

Training on Web Scraping Prices for CPI

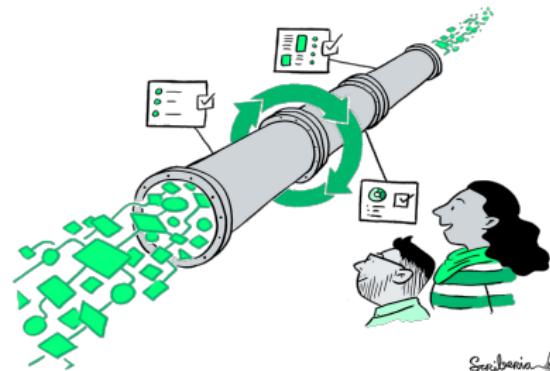
Reproducible Analytical Pipelines

Christophe Bontemps & Serge Goussev



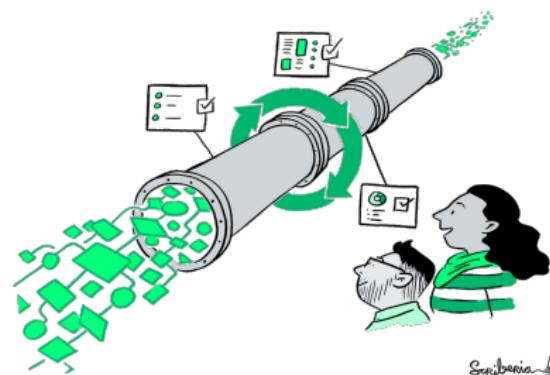
WHY ARE REPRODUCIBLE ANALYTICAL PIPELINES GOOD FOR YOU?

- ▶ It will make your (*working*) life easier



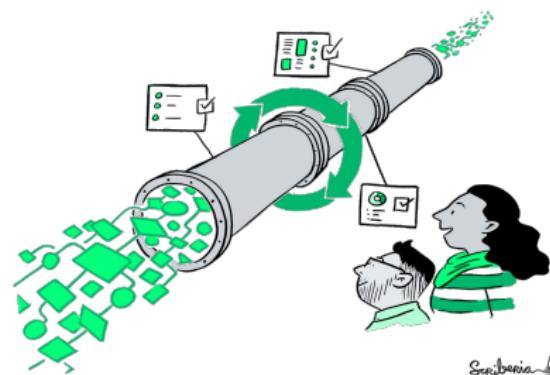
WHY ARE REPRODUCIBLE ANALYTICAL PIPELINES GOOD FOR YOU?

- ▶ It will make your (*working*) life easier
- ▶ It is a satisfying way to work



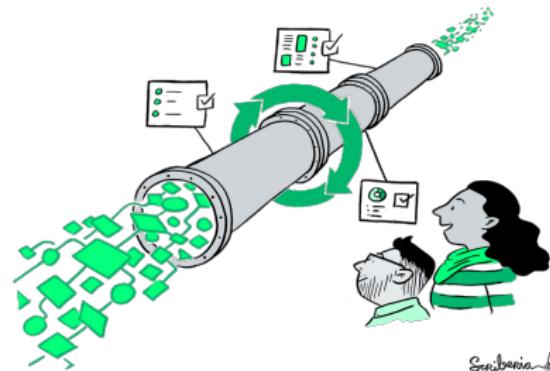
WHY ARE REPRODUCIBLE ANALYTICAL PIPELINES GOOD FOR YOU?

- ▶ It will make your (*working*) life easier
- ▶ It is a satisfying way to work
- ▶ It is a time-saver



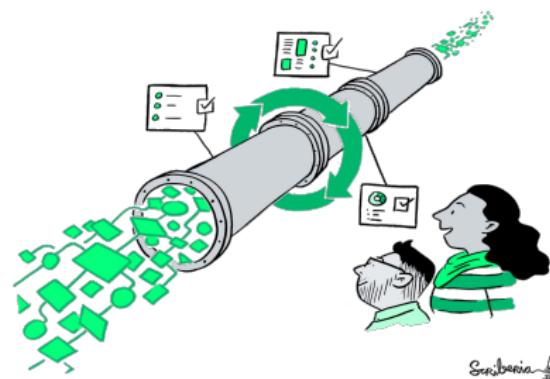
WHY ARE REPRODUCIBLE ANALYTICAL PIPELINES GOOD FOR YOU?

- ▶ It will make your (*working*) life easier
- ▶ It is a satisfying way to work
- ▶ It is a time-saver
- ▶ It helps tracking mistakes



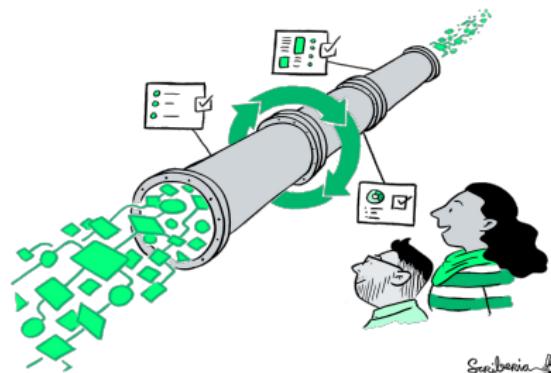
WHY ARE REPRODUCIBLE ANALYTICAL PIPELINES GOOD FOR YOU?

- ▶ It will make your (*working*) life easier
- ▶ It is a satisfying way to work
- ▶ It is a time-saver
- ▶ It helps tracking mistakes
- ▶ It helps collaborating



WHY ARE REPRODUCIBLE ANALYTICAL PIPELINES GOOD FOR YOU?

- ▶ It will make your (*working*) life easier
- ▶ It is a satisfying way to work
- ▶ It is a time-saver
- ▶ It helps tracking mistakes
- ▶ It helps collaborating
- ▶ It will enhance your skills (and make you famous?)



Motivation



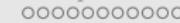
Issues



RAP



Principles



Version Control



Takeaways

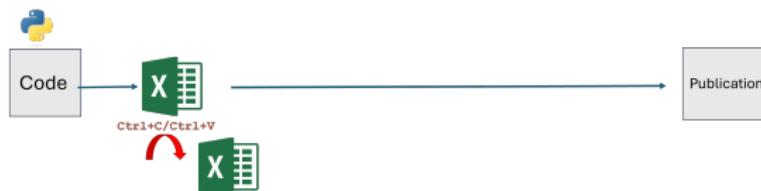


Resources



USUAL PRACTICE: THEORY VS REALITY

USUAL PRACTICE: THEORY VS REALITY



Send that file by email to your collaborators.

USUAL PRACTICE: THEORY VS REALITY



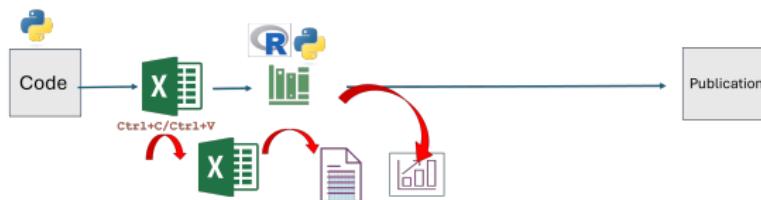
Someone in your team can start writing some insights.

USUAL PRACTICE: THEORY VS REALITY



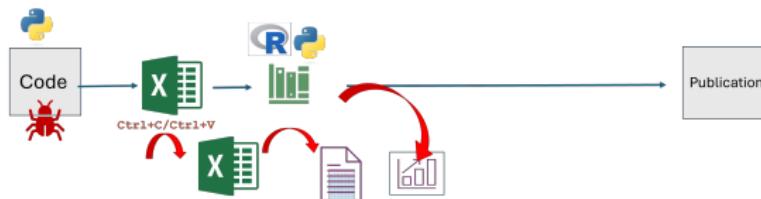
You, or someone else, start an analysis ...

USUAL PRACTICE: THEORY VS REALITY



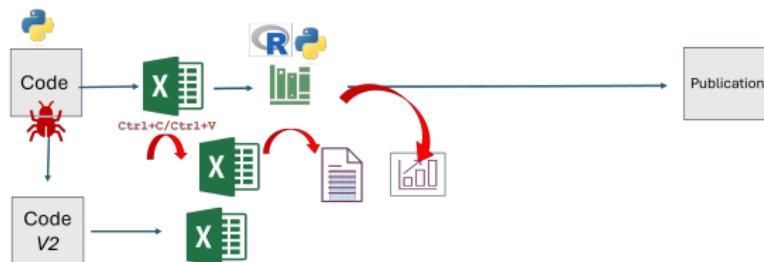
... producing some outputs (graphics, tables, etc..)

USUAL PRACTICE: THEORY VS REALITY



But wait... Oh no! There's a bug in the code!

USUAL PRACTICE: THEORY VS REALITY



So here is version 2, and another Excel file

USUAL PRACTICE: THEORY VS REALITY



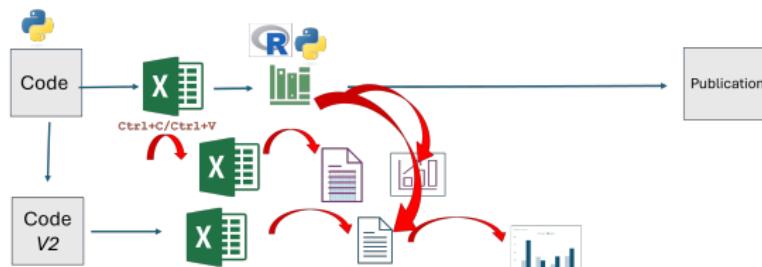
And new insights ...

USUAL PRACTICE: THEORY VS REALITY



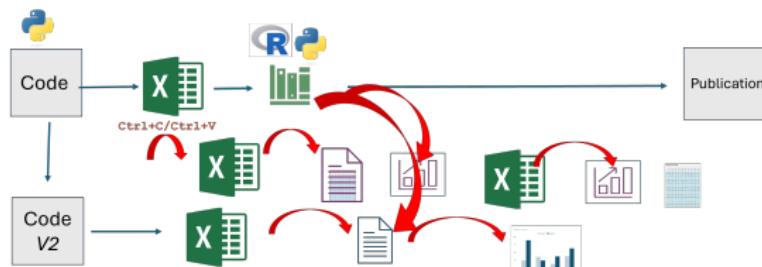
..and a new analysis based on the second Excell file

USUAL PRACTICE: THEORY VS REALITY



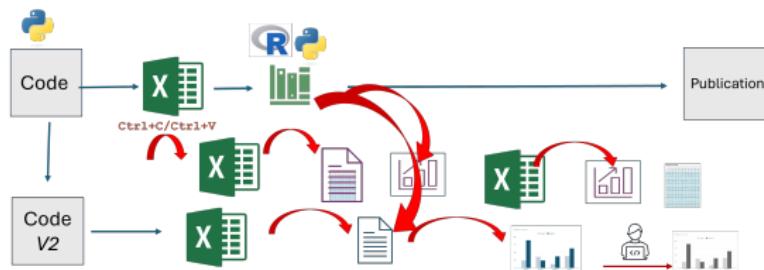
And new outputs, new graphics, etc.

USUAL PRACTICE: THEORY VS REALITY



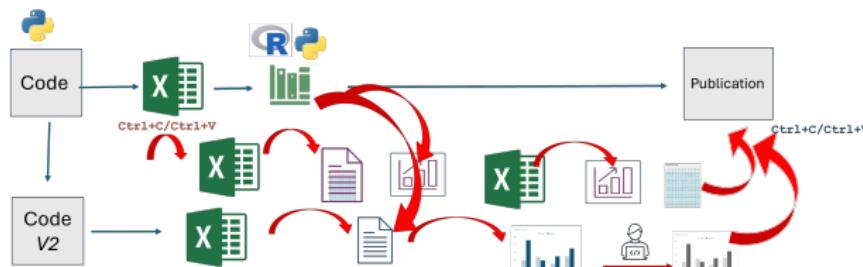
Also, using other data sets (classifications, scanner data)

USUAL PRACTICE: THEORY VS REALITY



Maybe someone adapts graphics to NSO reports style

USUAL PRACTICE: THEORY VS REALITY



Finally, copy/paste everything into the final report

USUAL PRACTICE: THEORY VS REALITY



In the end, this what you have produced!

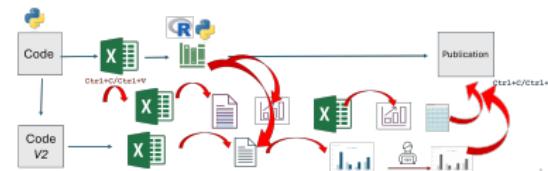
USUAL PRACTICE: IN THE END

- ▶ Lots of files



USUAL PRACTICE: IN THE END

- ▶ Lots of files
- ▶ Cut and paste is not a reliable, reproducible approach!



USUAL PRACTICE: IN THE END

- ▶ Lots of files
- ▶ Cut and paste is not a reliable, reproducible approach!
- ▶ Your brain may remember..



