



Deloitte
Data Science
Academy

Lesson 1: Data Science Intro & Python 101

Data Science Introduction – use cases and methods overview, Data Science Toolkit

Python programming – variables, data types and objects, operators, classes, flow control, libraries, IDEs, strings and functions, etc.

Data exploration with „pure“ python

18 / 08 / 2020



Expectations?

- What are your expectations?
- What do you want to learn, why did you choose this course?
- What is your background, how familiar are you with SQL, Programming, Data Science?
- What tools do you use on daily basis?



Goal

Move from zero level to intermediate level in data science.

Learn Python

Use Python and its data science tools

Develop segmentation and classification models

Effectively visualize data in Python

How to present results to business

How to evaluate models

Deloitte
Data Science
Academy

Lecturers' introduction



Pavel Milička

AI Specialist Lead

- Master degree in Artificial Intelligence from Czech Technical University
- Co-author of 3 research papers – Bio-inspired hexapod control, Bilevel optimization
- Model for predicting performance based on CVs.
- Expedition optimization for major steel manufacturer.
- Technical owner of Deloitte Dynamic Pricing solution.
- Leading project for Debt Collection Optimization for foreign Health Insurer.
- Collaborating on automated open source intelligence system.
- Creator of internal Task Mining solution from activity logs.

Content

1	Introduction to Data Science, Data Science Tools, Python as a programming language, Python Basics, Git – code collaboration, Pure Python Data Exploration	HA1
2	Features and, Clustering, behavioral segmentation, Hands on training for segmentation with Python	HA2
3	Propensity models, single factor analysis, binning, binary classifiers, Hands on training for data exploration, classification and regression	HA3
4	Introduction to NLP and Computer Vision, Selected topics	HA4

Format

Overview

4 **Days**
During which we go through
Data Science, Python, NLP, and much
more

6 **Hours**
Every lecture has 6 hours.

4 **Home assignments**
You need **all 4 home assignments**
accepted to qualify for a final
certificate.



**Calls and emails outside the
room, PC for hands-on
exercises only.**

Major theme

During the course, we will work with the open-source Inside Airbnb Prague dataset (<http://insideairbnb.com/>) along others, containing information about listed accommodations, hosts and reviews. We will develop a **1) segmentation of the AirBnB hosts** with the aim to better understand the host types and motivation, a **2) a regression model of listings**, so we can estimate a suitable price if we were to buy a flat for AirBnB rental purposes. Apart from the core airbnb dataset we will work with simpler processed datasets provided by the scikit package to demonstrate some of the machine learning algorithms in a simple manner.

We will work it end to end from data crunching, calculating predictors, target definition and feature engineering to modeling and final visualizations.

It is possible to come up with your own problems if the scale of the problem is comparable.

Today's program

1	Introduction to Data Science, Data Science Tools, Keboola introduction, Python basics, Git – Code collaboration, Pure Python data exploration	HA1
2	Features and dataset preparation, Clustering, Behavioral segmentation, Hands on training for data exploration and clustering with Python	HA2
3	Regression and Classification, Hands on training for data exploration, classification and regression	HA3
4	Introduction to NLP and Computer Vision, AutoML	HA4

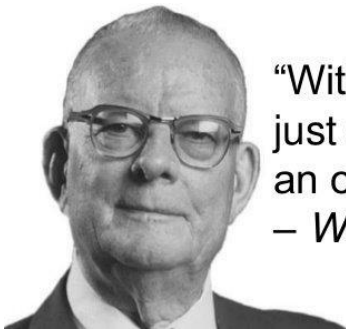


Data Science

Quotes on Data Science

Data is the new oil.
Clive Humby

Data is the new oil?
No: Data is the new
soil. David
McCandless



“Without data you’re
just another person with
an opinion.”
– *W. Edwards Deming*

Information is the oil of
the 21st century, and
analytics is the
combustion engine.
Peter Sondergaard, SVP,
Gartner Research



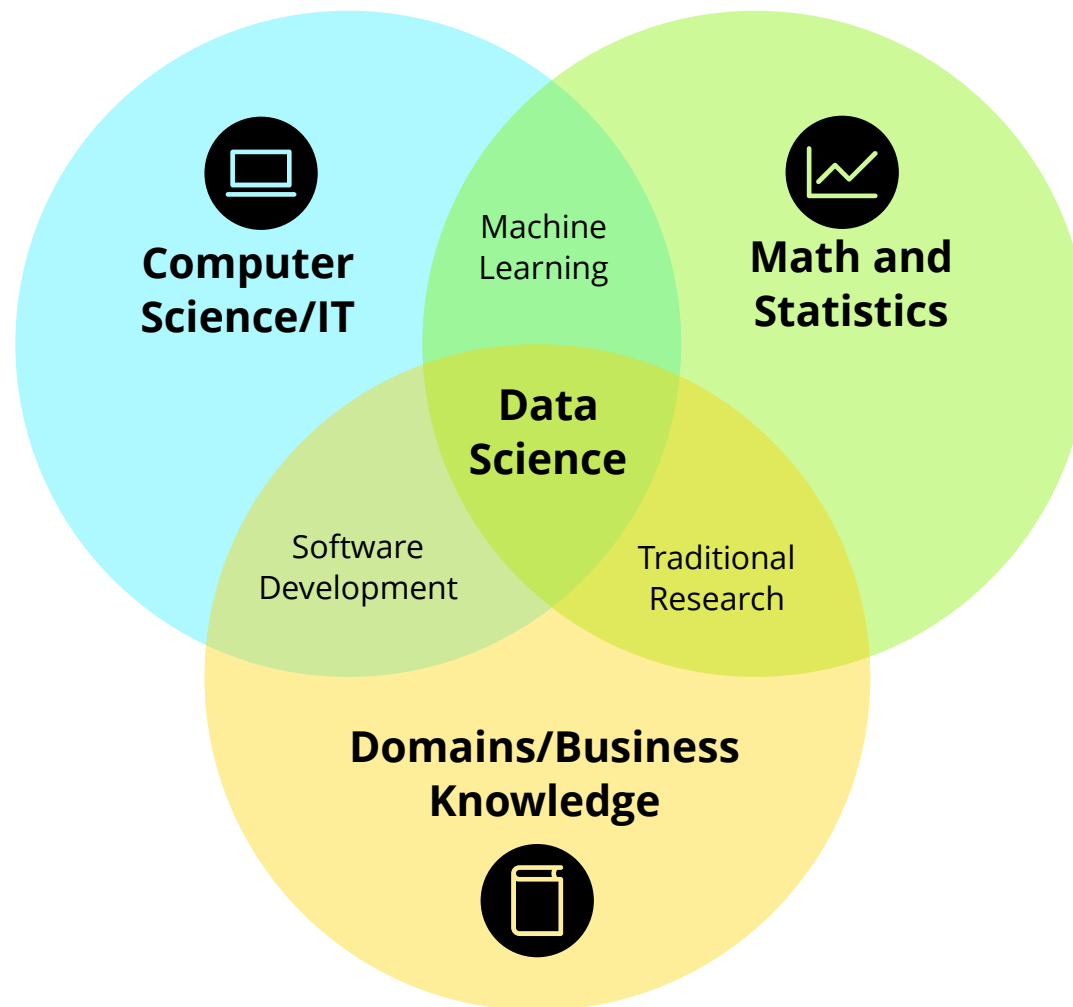
The world is
one big data
problem.
Andrew
McAfee

What is Data Science

Data science is the extensive use of **data**, statistical and quantitative **analysis**, exploratory, predictive models, and fact based management to drive decisions and **actions**.

Data scientists come with flavours.

- Data Scientist
- ML Engineer
- Business Analyst
- ...



What is Data Science

Foresight

Understand signals to shape the future

Foresight

Prescriptive

Optimization

Predictive

Predictive modeling

Statistical analysis

Insight

Use data to drive changes here and now

Insight

Descriptive statistics

Exceptions and alerts

Hindsight

Conduct "rear-view mirror" assessments

Hindsight

Descriptive

Slice and dice, drill downs

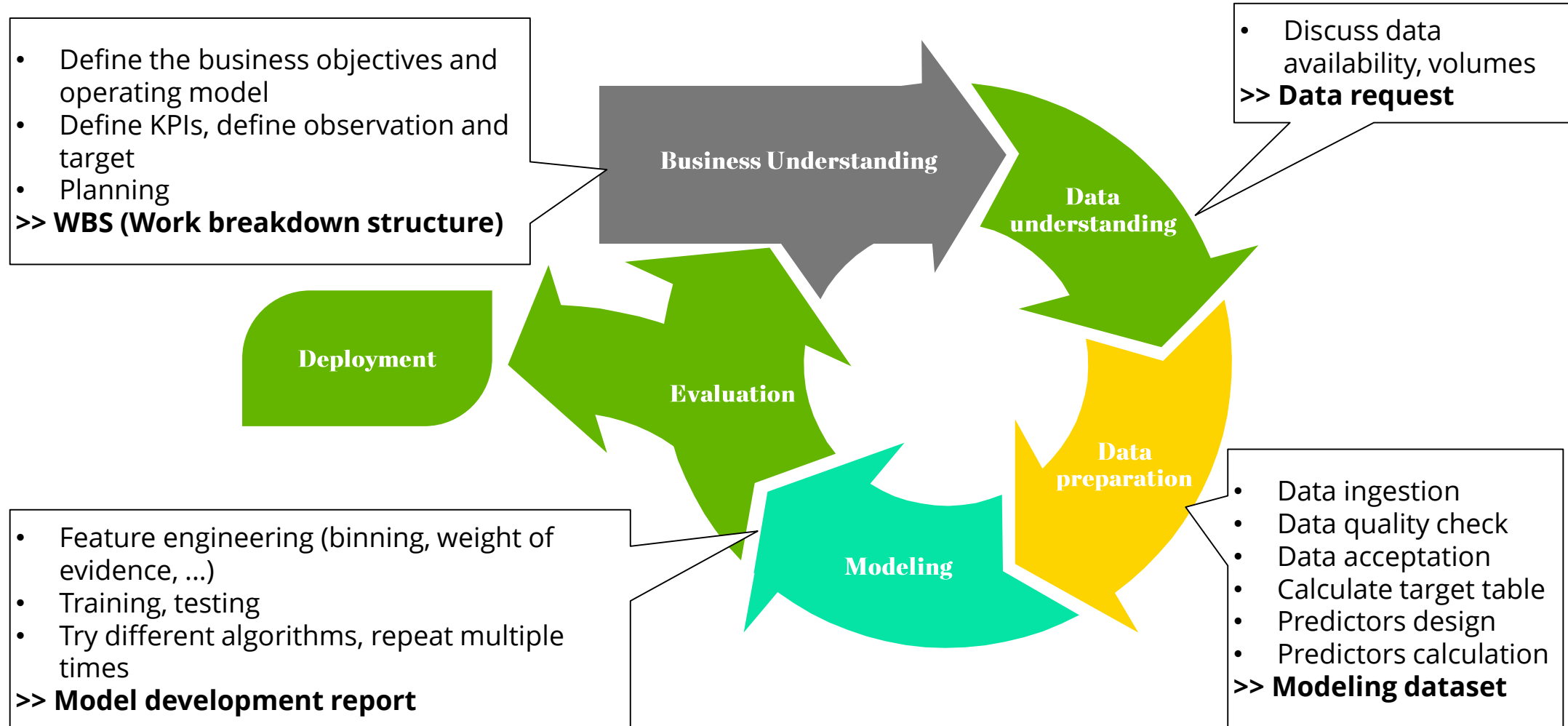
Management reporting

Enterprise data management

Value

CRISP-DM Framework

Applicable to all Data Science jobs



Data Science:

