Python – A shot of inspiration for day to day analysis

Vipin Ajayakumar

May 31, 2017

Contents

My programming background

- Comp Sci at A-levels (fundamentals, intro to Visual Basic)

My programming background

- Comp Sci at A-levels (fundamentals, intro to Visual Basic)
- Aerospace engineering degree (not taught much comp sci)

My programming background

- Comp Sci at A-levels (fundamentals, intro to Visual Basic)
- Aerospace engineering degree (not taught much comp sci)
- Used VBA in an internship to automate a data entry task

My programming background

- Comp Sci at A-levels (fundamentals, intro to Visual Basic)
- Aerospace engineering degree (not taught much comp sci)
- Used VBA in an internship to automate a data entry task
- Used Matlab for thesis to model lift with discrete vortices

My programming background

- Comp Sci at A-levels (fundamentals, intro to Visual Basic)
- Aerospace engineering degree (not taught much comp sci)
- Used VBA in an internship to automate a data entry task
- Used Matlab for thesis to model lift with discrete vortices
- Self-taught Python in final attachment of grad scheme to automate a data analysis task

My programming background

- Comp Sci at A-levels (fundamentals, intro to Visual Basic)
- Aerospace engineering degree (not taught much comp sci)
- Used VBA in an internship to automate a data entry task
- Used Matlab for thesis to model lift with discrete vortices
- Self-taught Python in final attachment of grad scheme to automate a data analysis task
- Currently, using Python for day to day tasks and for improvements

My programming background

- Comp Sci at A-levels (fundamentals, intro to Visual Basic)
- Aerospace engineering degree (not taught much comp sci)
- Used VBA in an internship to automate a data entry task
- Used Matlab for thesis to model lift with discrete vortices
- Self-taught Python in final attachment of grad scheme to automate a data analysis task
- Currently, using Python for day to day tasks and for improvements
- Learning JavaScript, LaTeX, Python, Machine Learning etc. at home

My programming background

- Comp Sci at A-levels (fundamentals, intro to Visual Basic)
- Aerospace engineering degree (not taught much comp sci)
- Used VBA in an internship to automate a data entry task
- Used Matlab for thesis to model lift with discrete vortices
- Self-taught Python in final attachment of grad scheme to automate a data analysis task
- Currently, using Python for day to day tasks and for improvements
- Learning JavaScript, LaTeX, Python, Machine Learning etc. at home
- In summary, barely scratched the surface, but I can share the little I know. :)

My programming background

- Comp Sci at A-levels (fundamentals, intro to Visual Basic)
- Aerospace engineering degree (not taught much comp sci)
- Used VBA in an internship to automate a data entry task
- Used Matlab for thesis to model lift with discrete vortices
- Self-taught Python in final attachment of grad scheme to automate a data analysis task
- Currently, using Python for day to day tasks and for improvements
- Learning JavaScript, LaTeX, Python, Machine Learning etc. at home
- In summary, barely scratched the surface, but I can share the little I know. :)
- If I can code, then surely you can do it even better!

Why code rather than use MS Excel?

- Increasing complexity of analyses

Source

Why code rather than use MS Excel?

- Increasing complexity of analyses
- Spreadsheets cannot cope

Source

Why code rather than use MS Excel?

- Increasing complexity of analyses
- Spreadsheets cannot cope
- Save time

Source

Why code rather than use MS Excel?

- Increasing complexity of analyses
- Spreadsheets cannot cope
- Save time
- Code exposes logic, a spreadsheet exposes the data

Source

Why code rather than use MS Excel?

- Increasing complexity of analyses
- Spreadsheets cannot cope
- Save time
- Code exposes logic, a spreadsheet exposes the data
- Reduce human error

Source

Why code rather than use MS Excel?

- Increasing complexity of analyses
- Spreadsheets cannot cope
- Save time
- Code exposes logic, a spreadsheet exposes the data
- Reduce human error
- Improve capability

Source

Why code rather than use MS Excel?

- Increasing complexity of analyses
- Spreadsheets cannot cope
- Save time
- Code exposes logic, a spreadsheet exposes the data
- Reduce human error
- Improve capability
- Enable continuous abstraction

Source

Why code rather than use MS Excel?

