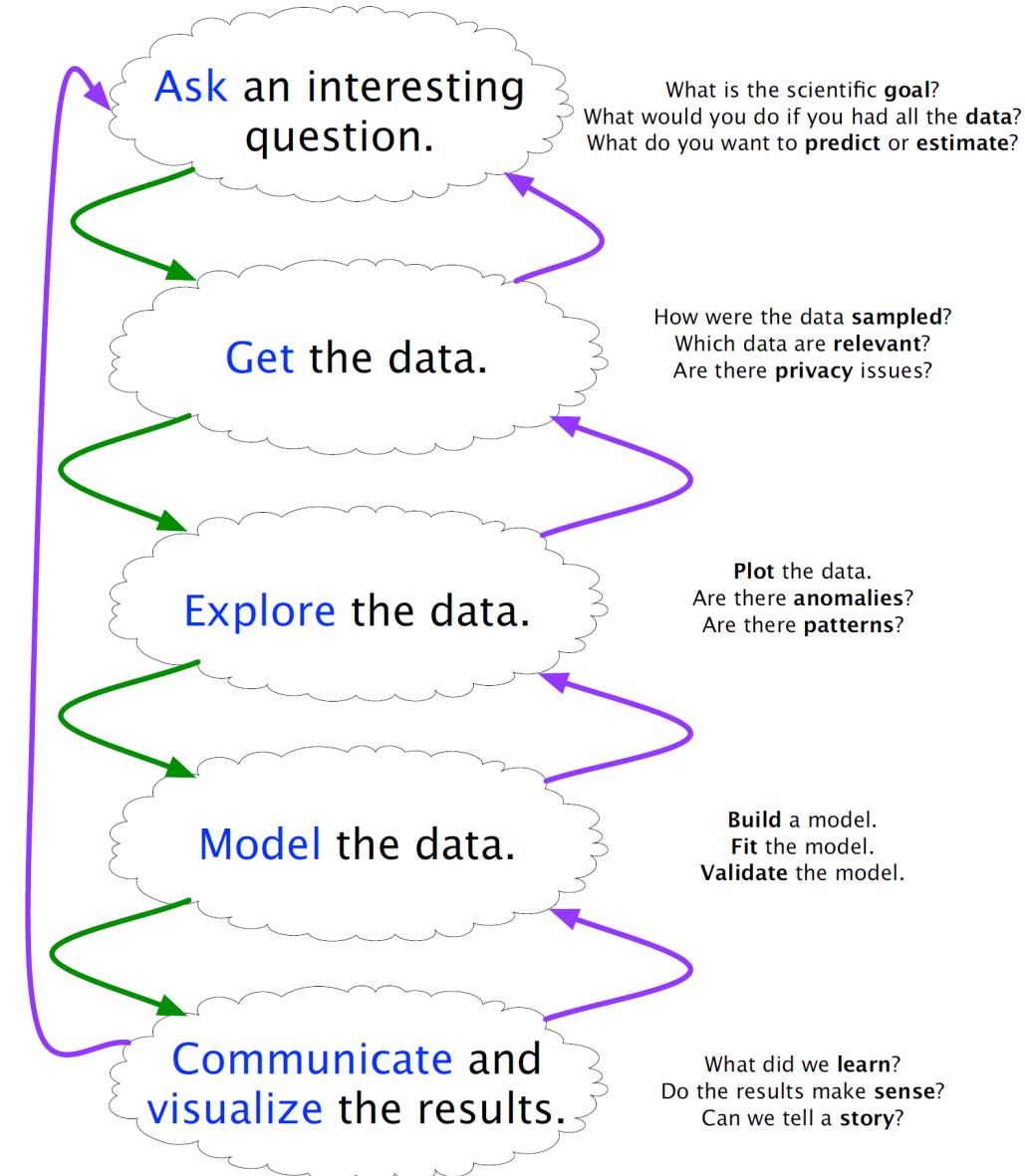
Data engineers are responsible for creating those pipelines.



# DATA SCIENCE (ANALYTICS) VERSUS DATA ENGINEERING (PROBLEM SOLVING)

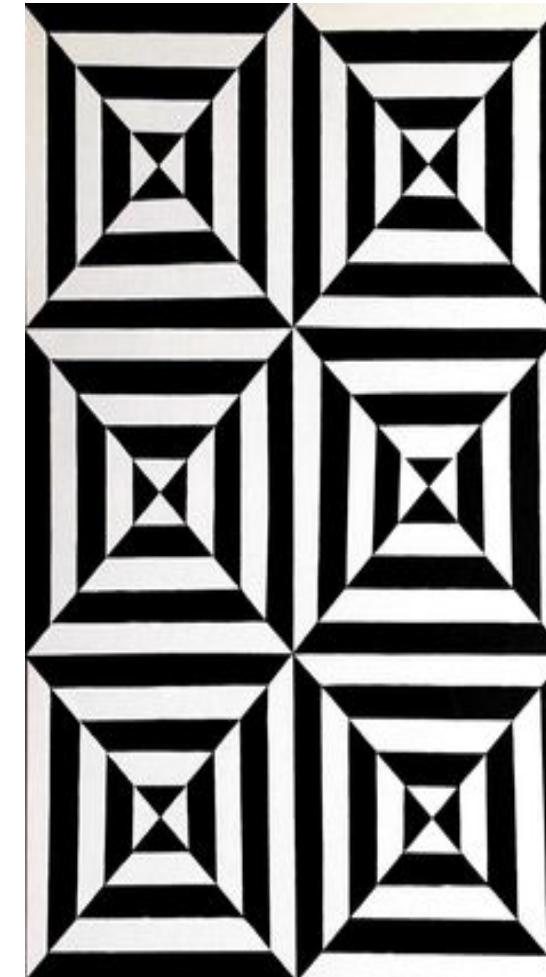
[0] Collection	Big Data (Acquisition/Aggregation) Empirical (Sensor/IoT Measuring/Sampling)	Gathering
[1] Access + Retrieval	Ownership (Open/Closed) Storage (Cloud/Database)	Ingesting
[2] Preparation + Wrangling (Munging)	Loading Feature Extraction/Reduction Normalization Transformation Conversion  Graphical (spatial) Ontological (Language) Semantic (text) Rule-based/Algorithmic Quantitative/Qualitative Numerical/Categorical/Symbolic	Processing
[3] Exploration	Mining (Heuristics/Statistics/Descriptive/Prescriptive) Construct Useful Insights/Trends/Patterns/Diagnosis (Information)	Discovering
[4] Analysis + Machine-Learning	Parameter Selection + Representation Summarization Problem Solving Diagnostic Prediction Encryption  Visualization Virtualization	Conceptualizing
[5] Abstraction	Performance (Measure/Monitor) Evaluation & Review Decision & Advise or Prescription (Interactive/Passive) Story Telling Prototyping	Modelling
[6] Organization + Managing		Presenting
[7] Automation + Reporting		Applying

# [the] Data Science Process



<https://github.com/cs109/2015/blob/master/Lectures/01-Introduction.pdf>

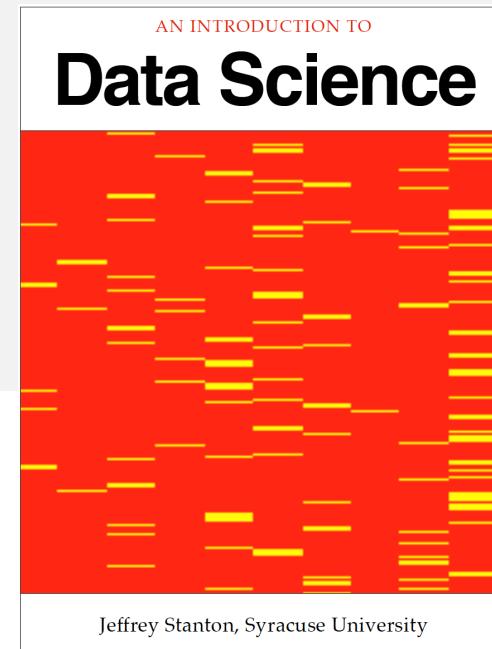
# {02} Data Types



# DATA versus INFORMATIE

## Data [gegevens]

Raw Facts  
No Context  
Numbers  
Symbols

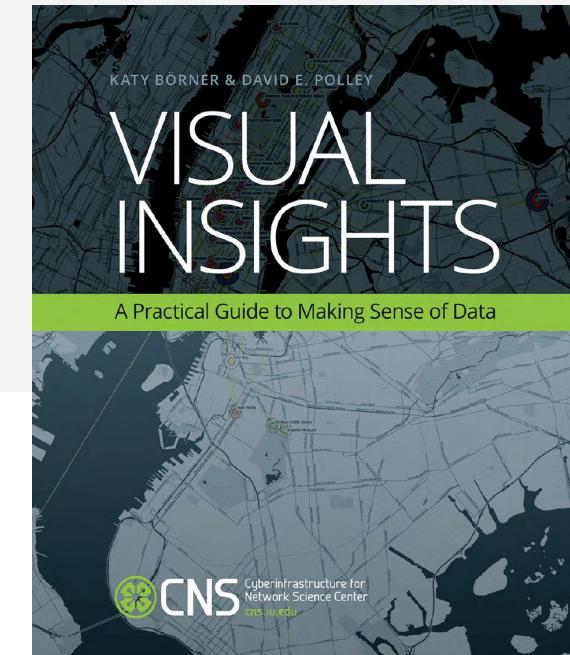


## Information

**Data with structure = processed data = MODEL**

**Value-added to Data**

- Summarised
- Organised
- Analysed



# Structured versus Unstructured

## Data Structuring

### Structured (organized) data:

This is data that can be thought of as observations and characteristics. It is usually organized using a table method (rows and columns).

### Unstructured (unorganized) data:

This data exists as a free entity and does not follow any standard organization hierarchy.

#### Structured Data

High Degree of organization, such as a relational database

Column	Value
Patient	Joe Brown
Date of Birth	02/13/1972
Date Admitted	02/05/2014

#### Unstructured Data

Information that is difficult to organize using traditional mechanisms

“The patient came in complaining of chest pain, shortness of breath, and lingering headaches...smokes 2 packs a day... family history of heart disease...has been experiencing similar symptoms for the past 12 hours....”

# Generalized form of data structure

## Data Table [DATA MATRIX]

A generalized version of the data table is shown.

This table can represent any number of observations described over multiple variables.

This table describes a series of observations (from  $o_1$  to  $o_n$ ) where each observation is described using a series of variables (from  $x_1$  to  $x_p$ ). A value is provided for each variable of each observation.

	Variables					
Observations	$x_1$	$x_2$	$x_3$	...	$x_p$	
$o_1$	$x_{11}$	$x_{12}$	$x_{13}$	...	$x_{1p}$	
$o_2$	$x_{21}$	$x_{22}$	$x_{23}$	...	$x_{2p}$	
$o_3$	$x_{31}$	$x_{32}$	$x_{33}$	...	$x_{3p}$	
...	...	...	...	...	...	...
$o_n$	$x_{n1}$	$x_{n2}$	$x_{n3}$	...	$x_{np}$	

Most data that exists in text form, including server logs and Facebook posts, is unstructured

Scientific observations, as recorded by careful scientists, are kept in a very neat and organized (structured) format: THE DATA TABLE

A genetic sequence of chemical nucleotides [ACGTATTGCA] is unstructured even if the order of the nucleotides matters

# Generalized form of data structure

Observations	Variables					
	$x_1$	$x_2$	$x_3$	...	$x_p$	$x_{1p}$
$o_1$	$x_{11}$	$x_{12}$	$x_{13}$	...	$x_{1p}$	$x_{1p}$
$o_2$	$x_{21}$	$x_{22}$	$x_{23}$	...	$x_{2p}$	$x_{2p}$
$o_3$	$x_{31}$	$x_{32}$	$x_{33}$	...	$x_{3p}$	$x_{3p}$
...	...	...	...	...	...	...
$o_n$	$x_{n1}$	$x_{n2}$	$x_{n3}$	...	$x_{np}$	$x_{np}$

## Data Table [DATA MATRIX]

A generalized version of the data table is shown.

This table can represent any number of **observations** described over multiple **variables**.

This table describes a series of observations (from  $o_1$  to  $o_n$ ) where each observation is described using a series of variables (from  $x_1$  to  $x_p$ ). A value is provided for each variable of each observation.

Patient ID	Treated	Age	Outcome	Random
1	Yes	Young	Positive	0.24
2	No	Young	Positive	0.85
3	Yes	Old	Negative	0.64
4	No	Old	Negative	0.70
5	No	Old	Negative	0.87
6	No	Old	Negative	0.72
7	No	Old	Negative	0.86
8	No	Young	Negative	0.16
9	No	Young	Positive	0.17

# Data [gegevens]

Raw Facts

No Context

Numbers

Symbols

Data comes from the Latin word, "datum," meaning a "thing given."

Although the term "data" has been used since as early as the 1500s, modern usage started in the 1940s and 1950s as practical electronic computers began to input, process, and output data.