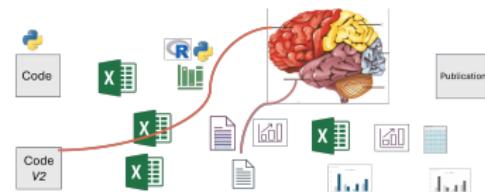
USUAL PRACTICE: IN THE END

- ▶ Lots of files
 - ▶ Cut and paste is not a reliable, reproducible approach!
 - ▶ Your brain may remember..
...all the steps...



USUAL PRACTICE: IN THE END

- ▶ Lots of files
 - ▶ Cut and paste is not a reliable, reproducible approach!
 - ▶ Your brain may remember..
...all the steps...
.. in the right order..



USUAL PRACTICE: IN THE END

- ▶ Lots of files
 - ▶ Cut and paste is not a reliable, reproducible approach!
 - ▶ Your brain may remember..
...all the steps...
.. in the right order..
...all of them !



USUAL PRACTICE: IN THE END

- ▶ Lots of files
- ▶ Cut and paste is not a reliable, reproducible approach!
- ▶ Your brain may remember..
...all the steps...
.. in the right order..
...all of them !
- ▶ Or use (bad) "tools"



WHAT ARE THE ISSUES?

- ▶ Errors due to cut and paste

Excel: Why using Microsoft's tool caused Covid-19 results to be lost

By Leo Kelton
Technology desk editor

5 October 2020



The badly thought-out use of Microsoft's Excel software was the reason nearly 16,000 coronavirus cases went unreported in England.

And it appears that Public Health England (PHE) was to blame, rather than a third-party contractor.

WHAT ARE THE ISSUES?

- ▶ Errors due to cut and paste
- ▶ Errors are difficult to track

Excel: Why using Microsoft's tool caused Covid-19 results to be lost

By Leo Kelion
Technology desk editor

© 5 October 2020

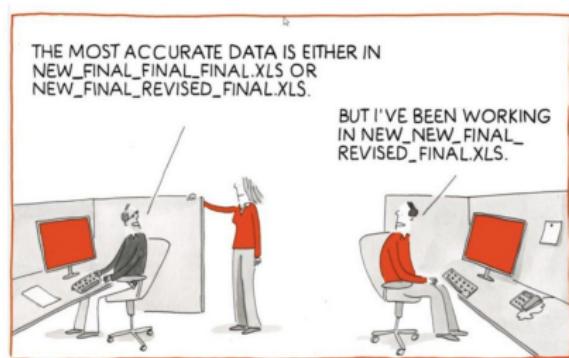


The badly thought-out use of Microsoft's Excel software was the reason nearly 16,000 coronavirus cases went unreported in England.

And it appears that Public Health England (PHE) was to blame, rather than a third-party contractor.

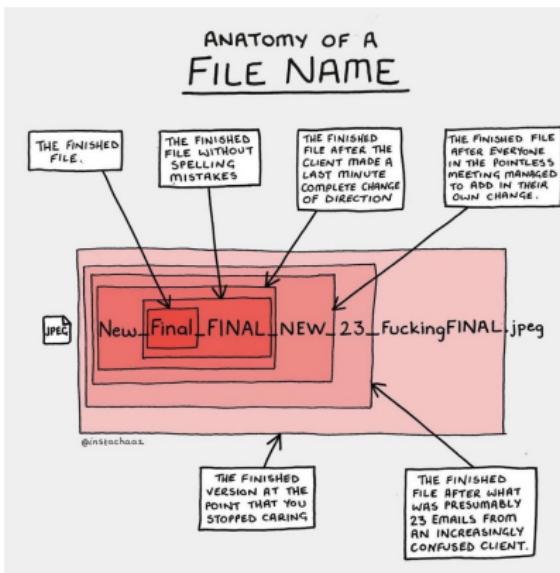
WHAT ARE THE ISSUES?

- ▶ Errors due to cut and paste
- ▶ Errors are difficult to track
- ▶ Each operator has his/her own approach



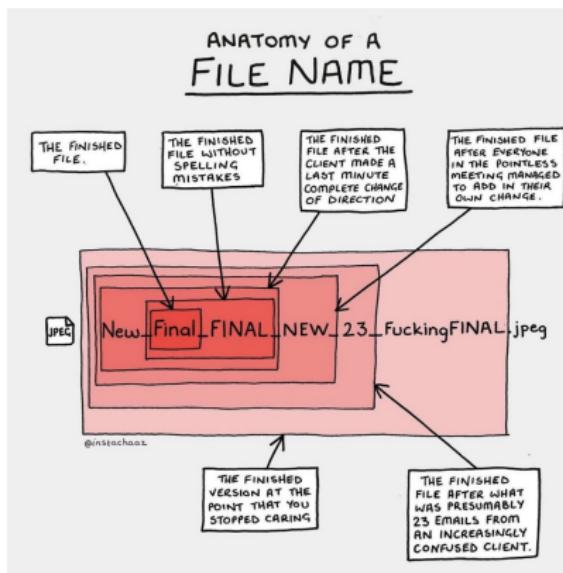
WHAT ARE THE ISSUES?

- ▶ Errors due to cut and paste
- ▶ Errors are difficult to track
- ▶ Each operator has his/her own approach
- ▶ Several versions of code may coexist



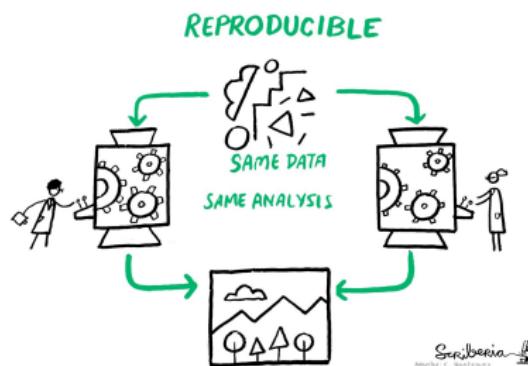
WHAT ARE THE ISSUES?

- ▶ Errors due to cut and paste
- ▶ Errors are difficult to track
- ▶ Each operator has his/her own approach
- ▶ Several versions of code may coexist
- ▶ The steps aren't recorded



WHAT ARE THE ISSUES?

- ▶ Errors due to cut and paste
- ▶ Errors are difficult to track
- ▶ Each operator has his/her own approach
- ▶ Several versions of code may coexist
- ▶ The steps aren't recorded
- ▶ Reproducibility is not granted



FUNDAMENTAL PRINCIPLES OF OFFICIAL STATISTICS

- ▶ Clear mention of the processes used to produce statistics



Fundamental Principles of Official Statistics*

For more information: unstats.un.org

The General Assembly
“Review recent resolutions” of the General Assembly and the Economic and Social Council highlighting the fundamental importance of official statistics for the national and international statistical agencies.

Bearing in mind the critical role of high-quality official statistical information in analysis and informed policy decision-making in support of sustainable development, peace and security as well as for mutual understanding, trade among the States and peoples of an increasingly connected world, demanding openness and transparency.

Recommending also that the essential trust of the public in the integrity of official statistical systems and confidence in statistics depend to a large extent on respect for the fundamental principles of official statistics as the basis of any society seeking to understand itself and respect the rights of its members, and in this context that professional independence and accountability of statistical agencies are crucial.

Stressing that, in order to be effective, the fundamental values and principles that govern the production of official statistics by legal and institutional frameworks and be respected at all political levels and by all stakeholders in national statistical systems,

Endorse the Fundamental Principles of Official Statistics, as adopted by the Statistical Commission in 1994* and reaffirmed in 2013, and endorsed by the Economic and Social Council in its resolution 205/21 of 24 July 2013.

* General Assembly resolution 65/21 adopted on 29 January 2014. The “Review” of the Principles was part of the original text.

These include General Assembly resolution 46/200 on the role of official statistics and Economic and Social Council resolution 205/13 on the 2013 World Programme of Action for Statistical Development on strengthening statistical capacity and 2013/2014 on the Fundamental Principles of Official Statistics.

For a one-page summary of the history of the initial adoption of the Fundamental Principles in 1994, see the document “Review of the Statistical Commission on its special session (Official Report) of the Economic and Social Council, 1994: Sustainable development and the role of official statistics.” The Fundamental Principles and their history is available from the website of the Statistics Division.

Principle 1: Relevance, Impartiality, and Equal Access
Official statistics provide an indispensable element in the information system of a democratic society, serving the Government, the economy and the public with data about the economic, demographic, social and environmental conditions. Official statistics that meet the test of practical utility are to be compiled and made available on an impartial basis by official statistical agencies to honour citizens' entitlement to public information.

Principle 2: Professional Standards, Scientific Principles, and Professional Ethics
To retain trust in official statistics, the statistical agencies need to demonstrate that they apply professional conventions, including scientific principles and professional ethics, on the methods and procedures for the collection, processing, storage and presentation of statistical data.

Principle 3: Accountability and Transparency
To facilitate a correct interpretation of the data, the statistical agencies are to present information according to scientific standards on the sources, methods and procedures of the statistics.

Principle 4: Prevention of Misuse
The statistical agencies are entitled to comment on erroneous interpretation and misuse of statistics.

Principle 5: Sources of Official Statistics
Data for statistical purposes may be drawn from all types of sources, including administrative records. Statistical agencies are to choose the source with regard to quality, timeliness, costs and the burden on respondents.

Principle 6: Confidentiality
Individual data collected by statistical agencies for statistical compilation, whether they refer to natural or legal persons, are to be strictly confidential and used exclusively for statistical purposes.

Principle 7: Legitimacy
The laws, regulations and measures under which the statistical systems operate are to be made public.

Principle 8: National Coordination
Coordination among statistical agencies or within countries is essential to achieve consistency and efficiency in the statistical system.

Principle 9: Use of International Standards
The use by statistical agencies in each country of international concepts, classifications and methods promotes the consistency and efficiency of statistical systems at all official levels.

Principle 10: International Cooperation
Bilateral and multilateral cooperation in statistics contributes to the improvement of systems of official statistics in countries.

FUNDAMENTAL PRINCIPLES OF OFFICIAL STATISTICS

- ▶ Clear mention of the **processes** used to produce statistics
- ▶ *To retain trust in official statistics, the statistical agencies need to decide according to strictly professional considerations, including scientific principles and professional ethics, on the methods and procedures for the collection, processing, storage and presentation of statistical data.*



Fundamental Principles of Official Statistics*

For more information: unstats.un.org

The General Assembly
Reviewing recent resolutions² of the General Assembly and the Economic and Social Council highlighting the fundamental importance of official statistics for the national development process, the statistical agencies:
Desiring in view the critical role of high-quality official statistical information in analysis and informed policy decision-making in support of sustainable development, peace and security as well as for mutual knowledge and trade among the States and peoples of an increasingly connected world, demanding openness and transparency;
Reiterating in mind also that the essential trust of the public in the integrity of official statistical systems and confidence in statistics depend to a large extent on trust for the fundamental principles of official statistics on the basis of any society seeking to understand itself and respect the rights of its members, and in this context that professional independence is an accountability of statistical agencies crucial;
Stressing that, in order to be effective, the fundamental values and principles that govern the production of official statistics must be legal and institutional frameworks and be respected at all political levels and by all stakeholders in national statistical systems;

Endorse the Fundamental Principles of Official Statistics, as adopted by the Statistical Commission in 1994³ and reaffirmed in 2013, and endorsed by the Economic and Social Council in its resolution 205/21 of 24 July 2013.

* General Assembly resolution 65/217 adopted on 29 January 2014. The "Values of the Principles" are part of the original text.
These include General Assembly resolution 46/200 on the role of official statistics and Economic and Social Council resolution 205/13 on the 2013 World Population Conference, both of which stress the importance of strengthening statistical capacity and 2030/2030 on the implementation of the Fundamental Principles of Official Statistics.

² For further reading, please see the report of the initial adoption of the Fundamental Principles in 1994, see the report of the Statistical Commission on its special session (Official Report) of the Economic and Social Council, 1994. *Supplementary Information on the Adoption of the Fundamental Principles and their history* is available from the website of the Statistics Division.

Principle 1: Relevance, Impartiality, and Equal Access
Official statistics provide an indispensable element in the information system of a democratic society, serving the Government, the economy and the public with data about the economic, demographic, social and environmental conditions. Official statistics that meet the test of practical utility are to be compiled and made available on an impartial basis by official statistical agencies to honour citizens' entitlement to public information.

Principle 2: Professional Standards, Scientific Principles, and Professional Ethics
To retain trust in official statistics, the statistical agencies must follow professional standards and scientific conventions, including scientific principles and professional ethics, on the methods and procedures for the collection, processing, storage and presentation of statistical data.

Principle 3: Accountability and Transparency
To facilitate a correct interpretation of the data, the statistical agencies are to present information according to scientific standards on the sources and methods and procedures of the statistics.

Principle 4: Prevention of Misuse
The statistical agencies are entitled to comment on erroneous interpretation and misuse of statistics.

Principle 5: Sources of Official Statistics
Data for statistical purposes may be drawn from all types of sources, including administrative records, surveys and experiments, provided that the statistical agencies are to choose the source with regard to quality, timeliness, costs and the burden on respondents.

Principle 6: Confidentiality
Individual data collected by statistical agencies for statistical compilation, whether they refer to natural or legal persons, are to be strictly confidential and used exclusively for statistical purposes.