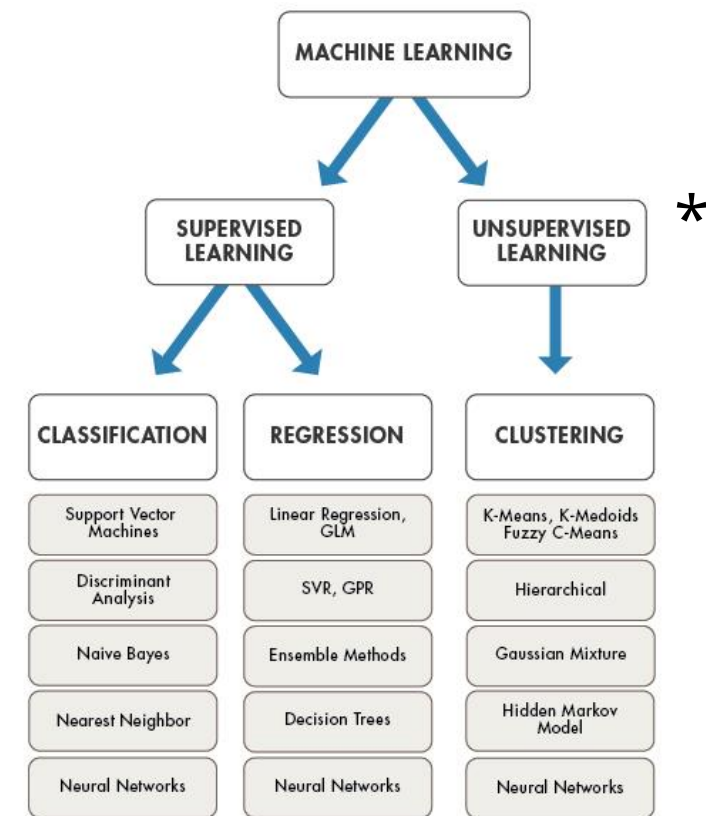
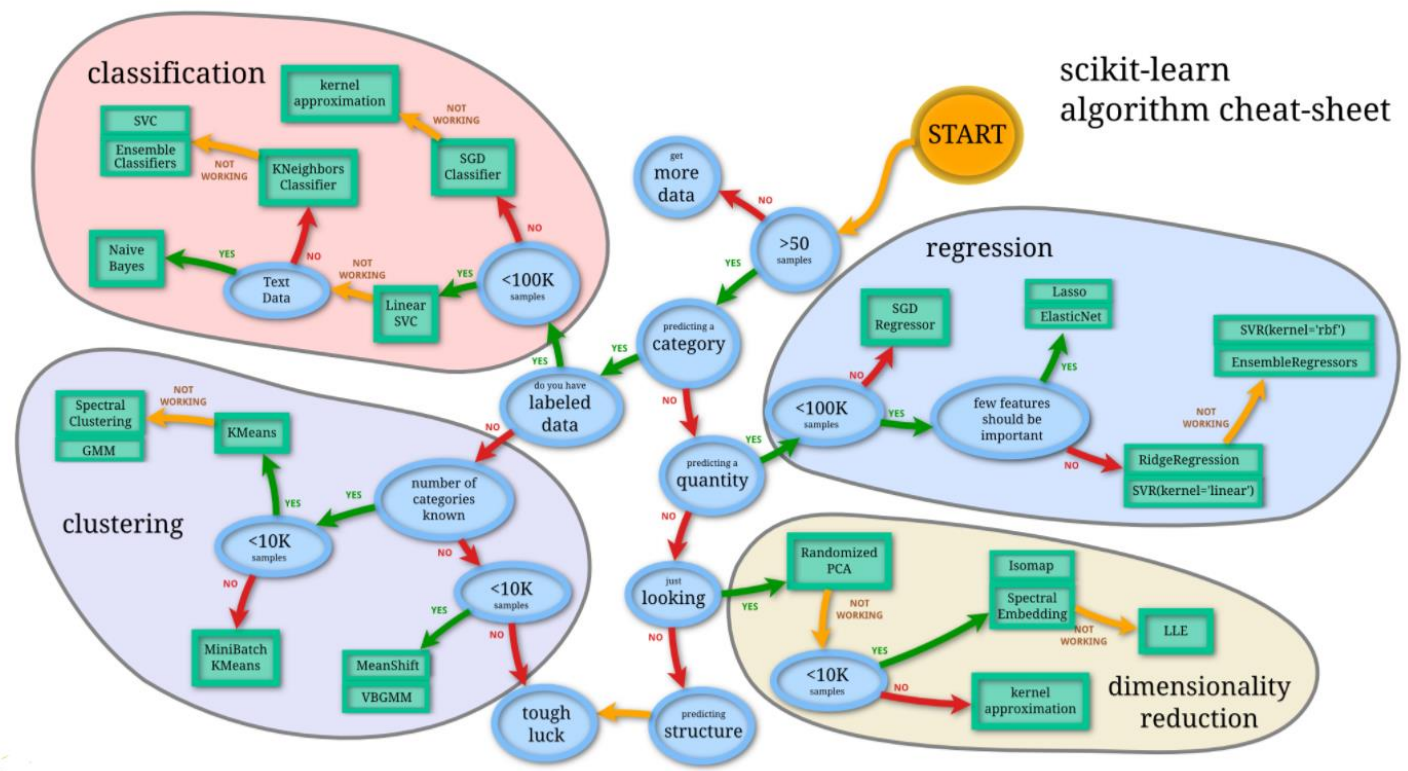
Tools, use cases,
algorithms



Data Science use cases

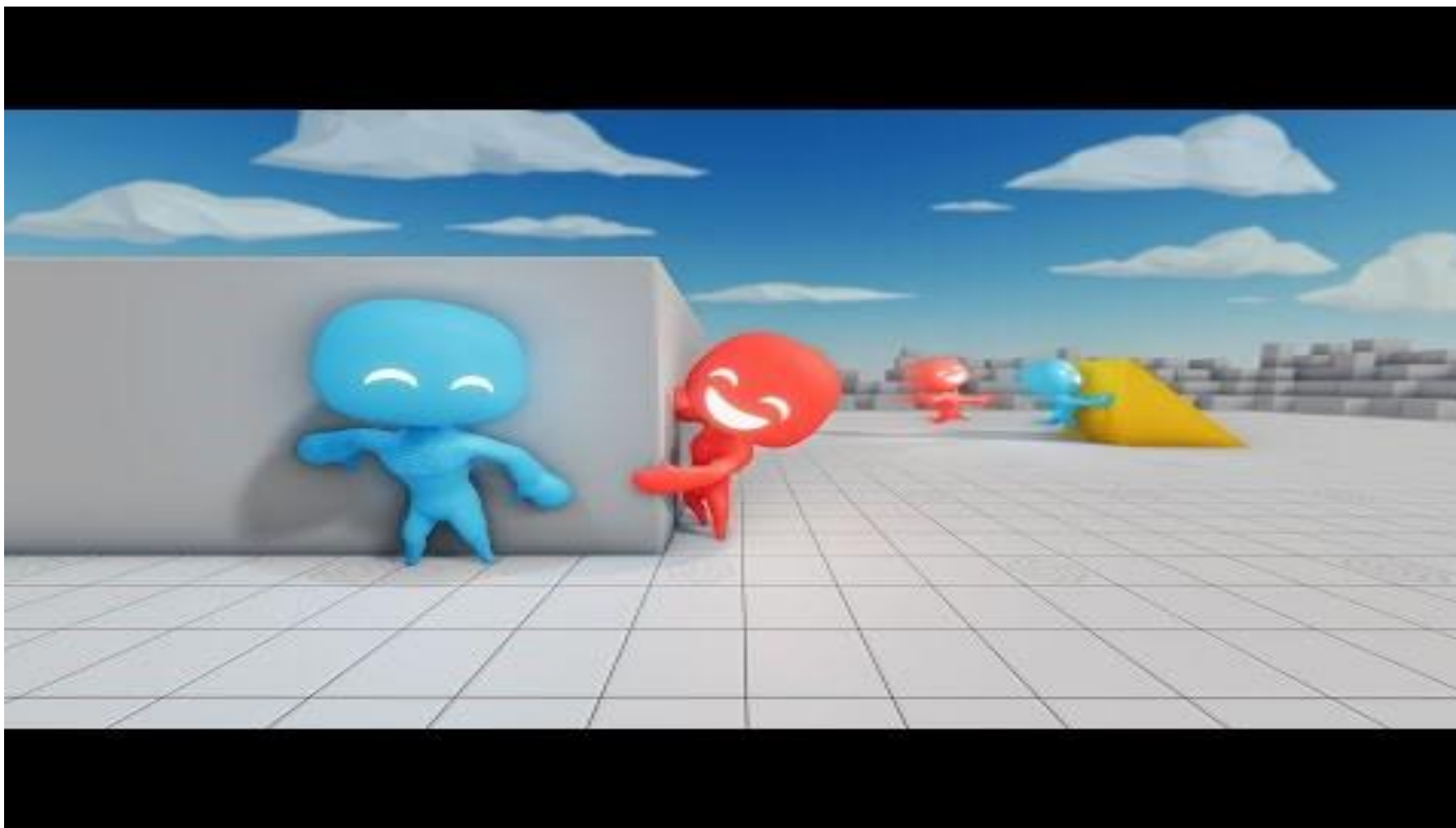
Need	Acquisition		Cross sell	Retention	Pricing	Supply chain	Risk
Solution	Building data science capability		Customer targeting	Retention	HR analytics	Dynamic pricing	Fraud
Product	Propensity to Save	Campaign Optimization	Customer Lifetime Value	Demand prediction	Energy prediction	Credit Risk Scorecard	Predictive IT Operations
	Segmentation	Propensity to Buy		Predictor Factory		Predictive maintenance	
	Next Best Offer		Propensity to Churn	Customer Wallet	Reporting		
Algorithm	Clustering	Binary classification	Mathematical Programming	Regression	Survival Analysis	Distributed Machine Learning	Gradient boosting
	Sequential Pattern Recognition	Decision Trees	Markov Chains	Social Network Analysis	Natural Language processing	Voice To Text	Multi armed bandit