98734975471894614398734578

20875980542158009258202908

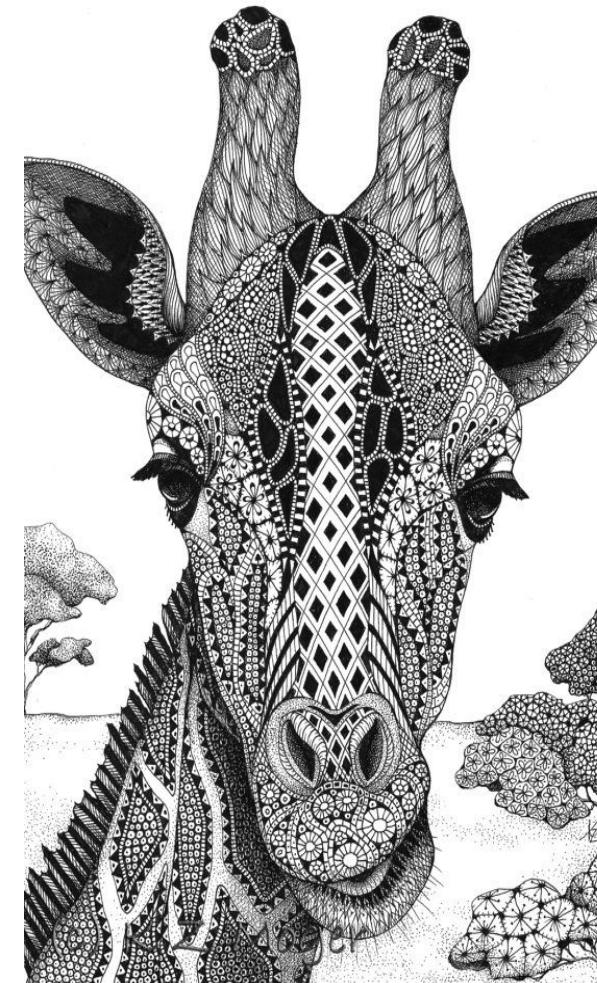
12349823094823048002343423

98734975471894614398734578

20875980542158009258202908

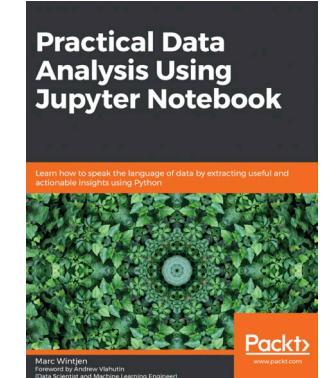
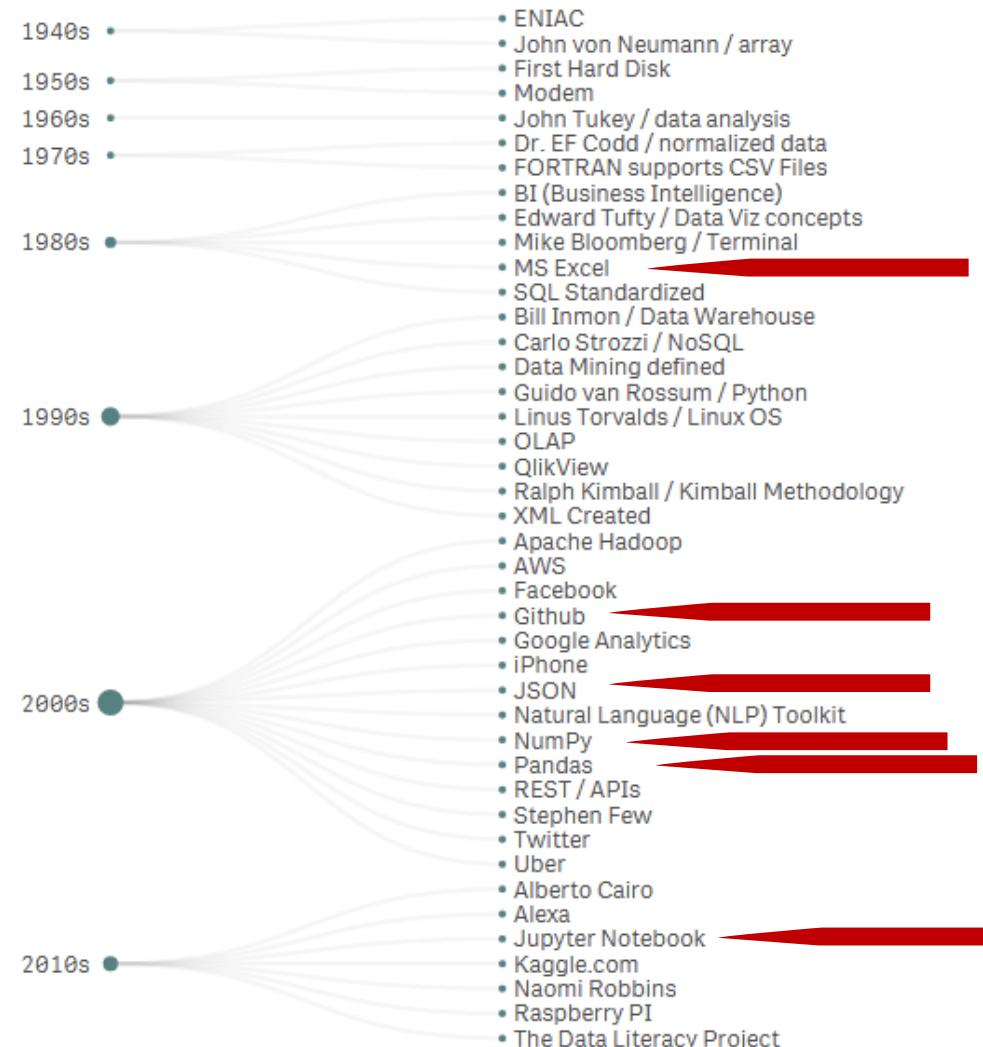
12349823094823048002343423

# {03} Tools



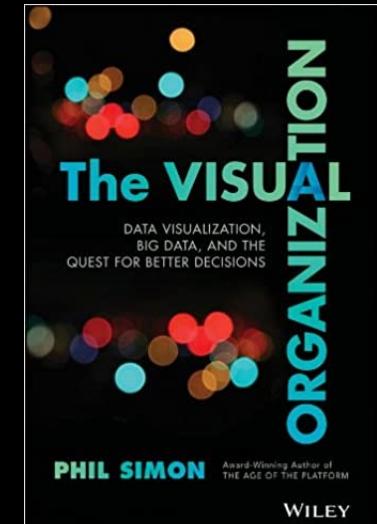
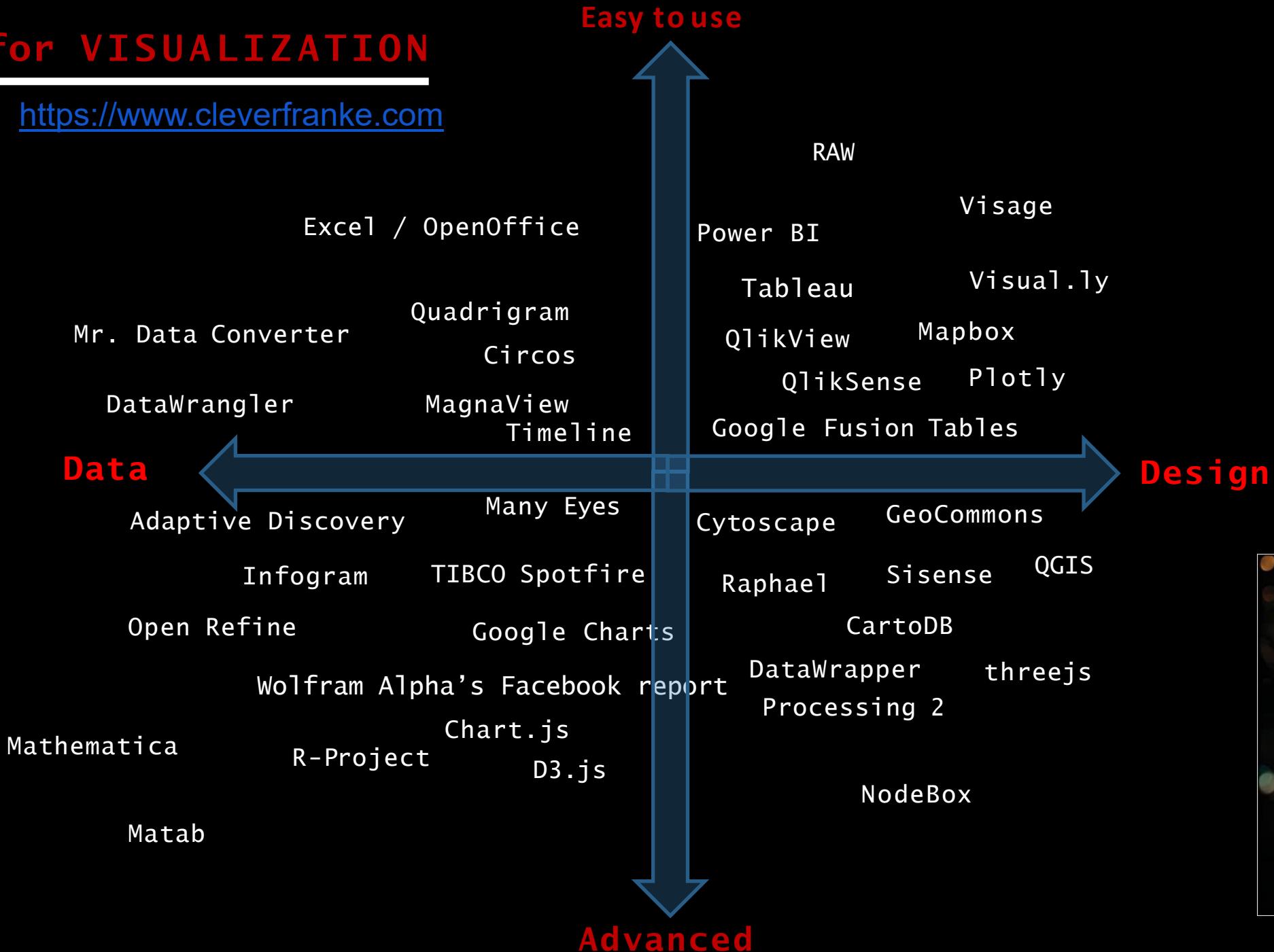
# DATA Science Tools

## A historical context



# TOOLS for VISUALIZATION

<https://www.cleverfranke.com>



# {04} DIY



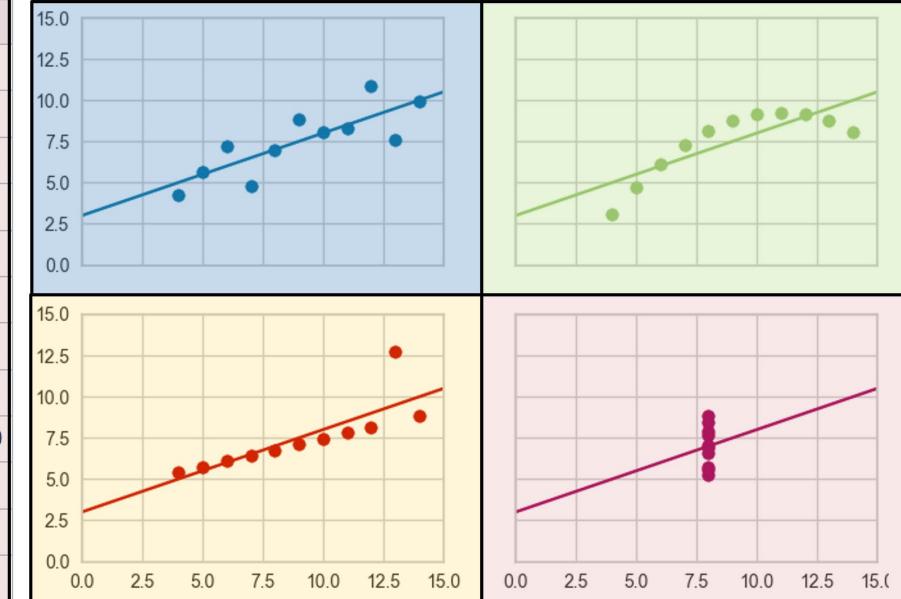
# {step Zero}

Dit ga je zelf  
online {coderen}  
& vizualizeren

[ 00, 10, 20 ],  
[ 10, 00, 10 ],  
[ 10, 10, 00 ]

## Anscombe's quartet

I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89



# {DATA SCIENCE REFERENCES of Anscombes Quartet}

[https://github.com/andrewhesterington/python-  
projects/blob/master/Blog%20Anscombe's%20Quartet/Anscombe's%20Quartet.ipynb](https://github.com/andrewhesterington/python-projects/blob/master/Blog%20Anscombe's%20Quartet/Anscombe's%20Quartet.ipynb)

<https://github.com/DistrictDataLabs/yellowbrick>

[https://matplotlib.org/stable/gallery/specialty\\_plots/anscombe.html](https://matplotlib.org/stable/gallery/specialty_plots/anscombe.html)

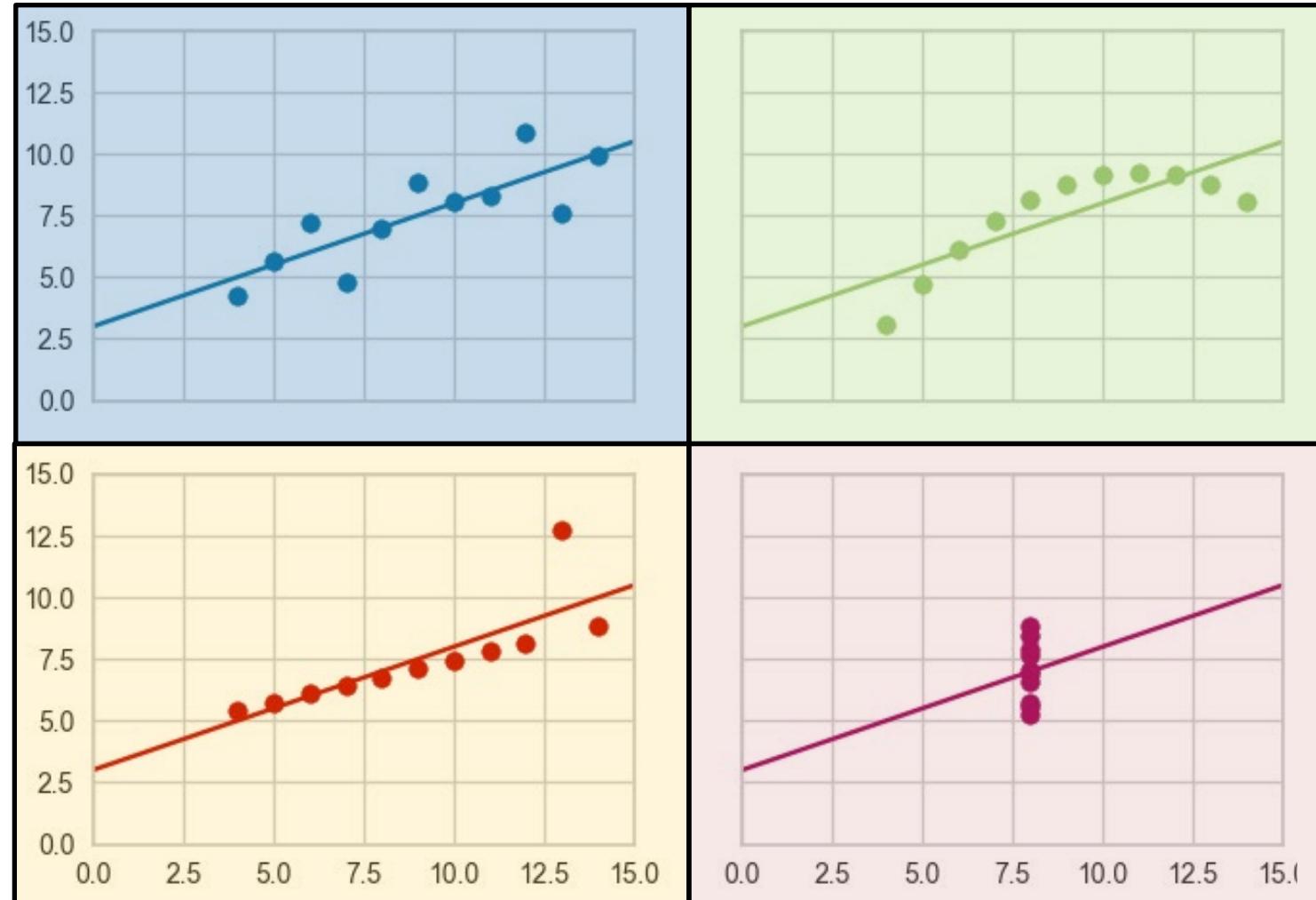
<https://www.scikit-yb.org/en/latest/api/anscombe.html>

<https://bmeyers.github.io/SolvingAnscombesQuartet/>

# Anscombe's quartet

Anscombe's quartet

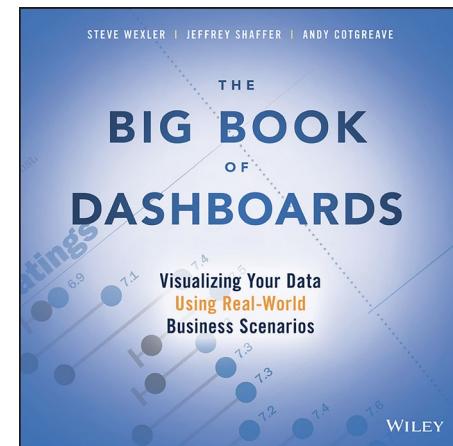
I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89