Principle 7: Legitimacy
The laws, regulations and measures under which the statistical systems operate are to be made public.

Principle 8: National Coordination
Coordination among statistical agencies or within countries is essential to achieve consistency and efficiency in the statistical system.

Principle 9: Use of International Standards
The use by statistical agencies in each country of international concepts, classifications and methods promotes the consistency and efficiency of statistical systems at all official levels.

Principle 10: International Cooperation
Bilateral and multilateral cooperation in statistics contributes to the improvement of systems of official statistics in countries.

FUNDAMENTAL PRINCIPLES OF OFFICIAL STATISTICS

- ▶ Clear mention of the **processes** used to produce statistics
- ▶ *To retain trust in official statistics, the statistical agencies need to decide according to strictly professional considerations, including scientific principles and professional ethics, on the methods and procedures for the collection, processing, storage and presentation of statistical data.*
- ▶ In short, **processes** are important!



Fundamental Principles of Official Statistics*

For more information: unstats.un.org

Principle 1: Relevance, Impartiality, and Equal Access
Official statistics provide an indispensable element in the information system of a democratic society, serving the Government, the economy and the public with data about the economic, demographic, social and environmental conditions. These official statistics that meet the test of practical utility are to be compiled and made available on an impartial basis by official statistical agencies to honour citizens' entitlement to public information.

Principle 2: Professional Standards, Scientific Principles, and Professional Ethics
To retain trust in official statistics, the statistical agencies need to demonstrate adherence to strict professional conventions, including scientific principles and professional ethics, on the methods and procedures for the collection, processing, storage and presentation of statistical data.

Principle 3: Accountability and Transparency
To facilitate a correct interpretation of the data, the statistical agencies are to present information according to scientific standards on the sources, methods and procedures of the statistics.

Principle 4: Prevention of Misuse
The statistical agencies are entitled to comment on erroneous interpretation and misuse of statistics.

Principle 5: Sources of Official Statistics
Data for statistical purposes may be drawn from all types of sources, including administrative records, surveys and experiments. Statistical agencies are to choose the source with regard to quality, timeliness, costs and the burden on respondents.

Principle 6: Confidentiality
Individual data collected by statistical agencies for statistical compilation, whether they refer to natural or legal persons, are to be strictly confidential and used exclusively for statistical purposes.

Principle 7: Legitimacy
The laws, regulations and measures under which the statistical systems operate are to be made public.

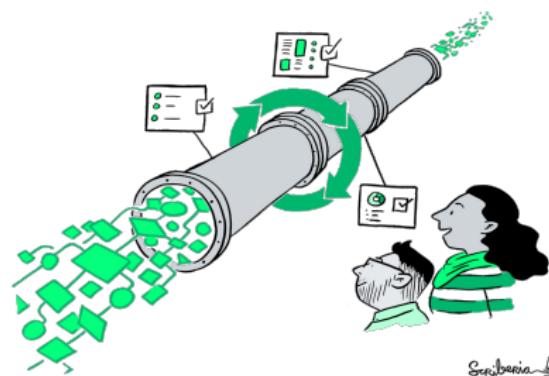
Principle 8: National Coordination
Coordination among statistical agencies or within countries is essential to achieve consistency and efficiency in the statistical system.

Principle 9: Use of International Standards
The use by statistical agencies in each country of international concepts, classifications and methods promotes the consistency and efficiency of statistical systems at all official levels.

Principle 10: International Cooperation
Bilateral and multilateral cooperation in statistics contributes to the improvement of systems of official statistics in countries.

WHAT IS A REPRODUCIBLE ANALYTICAL PIPELINE?

- ▶ It is a process



WHAT IS A REPRODUCIBLE ANALYTICAL PIPELINE?

- ▶ It is a process
- ▶ It is (*quite*) automated



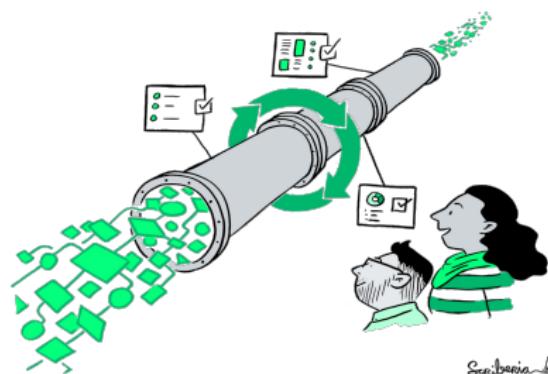
WHAT IS A REPRODUCIBLE ANALYTICAL PIPELINE?

- ▶ It is a process
- ▶ It is (*quite*) automated
- ▶ It is (*easily*) reproducible



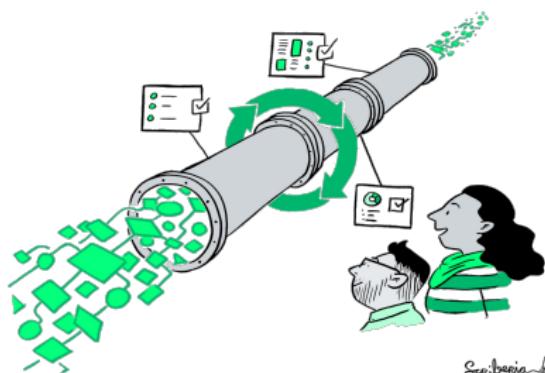
WHAT IS A REPRODUCIBLE ANALYTICAL PIPELINE?

- ▶ It is a process
- ▶ It is (*quite*) automated
- ▶ It is (*easily*) reproducible
- ▶ It minimises mistakes



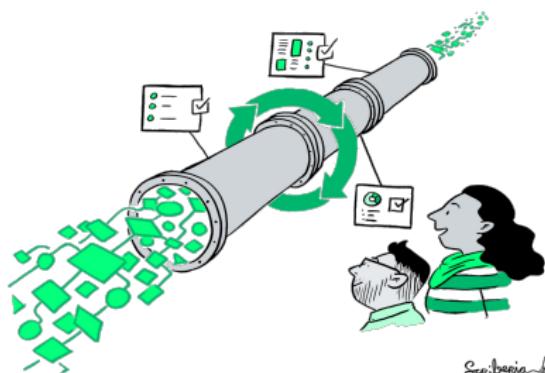
WHAT IS A REPRODUCIBLE ANALYTICAL PIPELINE?

- ▶ It is a process
- ▶ It is (*quite*) automated
- ▶ It is (*easily*) reproducible
- ▶ It minimises mistakes
- ▶ It leads to fast processes
(see Vanuatu Experience)



WHAT IS A REPRODUCIBLE ANALYTICAL PIPELINE?

- ▶ It is a process
- ▶ It is (*quite*) automated
- ▶ It is (*easily*) reproducible
- ▶ It minimises mistakes
- ▶ It leads to fast processes
(see Vanuatu Experience)
- ▶ It builds trust



WHAT DOES A RAP LOOK LIKE?



C

Ideally, Input (website) and output (report) are linked

WHAT DOES A RAP LOOK LIKE?



C

Many steps are needed to create a report

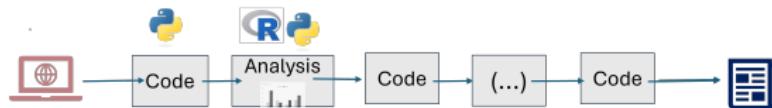
WHAT DOES A RAP LOOK LIKE?



C

All steps should be linked in a structured process

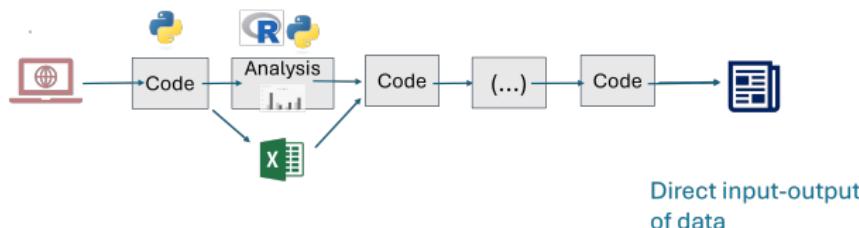
WHAT DOES A RAP LOOK LIKE?



C

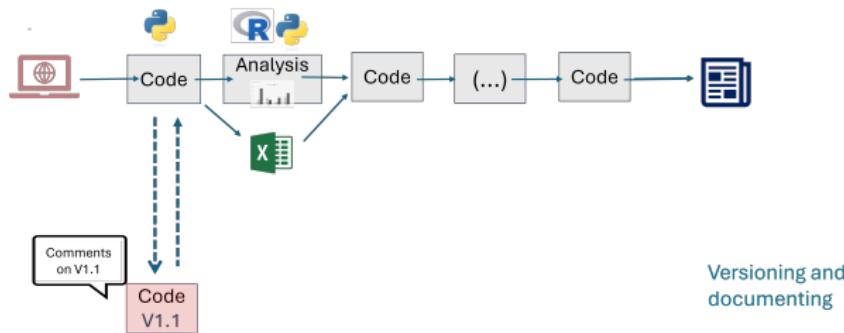
And only through code (Python, R, *Stata*), no copy/paste

WHAT DOES A RAP LOOK LIKE?



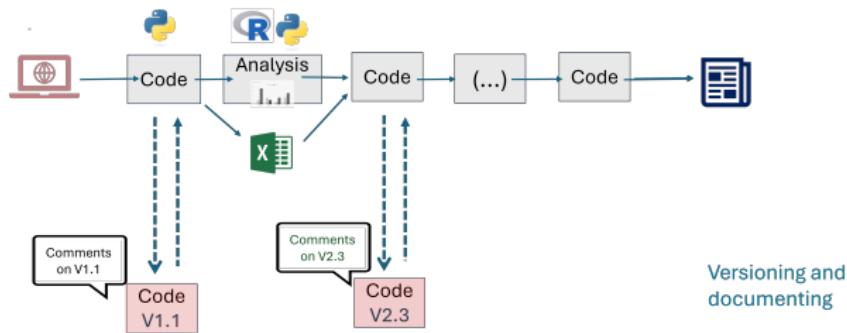
There may be side-products, but with explicit output-input links

WHAT DOES A RAP LOOK LIKE?



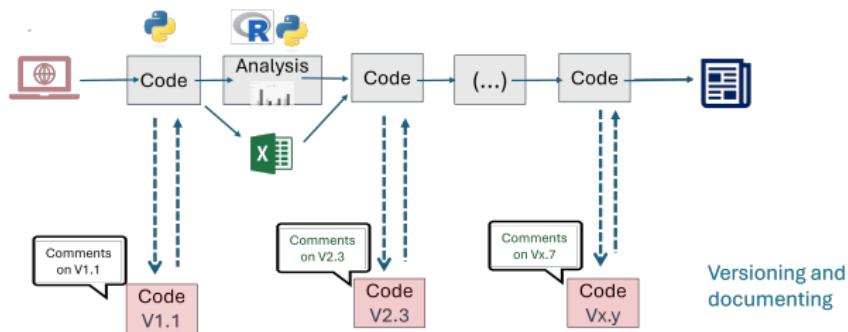
If needed, code can be updated (new versions)

WHAT DOES A RAP LOOK LIKE?



And comments added for each change

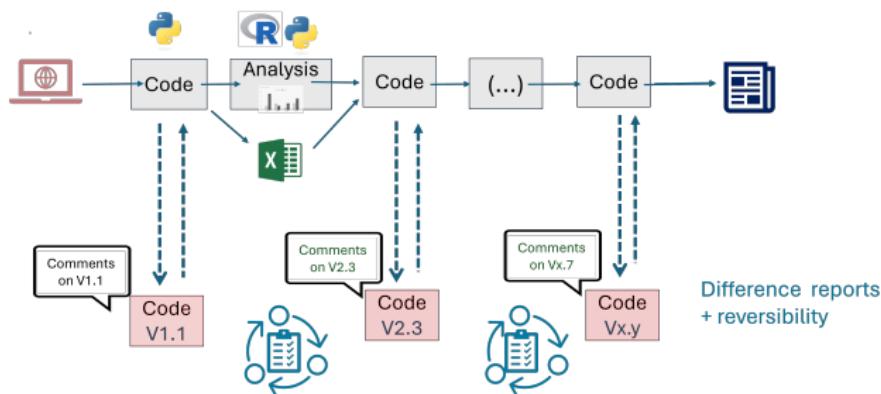
WHAT DOES A RAP LOOK LIKE?



C

Documentation on the process builds up with code changes

WHAT DOES A RAP LOOK LIKE?



C

Other contributors are welcome!