Algorithms overview

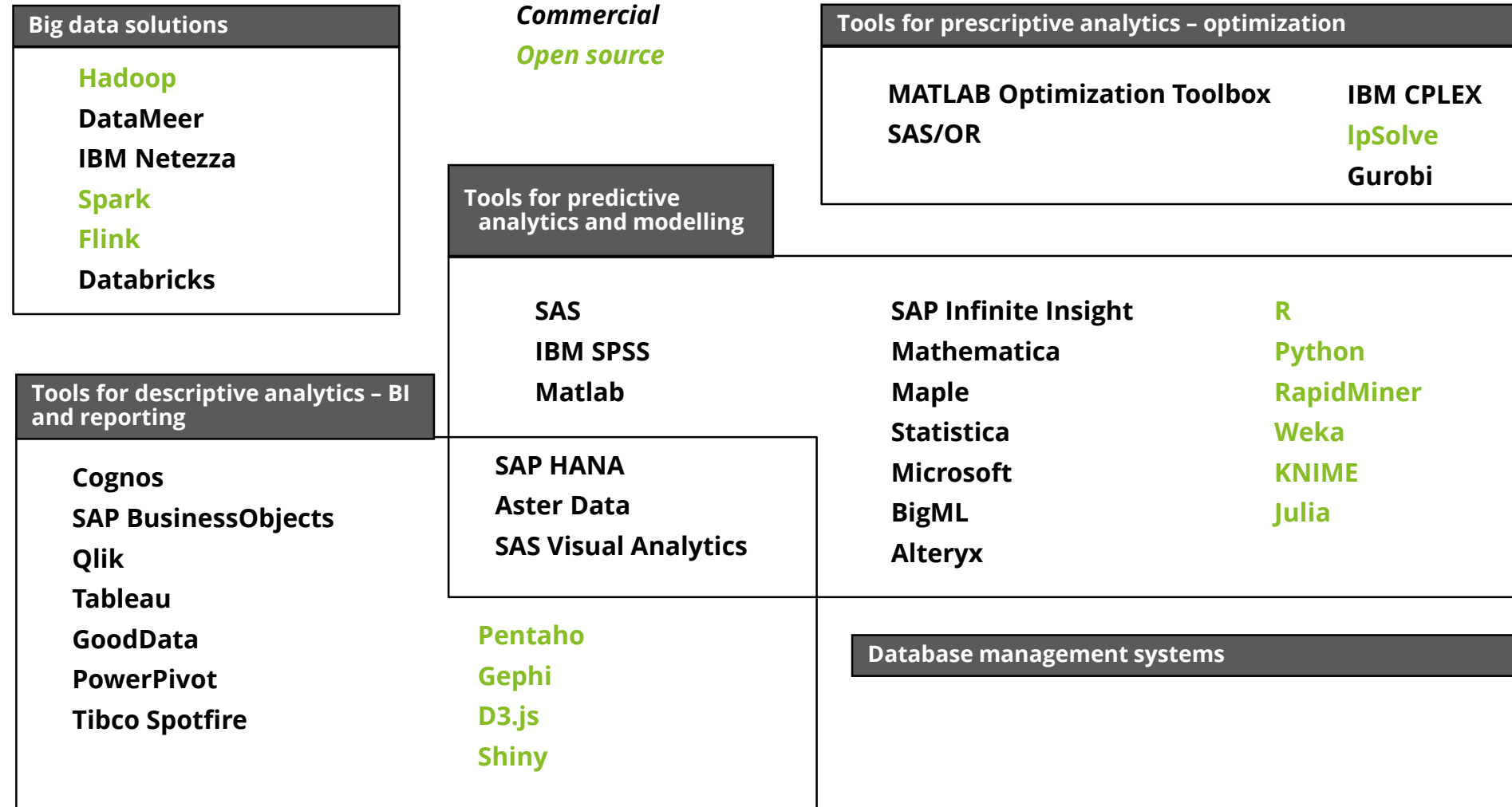


* **Reinforcement** (learning from mistakes/rewards) is often mentioned as a third type of ML.

Reinforcement learning

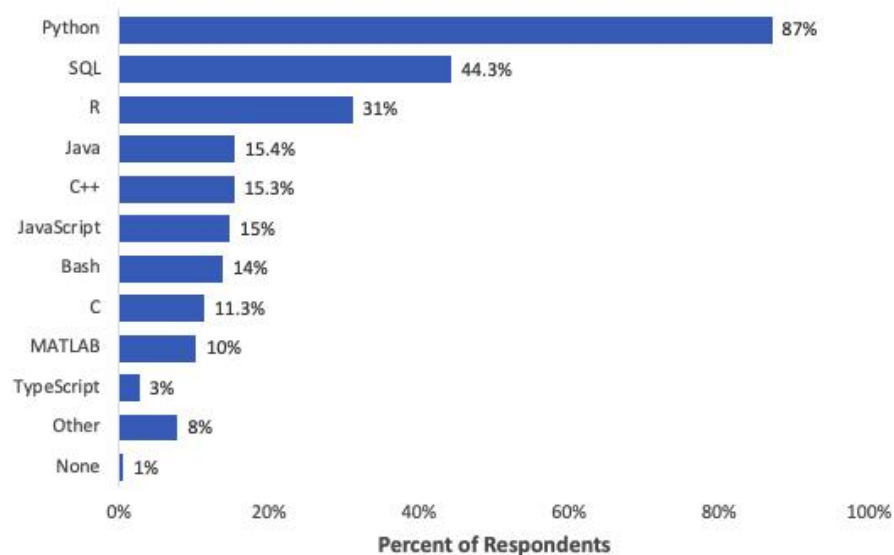


Tools for Data Science



Most popular tools

What programming languages do you use on a regular basis?

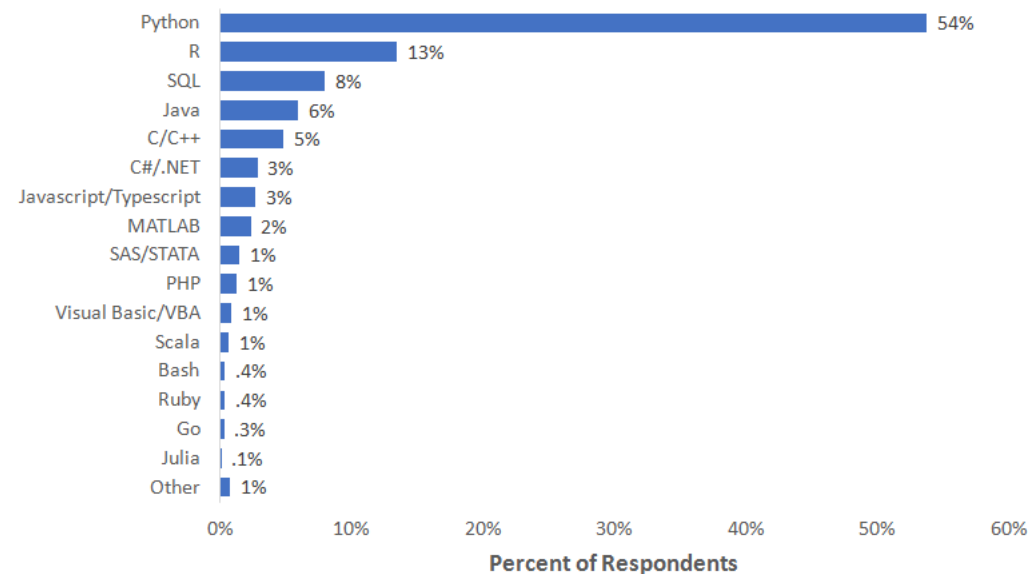


Note: Data are from the 2019 Kaggle ML and Data Science Survey. You can learn more about the study here: <https://www.kaggle.com/c/kaggle-survey-2019>. A total of 19717 respondents completed the survey; the percentages in the graph are based on a total of 14762 respondents who provided an answer to this question.



Copyright 2020 Business Over Broadway

What specific programming language do you use most often?



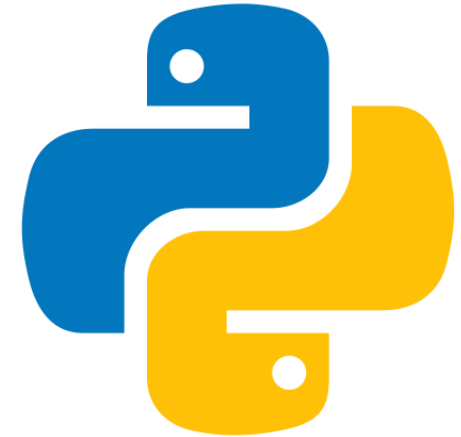
Note: Data are from the 2018 Kaggle ML and Data Science Survey. You can learn more about the study here: <http://www.kaggle.com/kaggle/kaggle-survey-2018>. A total of 23859 respondents completed the survey; the percentages in the graph are based on a total of 15222 respondents who provided an answer to this question.



Copyright 2019 Business Over Broadway

Python

"Python is powerful... and fast; plays well with others; runs everywhere; is friendly & easy to learn; is Open." **python.org**