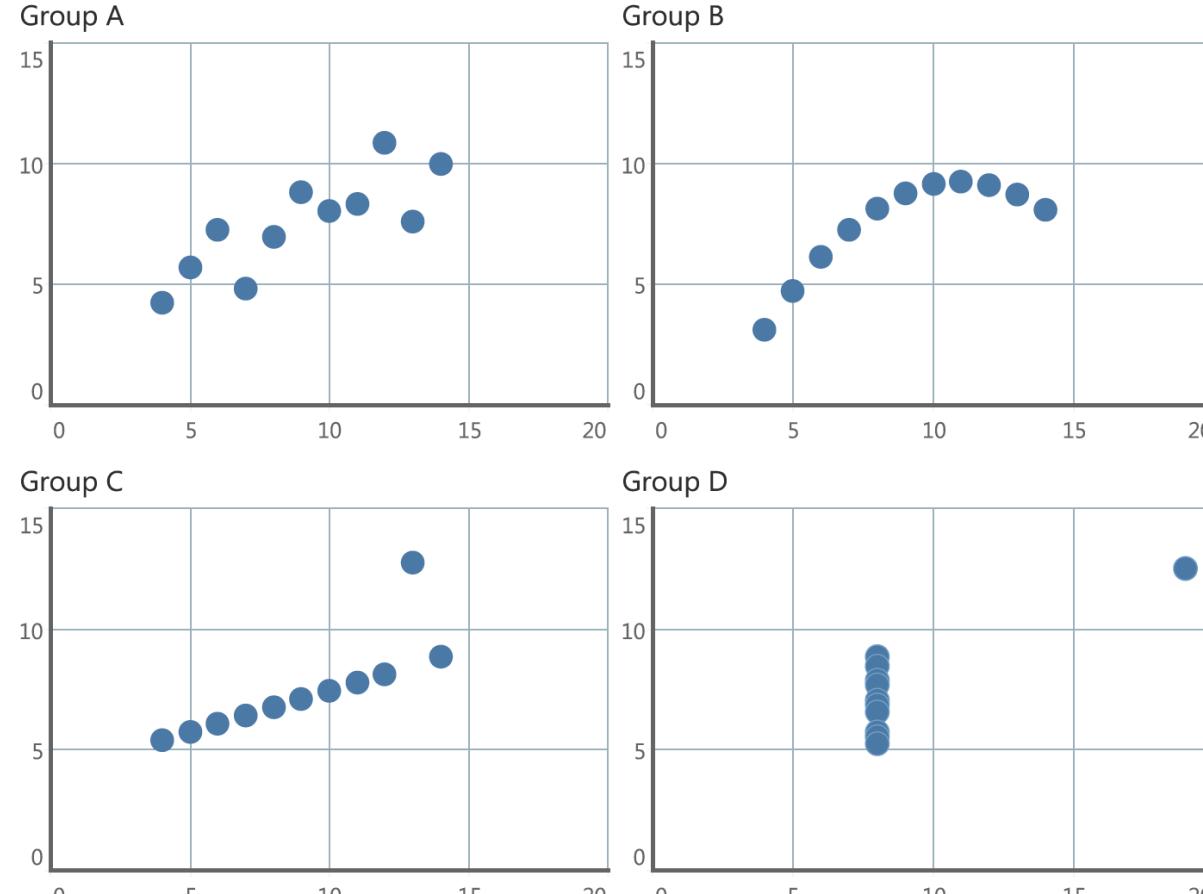
# Anscombe's quartet

**TABLE 1.1** Table with four groups of numbers: What do they tell you?

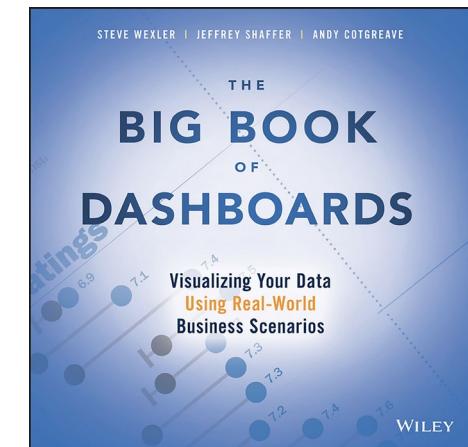
Group A		Group B		Group C		Group D	
x	y	x	y	x	y	x	y
10.00	8.04	10.00	9.14	10.00	7.46	8.00	6.58
8.00	6.95	8.00	8.14	8.00	6.77	8.00	5.76
13.00	7.58	13.00	8.74	13.00	12.74	8.00	7.71
9.00	8.81	9.00	8.77	9.00	7.11	8.00	8.84
11.00	8.33	11.00	9.26	11.00	7.81	8.00	8.47
14.00	9.96	14.00	8.10	14.00	8.84	8.00	7.04
6.00	7.24	6.00	6.13	6.00	6.08	8.00	5.25
4.00	4.26	4.00	3.10	4.00	5.39	19.00	12.50
12.00	10.84	12.00	9.13	12.00	8.15	8.00	5.56
7.00	4.82	7.00	7.26	7.00	6.42	8.00	7.91
5.00	5.68	5.00	4.74	5.00	5.73	8.00	6.89



# Anscombe's quartet



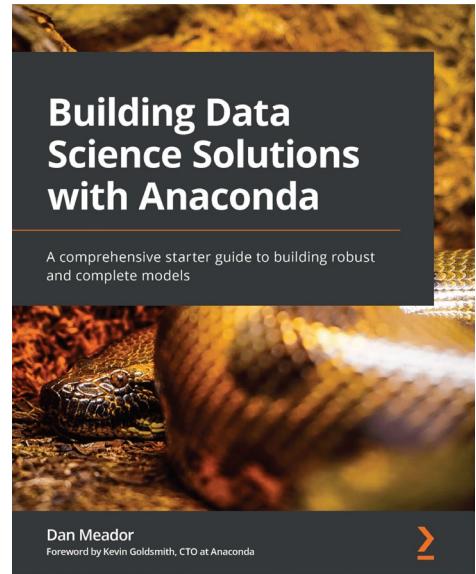
RE 1.1 Now can you see a difference in the four groups?



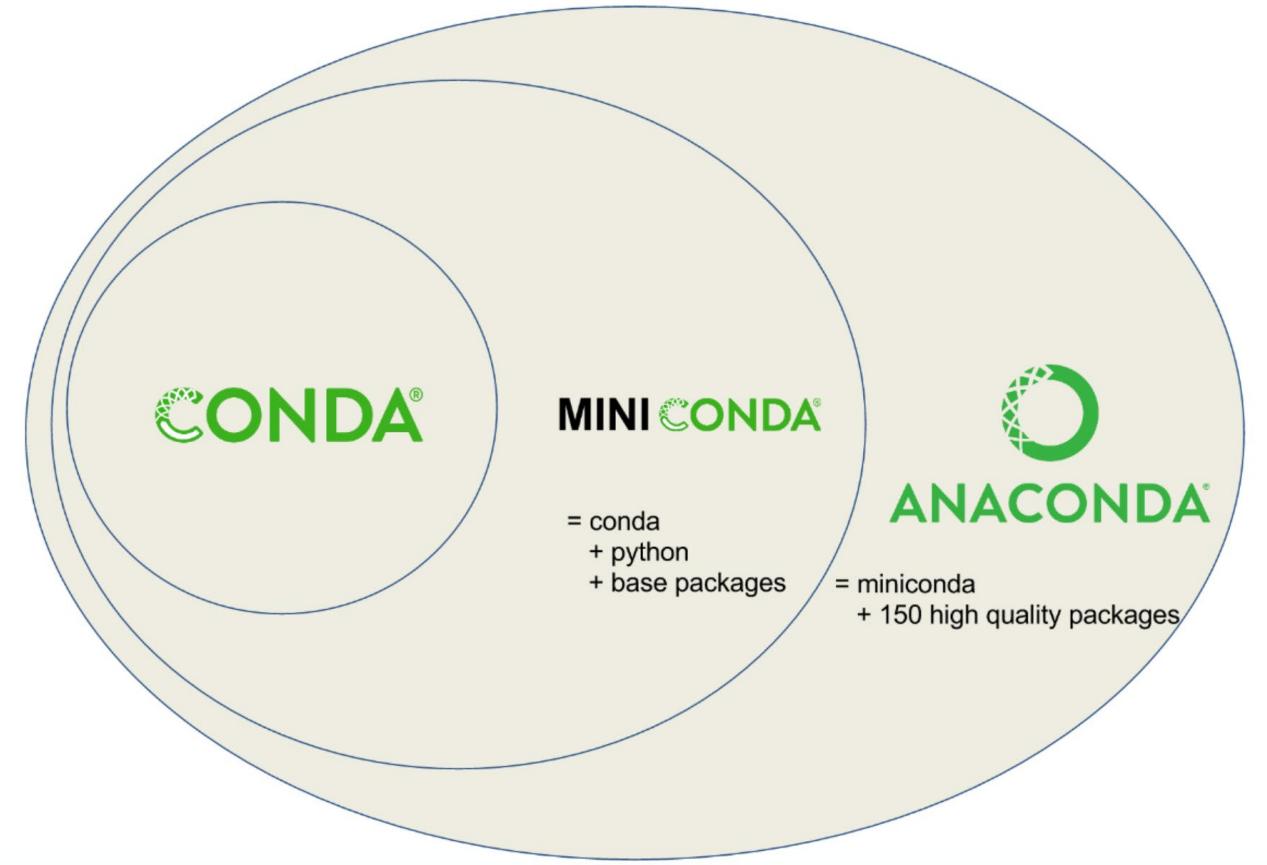
# {step ONE}

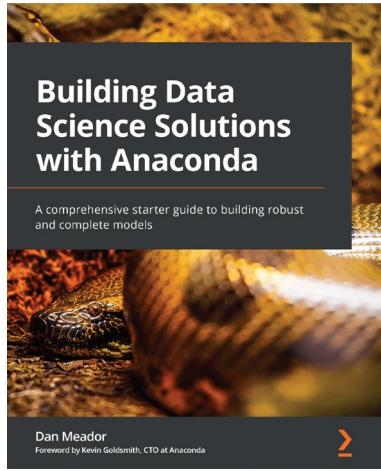
## Get Acces to a online {coding} TOOL





# ANACONDA





# TERMINOLOGY

**API** : *Application Programming Interface* is a computing interface which defines interactions between multiple software intermediaries.

**GUI** : *Graphical User Interface* is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicators, such as primary notation, instead of text-based user interfaces, typed command labels or text navigation.

**IDE** : An *Integrated Development Environment* is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools, and a debugger.

**Library** : Collection of software packages, i.e., Numpy, Matplotlib, Seaborn and TensorFlow.

**Package** : Collection of modules.

**Module** : A set of functions, types, classes, etc. Anything ending in .py.

**Command line Console/Terminal** : A console (also called a “shell”) is a command line interpreter that takes input from the user and interprets it. Cannot save or store information.

# API v.s. SDK v.s. IDE

## Application Programming Interface (API)

A library of functions and methods. You don't need to know how it works, but you have to know how to call them.

## Software Development Kit (SDK)

Many tools are included in SDK. Different platform, different SDK.

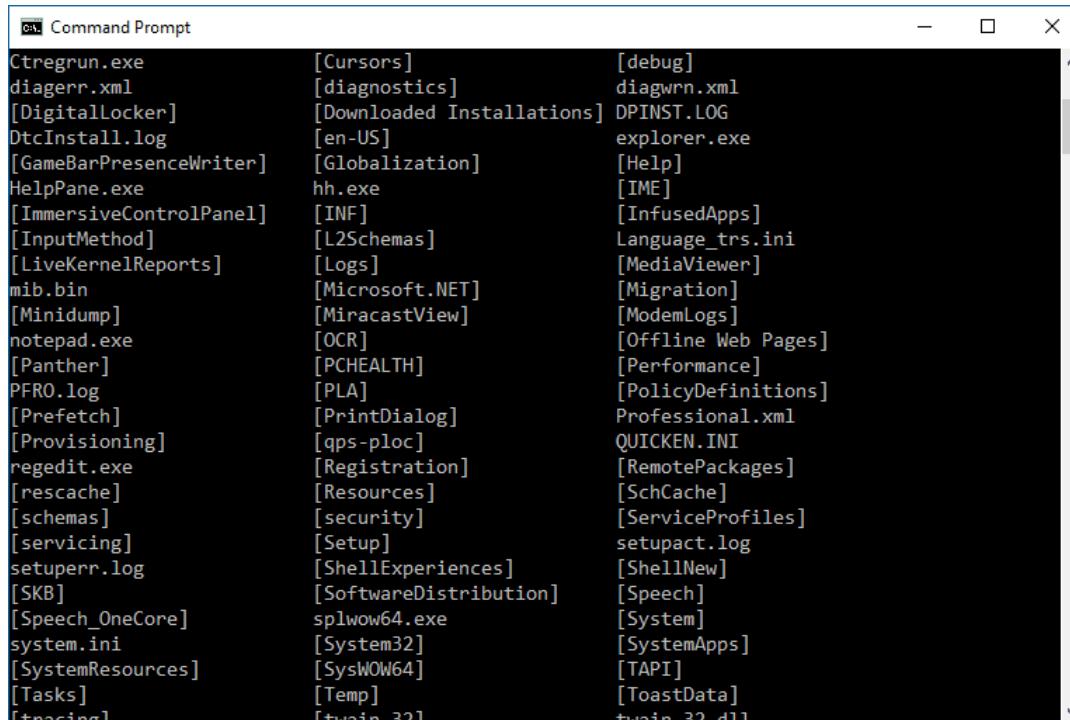
## Integrated Development Environment (IDE)

Usually includes code editor, debugger, compiler.

# Command Line Terminal (windows)

## List of Command Prompt Commands

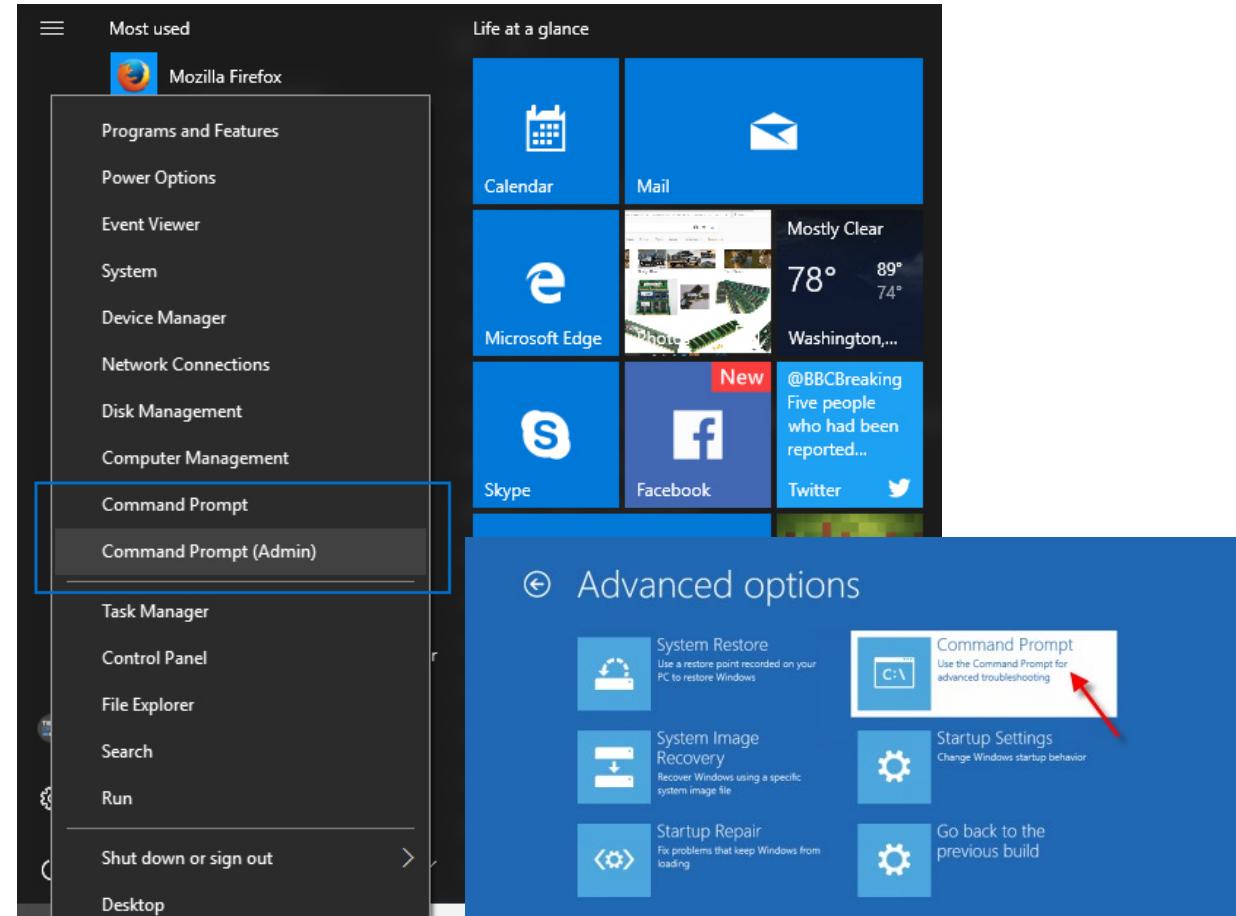
Complete list of CMD commands for Windows



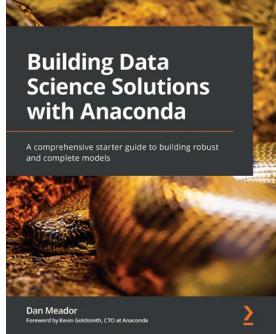
```
C:\ Command Prompt
Ctregrun.exe [Cursors] [debug]
diagerr.xml [diagnostics] diagwrn.xml
[DigitalLocker] [Downloaded Installations] DPINST.LOG
DtcInstall.log [en-US] explorer.exe
[GameBarPresenceWriter] [Globalization] [Help]
HelpPane.exe hh.exe [IME]
[ImmersiveControlPanel] [INF] [InfusedApps]
[InputMethod] [L2Schemas] Language_trs.ini
[LiveKernelReports] [Logs] [MediaViewer]
mib.bin [Microsoft.NET] [Migration]
[Minidump] [MiracastView] [ModemLogs]
notepad.exe [OCR] [Offline Web Pages]
[Panther] [PCHEALTH] [Performance]
PFRO.log [PLA] [PolicyDefinitions]
[Prefetch] [PrintDialog] Professional.xml
[Provisioning] [gps-plc] QUICKEN.INI
regedit.exe [Registration] [RemotePackages]
[rescache] [Resources] [SchCache]
[schemas] [security] [ServiceProfiles]
[servicing] [Setup] setupact.log
setuperr.log [ShellExperiences] [ShellNew]
[SKB] [SoftwareDistribution] [Speech]
[Speech_OneCore] splwow64.exe [System]
system.ini [System32] [SystemApps]
[SystemResources] [SysWOW64] [TAPI]
[Tasks] [Temp] [ToastData]
[tracing] [twain_32] twain_32.dll
```

The [Command Prompt](#) in Windows provides access to over 280 [commands](#)! These commands are used to do certain [operating system](#) tasks from a [command line interface](#) instead of the graphical Windows interface we use most of the time.