Motivation
oooo

Issues
oo

RAP
oo

Principles
●oooooooooooo

Version Control
oooooooooooo

Takeaways
o

Resources
o

RAP PRINCIPLES:

Motivation
○○○○

Issues
○○

RAP
○○

Principles
●○○○○○○○○○○

Version Control
○○○○○○○○○○○○

Takeaways
○

Resources
○

RAP PRINCIPLES:

- ▶ Automation (*as much as you can*)

RAP PRINCIPLES:

- ▶ Automation (*as much as you can*)
- ↪ Avoid manual work

RAP PRINCIPLES:

- ▶ Automation (*as much as you can*)
- ↪ Avoid manual work
- ▶ Reusable (modular) code

RAP PRINCIPLES:

- ▶ Automation (*as much as you can*)
- ↪ Avoid manual work
- ▶ Reusable (modular) code
- ↪ Build blocs, update blocs, change blocs, test blocs

RAP PRINCIPLES:

- ▶ Automation (*as much as you can*)
- ↪ Avoid manual work
- ▶ Reusable (modular) code
- ↪ Build blocs, update blocs, change blocs, test blocs
- ▶ Transparency

RAP PRINCIPLES:

- ▶ Automation (*as much as you can*)
- ↪ Avoid manual work
- ▶ Reusable (modular) code
- ↪ Build blocs, update blocs, change blocs, test blocs
- ▶ Transparency
- ↪ Show what you, do what you say

RAP PRINCIPLES:

- ▶ Automation (*as much as you can*)
- ↪ Avoid manual work
- ▶ Reusable (modular) code
- ↪ Build blocs, update blocs, change blocs, test blocs
- ▶ Transparency
- ↪ Show what you, do what you say
- ▶ Use open source tools

RAP PRINCIPLES:

- ▶ Automation (*as much as you can*)
- ↪ Avoid manual work
- ▶ Reusable (modular) code
- ↪ Build blocs, update blocs, change blocs, test blocs
- ▶ Transparency
- ↪ Show what you, do what you say
- ▶ Use open source tools
- ↪ Free, reusable, huge community

RAP PRINCIPLES:

- ▶ Automation (*as much as you can*)
- ↪ Avoid manual work
- ▶ Reusable (modular) code
- ↪ Build blocs, update blocs, change blocs, test blocs
- ▶ Transparency
- ↪ Show what you, do what you say
- ▶ Use open source tools
- ↪ Free, reusable, huge community
- ▶ Version control

RAP PRINCIPLES:

- ▶ Automation (*as much as you can*)
- ↪ Avoid manual work
- ▶ Reusable (modular) code
- ↪ Build blocs, update blocs, change blocs, test blocs
- ▶ Transparency
- ↪ Show what you, do what you say
- ▶ Use open source tools
- ↪ Free, reusable, huge community
- ▶ Version control
- ↪ Easy to track code, easy to share, easy to update, ...

RAP PRINCIPLES:

- ▶ Automation (*as much as you can*)
- ↪ Avoid manual work
- ▶ Reusable (modular) code
- ↪ Build blocs, update blocs, change blocs, test blocs
- ▶ Transparency
- ↪ Show what you, do what you say
- ▶ Use open source tools
- ↪ Free, reusable, huge community
- ▶ Version control
- ↪ Easy to track code, easy to share, easy to update, ...
- ▶ Good coding practices

RAP PRINCIPLES:

- ▶ Automation (*as much as you can*)
- ↪ Avoid manual work
- ▶ Reusable (modular) code
- ↪ Build blocs, update blocs, change blocs, test blocs
- ▶ Transparency
- ↪ Show what you, do what you say
- ▶ Use open source tools
- ↪ Free, reusable, huge community
- ▶ Version control
- ↪ Easy to track code, easy to share, easy to update, ...
- ▶ Good coding practices
- ↪ Write for humans, not for machines

RAP PRINCIPLES:

- ▶ Automation (*as much as you can*)
- ↪ Avoid manual work
- ▶ Reusable (modular) code
- ↪ Build blocs, update blocs, change blocs, test blocs
- ▶ Transparency
- ↪ Show what you, do what you say
- ▶ Use open source tools
- ↪ Free, reusable, huge community
- ▶ Version control
- ↪ Easy to track code, easy to share, easy to update, ...
- ▶ Good coding practices
- ↪ Write for humans, not for machines
- ▶ Testing

RAP PRINCIPLES:

- ▶ Automation (*as much as you can*)
- ↪ Avoid manual work
- ▶ Reusable (modular) code
- ↪ Build blocs, update blocs, change blocs, test blocs
- ▶ Transparency
- ↪ Show what you, do what you say
- ▶ Use open source tools
- ↪ Free, reusable, huge community
- ▶ Version control
- ↪ Easy to track code, easy to share, easy to update, ...
- ▶ Good coding practices
- ↪ Write for humans, not for machines
- ▶ Testing
- ▶ Peer-review

Motivation
oooo

Issues
oo

RAP
oo

Principles
o●oooooooooooo

Version Control
oooooooooooo

Takeaways
o

Resources
o

RAP PRINCIPLES:

RAP PRINCIPLES:

These principles translate into:

Good Practices

+

Good Tools

RAP PRINCIPLES:

These principles translate into:

Good Practices

+

Good Tools

- We'll detail some of these practices and tools

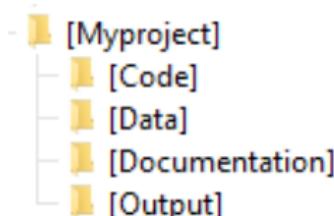
GOOD PRACTICES: ORGANIZE YOUR WORK

Have a clear directory structure

GOOD PRACTICES: ORGANIZE YOUR WORK

Have a clear directory structure

- ▶ Separate files into data, code, docs, etc.

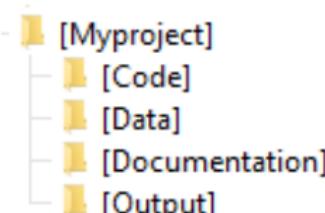


Example of a well-organized directory structure.

GOOD PRACTICES: ORGANIZE YOUR WORK

Have a clear directory structure

- ▶ Separate files into data, code, docs, etc.
- ▶ Make directories portable (relative path)



Example of a well-organized directory structure.

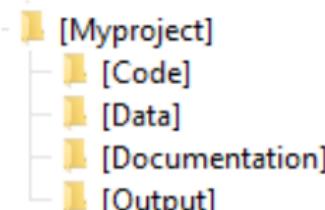
Usual

```
mydata =  
pd.read_csv("c://ESCAP/Webscraping/Data/WebData.csv")
```

GOOD PRACTICES: ORGANIZE YOUR WORK

Have a clear directory structure

- ▶ Separate files into data, code, docs, etc.
- ▶ Make directories portable (relative path)



Example of a well-organized directory structure.

Usual

```
mydata =  
pd.read_csv("c://ESCAP/Webscraping/Data/WebData.csv")
```

Better

```
Assuming your code is in c://ESCAP/Webscraping/Code/  
mydata = pd.read_csv("../Data/WebData.csv")
```

GOOD PRACTICES: ORGANIZE YOUR WORK

Use naming conventions: For files/code

- ▶ Avoid lazy names

Usual

prog1.ipynb
prog2.ipynb
Stat.ipynb
progC.ipynb
progP.ipynb

GOOD PRACTICES: ORGANIZE YOUR WORK

Use naming conventions: For files/code

- ▶ Avoid lazy names
- ▶ Meaningful files names

Usual	Better
prog1.ipynb	Scraping_Data.ipynb
prog2.ipynb	Cleaning_Data.ipynb
Stat.ipynb	Stats_Tables.ipynb
progC.ipynb	Classification.ipynb
progP.ipynb	Price_CPI.ipynb

GOOD PRACTICES: ORGANIZE YOUR WORK

Use naming conventions: For files/code

- ▶ Avoid lazy names
- ▶ Meaningful files names
- ▶ Order of execution

Usual	Even better
prog1.ipynb	01_Scraping_data.ipynb
prog2.ipynb	02_Cleaning_data.ipynb
Stat.ipynb	03_Classification.ipynb
progC.ipynb	04_Stats_Tables.ipynb
progP.ipynb	04_Price_CPI.ipynb

GOOD PRACTICES: ORGANIZE YOUR WORK

Use naming conventions: For outputs

- ▶ Avoid numbering
- Usual
 - Table1.pdf
 - Table2.pdf
 - Graph.jpg
 - Model.csv

GOOD PRACTICES: ORGANIZE YOUR WORK

Use naming conventions: For outputs

- ▶ Avoid numbering
- ▶ Explicit type of output

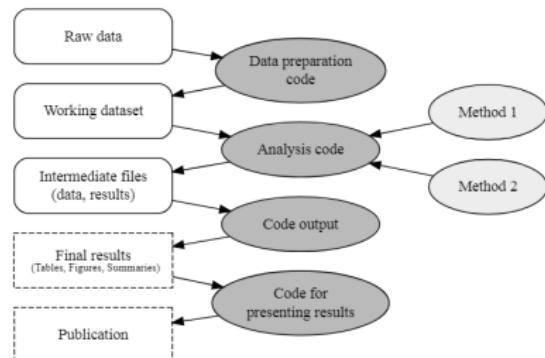
Usual
Table1.pdf
Table2.pdf
Graph.jpg
Model.csv

Better
Stat_Desc_Table.pdf
Price_Stat_Table.pdf
Dress_Prices_Graphic.jpg
All_prices_Results.csv

GOOD PRACTICES FOR AUTOMATION

Keep track of the workflow:

- ▶ Cut and paste should be avoided

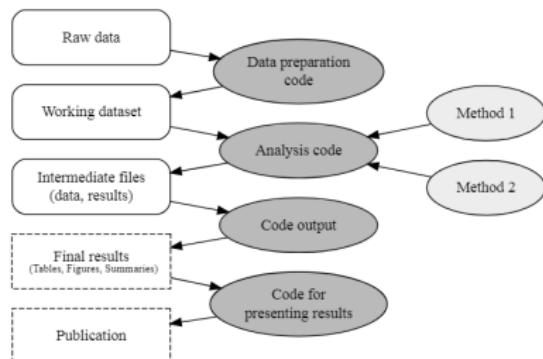


Example of a simple workflow.

GOOD PRACTICES FOR AUTOMATION

Keep track of the workflow:

- ▶ Cut and paste should be avoided
- ▶ Every step of the process is coded

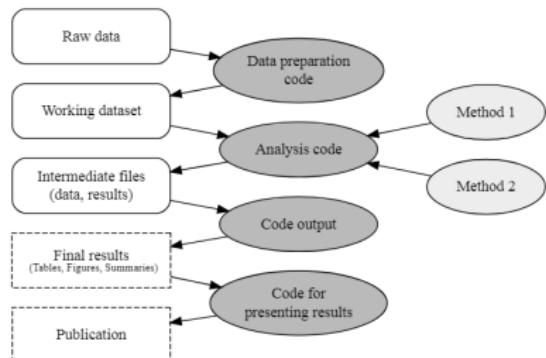


Example of a simple workflow.

GOOD PRACTICES FOR AUTOMATION

Keep track of the workflow:

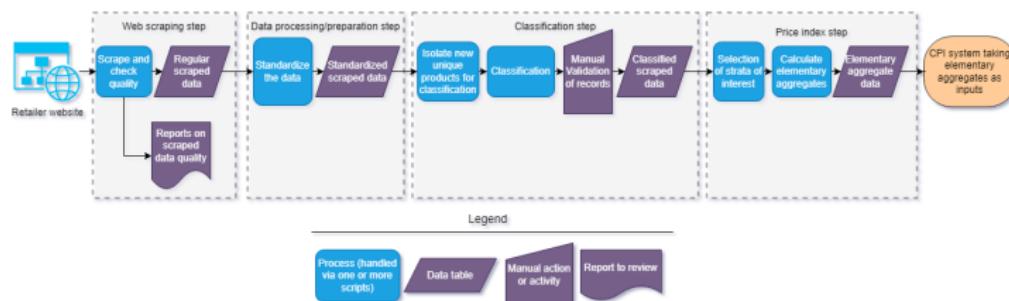
- ▶ Cut and paste should be avoided
- ▶ Every step of the process is coded
- ▶ Manage (and draw) the workflow



Example of a simple workflow.

GOOD PRACTICES FOR AUTOMATION

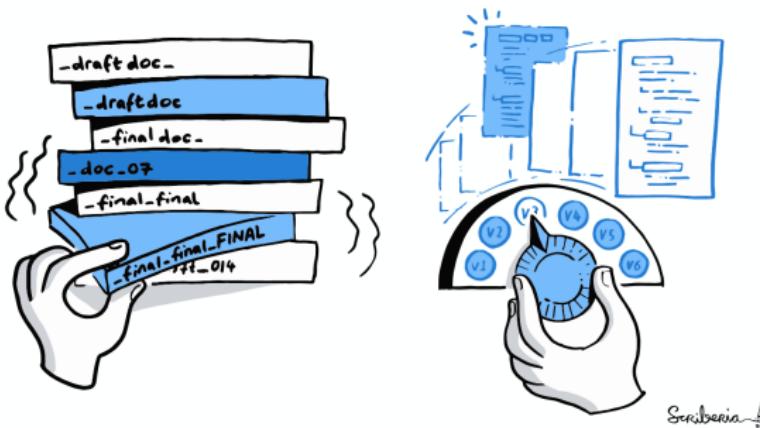
Keep track of the workflow:
Here is our workflow:



Created by Serge Goussev.