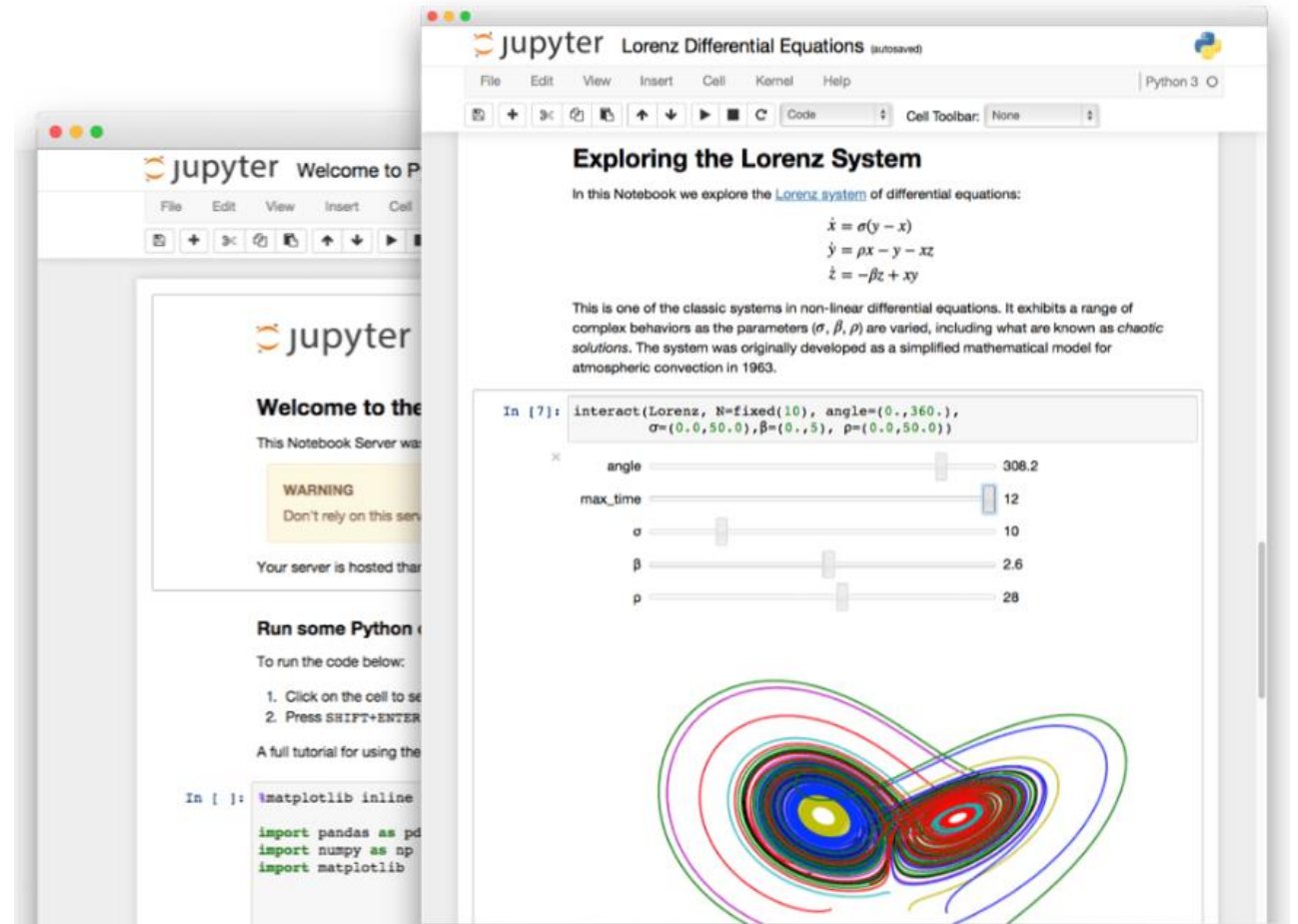


- High level programming language with strong support for data science. Created in 1991. Philosophy of code readability. Open source.
- Dynamic typing, automatic memory management. Procedural, object oriented, functional.
- IDEs (Spyder, PyCharm, VSCode, ...)
- Large community

IDEs - Jupyter notebooks / Jupyter lab

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.

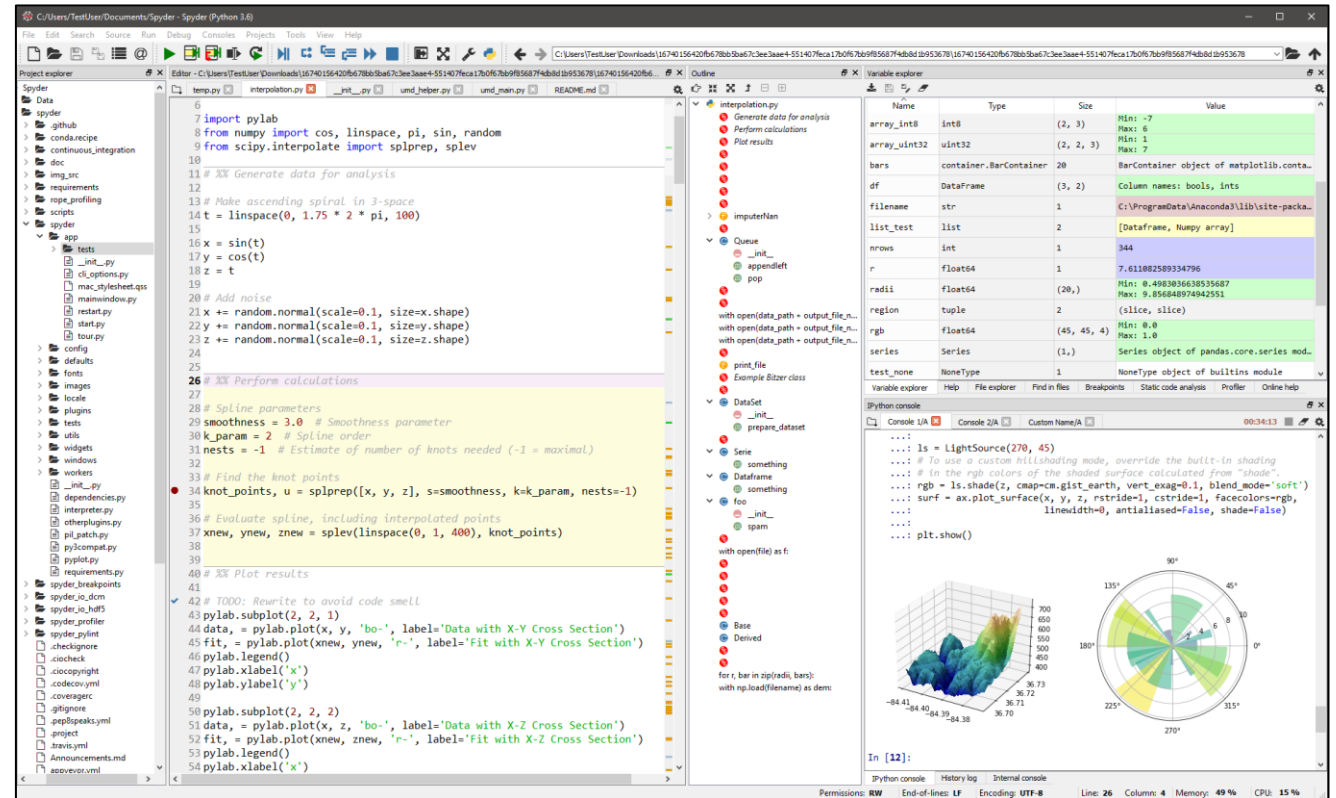
<https://jupyter.org/>



<https://jupyter.org/>

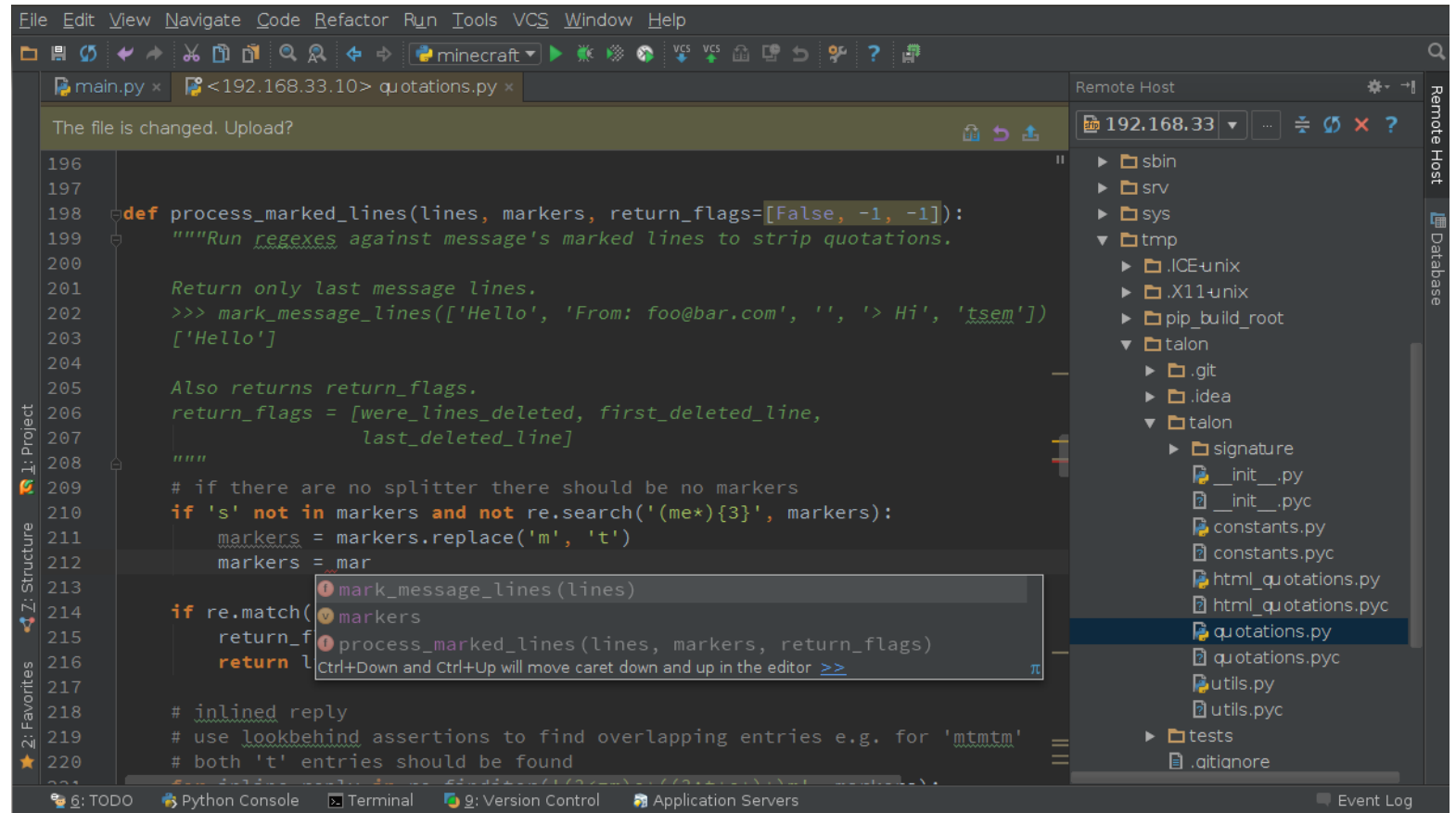
IDEs – Spyder

- Scientific Python IDE
- Similar to RStudio
- Loved by R & Python bilingual data scientists
- Part of Anaconda distribution
- Version 4 released in 2020 bringing in many new features