**Note:** It's important to know that the commands in Windows 10, 8, 7, Vista, and XP are called *CMD commands* or *Command Prompt commands*, and the commands in Windows 98/95 and MS-DOS are called *DOS commands*.



<https://www.lifewire.com/command-prompt-tricks-and-hacks-2618104>



Building Data  
Science Solutions  
with Anaconda

A comprehensive starter guide to building robust  
and complete models

Dan Meador  
Foreword by Kevin Goldsmith, CTO of Anaconda

# Command-Line Tools

## → Power Shell {windows}

### PowerShell Cheat Sheet **comparitech**

#### Basic Commands

**Cmdlet** Commands built into shell written in .NET  
**Functions** Commands written in PowerShell language  
**Parameter** Argument to a Cmdlet/Function/Script  
**Alias** Shortcut for a Cmdlet or Function  
**Scripts** Text files with .ps1 extension  
**Applications** Existing windows programs  
**Pipelines** Pass objects Get-process word | Stop-Process  
**Ctrl+c** Interrupt current command  
**Left/right** Navigate editing cursor  
**Ctrl+left/right** Navigate a word at a time  
**Home / End** End Move to start / end of line  
**Up/down** Move up and down through history  
**Insert** Toggles between insert/overwrite mode  
**F7** Command history in a window  
**Tab / Shift-Tab** Command line completion

#### Variables

\$var = "string" Assign variable  
\$@,\$# = 0 or \$a,\$b = 'a','b' Assign multiple variables  
\$a,\$b = \$b,\$a Flip variables  
\$var=[int]\$5 Strongly typed variable

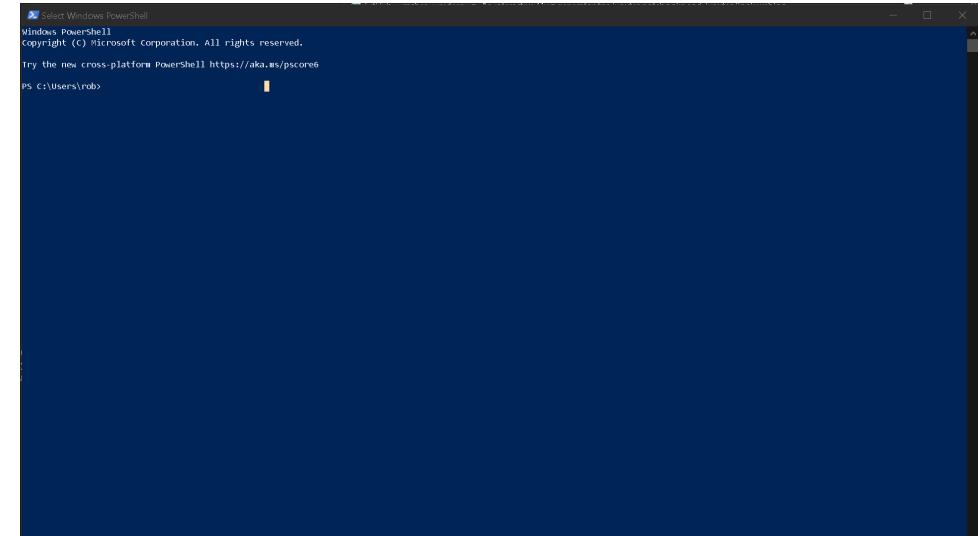
#### Help

Get-Command Get all commands  
Get-Command -Module RGHS Get all commands in RGHS module  
Get-Command Get-p\* Get all commands starting with get-p  
Get-help get-process Get help for command  
Get-Process | Get-Member Get members of the object  
Get-Process|format-list -properties \* Get-Process as list with all properties

#### Scripts

Set-ExecutionPolicy -ExecutionPolicy Bypass Set execution policy to allow all scripts  
."\\c-is-ts-91\\$scripts\script.ps1" Run Script.ps1 script in current scope  
&."\\c-is-ts-91\\$scripts\script.ps1" Run Script.ps1 script in scope  
.\\$profile Run Script.ps1 script in script scope  
\\$profile Your personal profile that runs at launch

Import, Export, Convert	Common cmdlets	Common Aliases
Export-CliXML Import-CliXML ConvertTo-XML Export-Csv ConvertTo-CSV ConvertFrom-CSV	cd, chdir, sl cat, gc, type ac sc copy, cp, cpi del, erase, rd, ri, rm, rmdir mi, move, mv si ni sleep sajb compare, diff group curl, iwr, wget measure nal rvpa rjab set, sv shcm sort sasv saps, start sujb wjb ?, where echo, write	Set-Location Get-Content Add-Content Set-Content Copy-Item Remove-Item Move-Item Set-Item New-Item Start-Sleep Start-Job Compare-Object Group-Object Invoke-WebRequest Measure-Object New-Alias Resolve-Path Resume-Job Set-Variable Show-Command Sort-Object Start-Service Start-Process Suspend-Job Wait-Job Where-Object Write-Output
Flow Control	Comments, Escape Characters	Arrays Objects
If{} Else{} Else{} For(\$i=0; \$i -lt 10; \$i++) Foreach(\$file in dir C:\){\$file.name} 1..10   foreach{\$_}	#Comment Comment <#comment#> Multiline Comment '''test''' Escape char` ' Tab 'n New line ' Line continue	\$arr = "a", "b" Array of strings \$arr = @() Empty array \$arr[5] Sixth array element \$arr[3..1] Last three array elements \$arr[1..6..9] Elements at index 1,4, 6-9 \$arr[1] += 200 Add to array item value \$z = \$arr + \$arr Two arrays into single array [pscustomobject]@{x=1;z=2} Create custom object (Get-Date).Date Date property of object
Parameters	Assignment, Logical, Comparison	Writing output and reading
-Confirm -WhatIf	Prompt whether to take action Displays what command would do	"This displays a string" String is written directly to output Write-Host "color" -ForegroundColor Red -NoNewLine String with colors, no new line at end \$age = Read-host "Please enter your age" Set \$age variable to input from user \$pwd = Read-host "Please enter your password" -asSecureString Read in \$pwd as secure string Clear-Host Clear console
Help		



[https://github.com/SQLPlayer/CheatSheets/blob/master/Powershell-cheatsheet%20\(Comparitech\).pdf](https://github.com/SQLPlayer/CheatSheets/blob/master/Powershell-cheatsheet%20(Comparitech).pdf)

# Command Line Terminal

## Introduction to the Mac OS X Command Line

### How to open the command line.

Before you can use it, you need to be able to find it.

So what we need to do is open the terminal. On OS X, open your Applications folder, then open the Utilities folder. Open the Terminal application. You may want to add this to your dock. I like to launch terminal by using Spotlight search in OS X, searching for “terminal”.



<http://blog.teamtreehouse.com/introduction-to-the-mac-os-x-command-line>

### Anatomy of the Console

First let's clarify a few terms.

**Console:** This is the system as a whole. This is both the command line as well as the output from previous commands.

**Command Line:** This is the actual line in a console where you type your command.

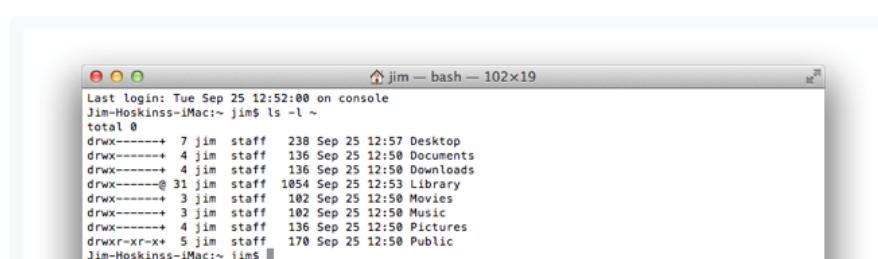
**Prompt:** This is the beginning of the command line. It usually provides some contextual information like who you are, where you are and other useful info. It typically ends in a \$ . After the prompt is where you will be typing commands.

**Terminal:** This is the actual interface to the console. The program we use to interact with the console is actually a “terminal emulator”, providing us the experience of typing into an old school terminal from the convenience of our modern graphical operating system.

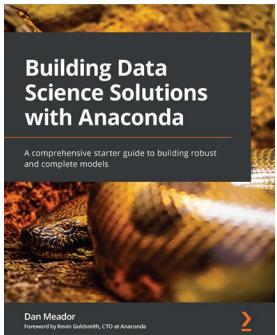
### Running a Command.

Nearly all commands follow a common pattern with 3 main parts. The program, the options, and the arguments. Let's see an example.

```
$ ls -l ~
```

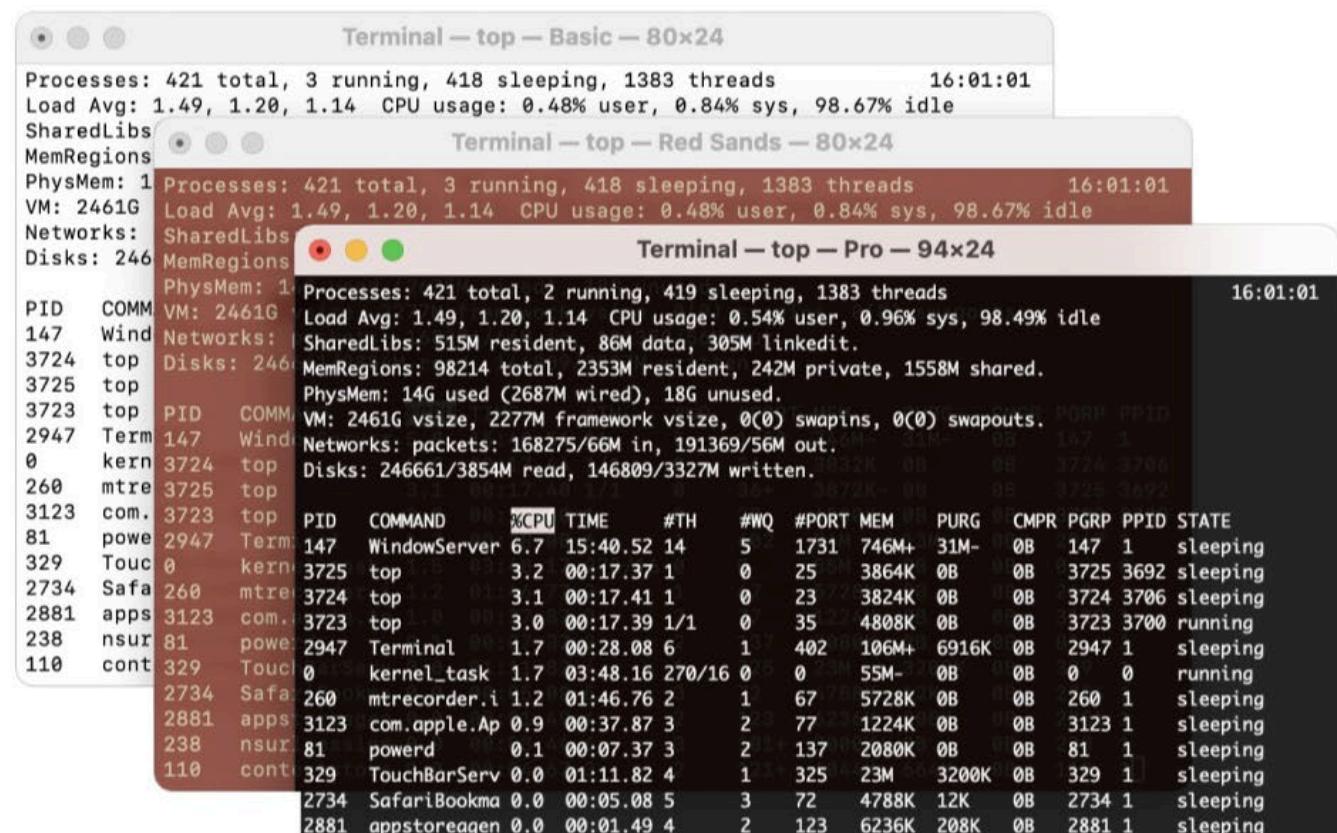
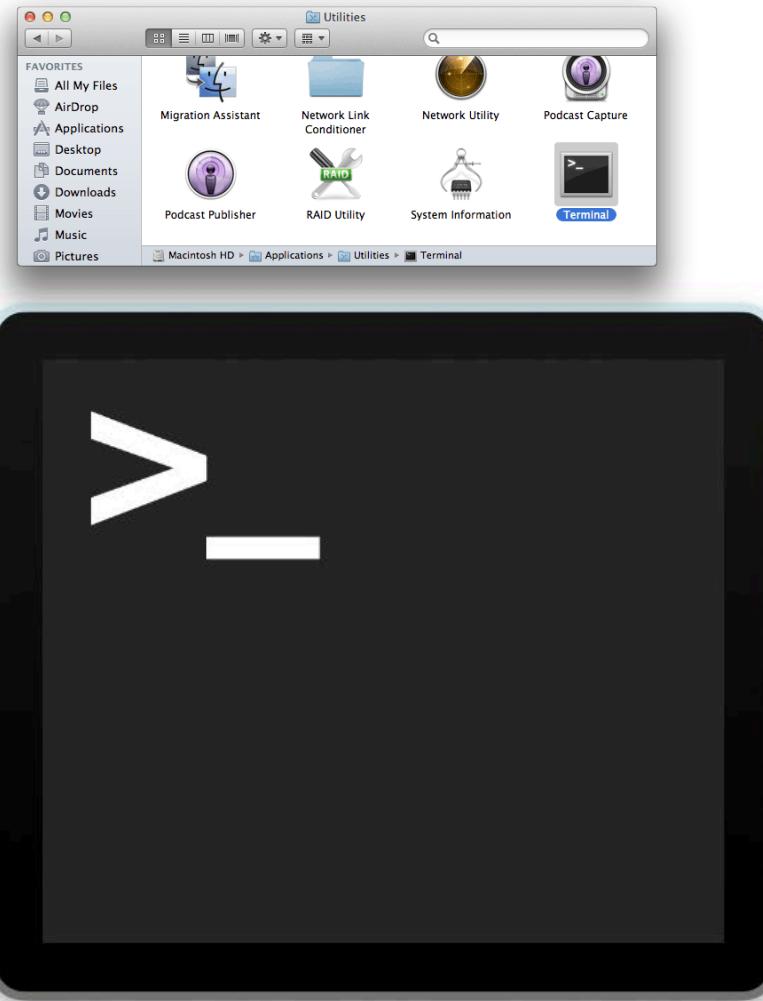


```
Last login: Tue Sep 25 12:52:00 on console
jim-Hoskinss-iMac:~ jim$ ls -l ~
total 0
drwx-----+ 7 jim staff 238 Sep 25 12:57 Desktop
drwx-----+ 4 jim staff 136 Sep 25 12:58 Documents
drwx-----+ 4 jim staff 136 Sep 25 12:58 Downloads
drwx-----@ 31 jim staff 1054 Sep 25 12:53 Library
drwx-----+ 3 jim staff 102 Sep 25 12:50 Movies
drwx-----+ 3 jim staff 102 Sep 25 12:50 Music
drwx-----+ 4 jim staff 136 Sep 25 12:50 Pictures
drwxr-xr-x+ 5 jim staff 170 Sep 25 12:50 Public
jim-Hoskinss-iMac:~ jim$
```