VERY GOOD PRACTICES

Use a version control system (Git/GitHub)



More on Version Control later

GOOD CODING PRACTICES: CODE FOR OTHERS

Program with style:

Use literate programming

*“Let us concentrate rather on explaining to humans
what we want the computer to do”*

D. Knuth (1984)

GOOD CODING PRACTICES: CODE FOR OTHERS

Program with style:

"(. . .) code is read much more often than it is written"

Guido van Rossum (2013 -PEP8)

PEP stands for *Python Enhancement Proposals*

GOOD CODING PRACTICES: CODE FOR OTHERS

Program with style:

Use conventions on layout (Comments, indentation,...)

Contents

- Introduction
- A Foolish Consistency is the Hobgoblin of Little Minds
- Code Lay-out
 - Indentation
 - Tabs or Spaces?
 - Maximum Line Length
 - Should a Line Break Before or After a Binary Operator?
 - Blank Lines
 - Source File Encoding
 - Imports
 - Module Level Dunder Names
- String Quotes
- Whitespace in Expressions and Statements
 - Pet Peeves
 - Other Recommendations
- When to Use Trailing Commas
- Comments
 - Block Comments
 - Inline Comments
 - Documentation Strings
- Naming Conventions
 - Overriding Principle
 - Descriptive: Naming Styles
 - Prescriptive: Naming Conventions
 - Names to Avoid
 - ASCII Compatibility
 - Package and Module Names
 - Class Names
 - Type Variable Names
 - Exception Names
 - Global Variable Names
 - Function and Variable Names
 - Function and Method Arguments
 - Method Names and Instance Variables
 - Constants
 - Designing for Inheritance
 - Public and Internal Interfaces
 - Progressing Recommendations
 - Future Enhancements

PEP 8 – Style Guide for Python Code

Author: Guido van Rossum <guido at python.org>, Barry Warsaw <barry at python.org>, Alyssa Coghlan <cohoglan at gmail.com>

Status: Active

Type: Process

Created: 05-Jul-2001

Post-History: 05-Jul-2001, 01-Aug-2013

► Table of Contents

Introduction

This document gives coding conventions for the Python code comprising the standard library in the main Python distribution. Please see the companion informational PEP describing style guidelines for the C code in the C implementation of Python.

This document and [PEP 257](#) (Docstring Conventions) were adapted from Guido's original Python Style Guide essay, with some additions from Barry's style guide [2].

This style guide evolves over time as additional conventions are identified and past conventions are rendered obsolete by changes in the language itself.

Many projects have their own coding style guidelines. In the event of any conflicts, such project-specific guides take precedence for that project.

A Foolish Consistency is the Hobgoblin of Little Minds

One of Guido's key insights is that code is read much more often than it is written. The guidelines provided here are intended to improve the readability of code and make it consistent across the wide spectrum of Python code. As PEP 20 says, "Readability counts".

A style guide is about consistency. Consistency with this style guide is important. Consistency within a project is more important. Consistency within one module or function is the most important.

However, know when to be inconsistent – sometimes style guide recommendations just aren't applicable. When you run into one of those situations, look at other examples and decide what other style guide(s) that particular ask!

GOOD CODING PRACTICES: CODE FOR OTHERS

Program with style

- ▶ Avoid ambiguities

Usual

```
df['sex'] = np.where(df['gender'] ==  
'1001', 1, 2)
```

Better

```
df['female'] = np.where(df['gender'] ==  
'1001', 1, 0)  
df['male'] = np.where(df['gender'] !=  
'1001', 1, 0)
```

GOOD CODING PRACTICES: CODE FOR OTHERS

Program with style

- ▶ Avoid ambiguities
- ▶ Avoid changing units

Usual

```
df['gdp'] = df['gdp'] / 118.722
```

GOOD CODING PRACTICES: CODE FOR OTHERS

Program with style

- ▶ Avoid ambiguities
- ▶ Avoid changing units

Usual

```
df['gdp'] = df['gdp'] / 118.722
```

Better

```
df['gdp_US'] = df['gdp'] / 118.722
```

GOOD CODING PRACTICES: CODE FOR OTHERS

Program with style

- ▶ Avoid ambiguities
- ▶ Avoid changing units

Usual

```
df['gdp'] = df['gdp'] / 118.722
```

Even better

```
US_Vanu_exch_rate = 118.722
df['gdp_US'] = df['gdp'] /
US_Vanu_exch_rate
```

GOOD PRACTICES: MODULARITY

Create reusable objects

- ▶ Store values

Usual

```
Current_Data = Mydata[Mydata['year'] == 2023]
```

GOOD PRACTICES: MODULARITY

Create reusable objects

- ▶ Store values
 - Avoid repetitions
- Usual
 - Current_Data = Mydata[Mydata['year'] == 2023]
- Better
 - Current_year = 2023
 - Current_Data= Mydata[Mydata['year'] == Current_year]

GOOD PRACTICES: MODULARITY

Create reusable objects

- ▶ Store values
- Avoid repetitions

Usual

```
# - Exports for Beef -
data = Mydata[Mydata['export'] == 'Beef']
plt.plot(data['Year'], data['Value'])
plt.title('Export for Beef')

# - Also for Kava -
data = Mydata[Mydata['export'] == 'Kava']
plt.plot(data['Year'], data['Value'])
plt.title('Export for Kava')

# - Also for ... -
```

GOOD PRACTICES: MODULARITY

Create reusable objects

- ▶ Store values
- Avoid repetitions
- ▶ Use functions

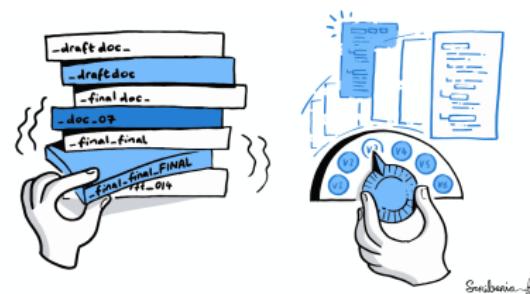
Better

```
# Defining a generic function
def plot_export(export_type):
    data = Mydata[Mydata['export'] ==
                           export_type]
    plt.plot(data['Year'], data['Value'])
    plt.title(f'Export for {export_type}')
    plt.show()
# Applying function to several products
plot_export("Beef")
plot_export("Kava")
```

VERSION CONTROL KEEPS TRACKS OF YOUR WORK

Tracking three W questions:

What changes?



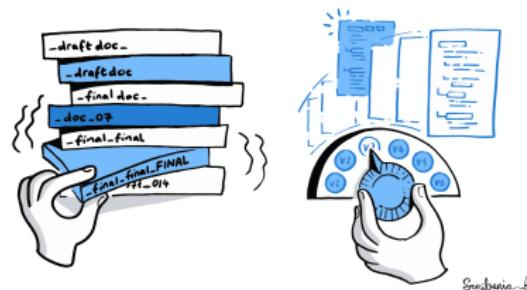
Source: The Turing Way project

VERSION CONTROL KEEPS TRACKS OF YOUR WORK

Tracking three W questions:

What changes?

Who made the changes?



Source: The Turing Way project

VERSION CONTROL KEEPS TRACKS OF YOUR WORK

Tracking three W questions:

What changes?

Who made the changes?

When were the changes made?



Source: The Turing Way project

VERSION CONTROL KEEPS TRACKS OF YOUR WORK

Tracking three W questions:

What changes?

Who made the changes?

When were the changes made?



Source: The Turing Way project

TRANSPARENCY, ACCOUNTABILITY & REPRODUCIBILITY

- ▶ Version control provides a detailed history of changes

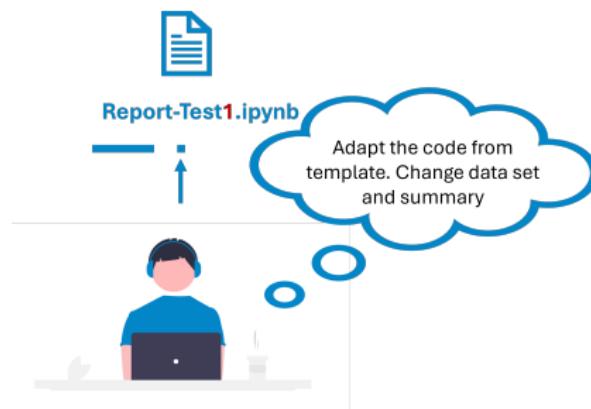
TRANSPARENCY, ACCOUNTABILITY & REPRODUCIBILITY

- ▶ Version control provides a detailed history of changes
- ▶ Each modification is attributed to a specific user

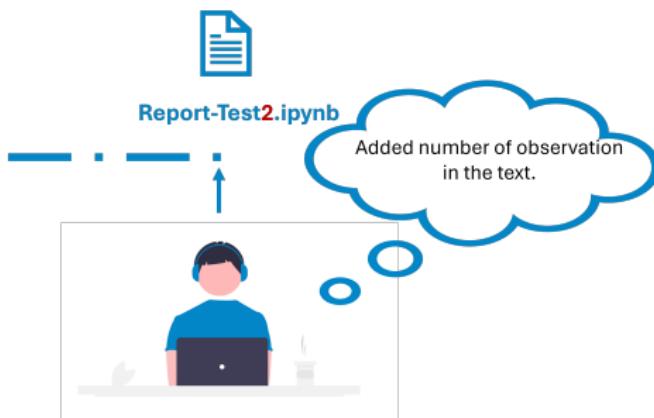
TRANSPARENCY, ACCOUNTABILITY & REPRODUCIBILITY

- ▶ Version control provides a detailed history of changes
- ▶ Each modification is attributed to a specific user
- ▶ Promotes accountability, transparency & reproducibility