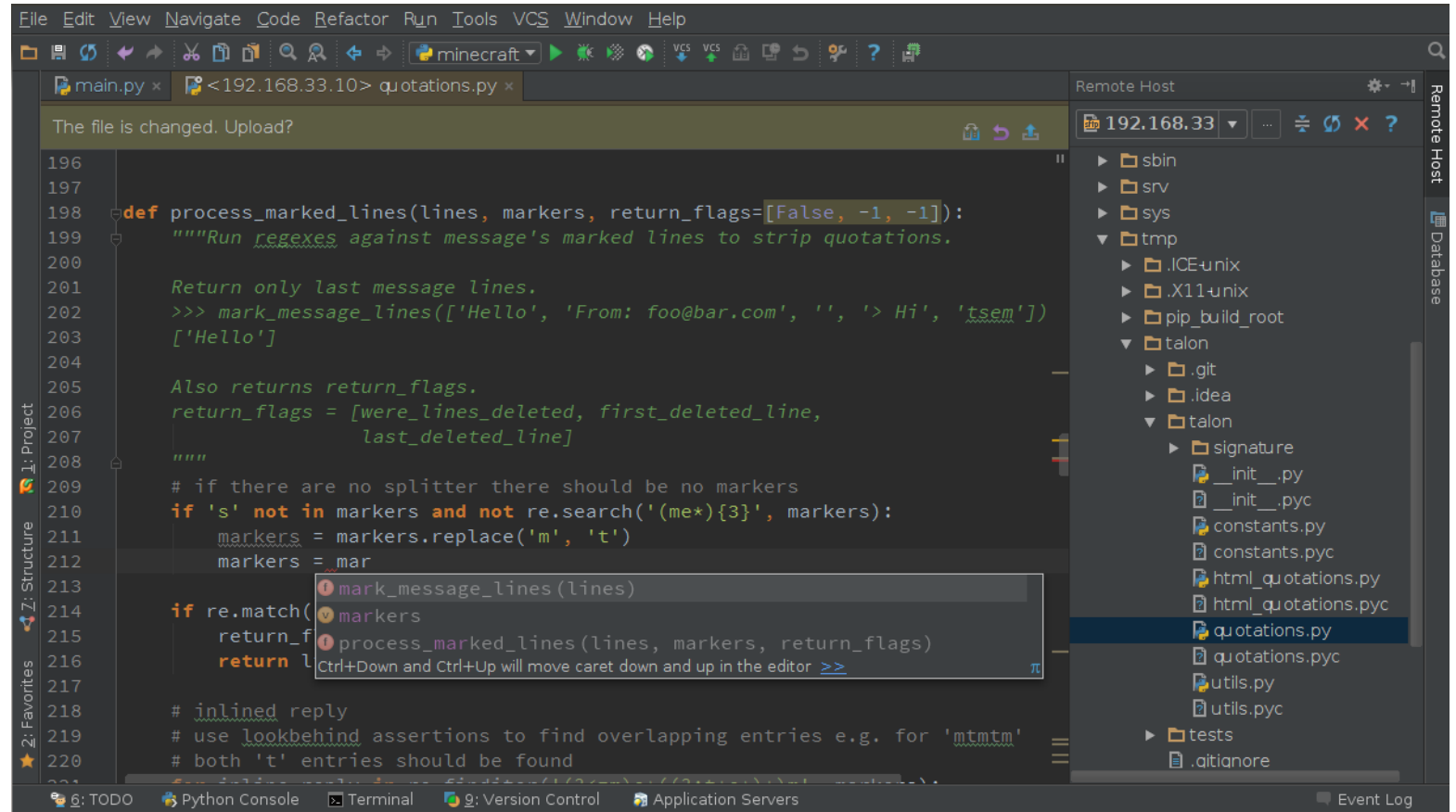
IDEs – PyCharm

- Rich IDE made by JetBrains (Czech company)
- Free and Professional edition
- Mainly focused for software developers (but it has a scientific mode similar to Spyder)



IDEs – VSCode

- Open Source
- Versatile, options are limitless as you can download extensions for remote development, notebook display, etc.
- Git integration



Python core Data Science packages

- Python can be augmented by using packages
- On 2020-30-07, the PyPI package repository features 253,532 packages
- Packages for everything – Web development (Django, Flask, ...), Natural Language Processing (gensim, spacy, nltk, ...), Web scraping (scrapy, selenium, BeautifulSoup...) etc.

NumPy 




pandas

matplotlib 

seaborn

R, RStudio and packages

- R is a language and environment for statistical computing and graphics.
- Shiny package (rapid dashboard development (suitable for web - <https://shiny.rstudio.com/>))
- Superior at specific statistical tasks
- IDEs (RStudio, Jupyter, ...)
- Smaller community than Python



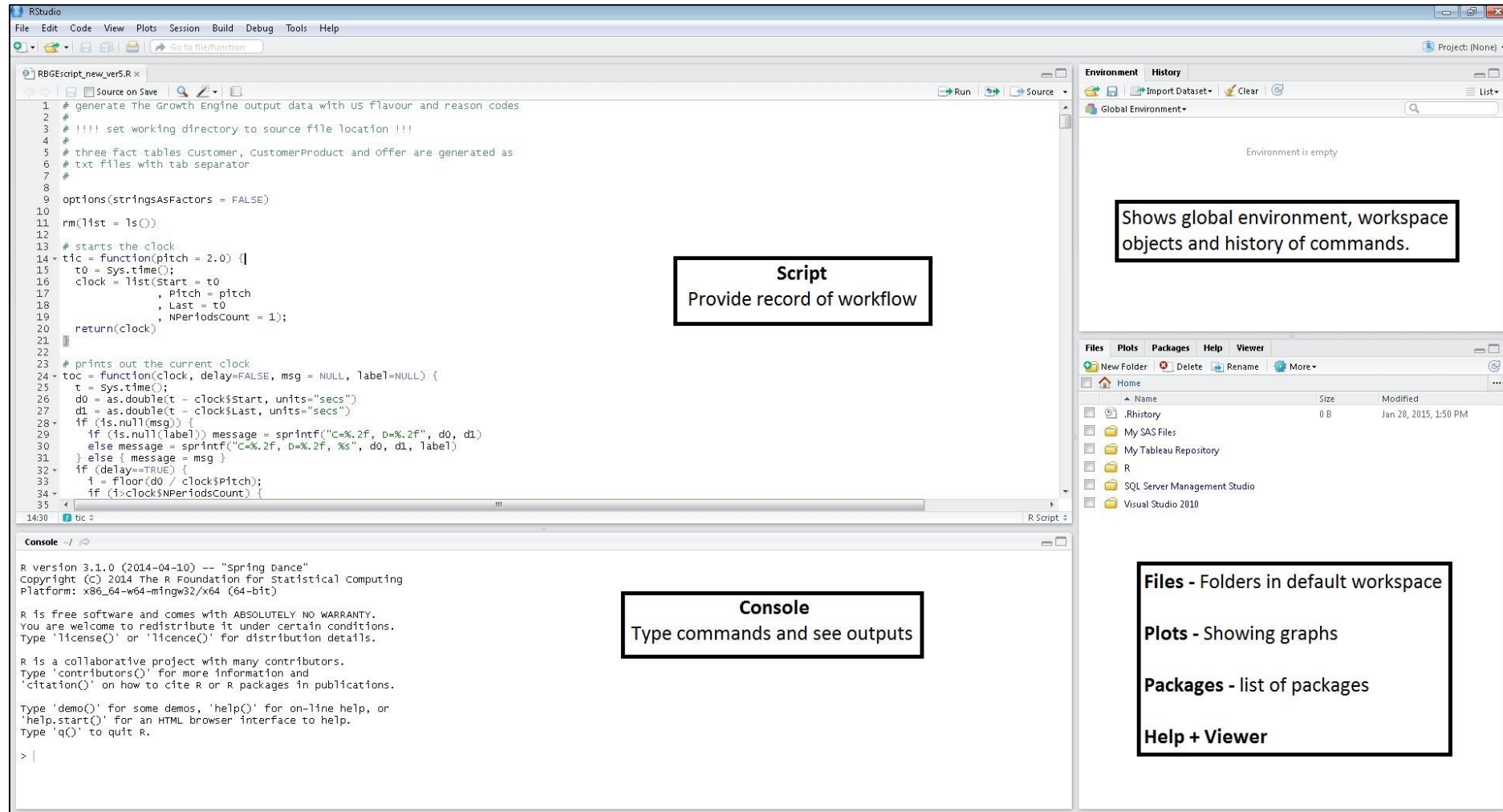
```
set.seed(653) # Set seed in order to provide reproducibility

# Create example data
N <- 10000 # Sample size of 10000
y <- rnorm(N) # y without any missing values
x <- 0.5 * y + rnorm(N) # x correlated with y

# Create missings according to the MCAR response mechanism
MCAR_missings <- rbinom(N, 1, 0.25) == 1 # 25% of Y are set to mis

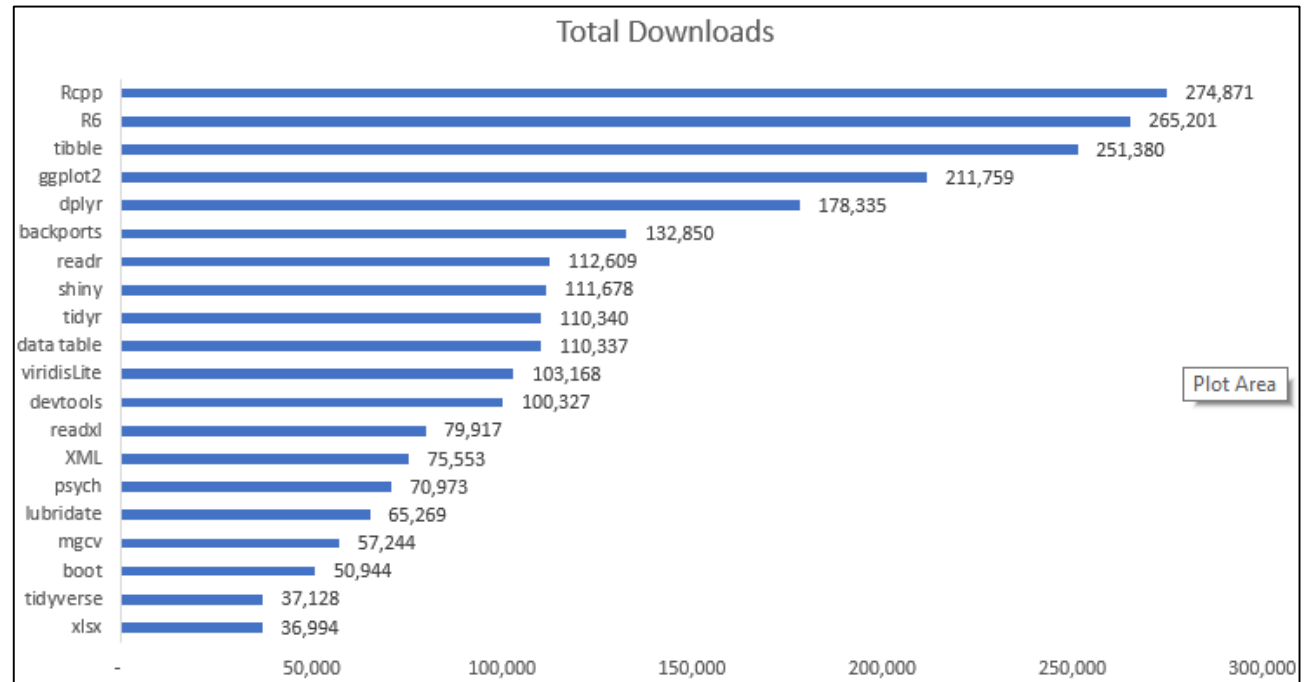
# Missing values according to the MAR response mechanism
x_normalized <- (x - min(x)) / (max(x) - min(x)) # Normalize x to
x_normalized <- x_normalized^2 # x_normalized to the power of 2 in
MAR_missings <- rbinom(N, 1, x_normalized) == 1 # Use x_normalized
```

RStudio



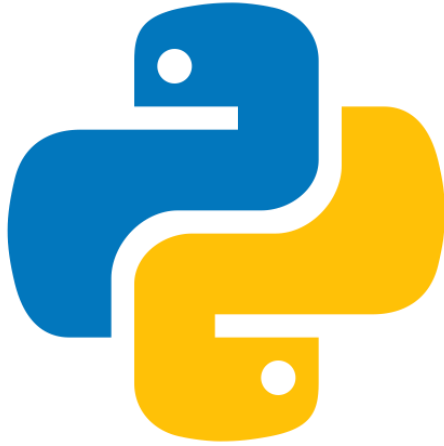
R packages

- Packages extend the functionalities of R
- As of 2017-12-18, the CRAN package repository features 12003 available packages
- Everybody can write and publish its own package



Source www.kdnuggets.com

Python x R



x



Better in programming
Faster execution
pandas, matplotlib, django
More popular, growing faster
Larger community

Better in statistics
dplyr, ggplot, shiny
Shiny dashboard package!