# Command-Line Tools

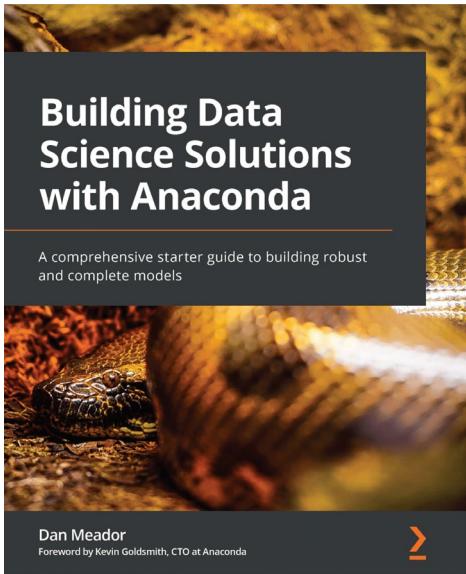
## → Terminal {MacOS}



# Command-Line Terminal

Online Terminals					
 CentOS	 Ipython	 Python-3	 Lua	 Memcached	 Mongo DB
 MySQL	 Node.js	 NumPy	 Oracle	 Octave	 PowerShell
 PHP	 R Programming	 Redis	 Ruby	 Scipy	 Sympy

<http://www.tutorialspoint.com/codingground.htm>



## Building Data Science Solutions with Anaconda

A comprehensive starter guide to building robust and complete models



Dan Meador  
Foreword by Kevin Goldsmith, CTO at Anaconda



# Installers & Packages

## Anaconda Installers

Contains over 200 curated packages that are securely built, highly optimized, and tested together to ensure compatibility.

[View All Installers](#)

## Anaconda Packages

Install more packages with [conda](#). Anaconda, Inc. maintains 1,000+ professionally built packages for data science.

[View All Packages](#)

## Miniconda Installers

The minimal way to bootstrap conda onto a system. Contains only conda and its dependencies.

[View All Installers](#)

## Installing Anaconda



### Mac

1. Open <https://www.anaconda.com/download> in your web browser.
2. Download the Anaconda installer for macOS (select the Graphical installer, not the Command Line installer).
3. Install Anaconda using all of the defaults for installation.

Make sure to click **Install for me only** if the installer says: *You cannot install Anaconda3 in this location. The Anaconda3 installer does not allow its software to be installed here.*

Refer to the [detailed step-by-step instructions for Mac](#) if needed.



### Windows

1. Open <https://www.anaconda.com/download> in your web browser.
2. Download the Anaconda installer for Windows.
3. Install Anaconda using all of the defaults for installation.

Refer to the [detailed step-by-step instructions for Windows](#) if needed.

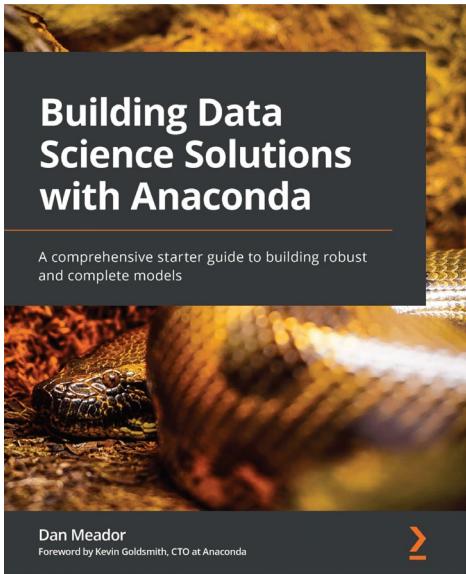


### Linux

1. Open <https://www.anaconda.com/download> in your web browser.
2. Download the Anaconda installer (x86 version) for Linux, make a note of the filename.
3. Open a Terminal window, and type bash followed by the name of the file you downloaded, for example:  
`bash ~/Downloads/Anaconda3-2020.07-Linux-x86_64.sh`

Refer to the [detailed step-by-step instructions for Linux](#) if needed.

<https://wiki.math.ntnu.no/anaconda/start#:~:text=Windows%3A%20You'll%20find%20Anaconda,run%20the%20command%20anaconda%2Dnavigator%20.>



## Anaconda Installers WWW links

### Windows

#### Python 3.9

[64-Bit Graphical Installer \(594 MB\)](#)

[32-Bit Graphical Installer \(488 MB\)](#)

### MacOS

#### Python 3.9

[64-Bit Graphical Installer \(591 MB\)](#)

[64-Bit Command Line Installer \(584 MB\)](#)

[64-Bit \(M1\) Graphical Installer \(316 MB\)](#)

[64-Bit \(M1\) Command Line Installer \(305 MB\)](#)

### Linux

#### Python 3.9

[64-Bit \(x86\) Installer \(659 MB\)](#)

[64-Bit \(Power8 and Power9\) Installer \(367 MB\)](#)

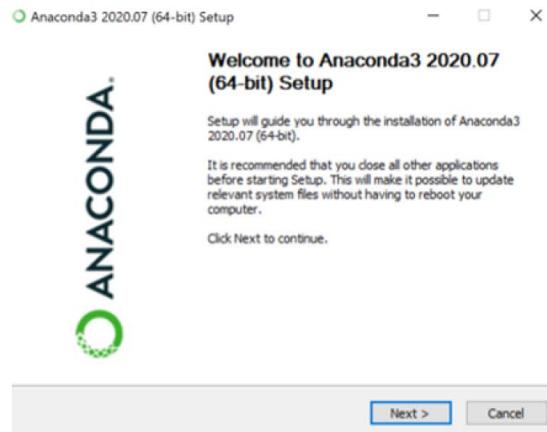
[64-Bit \(AWS Graviton2 / ARM64\) Installer \(568 MB\)](#)

[64-bit \(Linux on IBM Z & LinuxONE\) Installer \(280 MB\)](#)

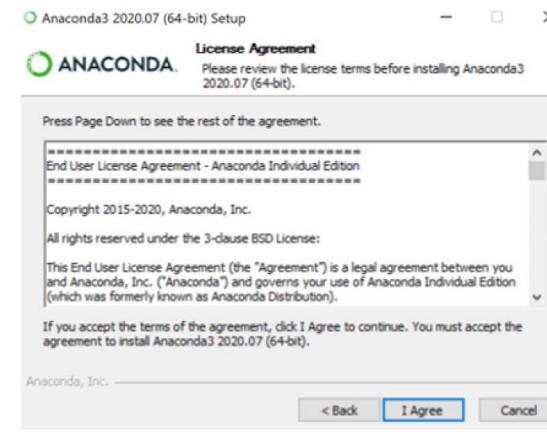
<https://www.anaconda.com/products/distribution#Downloads>

# Windows installation

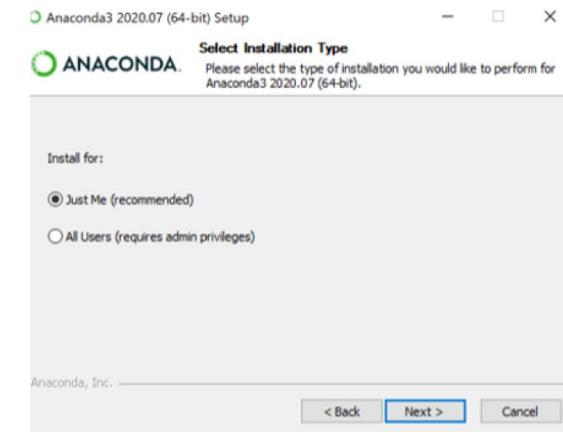
(a) Welcome Screen



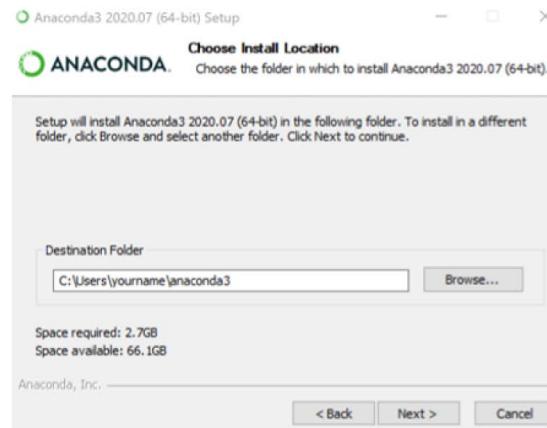
(b) Read/accept license agreement



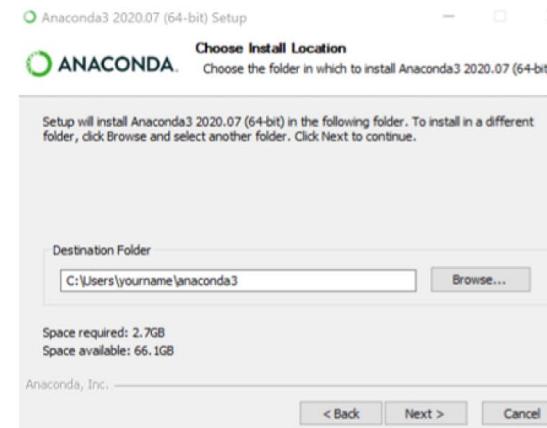
(c) Installation type



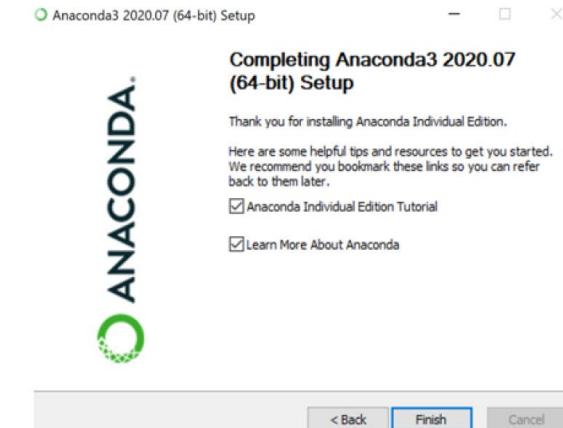
(d) Choosing location for installation



(e) Advanced installation options

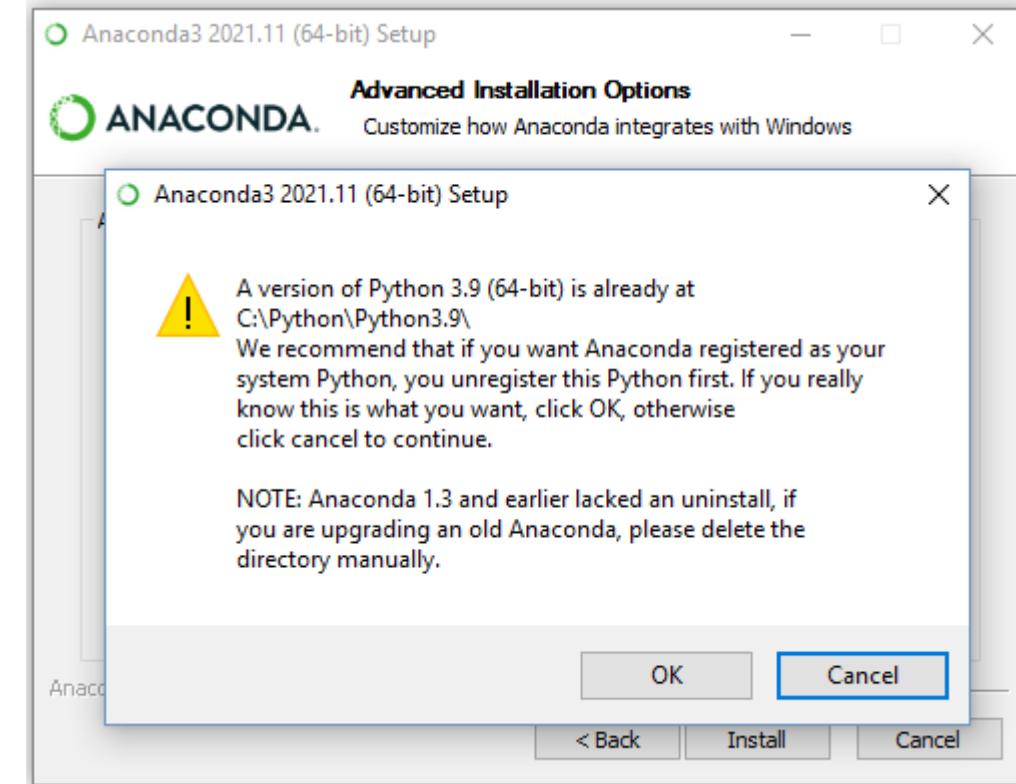


(f) Finish installation and close wizard



# Windows installation

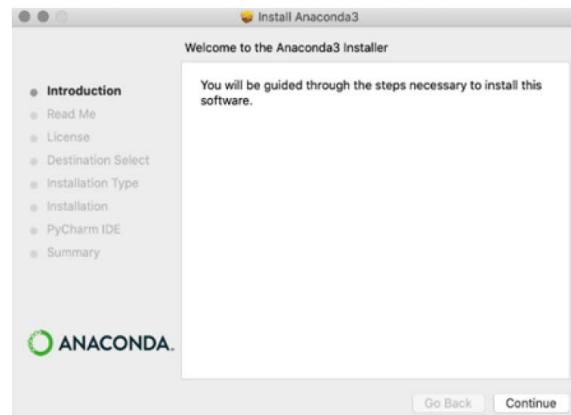
## +complications+



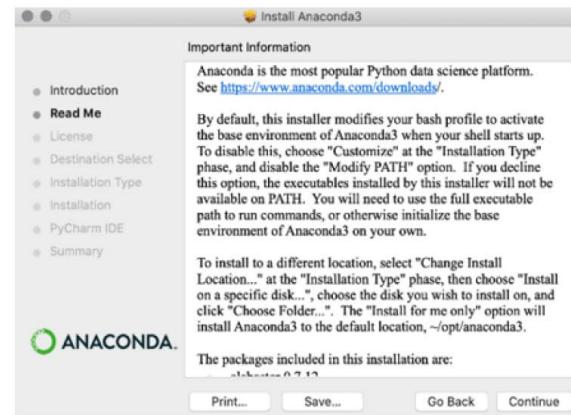
<https://gurubhaipoint.com/how-to-download-install-anaconda-python-on-windows-64-32-bit/>