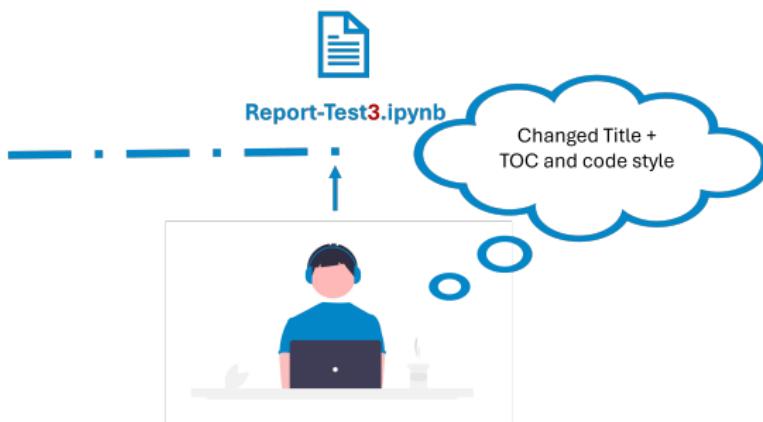
FILE EVOLUTION WITHOUT VERSION CONTROL



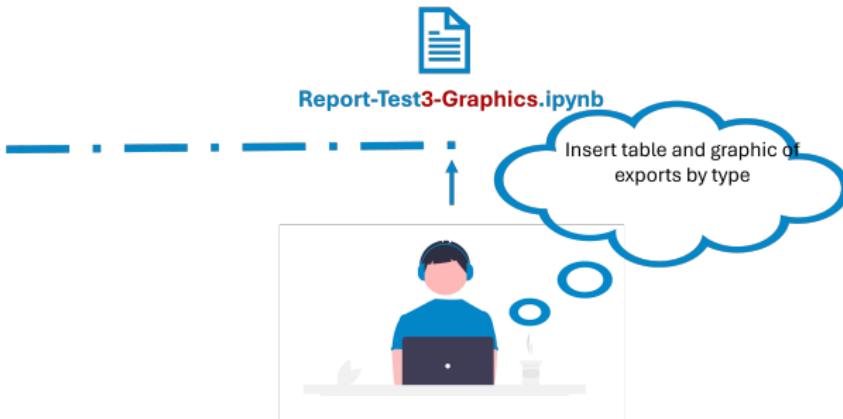
FILE EVOLUTION WITHOUT VERSION CONTROL



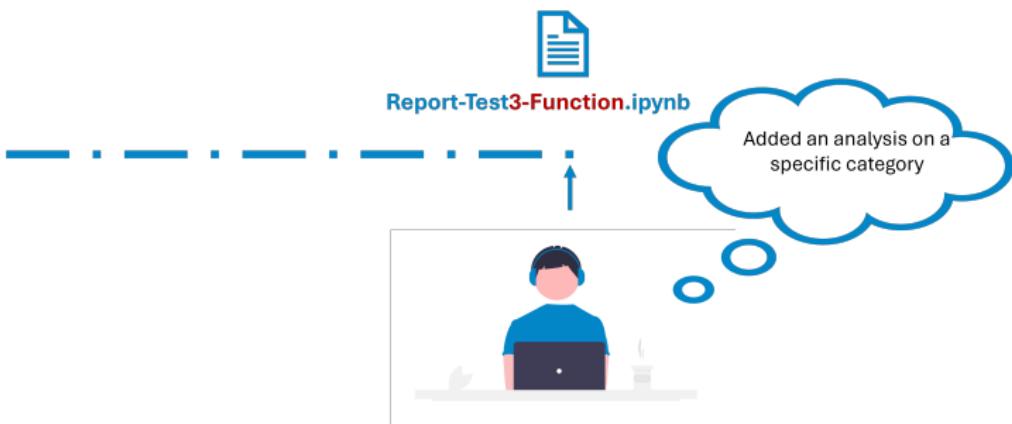
FILE EVOLUTION WITHOUT VERSION CONTROL



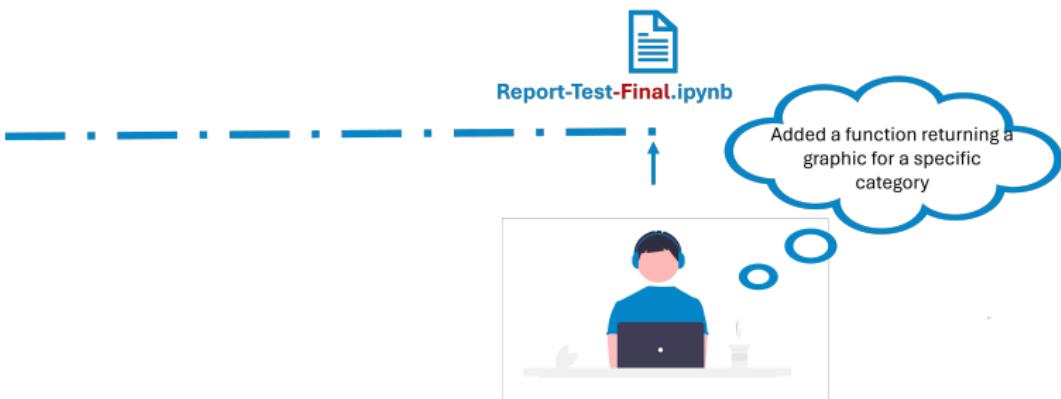
FILE EVOLUTION WITHOUT VERSION CONTROL



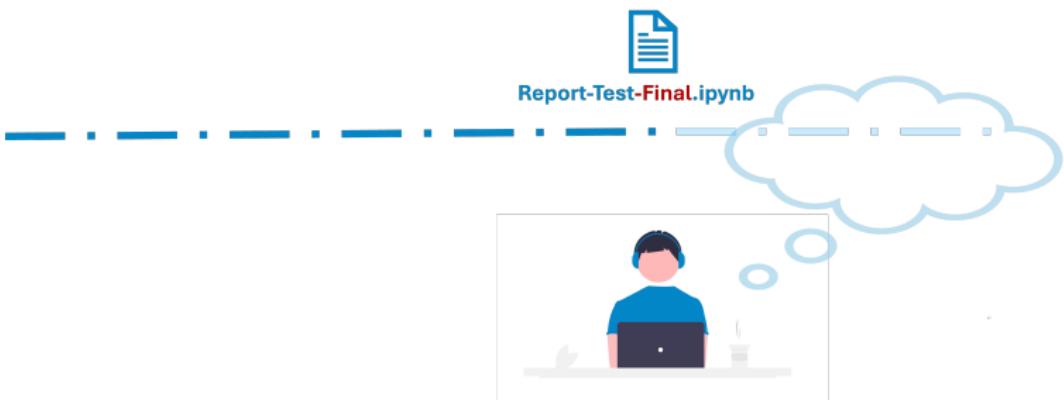
FILE EVOLUTION WITHOUT VERSION CONTROL



FILE EVOLUTION WITHOUT VERSION CONTROL



FILE EVOLUTION WITHOUT VERSION CONTROL



FILE EVOLUTION WITHOUT VERSION CONTROL

Usual ways to keep track of changes:

FILE EVOLUTION WITHOUT VERSION CONTROL

Usual ways to keep track of changes:

- ▶ New file after each change



[Report-Test1.ipynb](#)



[Report-Test3-Graphics.ipynb](#)



[Report-Test2.ipynb](#)



[Report-Test3-Function.ipynb](#)



[Report-Test3.ipynb](#)



[Report-Test-Final.ipynb](#)

FILE EVOLUTION WITHOUT VERSION CONTROL

Usual ways to keep track of changes:

- ▶ New file after each change
- Need to open each file to see the change



[Report-Test1.ipynb](#)



[Report-Test3-Graphics.ipynb](#)



[Report-Test2.ipynb](#)



[Report-Test3-Function.ipynb](#)



[Report-Test3.ipynb](#)



[Report-Test-Final.ipynb](#)

FILE EVOLUTION WITHOUT VERSION CONTROL

Usual ways to keep track of changes:

- ▶ New file after each change
- Need to open each file to see the change
- Names have to be explicit



[Report-Test1.ipynb](#)



[Report-Test3-Graphics.ipynb](#)



[Report-Test2.ipynb](#)



[Report-Test3-Function.ipynb](#)



[Report-Test3.ipynb](#)



[Report-Test-Final.ipynb](#)

FILE EVOLUTION WITHOUT VERSION CONTROL

Usual ways to keep track of changes:

- ▶ New file after each change
- ↳ Need to open each file to see the change
- ↳ Names have to be explicit
- ▶ Only the last file with lots of comments



Report-Test3-Graphics-
Functions-Final-
Chris.ipynb

FILE EVOLUTION WITHOUT VERSION CONTROL

Usual ways to keep track of changes:

- ▶ New file after each change
- ↳ Need to open each file to see the change
- ↳ Names have to be explicit
- ▶ Only the last file with lots of comments
- ▶ Not fulfilling the 3 W...



Report-Test3-Graphics-
Functions-Final-
Chris.ipynb

FILE EVOLUTION WITH VERSION CONTROL

You can see exactly what has been going on!

Showing 2 changed files with 325 additions and 24 deletions.

Filter changed files

prices_scrape/notbooks

session11_code.html

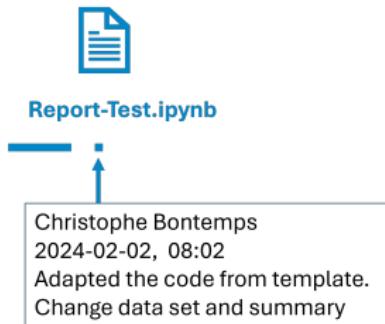
session11_code.pybm

Whitespace Ignore whitespace Split Unified

Line Number	Code	Line Number	Code
7531	</div>	7558	</div>
7532	<div class="jp-InputArea jp-Cell-InputArea"><div class="jp-InputPrompt jp-InputArea-prompt">	7559	<div class="jp-InputArea jp-Cell-InputArea"><div class="jp-InputPrompt jp-InputArea-prompt">
7533	</div><div class="jp-RenderedHTMLCommon jp-RenderedMarkdown jp-MarkdownOutput" data-mime-type="text/markdown">	7560	</div><div class="jp-RenderedHTMLCommon jp-RenderedMarkdown jp-MarkdownOutput" data-mime-type="text/markdown">
7534	+ <p>As usual, one first need some packages to be loaded</p>	7561	+ <h3 id="Defining-the-Website-URL">Defining the Website URL</h3>#It is good to have a look at the website beforehand and navigate a bit to see if the structure looks easy to navigate, and formatting consistent</p>
7535	</div>	7562	</div>
7536	</div>	7563	</div>
7537	</div>	7564	</div>
7538	+ @@ -7543,9 +7570,8 @@ <h2 id="Initial-scrap:Only-one-page">Initial scrap: Only one page<a class="anch	7565	<div class="jp-InputArea jp-Cell-InputArea-prompt">[/]</div>
7539	<div class="jp-InputPrompt jp-InputArea-prompt">[/]</div>	7566	<div class="jp-CodeMirrorEditor jp-Editor jp-InputArea-editor" data-type="inline">
7540	<div class="jp-CodeMirrorEditor jp-Editor jp-InputArea-editor" data-type="inline">	7567	<div class="cm-editor cm-jupyter">
7541	<div class="highlight hi-python3"><pre>	7568	+ <div class="highlight hi-python3"><pre>
7542	import as	7569	# Define the URL of the website
7543	pd</pre></div>	7570	+ #= https://www.farmers.co.nz/women/fashion/tops
7544	</div>	7571	</div>
7545	</div>	7572	<div class="cm-editor cm-jupyter">
7546	+ <div class="highlight hi-python3"><pre>	7573	+ <div class="highlight hi-python3"><pre>
7547	from as	7574	# Define the URL of the website
7548	import as	7575	+ #= https://www.farmers.co.nz/women/fashion/tops
7549	pandasaspd	7576	</pre></div>
7550	</pre></div>	7577	</div>
7551	</div>	7578	</div>
7552	+ @@ -7558,7 +7584,7 @@ <h2 id="Initial-scrap:Only-one-page">Initial scrap: Only one page<a class="anch	7579	<div class="jp-InputArea jp-Cell-InputArea"><div class="jp-InputPrompt jp-InputArea-prompt">
7553	<div class="jp-InputArea jp-Cell-InputArea"><div class="jp-InputPrompt jp-InputArea-prompt">	7580	</div><div class="jp-RenderedHTMLCommon jp-RenderedMarkdown jp-MarkdownOutput" data-mime-type="text/markdown">
7554	</div><div class="jp-RenderedHTMLCommon jp-RenderedMarkdown jp-MarkdownOutput" data-mime-type="text/markdown">	7581	+ <h3 id="Testing-the-website">Testing the website</h3>#
7555	</div>	7582	</div>
7556	</div>	7583	</div>
7557	</div>	7584	</div>
7558	</div>	7585	<div class="jp-InputArea jp-Cell-InputArea"><div class="jp-InputPrompt jp-InputArea-prompt">
7559	</div>	7586	</div><div class="jp-RenderedHTMLCommon jp-RenderedMarkdown jp-MarkdownOutput" data-mime-type="text/markdown">
7560	</div>	7587	+ <h3 id="Testing-the-website">Testing the website</h3>#

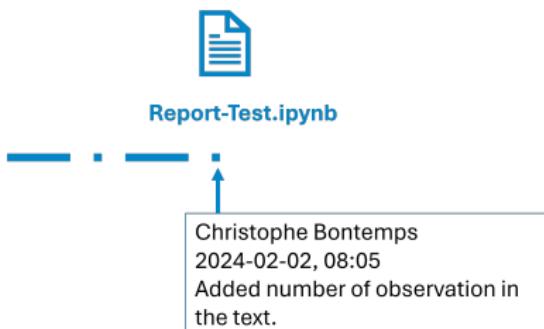
FILE EVOLUTION WITH VERSION CONTROL

Record a message (*commit*) for each change!



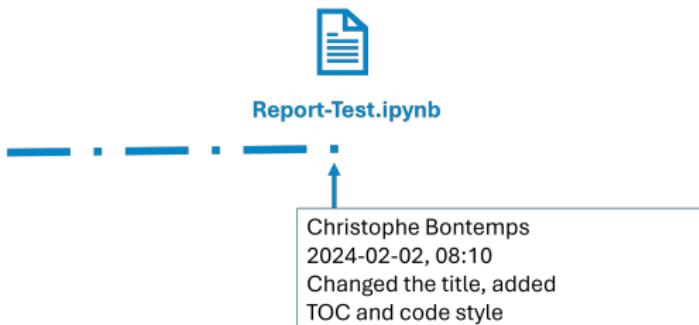
FILE EVOLUTION WITH VERSION CONTROL

Record a message (*commit*) for each change!



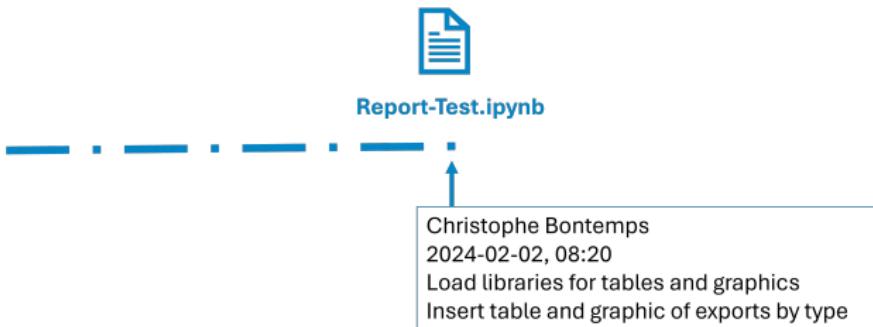
FILE EVOLUTION WITH VERSION CONTROL

Record a message (*commit*) for each change!



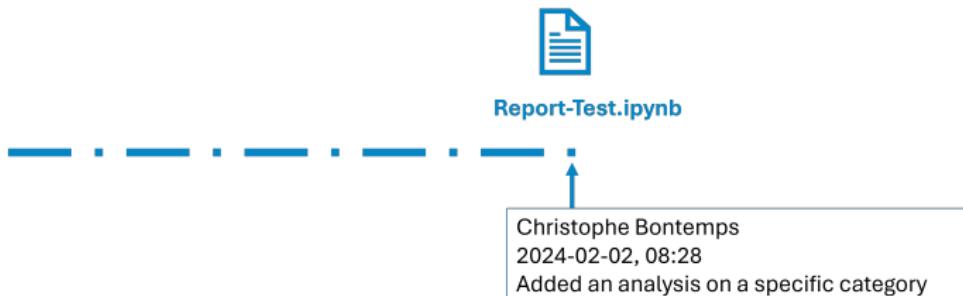
FILE EVOLUTION WITH VERSION CONTROL

Record a message (*commit*) for each change!