Anaconda and Python environments

- Installing your own python environment
 - <https://www.anaconda.com/>
- Conda environments
 - `conda env list`
 - `conda create -n=<environment_name> python=3.8`
 - `conda activate <environment_name>`
- Package management
 - `pip install <package_name>`
 - `pip freeze`
 - `(conda install <package_name>)`

Python Basics

The background of the slide is a dark blue field filled with a complex network of glowing blue nodes and connecting lines. The nodes vary in size and brightness, with some appearing as large, bright white-blue spheres and others as smaller, dimmer points. The lines connecting them are thin and light blue, creating a web-like structure that suggests a digital or technological theme.

Python Basics hands-on

1. Open the D1_Python_101.ipynb notebook.



Version control systems – Git

Which file is the current one?

- 📄 Data_quality_control_script.sql
- 📄 Data_quality_control_script_Jan_new.sql
- 📄 Data_quality_control_script_latest.sql
- 📄 Data_quality_control_script_latest_adjusted_KP.sql
- 📄 Data_quality_control_script_old.sql
- 📄 Data_quality_control_script_v2_new.sql
- 📄 Data_quality_control_script_version1.sql
- 📄 Data_quality_control_script_version1_old.sql

Version control systems – Introduction

A version control system, or VCS, tracks the history of changes as people and teams collaborate on projects together. As the project evolves, teams can run tests, fix bugs, and contribute new code with the confidence that any version can be recovered at any time. Developers can review project history to find out:

- Which changes were made?
- Who made the changes?
- When were the changes made?
- Why were changes needed?

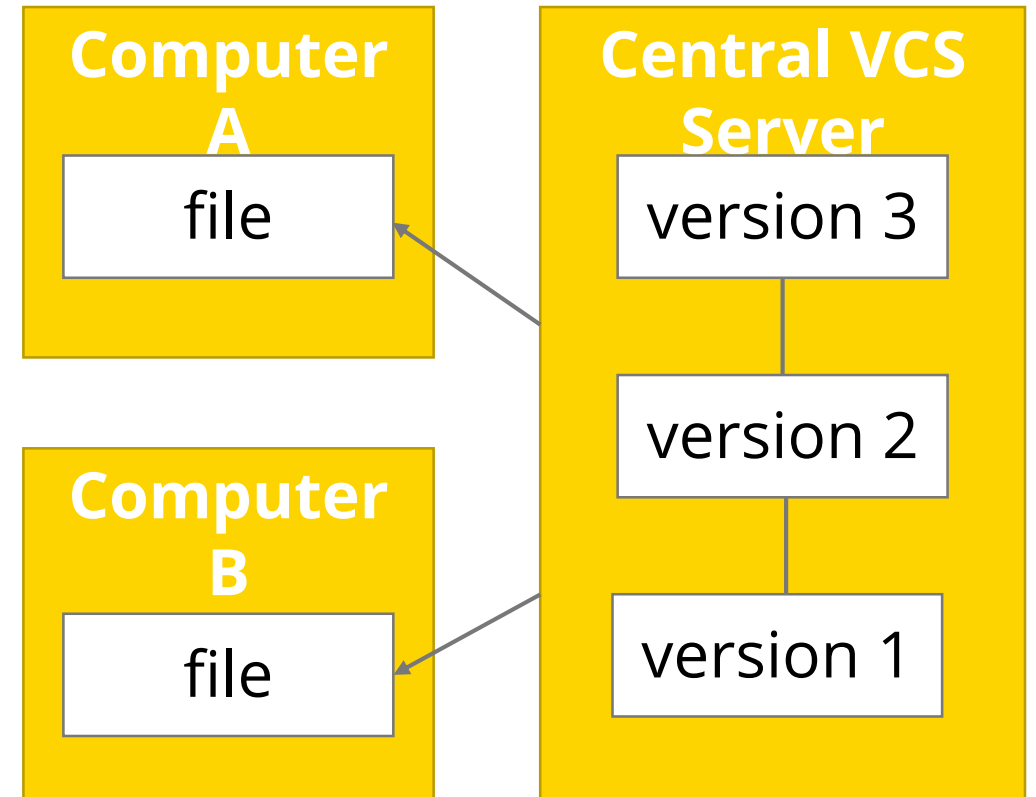
Version control systems – benefits

- Protects source code from both catastrophe and the casual degradation of human error and unintended consequences.
- Allows work parallelization
- Helps organize changes and repairs in the code and document why the changes were made
- No need for file suffixing, all changes are backed by commit messages stating what was done and why.

Most common version control systems are **git** and **SVN**.

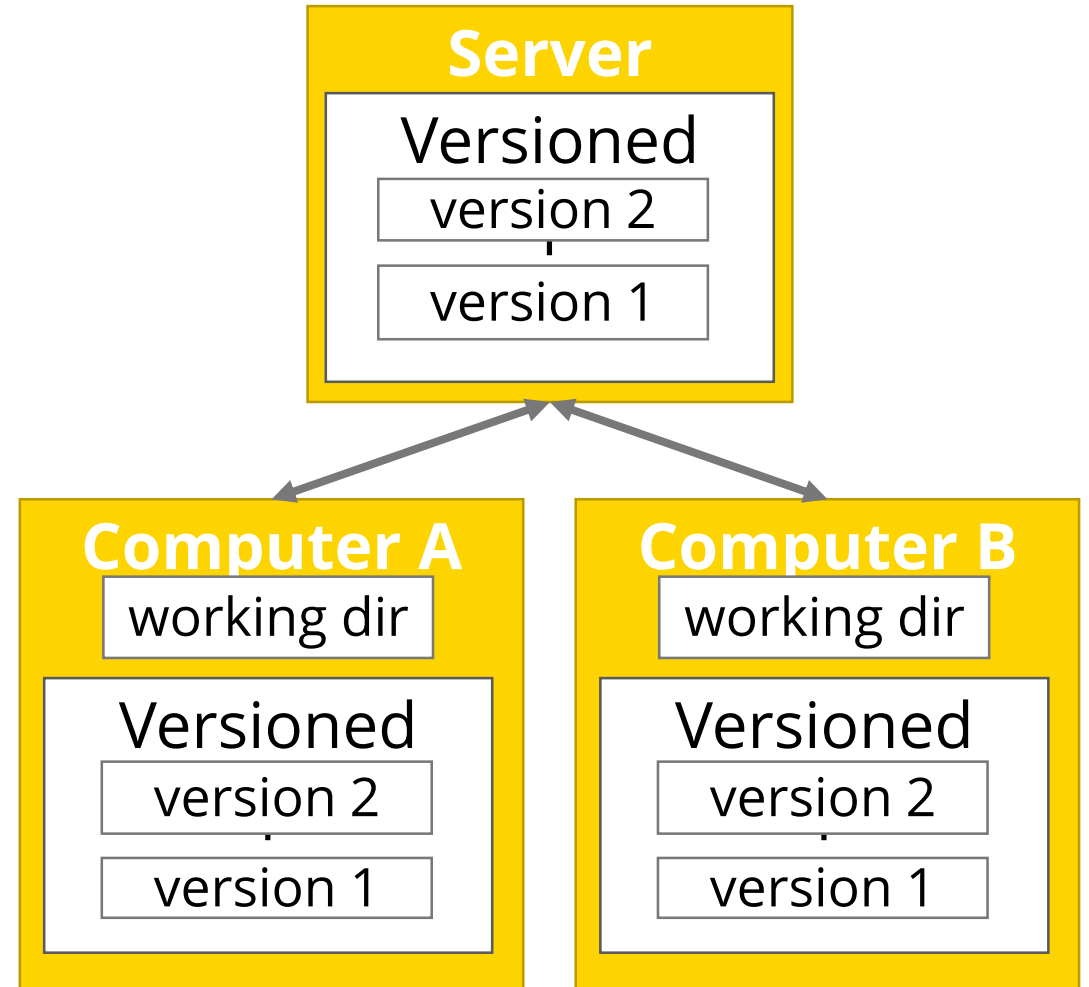
SVN – Subversion

- Example of **centralized** version control system
- Server holds the official copy of the code and files along with the whole history of the files and versioning
- Developers make **checkouts** from the server to their local environments, their local changes are not versioned
- When the changes are done, the files are **checked-in** back to the centralized server, check-ins increment the repository version

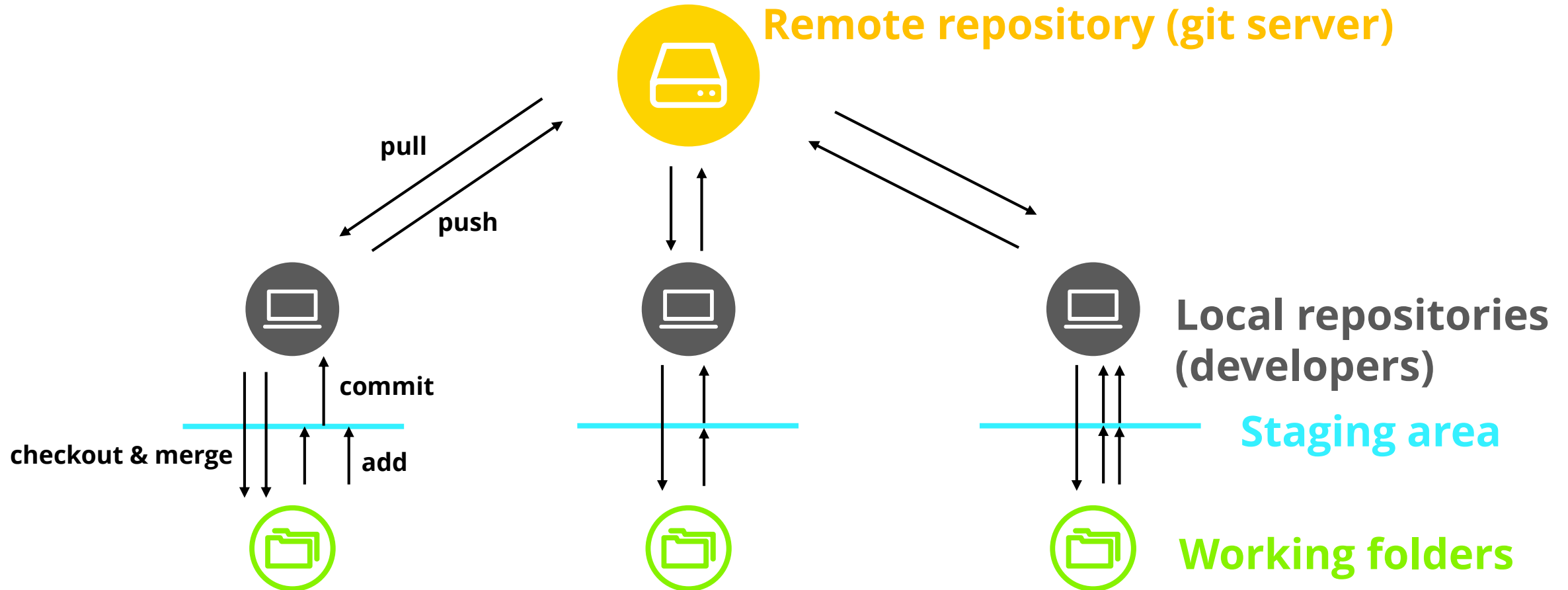


Git – Introduction

- Created by author of Linux
- Repository on a server – interact with the remote by **clone**, **pull** and **push**
- **Local repository** is a complete copy of the remote server
- Many of the operations are done locally:
 - change branches
 - commit changes to the local repository
 - versioning
- Pushing the local repository makes the versioning visible for the central repository