# MacOS installation

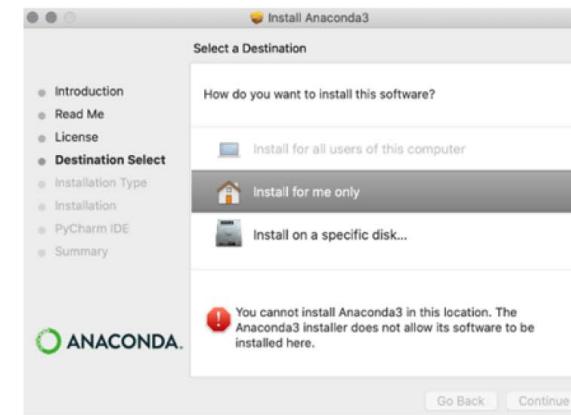
(a) Welcome Screen



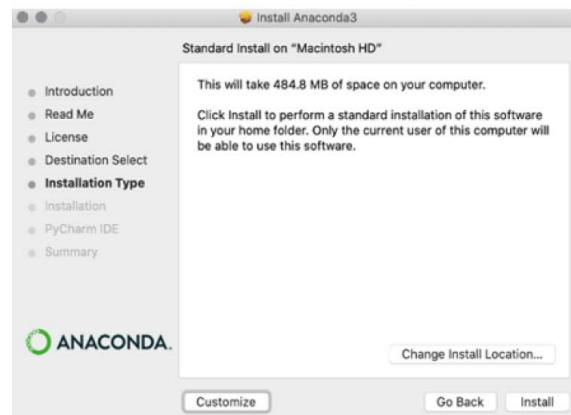
(b) Read/accept license agreement



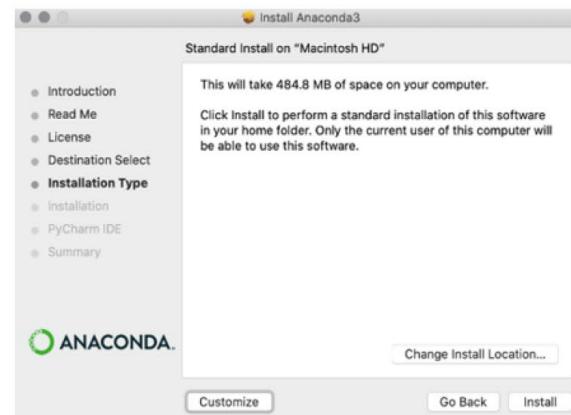
(c) Installation type



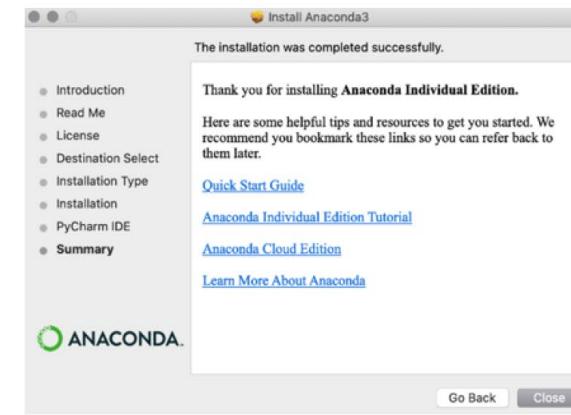
(d) Choosing location for installation



(e) Advanced installation options



(f) Finish installation and close wizard



<https://docs.anaconda.com/anaconda/install/mac-os/#macos-graphical-install>

# LINUX installation

<https://www.digitalocean.com/community/tutorials/how-to-install-the-anaconda-python-distribution-on-ubuntu-22-04>

// Tutorial //

## How To Install the Anaconda Python Distribution on Ubuntu 22.04

Published on April 26, 2022

Data Analysis Development Machine Learning Python Ubuntu 22.04



By [Lisa Tagliaferri](#) and [Tony Tran](#)



**Not using Ubuntu 22.04?**

Choose a different version or distribution.

Ubuntu 22.04 ▾

<https://docs.anaconda.com/anaconda/install/linux/>

# LINUX installation

The best way to install Anaconda is to download the latest Anaconda installer bash script, verify it, and then run it.

Find the latest version of Anaconda for Python 3 at the [Anaconda Downloads page](#). At the time of writing, the latest version is [2021.11](#), but you should use a later stable version if it is available.

Next, change to the `/tmp` directory on your server. This is a good directory to download ephemeral items, like the Anaconda bash script, which you won't need after running it.

```
$ cd /tmp
```

Copy

Use `curl` to download the link that you copied from the Anaconda website. You'll output this to a file called `anaconda.sh` for quicker use.

```
$ curl https://repo.anaconda.com/archive/Anaconda3-2021.11-Linux-x86_64.sh --output anaconda.sh
```

Copy

You can now verify the data integrity of the installer with cryptographic hash verification through the SHA-256 checksum. You'll use the `sha256sum` command along with the filename of the script:

```
$ sha256sum anaconda.sh
```

Copy

You'll receive output that looks similar to this:

Output

```
fedf9e340039557f7b5e8a8a86affa9d299f5e9820144bd7b92ae9f7ee08ac60 anaconda.sh
```

You should check the output against the hashes available at the [Anaconda with Python 3 on 64-bit Linux page](#) for your appropriate Anaconda version. As long as your output matches the hash displayed in the `sha256` row, you're good to go.

Now you can run the script:

```
$ bash anaconda.sh
```

Copy

You'll receive the following output:

Output

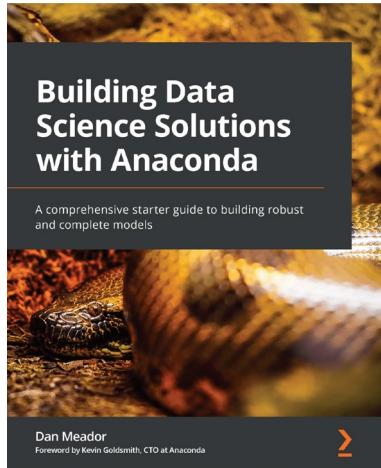
```
Welcome to Anaconda3 2021.11
```

```
In order to continue the installation process, please review the license agreement.
```

```
Please, press ENTER to continue
```

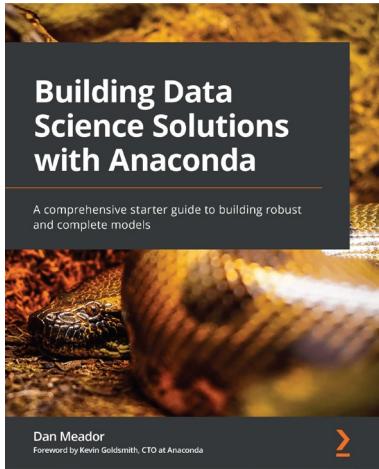
```
>>>
```

<https://www.digitalocean.com/community/tutorials/how-to-install-the-anaconda-python-distribution-on-ubuntu-22-04>



# ANACONDA package lists

	Python 3.9	Python 3.8	Python 3.7
64-bit Windows	<a href="#">64-bit Windows, Py3.9</a>	<a href="#">64-bit Windows, Py3.8</a>	<a href="#">64-bit Windows, Py3.7</a>
32-bit Windows	<a href="#">32-bit Windows, Py3.9</a>	<a href="#">32-bit Windows, Py3.8</a>	<a href="#">32-bit Windows, Py3.7</a>
64-bit macOS Intel x86	<a href="#">macOS Intel x86, Py3.9</a>	<a href="#">macOS Intel x86, Py3.8</a>	<a href="#">macOS Intel x86, Py3.7</a>
64-bit macOS Apple M1	<a href="#">macOS Apple M1, Py3.9</a>	<a href="#">macOS Apple M1, Py3.8</a>	Not supported
64-bit Linux	<a href="#">64-bit Linux, Py3.9</a>	<a href="#">64-bit Linux, Py3.8</a>	<a href="#">64-bit Linux, Py3.7</a>
64-bit Linux Graviton2 ARM64	<a href="#">Linux on ARM64, Py3.9</a>	<a href="#">Linux on ARM64, Py3.8</a>	<a href="#">Linux on ARM64, Py3.7</a>
64-bit Linux on IBM Z	<a href="#">Linux on IBM Z, Py3.9</a>	<a href="#">Linux on IBM Z, Py3.8</a>	<a href="#">Linux on IBM Z, Py3.7</a>
64-bit Linux on IBM Power CPUs	<a href="#">Linux on IBM Power, Py3.9</a>	<a href="#">Linux on IBM Power, Py3.8</a>	<a href="#">Linux on IBM Power, Py3.7</a>



# Understanding Conda & PIP

Pip is the command-line Package Installer for Python (PIP) Authority's recommended tool for installing packages from the [Python Package Index](#), PyPI.

Pip installs Python software packaged as wheels or source distributions. The latter may require that the system have compatible compilers, and possibly libraries, installed before invoking pip to succeed.

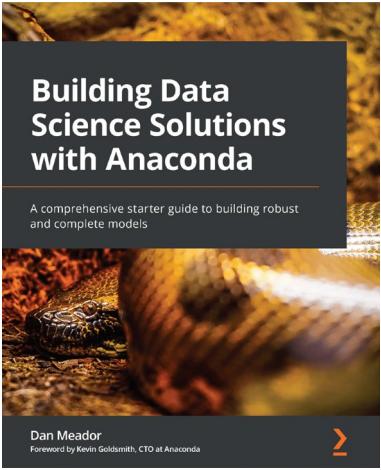
[https://en.wikipedia.org/wiki/Pip\\_\(package\\_manager\)](https://en.wikipedia.org/wiki/Pip_(package_manager))

Conda is a cross platform package and environment manager that installs and manages conda packages from the [Anaconda repository](#) as well as from the [Anaconda Cloud](#).

Conda packages are binaries. There is never a need to have compilers available to install them. Additionally conda packages are not limited to Python software.

They may also contain C or C++ libraries, R packages or any other software.

[https://en.wikipedia.org/wiki/Conda\\_\(package\\_manager\)](https://en.wikipedia.org/wiki/Conda_(package_manager))



# Packages for 64-bit Windows with Python 3.9

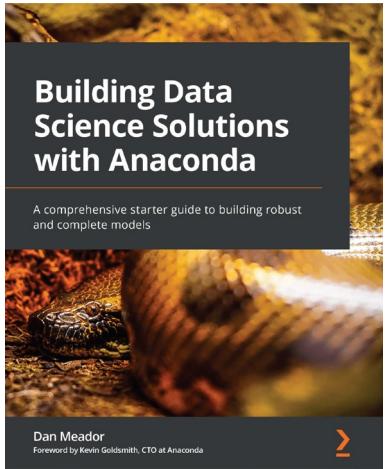
Platform: Windows 64-bit

Python version: 3.9

Number of supported packages: 678

Name	Version	Summary / License	In Installer
<a href="#">ipyw_jlab_nb_ext_conf</a>	0.1.0	A configuration metapackage for enabling Anaconda-bundled jupyter extensions / BSD	
<a href="#">libgcc_mutex</a>	0.1	Mutex for libgcc and libgcc-ng / None	
<a href="#">affine</a>	2.3.0	Matrices describing affine transformation of the plane. / BSD-3-Clause	
<a href="#">agate</a>	1.6.2	A data analysis library that is optimized for humans instead of machines. / COPYING	
<a href="#">agate-dbf</a>	0.2.2	agate-dbf adds read support for dbf files to agate. / MIT	
<a href="#">agate-excel</a>	0.2.5	agate-excel adds read support for Excel files (xls andxlsx) to agate. / MIT	
<a href="#">aiobotocore</a>	2.1.0	Async client for aws services using botocore and aiohttp / Apache-2.0	
<a href="#">aiofiles</a>	0.7.0	File support for asyncio / Apache 2.0	
<a href="#">aiohttp</a>	3.8.1	Async http client/server framework (asyncio) / Apache-2.0	
<a href="#">aiosignal</a>	1.2.0	aiosignal: a list of registered asynchronous callbacks / Apache-2.0	
<a href="#">alabaster</a>	0.7.12	Configurable, Python 2+3 compatible Sphinx theme. / BSD 3-Clause	
<a href="#">alembic</a>	1.7.5	A database migration tool for SQLAlchemy. / MIT	
<a href="#">anaconda-clean</a>	1.1.0	Delete Anaconda configuration files / BSD	
<a href="#">anaconda-client</a>	1.9.0	Anaconda Cloud command line client library / BSD 3-Clause	
<a href="#">anaconda-navigator</a>	2.1.4	Anaconda Navigator / proprietary - Continuum Analytics, Inc.	
<a href="#">anaconda-project</a>	0.10.2	Tool for encapsulating, running, and reproducing data science projects / BSD-3-Clause	
<a href="#">ansi2html</a>	1.5.2	Convert text with ANSI color codes to HTML or to LaTeX. / GPLv3+	

[https://docs.anaconda.com/anaconda/packages/py3.9\\_win-64/](https://docs.anaconda.com/anaconda/packages/py3.9_win-64/)



## anaconda / packages

[Packages](#)
[Files](#)
[Install Instructions](#)
<https://anaconda.org/anaconda/repo>

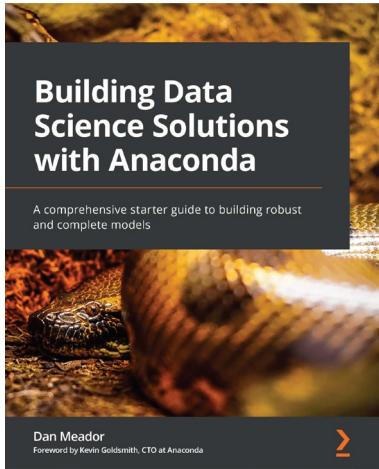
### Filters

Type: all ▾

Access: all ▾

Label: all ▾

Package Name	Access	Summary	Updated
<a href="#">tensorflow-base</a>	public	TensorFlow is a machine learning library, base package contains only tensorflow.	2022-09-05
<a href="#">snowflake-snowpark-python</a>	public	Snowflake Snowpark Python API	2022-09-05
<a href="#">snowflake-connector-python</a>	public	Snowflake Connector for Python	2022-09-05
<a href="#">pyspnego</a>	public	Windows Negotiate Authentication Client and Server	2022-09-05
<a href="#">mkl</a>	public	Math library for Intel and compatible processors	2022-09-05
<a href="#">neuralprophet</a>	public	Explainable Forecasting at Scale	2022-09-05
<a href="#">multidict</a>	public	multidict implementation	2022-09-05
<a href="#">mkl-include</a>	public	MKL headers for developing software that uses MKL	2022-09-05
<a href="#">ipykernel</a>	public	IPython Kernel for Jupyter	2022-09-05
<a href="#">dal-static</a>	public	Static libraries for Intel® oneDAL	2022-09-05
<a href="#">dal-devel</a>	public	Devel package for building things linked against Intel® oneDAL shared libraries	2022-09-05
<a href="#">dal</a>	public	Intel® oneDAL runtime libraries	2022-09-05
<a href="#">intel-openmp</a>	public	Math library for Intel and compatible processors	2022-09-05
<a href="#">dal-include</a>	public	Headers for building against Intel® oneDAL libraries	2022-09-05
<a href="#">daal-static</a>	public	Static libraries for DAAL	2022-09-05
<a href="#">daal-include</a>	public	Headers for building against DAAL libraries	2022-09-05
<a href="#">daal-devel</a>	public	Devel package for building things linked against DAAL shared libraries	2022-09-05
<a href="#">daal</a>	public	DAAL runtime libraries	2022-09-05
<a href="#">tapi</a>	public	TAPI is a Text-based Application Programming Interface	2022-09-05
<a href="#">clang</a>	public	Development headers and libraries for Clang	2022-09-04
<a href="#">psycopg2</a>	public	PostgreSQL database adapter for Python	2022-09-04
<a href="#">clang-14</a>	public	Development headers and libraries for Clang	2022-09-03
<a href="#">tensorflow-estimator</a>	public	TensorFlow Estimator is a high-level TensorFlow API that greatly simplifies machine learning programming.	2022-09-02
<a href="#">tensorflow</a>	public	TensorFlow is a machine learning library.	2022-09-02
<a href="#">tensorboard</a>	public	TensorFlow's Visualization Toolkit	2022-09-02
<a href="#">clang-tools</a>	public	Development headers and libraries for Clang	2022-09-02
<a href="#">python</a>	public	General purpose programming language	2022-09-02
<a href="#">python-regr-testsuite</a>	public	General purpose programming language	2022-09-02
<a href="#">python-clang</a>	public	Development headers and libraries for Clang	2022-09-02
<a href="#">libclang13</a>	public	Development headers and libraries for Clang	2022-09-02
<a href="#">libpython-static</a>	public	General purpose programming language	2022-09-02
<a href="#">libclang-cpp</a>	public	Development headers and libraries for Clang	2022-09-02
<a href="#">libclang</a>	public	Development headers and libraries for Clang	2022-09-02



# Managing packages using the command-line

[Home](#) » User guide » Tasks » Managing packages

[Edit on GitHub](#)

## Managing packages

- [Searching for packages](#)
- [Installing packages](#)
- [Installing similar packages](#)
- [Installing packages from Anaconda.org](#)
- [Installing non-conda packages](#)
- [Installing commercial packages](#)
- [Viewing a list of installed packages](#)
- [Listing package dependencies](#)
- [Updating packages](#)
- [Preventing packages from updating \(pinning\)](#)
- [Adding default packages to new environments automatically](#)
- [Removing packages](#)

### Note

There are many options available for the commands described on this page. For details, see [Command reference](#).