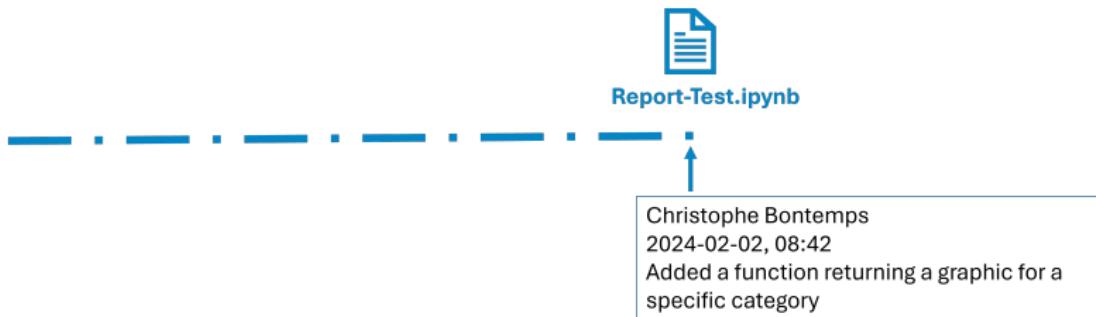
FILE EVOLUTION WITH VERSION CONTROL

Record a message (*commit*) for each change!



FILE EVOLUTION WITH VERSION CONTROL

Record a message (*commit*) for each change!



FILE EVOLUTION WITH VERSION CONTROL

Record a message (*commit*) for each change!



Report-Test.ipynb



THE HISTORY OF THE FILE IS RECORDED!

Each version is documented (with *commits*)



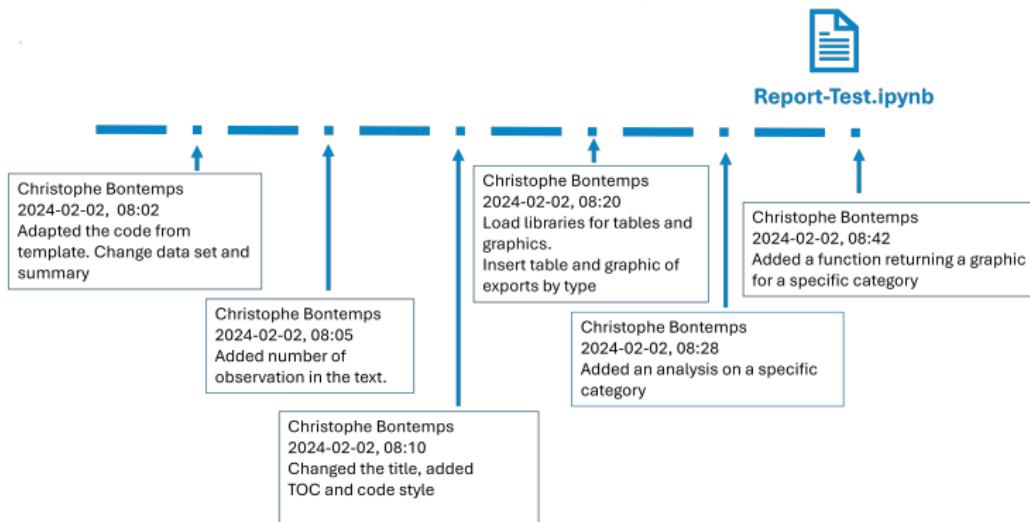
THE HISTORY OF THE FILE IS RECORDED!

Each version is documented (with *commits*)



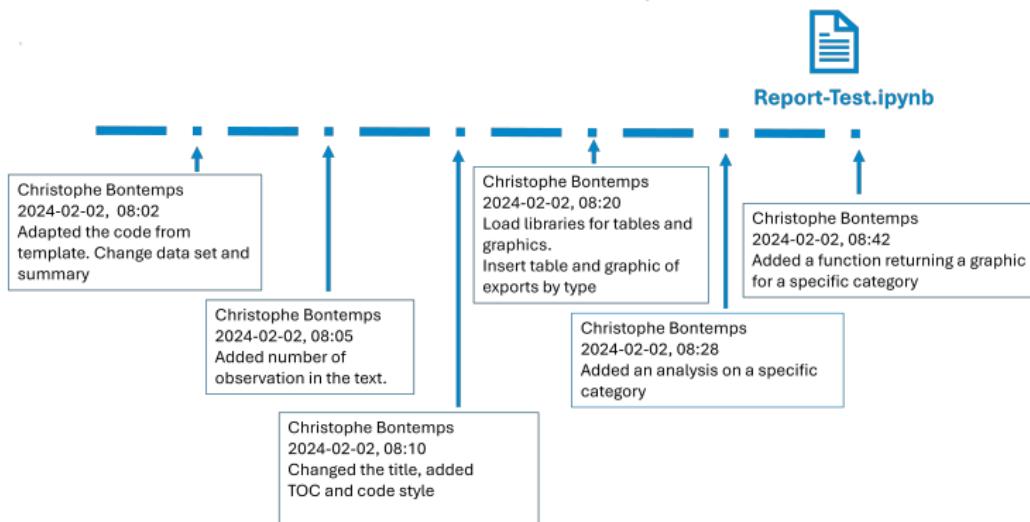
THE HISTORY OF THE FILE IS RECORDED!

Each version embeds the full history!



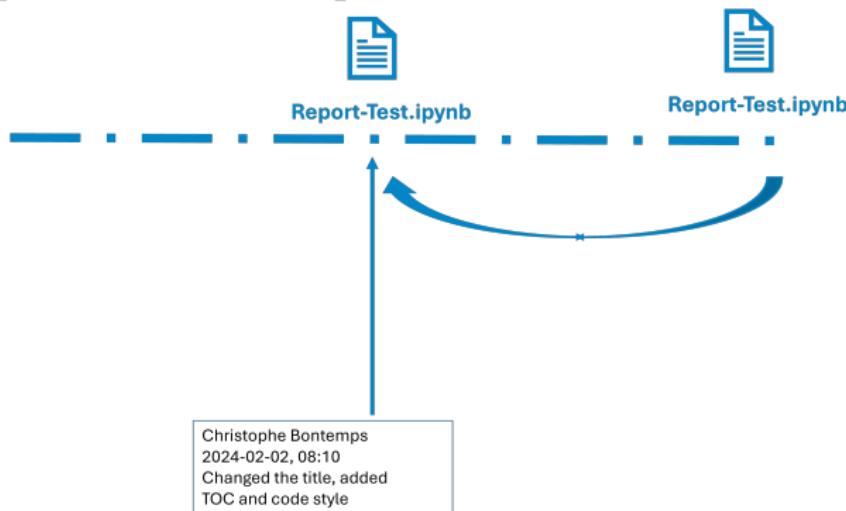
THE HISTORY OF THE FILE IS RECORDED!

Each version embeds the full history!



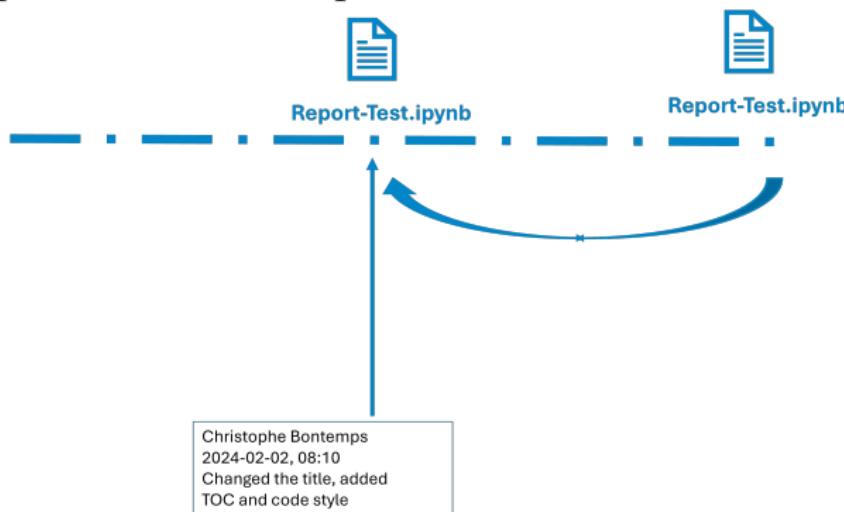
GOING BACK (*revert*) IS POSSIBLE

It is possible to review previous version...



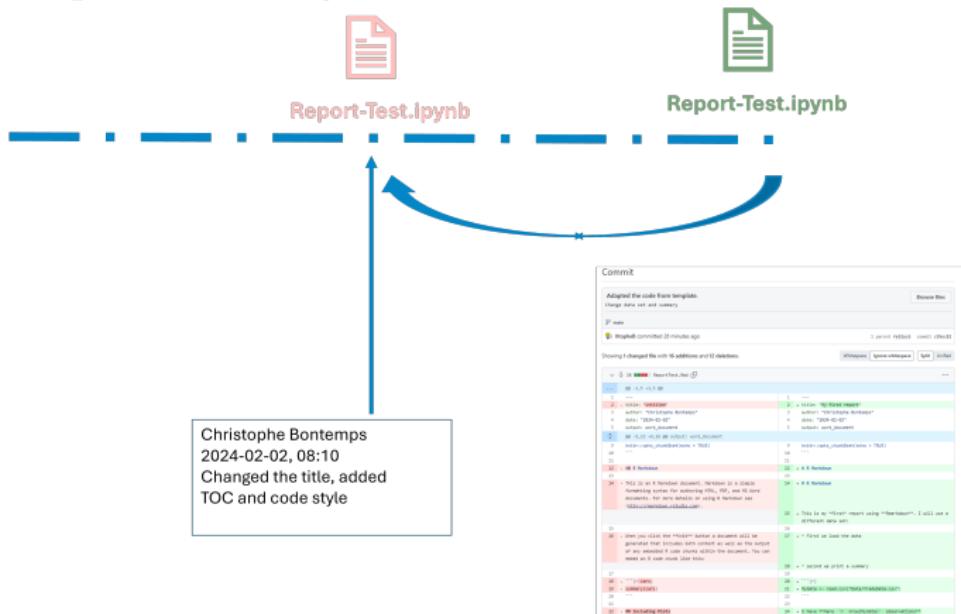
GOING BACK (*revert*) IS POSSIBLE

It is possible to review previous version...



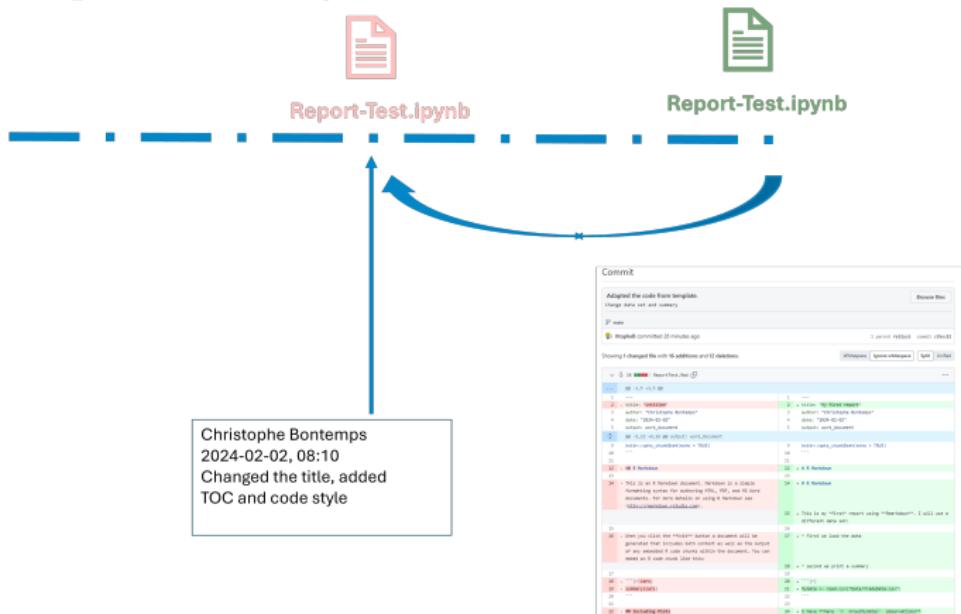
GOING BACK (*revert*) IS POSSIBLE

...to compare the changes...



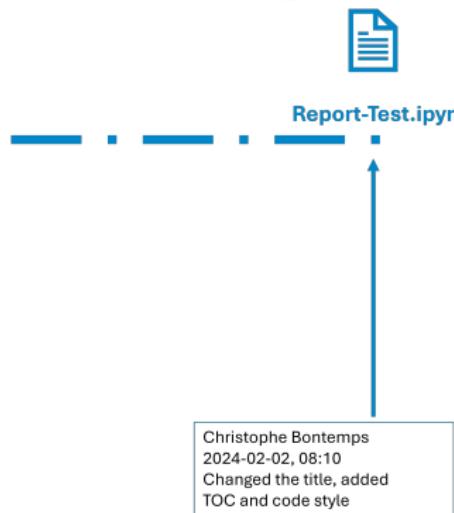
GOING BACK (*revert*) IS POSSIBLE

...to compare the changes...