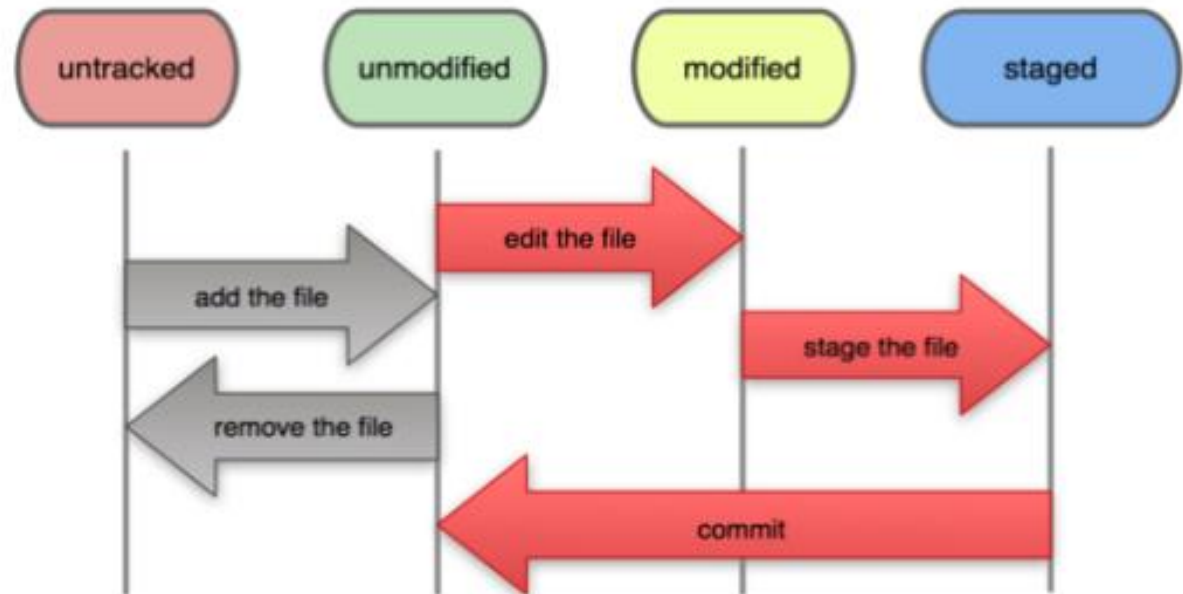


Git Schema



Local operations

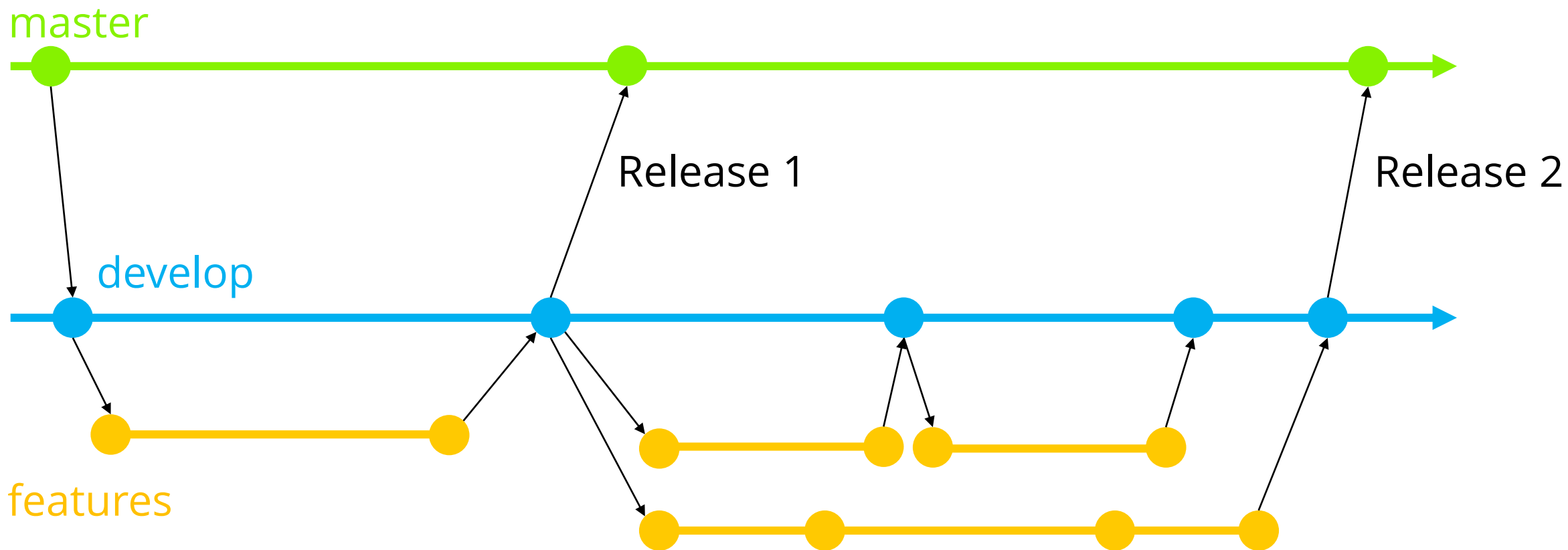
- **Add files** (git add) to tell git to version them
- **Stage files** create a snapshots of the files in he staging area
- **Commit** takes the files in the staging area and stores the snapshot permanently to the git repostiory



Branching

- Commits are associated with a branch.
- Default git branch is master.
- Branches are created from different branch (basically a copy)
- Branches allow to cluster commits belonging together (e.g. developers trying to implement specific feature vs others fixing a bug)
- Branches can be merged together when the new feature is ready and quality is assured to mess up the production branch.
- When two developers independently make conflicting changes, it creates a merge conflicts showing the conflicting parts with possibilities of resolution (use mine/use theirs/...)

Branching



Branching – Merge conflict

- File that where the same part was modified in two branches will result in **merge conflict**.
- If git is unable to resolve the conflict it adds <<< >>> sections to indicate where the problem occurred, so the developer can decide which code is still relevant and which is faulty.

```
<<<<<<< HEAD:index.html
<div id="footer">todo: message here</div>
=====
<div id="footer">
  thanks for visiting our site
</div>
>>>>>>> SpecialBranch:index.html
```

} branch 1's version

} branch 2's version

Remote operations

- **Clone** takes a repository from a server and makes a local repository at given destination

```
git clone git://git.kernel.org/pub/scm/.../linux.git my-linux
```

- **Push** takes the local changes and writes them to the server.

```
git pull origin master
```

- **Pull** gets most recent changes from the repository.

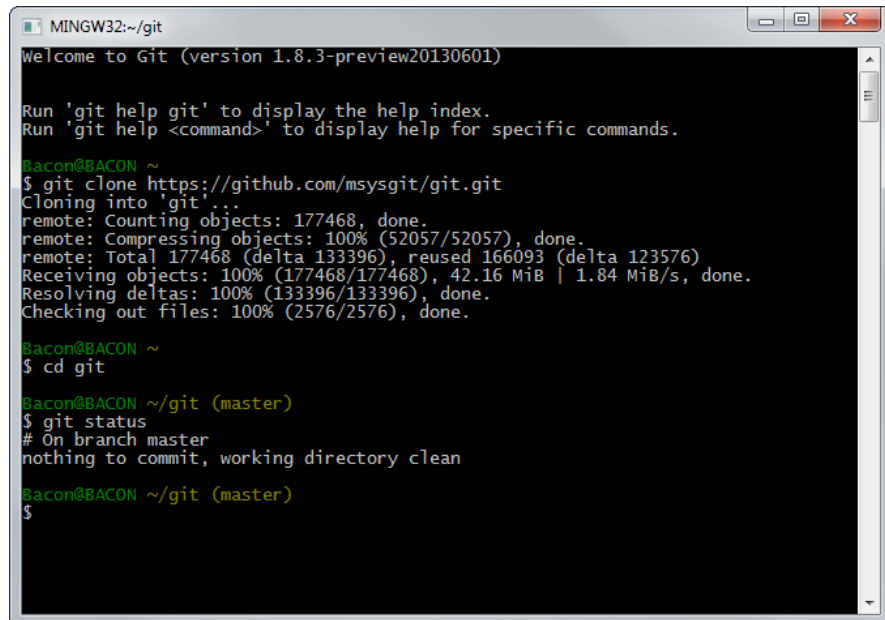
```
git pull origin master
```

Git – Workflow

1. **Create a branch:** Topic branches created from the canonical deployment branch (usually master) allow teams to contribute to many parallel efforts. Short-lived topic branches, in particular, keep teams focused and results in quick ships.
2. **Add commits:** Snapshots of development efforts within a branch create safe, revertible points in the project's history.
3. **Open a pull request:** Pull requests publicize a project's ongoing efforts and set the tone for a transparent development process.
4. **Discuss and review code:** Teams participate in code reviews by commenting, testing, and reviewing open pull requests. Code review is at the core of an open and participatory culture.
5. **Merge:** Upon clicking merge, GitHub automatically performs the equivalent of a local 'git merge' operation. GitHub also keeps the entire branch development history on the merged pull request.