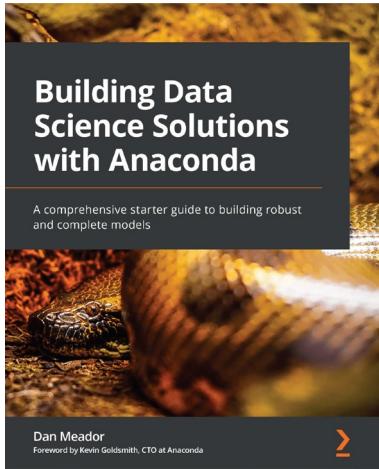
## Searching for packages

Use the terminal or an Anaconda Prompt for the following steps.

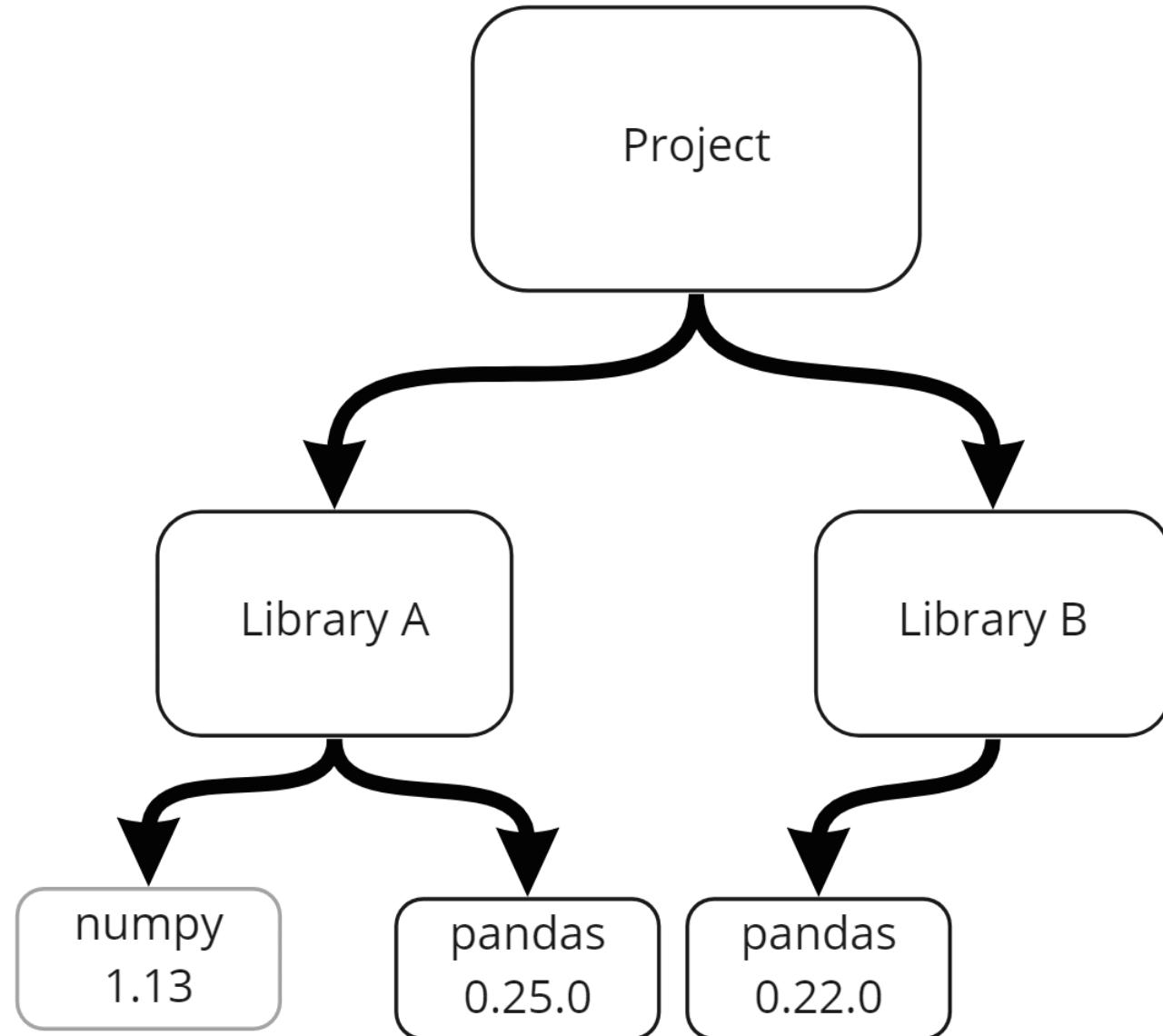
To see if a specific package, such as SciPy, is available for installation:

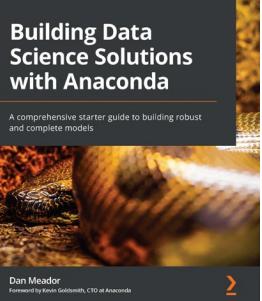
```
conda search scipy
```

<https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-pkgs.html>



Direct dependency  
  
Transitive dependency





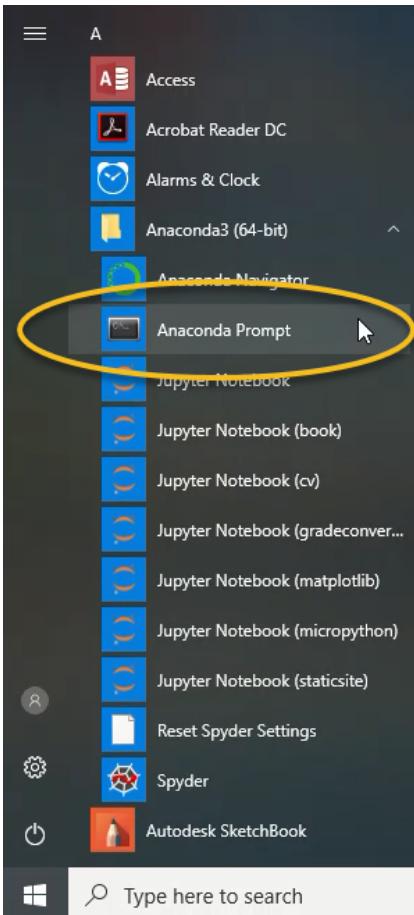
Building Data  
Science Solutions  
with Anaconda

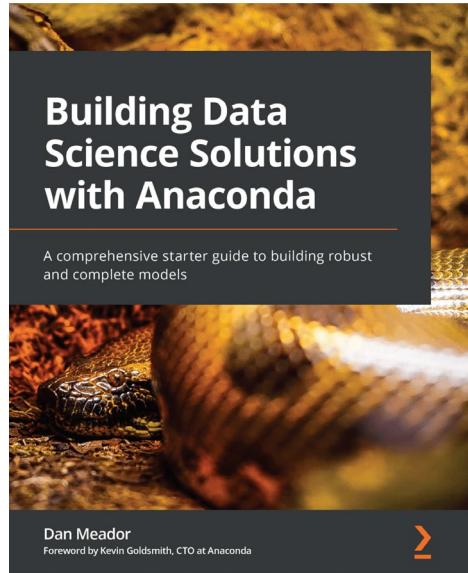
A comprehensive starter guide to building robust  
and complete models

Dan Meador  
Foreword by Kevin Goldsmith, CTO at Anaconda

# Command-Line Tools

## →Anaconda Prompt{windows}





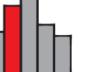
# ANACONDA NAVigator == GUI

Anaconda Navigator    File    Help    Sign in to Anaconda Cloud    Refresh

Home    Environments    Learning    Community    Documentation    Developer Blog

Applications on base (root)    Channels

↓

 <b>CMD.exe Prompt</b> 0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated <a href="#">Launch</a>	 <b>JupyterLab</b> 2.1.5 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. <a href="#">Launch</a>	 <b>jupyter Notebook</b> 6.0.3 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. <a href="#">Launch</a>	 <b>Powershell Prompt</b> 0.0.1 Run a Powershell terminal with your current environment from Navigator activated <a href="#">Launch</a>
 <b>Qt Console</b> 4.7.5 PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more. <a href="#">Launch</a>	 <b>Spyder</b> 4.1.4 Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features <a href="#">Launch</a>	 <b>Glueviz</b> 0.15.2 Multidimensional data visualization across files. Explore relationships within and among related datasets. <a href="#">Install</a>	 <b>Orange 3</b> 3.26.0 Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox. <a href="#">Install</a>

[Documentation](#)    [Developer Blog](#)

[Twitter](#)    [YouTube](#)    [GitHub](#)



# Package Management via GUI nav

The screenshot shows the Anaconda Navigator interface with the search bar set to 'pandas'. The results table displays 57 packages matching the search term, with one package selected: 'pandas' (version 1.4.2). The 'Apply' button at the bottom right is highlighted with a red circle and labeled '2'.

Name	Description	Version
modin-core	Speed up your pandas workflows by changing a single line of code	0.14.0
modin-dask	Speed up your pandas workflows by changing a single line of code	0.14.0
modin-ray	Speed up your pandas workflows by changing a single line of code	0.14.0
movingpandas		0.9rc3
<b>pandas</b>	High-performance, easy-to-use data structures and data analysis tools.	<b>1.4.2</b>
pandas-alive		0.2.4
pandas-compat		0.1.1
pandas-datapackage-reader		0.18.0
pandas-datareader	Up to date remote data access for pandas, works for multiple versions of pandas	0.9.0
pandas-dev-flaker		0.5.0
pandas-flavor		0.2.0
pandas-gbq		0.9.0

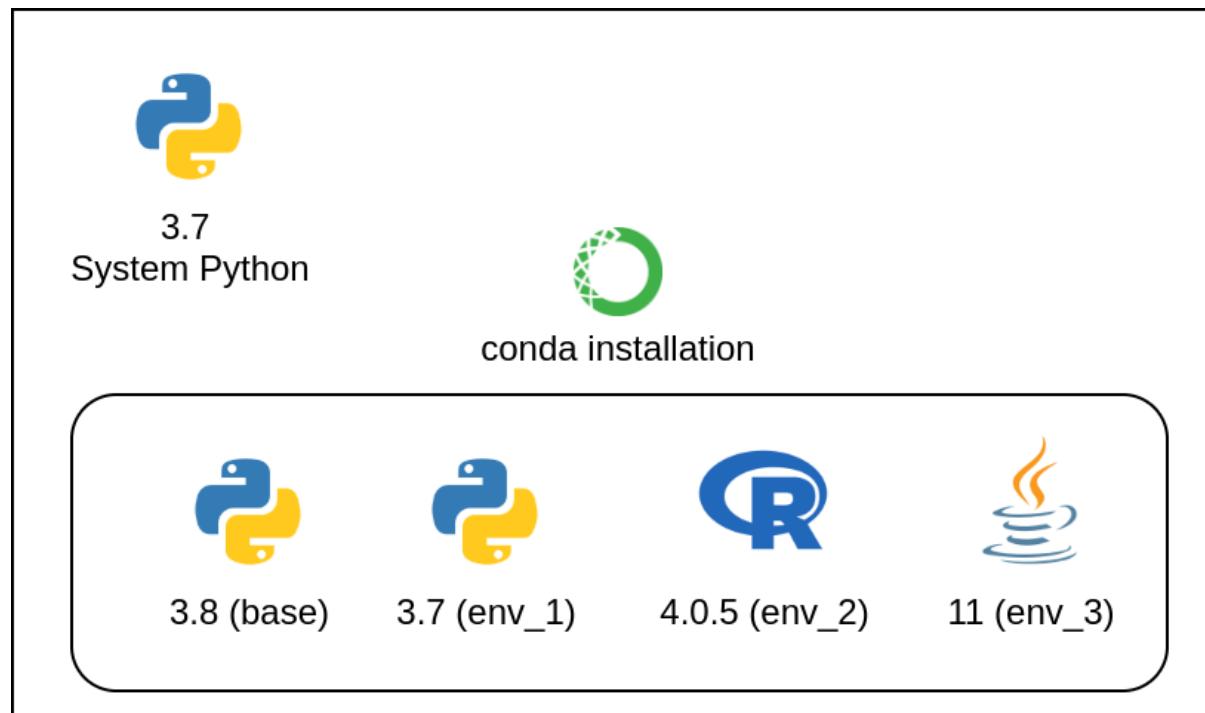
57 packages available matching "pandas" 1 package selected

Apply Clear



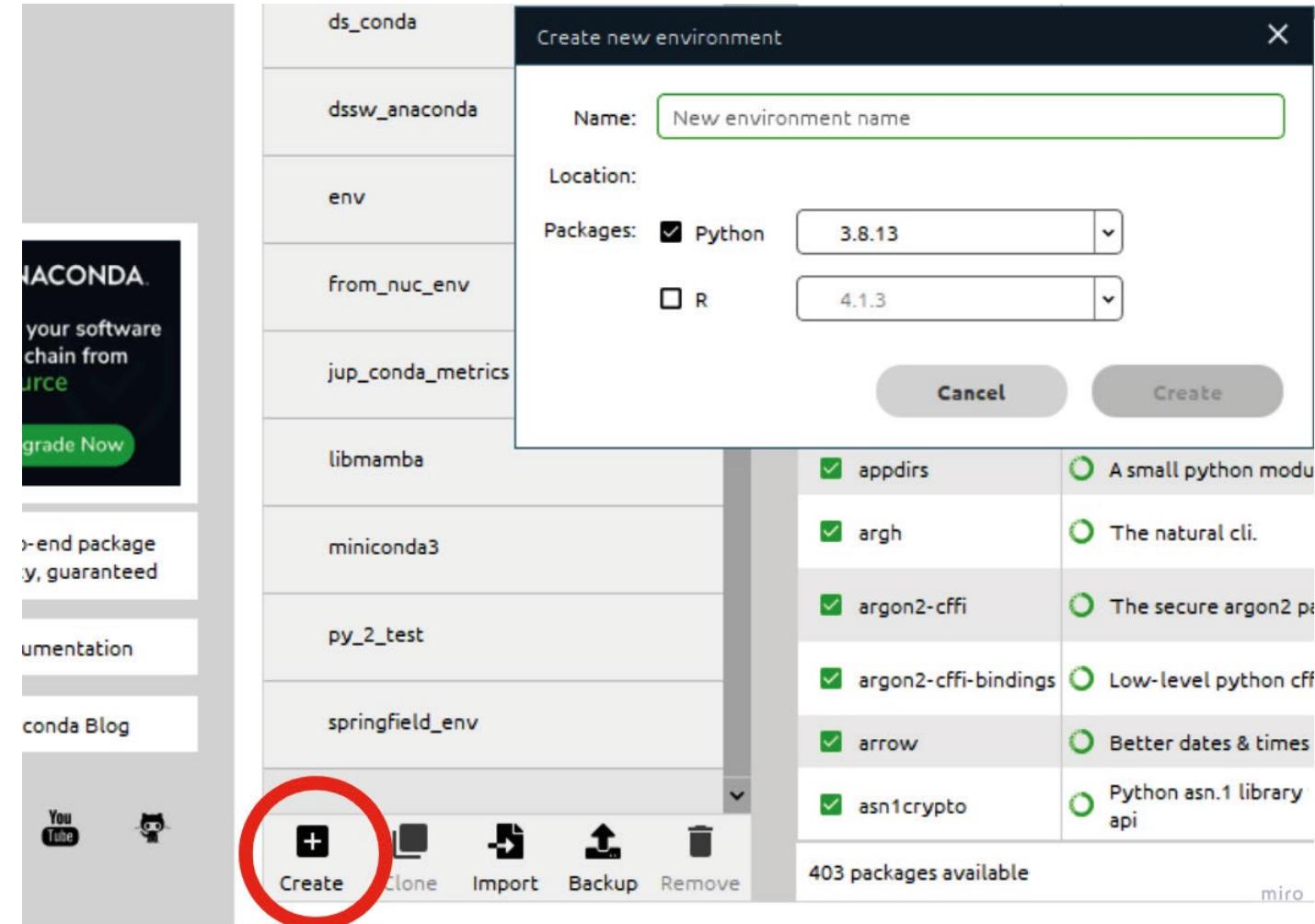
# Virtual Environment management

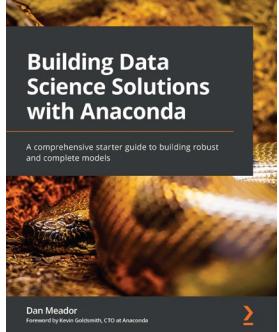
A virtual environment is just an **isolated  
installation of Python and its libraries.**