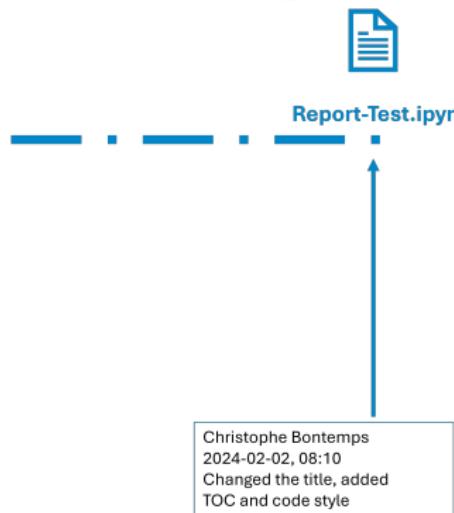
GOING BACK (*revert*) IS POSSIBLE

... and to revert to a previous version...



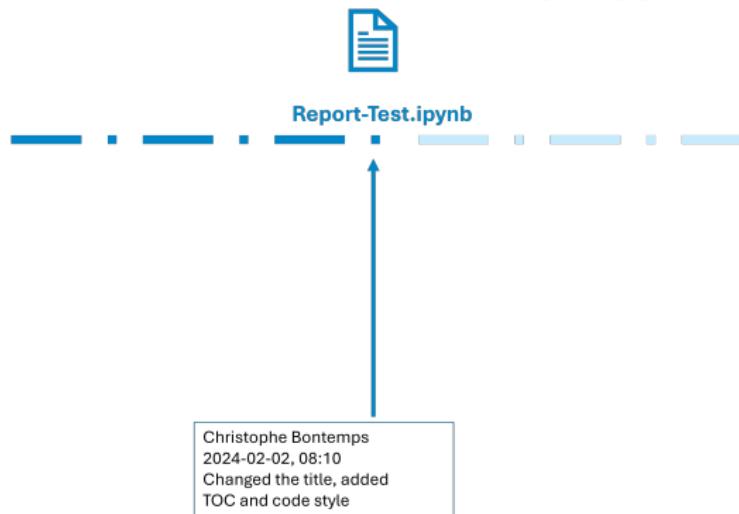
GOING BACK (*revert*) IS POSSIBLE

... and to revert to a previous version...



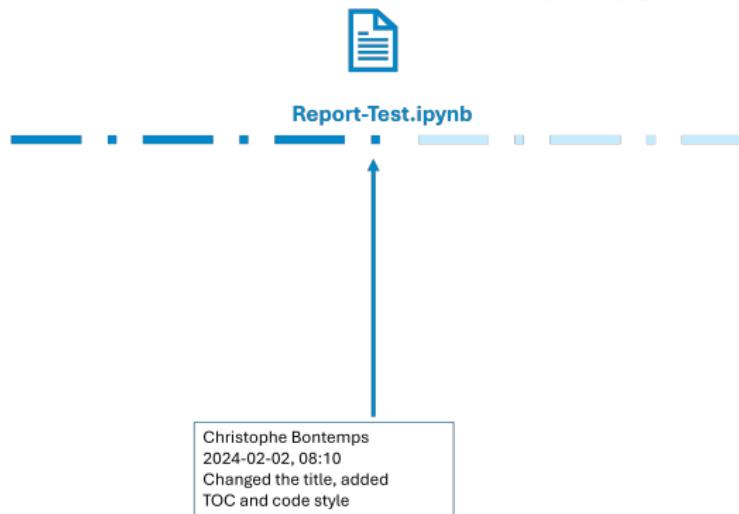
GOING BACK (*revert*) IS POSSIBLE

... or *restart* from there as if nothing happened



GOING BACK (*revert*) IS POSSIBLE

... or *restart* from there as if nothing happened



Motivation
oooo

Issues
oo

RAP
oo

Principles
oooooooooooo

Version Control
ooooooo●o

Takeaways
o

Resources
o

GOOD NEWS!

Version Control will help you

GOOD NEWS!

Version Control will help you

- ▶ Version Control is integrated in Visual Studio (& RStudio)

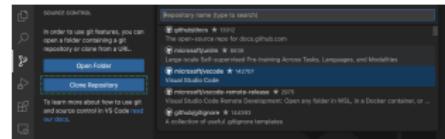


"Visual Studio Documentation"

GOOD NEWS!

Version Control will help you

- ▶ Version Control is integrated in Visual Studio (& RStudio)
- Simple operations are easy

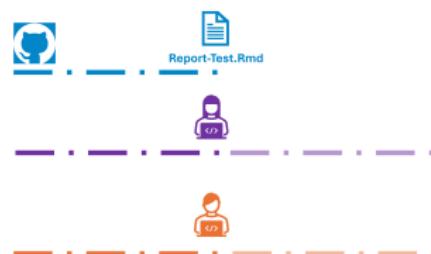


"Visual Studio Documentation"

GOOD NEWS!

Version Control will help you

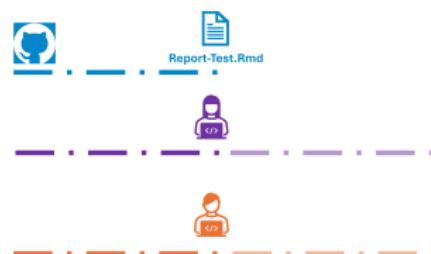
- ▶ Version Control is integrated in Visual Studio (& RStudio)
- ↪ Simple operations are easy
- ▶ Collaborate on a project



GOOD NEWS!

Version Control will help you

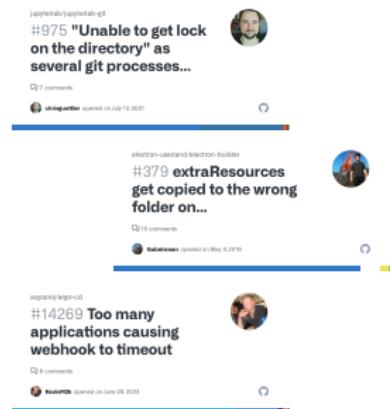
- ▶ Version Control is integrated in Visual Studio (& RStudio)
- ↪ Simple operations are easy
- ▶ Collaborate on a project
- ↪ Track changes of others



GOOD NEWS!

Version Control will help you

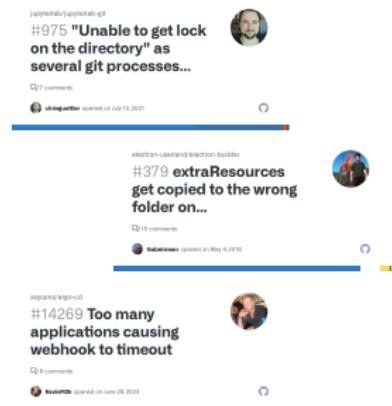
- ▶ Version Control is integrated in Visual Studio (& RStudio)
- Simple operations are easy
- ▶ Collaborate on a project
- Track changes of others
- ▶ Git seems “*unfriendly*” but it is your friend



GOOD NEWS!

Version Control will help you

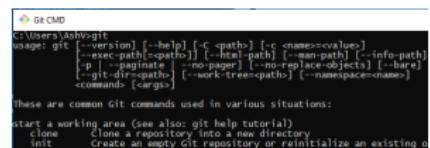
- ▶ Version Control is integrated in Visual Studio (& RStudio)
- Simple operations are easy
- ▶ Collaborate on a project
- Track changes of others
- ▶ Git seems “*unfriendly*” but it is your friend
- Takes time and patience



GOOD NEWS!

Version Control will help you

- ▶ Version Control is integrated in Visual Studio (& RStudio)
- ↪ Simple operations are easy
- ▶ Collaborate on a project
- ↪ Track changes of others
- ▶ Git seems “*unfriendly*” but it is your friend
- ↪ Takes time and patience
- ▶ Git works *mostly* in command mode



```
git CMD
l:\Users\Ashish\git
usage: git [--help] [--version] [-c <name>=<value>]
   [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
   [--git-dir[=<path>]] [--work-tree[=<path>]] [--namespace[=<name>]]
   [--command[=<args>]]

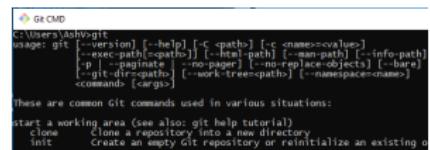
These are common Git commands used in various situations:
start a working area (see also: git help tutorial)
  clone      clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one
```

Ashish Vishwakarma

GOOD NEWS!

Version Control will help you

- ▶ Version Control is integrated in Visual Studio (& RStudio)
- ↪ Simple operations are easy
- ▶ Collaborate on a project
- ↪ Track changes of others
- ▶ Git seems “*unfriendly*” but it is your friend
- ↪ Takes time and patience
- ▶ Git works *mostly* in command mode
- ↪ You will learn that too!



```
git CMD
l:\Users\Ashish\git
usage: git [--help] [--version] [-c <name>=<value>]
   [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
   [--git-dir[=<path>]] [--work-tree[=<path>]] [--namespace[=<name>]
   [--command[=<args>]]

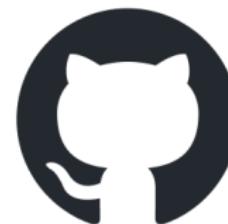
These are common Git commands used in various situations:
start a working area (see also: git help tutorial)
  clone      clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one
```

Ashish Vishwakarma

VERSION CONTROL IN A NUTSHELL

A Version Control systems:

- ▶ Keeps track of all changes



GitHub logo

VERSION CONTROL IN A NUTSHELL

A Version Control systems:

- ▶ Keeps track of all changes
- ▶ Ignores your data (`.gitignore`)



GitHub logo

VERSION CONTROL IN A NUTSHELL

A Version Control systems:

- ▶ Keeps track of all changes
- ▶ Ignores your data (`.gitignore`)
- ▶ Allows reviewing stages of development



GitHub logo

VERSION CONTROL IN A NUTSHELL

A Version Control systems:

- ▶ Keeps track of all changes
- ▶ Ignores your data (`.gitignore`)
- ▶ Allows reviewing stages of development
- ▶ Allow collaborating on projects



GitHub logo

VERSION CONTROL IN A NUTSHELL

A Version Control systems:

- ▶ Keeps track of all changes
- ▶ Ignores your data (`.gitignore`)
- ▶ Allows reviewing stages of development
- ▶ Allow collaborating on projects
- ▶ Comes with different tools (Git, GitHub, GitLab, etc..)



GitHub logo

VERSION CONTROL IN A NUTSHELL

A Version Control systems:

- ▶ Keeps track of all changes
- ▶ Ignores your data (`.gitignore`)
- ▶ Allows reviewing stages of development
- ▶ Allow collaborating on projects
- ▶ Comes with different tools (Git, GitHub, GitLab, etc..)
- ▶ Backups your work



GitHub logo

Motivation
oooo

Issues
oo

RAP
oo

Principles
oooooooooooo

Version Control
oooooooooooo

Takeaways
●

Resources
○

TAKEAWAYS

TAKEAWAYS

- ▶ There many levels of RAP (a full spectrum)

TAKEAWAYS

- ▶ There many levels of RAP (a full spectrum)
- ↪ Start small, be an advocate for others

TAKEAWAYS

- ▶ There many levels of RAP (a full spectrum)
 - ↳ Start small, be an advocate for others
 - ↳ Increase complexity when ready

TAKEAWAYS

- ▶ There many levels of RAP (a full spectrum)
- ↳ Start small, be an advocate for others
- ↳ Increase complexity when ready
- ▶ Good practices starts with our own practices

TAKEAWAYS

- ▶ There many levels of RAP (a full spectrum)
- ↪ Start small, be an advocate for others
- ↪ Increase complexity when ready
- ▶ Good practices starts with our own practices
- ↪ KISS: Keep it Simple, Stupid

TAKEAWAYS

- ▶ There many levels of RAP (a full spectrum)
 - ↳ Start small, be an advocate for others
 - ↳ Increase complexity when ready
- ▶ Good practices starts with our own practices
 - ↳ KISS: Keep it Simple, Stupid
- ▶ Automate little by little

TAKEAWAYS

- ▶ There many levels of RAP (a full spectrum)
 - ↳ Start small, be an advocate for others
 - ↳ Increase complexity when ready
- ▶ Good practices starts with our own practices
 - ↳ KISS: Keep it Simple, Stupid
- ▶ Automate little by little
- ▶ Version control is a life-changer

TAKEAWAYS

- ▶ There many levels of RAP (a full spectrum)
- ↳ Start small, be an advocate for others
- ↳ Increase complexity when ready
- ▶ Good practices starts with our own practices
- ↳ KISS: Keep it Simple, Stupid
- ▶ Automate little by little
- ▶ Version control is a life-changer
- ▶ Building a RAP is a collective process



USEFUL RESOURCES

- ▶ NHS Community of Practice
- ▶ This course website (created by Serge Goussev)
- ▶ Vanuatu Bureau of Statistics implementation of RAP
- ▶ SIAP's (free) online RAP course
- ▶ The UK government RAP website.
- ▶ UK best practice documentation.
- ▶ A free RAP course to teach you all you need to know.
- ▶ How the Data Science Campus sets its coding standards.
- ▶ A new open-source book from the Alan Turing institute setting out how to do reproducible data science.