Git Bash vs Git GUI

- Less intuitive
- More technical approach
- Better control



```
MINGW32:~/git
Welcome to Git (version 1.8.3-preview20130601)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

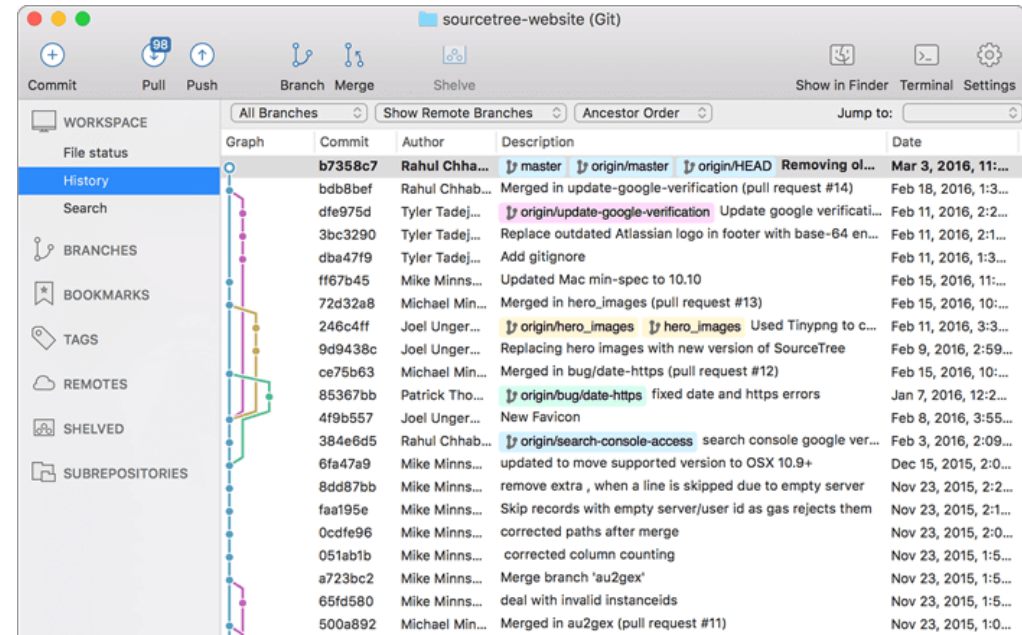
Bacon@BACON ~
$ git clone https://github.com/msysgit/git.git
Cloning into 'git'...
remote: Counting objects: 177468, done.
remote: Compressing objects: 100% (52057/52057), done.
remote: Total 177468 (delta 133396), reused 166093 (delta 123576)
Receiving objects: 100% (177468/177468), 42.16 MiB | 1.84 MiB/s, done.
Resolving deltas: 100% (133396/133396), done.
Checking out files: 100% (2576/2576), done.

Bacon@BACON ~
$ cd git

Bacon@BACON ~/git (master)
$ git status
# On branch master
nothing to commit, working directory clean

Bacon@BACON ~/git (master)
$
```

- Easy to start
- No need to know commands
- Blackbox



Git with GUI demo



GitLab



GitHub

- Git server repository.
- Ticketing system.
- Tickets (Issues) can be connected with branches – the issues can be discussed in the tickets.
- Assigning tickets to people. Issues labeling.
- When the developing is complete a merge request is made and the code is assigned to a person non-active in the issue fix. The person reviews the code and merge the changes into the production branch.
- Milestones.
- CI/CD
- ...

Git with GUI demo

1. Create a gitlab repository
2. Clone the repository
3. Add files
4. Push
5. Issue with branch and merge request
6. Switch to branch, add a file and change some of the already present
7. Stage and commit
8. Switch between branches (how working directory behaves)
9. Push changes
10. Merge and close the issue/branch

Git – Essential Commands

GIT Command	Description
git init	Create empty Git repo in specified directory. Run with no arguments to initialize the current directory as a git repository
git clone <repo>	Clone repo located at <repo> onto local machine. Original repo can be located on the local filesystem or on a remote machine via HTTP or SSH.
git branch	List all of the branches in your repo. Add a <branch> argument to create a new branch with the name <branch>.
git checkout -b <branch>	Create and check out a new branch named <branch>. Drop the -b flag to checkout an existing branch.
git add <directory>	Stage all changes in <directory> for the next commit. Replace <directory> with a <file> to change a specific file.
git commit -m "<message>"	Commit the staged snapshot, but instead of launching a text editor, use <message> as the commit message.
git status	List which files are staged, unstaged, and untracked.
git pull <remote>	Fetch the specified remote's copy of current branch and immediately merge it into the local copy.
git push <remote> <branch>	Push the branch to <remote>, along with necessary commits and objects. Creates named branch in the remote repo if it doesn't exist.

GIT Cheat Sheet: <https://www.atlassian.com/dam/jcr:8132028b-024f-4b6b-953e-e68fcce0c5fa/atlassian-git-cheatsheet.pdf>

GIT Official Documentation: <https://git-scm.com/docs/git-pull>

Git in CMD demo

1. git status, git branch in the repository
2. modify a file, add a file
3. git status
4. git commit +msg
5. git push
6. show the results on

Tasks

- Go to <https://learngitbranching.js.org/> and complete the introduction sequence

Git – Takeaway message

- Git is good for you, even though it may seem that it only adds work, in the end having track of all the changes and being able to develop parallelly outweighs the drawbacks by far.
- Helps with Quality Assurance workflow.
- Git is a versioning system vs GitLab/GitHub are cloud repository storage with further functions.
 - You do not have to be on cloud to use git.
- You do not need to work in team to benefit from git – you can use it solely to version your work.
 - One branch, just committing changes in time.
- It may seem overwhelming at first, but it is fairly simple once you get used to it. 😊

References:

- <https://git-scm.com/>
- <https://courses.cs.washington.edu/courses/cse403/13au/lectures/git.ppt.pdf>
- <https://www.slideshare.net/naimlatifi/gitpresentation-140814102916phpapp01>

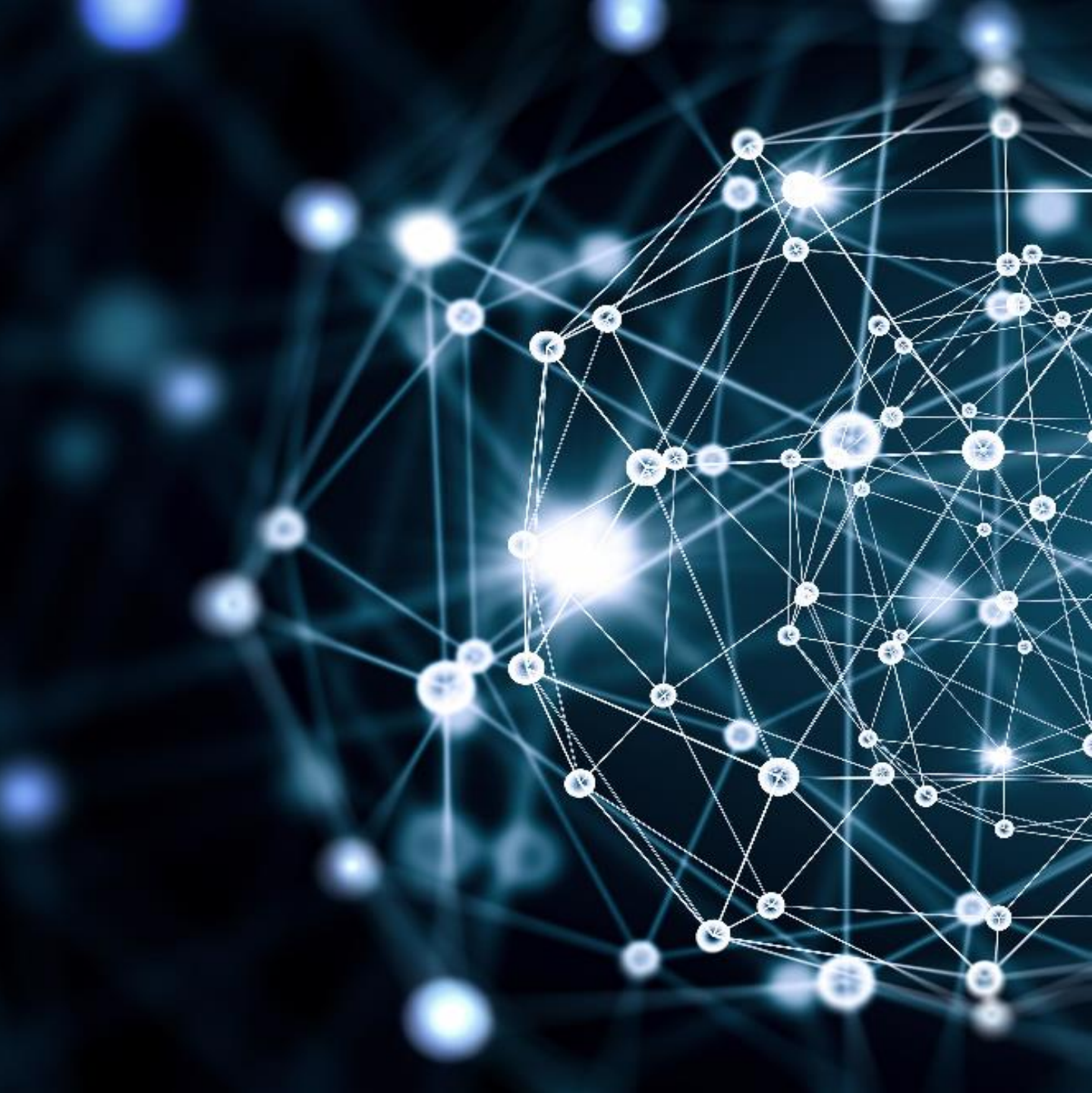
Pure Python Data Exploration



Pure Python Data Exploration

1. Upload the google playstore datasets
2. Open jupyter notebook D1_google_apps_empty.ipynb

Python Libraries



Python Libraries introduction

1. Open the D1_Python_libraries.ipynb notebook

Home Assignment HAI

Pick one of the following

1. Scrape a website with product you want to of your choosing and output a list of relevant information about the offers (e.g. <https://www.sbazar.cz/hledej/macbook?cena-dohodou=bez> , <https://www.mmreality.cz/nemovitosti/> , ...) (hint: BeautifulSoup)
2. For a list of stock symbols, get the last and future dividend payouts (e.g. <https://www.nasdaq.com/market-activity/stocks/t/dividend-history>) (hint: BeautifulSoup or find a free API)
3. Create a script (function) that states size of all files inside recursively and returns the path to the largest file. Raise error if given path is not a directory. Do this on local python installation in order to test it. (hint: os)
4. Any other ideas are welcome.

Save your script/notebok and send it to pmilicka@deloitteCE.com with subject **DSI_HA01_<surname>** for review.



Deloitte refers to one or more of Deloitte Touche Tohmatsu Limited, a UK private company limited by guarantee ("DTTL"), its network of member firms, and their related entities. DTTL and each of its member firms are legally separate and independent entities. DTTL (also referred to as "Deloitte Global") does not provide services to clients. Please see www.deloitte.com/cz/about to learn more about our global network of member firms.

Deloitte provides audit, consulting, legal, financial advisory, risk advisory, tax and related services to public and private clients spanning multiple industries. Deloitte serves four out of five Fortune Global 500® companies through a globally connected network of member firms in more than 150 countries and territories bringing world-class capabilities, insights, and high-quality service to address clients' most complex business challenges. To learn more about how Deloitte's approximately 245,000 professionals make an impact that matters, please connect with us on Facebook, LinkedIn, or Twitter.

Deloitte Central Europe is a regional organization of entities organized under the umbrella of Deloitte Central Europe Holdings Limited, the member firm in Central Europe of Deloitte Touche Tohmatsu Limited. Services are provided by the subsidiaries and affiliates of Deloitte Central Europe Holdings Limited, which are separate and independent legal entities. The subsidiaries and affiliates of Deloitte Central Europe Holdings Limited are among the region's leading professional services firms, providing services through more than 6,000 people in 44 offices in 18 countries.

© 2019. For information, contact Deloitte Czech Republic.