# Virtual Environment management via the GUI



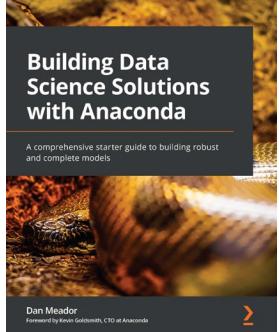


# Virtual Environment management via the Command-line Interface CLI

## Conda environment commands

The environment-specific commands are as follows:

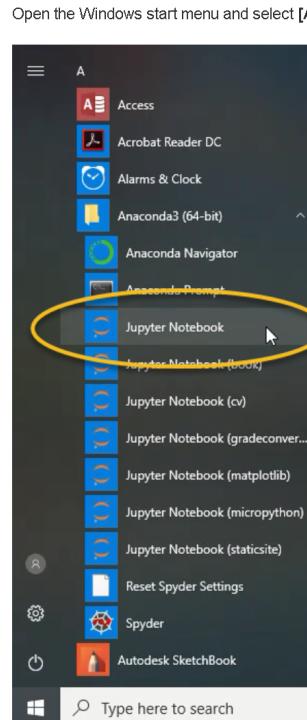
- `conda create -n <ENVIRONMENTname>`: Creates a new conda environment
- `conda activate <ENVIRONMENTname>`: Activates a conda environment
- `conda deactivate`: Deactivates the current conda environment
- `conda env export --file FILENAME`: Exports an environment's `yaml` file to the specific file



# Working with Jupyter Notebooks

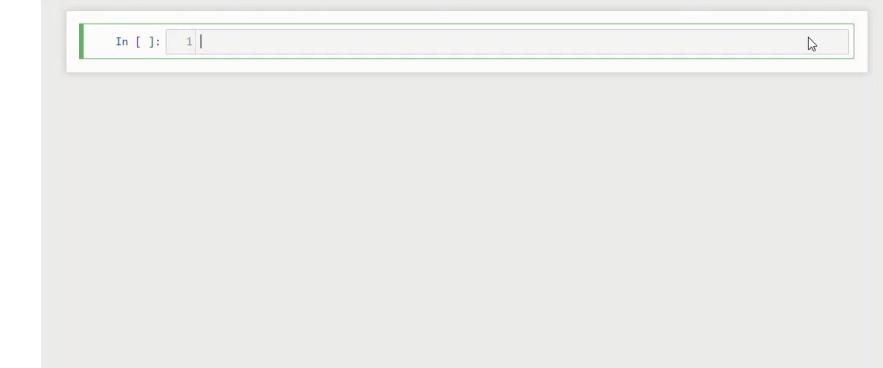
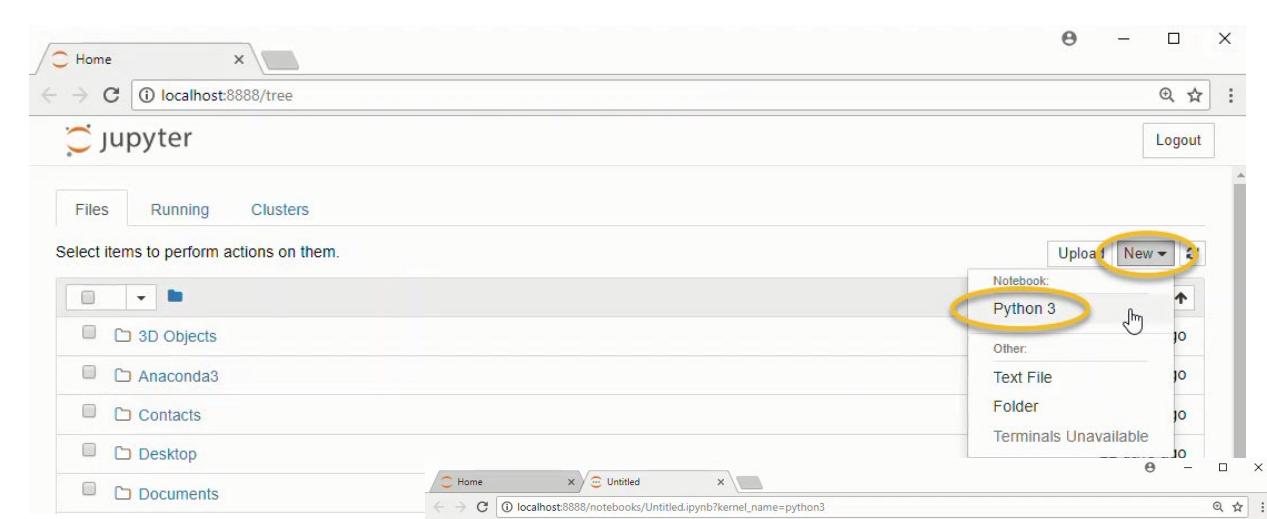
3 ways to open a Jupyter notebook:

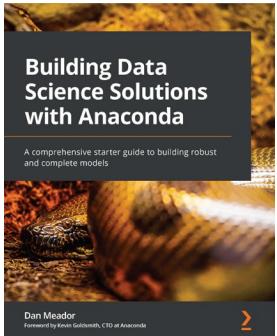
Windows Start Menu  
Anaconda Prompt  
Anaconda Navigator



This action opens the Jupyter file browser in a web browser tab.

In the upper right select [New] -> [Python 3]





# Working with Jupyter Notebooks

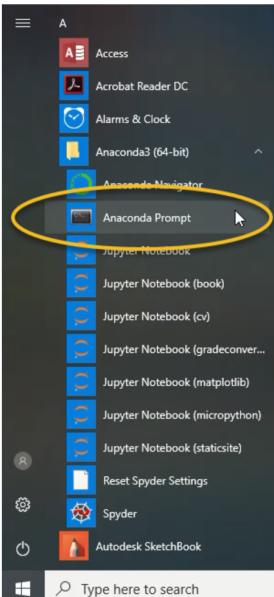
3 ways to open a **Jupyter notebook**:

Windows Start Menu  
Anaconda Prompt  
Anaconda Navigator

## Open a Jupyter Notebook with the Anaconda Prompt

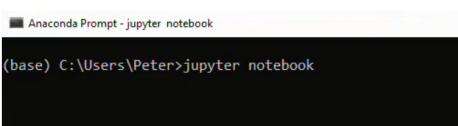
Another method to open a Jupyter notebook is to use the **Anaconda Prompt**.

Go to the Windows start menu and select **[Anaconda Prompt]** under **[Anaconda3]**.



If you don't see the **Anaconda Prompt** in the Windows Start Menu, then you need to install the Anaconda distribution of Python. Download Anaconda at the following link: [Anaconda.com/distribution](https://www.anaconda.com/distribution)

The **Anaconda Prompt** window should look something like the image below.



At the **Anaconda Prompt** type:

```
> jupyter notebook
```

This command starts the **Jupyter notebook** server. The output in the **Anaconda Prompt** will look something like the output shown below:

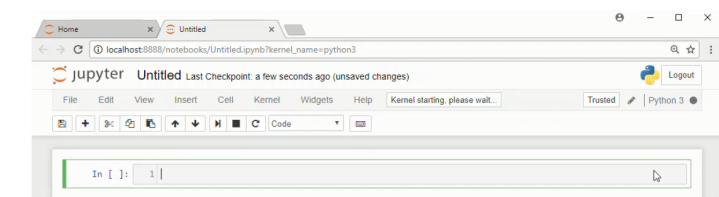
```
Copy/paste this URL into your browser when you connect ...  
to login with a token:  
http://localhost:8888/?token=6bdef677d3503fb2...  
[I 16:14:12.661 NotebookApp] Accepting one-time-token ...
```

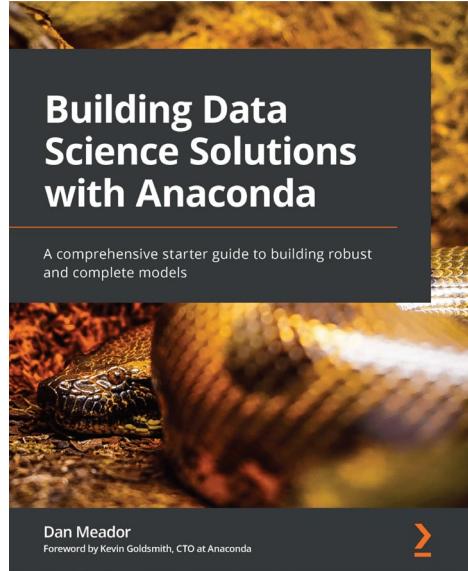
A web browser should open, and you should be able to see the **Jupyter file browser**. If a web browser doesn't open automatically, you can copy the web address from the **Anaconda Prompt** and paste it into a web browser's address bar.



In the upper right select **[New] -> [Python 3]**

You will see a new tab open in your web browser. This web browser page is a **Jupyter notebook**.





Dan Meador  
Foreword by Kevin Goldsmith, CTO at Anaconda

↗

# ANACONDA NAV== GUI

Anaconda Navigator

File Help

**ANACONDA NAVIGATOR**

Sign in to Anaconda Cloud Refresh

Applications on base (root) Channels

Home Environments Learning Community Documentation Developer Blog

**Jupyter Notebook**

0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated

**JupyterLab**

2.1.5 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

**Powershell Prompt**

0.0.1 Run a Powershell terminal with your current environment from Navigator activated

**CMD.exe Prompt**

0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated

**IP[y]: Qt Console**

4.7.5 PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

**Spyder**

4.1.4 Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

**Glueviz**

0.15.2 Multidimensional data visualization across files. Explore relationships within and among related datasets.

**Orange 3**

3.26.0 Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

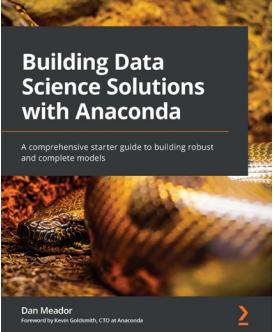
Launch Launch Launch Launch

Launch Launch Install Install

Documentation Developer Blog

Twitter YouTube GitHub

A large red arrow points down to the Jupyter Notebook icon.



# Jupyter Notebook

The screenshot shows the Jupyter Notebook interface. At the top, there is a navigation bar with icons for Home, Help, and Logout. Below the navigation bar, there are tabs for Files, Running (which is selected), and Clusters. A message says 'Select items to perform actions on them.' On the left, there is a file browser showing an 'Untitled Folder' containing an empty folder icon. In the center, it says 'The notebook list is empty.' On the right, there is a context menu with options for Upload, New (with sub-options Notebook, Text File, Folder, Terminal), and a search field. The 'Notebook:' section has 'Python 3' selected, which is highlighted with a red box.

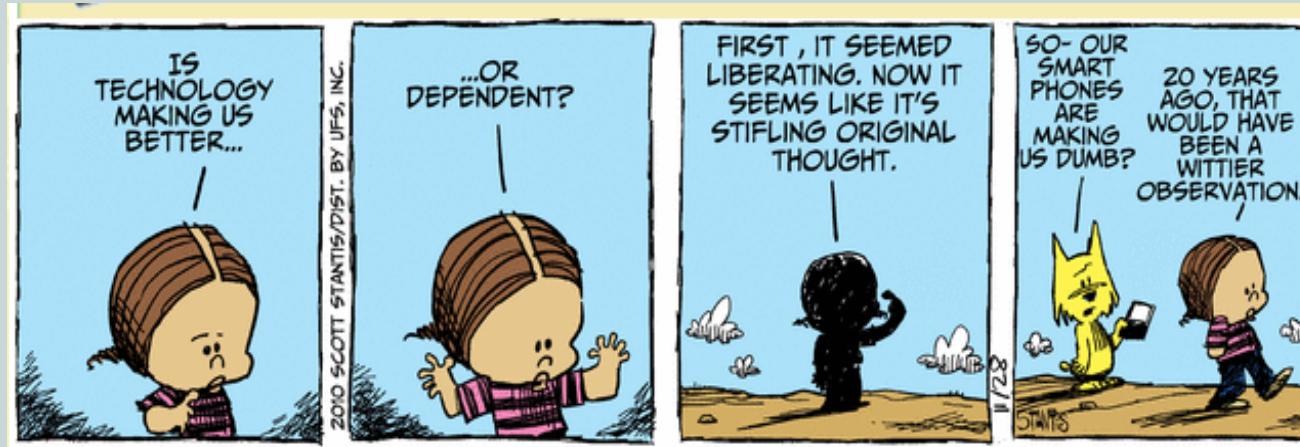
## Starting Jupyter Notebook

A *Jupyter Notebook* lets you combine code, explanatory text, mathematics, plots, and more in a single document. Notebook documents contain the inputs and outputs of your computation, and can serve as a complete computational record of a session, making it easy to share reproducible work.

To start *Jupyter Notebook*, first open *Anaconda Navigator* as described in the previous section, then click the **Launch** button below the *Jupyter Notebook* icon.

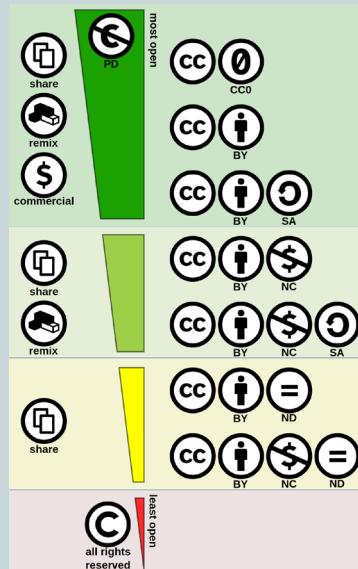
To start using Notebooks, read more about:

- [The Notebook user interface](#)
- [Notebook basics](#)
- [Running code](#)



This lesson was developed by:

RobFvdW  
Data Scientist  
Hogeschool Rotterdam  
Programma voor AI & Ethisiek



<https://creativecommons.org>

This lesson is licensed under a Creative Commons Attribution-Share-Alike license. You can change it, transmit it, show it to other people. Just always give credit to RFvdW.

[https://en.wikipedia.org/wiki/Creative\\_Commons\\_license](https://en.wikipedia.org/wiki/Creative_Commons_license)