- Increasing complexity of analyses
- Spreadsheets cannot cope
- Save time
- Code exposes logic, a spreadsheet exposes the data
- Reduce human error
- Improve capability
- Enable continuous abstraction
- Focus on the more important tasks

Source

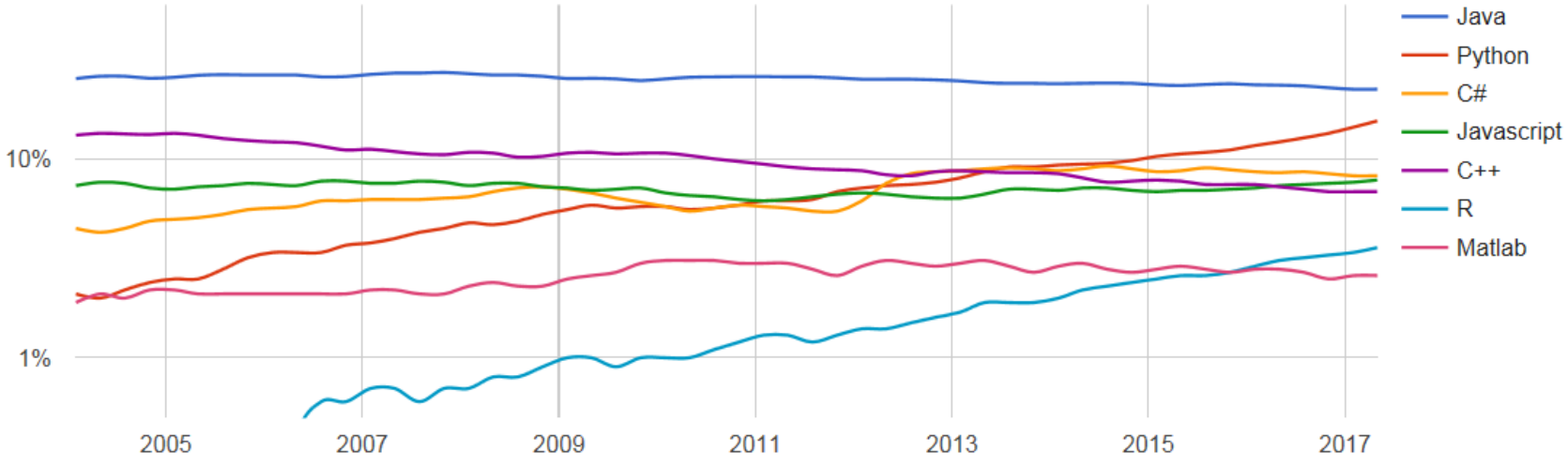
What is Python?

Courtesy of Wikipedia:

- Python is a programming language
- Created by Guido van Rossum in 1991
- Two relevant versions: Python 2.7 and Python 3.x
- Python 2.x will not be developed much more

Why Python?

- Python is very popular and still gaining in popularity.
- Therefore, it has a very active and supportive community.



PopularitY of Programming Language [Carbonnelle, 2016]

Why Python?

- Beginner friendly

Source

Why Python?

- Beginner friendly
- Beautiful syntax

Source

Why Python?

- Beginner friendly
- Beautiful syntax
- Dynamically typed

Source

Why Python?

- Beginner friendly
- Beautiful syntax
- Dynamically typed
- Object Oriented

Source

Why Python?

- Beginner friendly
- Beautiful syntax
- Dynamically typed
- Object Oriented
- Interpreted

Source

Why Python?

- Beginner friendly
- Beautiful syntax
- Dynamically typed
- Object Oriented
- Interpreted
- Machine learning support

Source

Why Python?

- Beginner friendly
- Beautiful syntax
- Dynamically typed
- Object Oriented
- Interpreted
- Machine learning support
- Cross platform

Source

Why Python?

- Beginner friendly
- Beautiful syntax
- Dynamically typed
- Object Oriented
- Interpreted
- Machine learning support
- Cross platform
- Open Source

Source

Why Python?

- Beginner friendly
- Beautiful syntax
- Dynamically typed
- Object Oriented
- Interpreted
- Machine learning support
- Cross platform
- Open Source
- Large and active Stack Overflow community

Source

Learning resources

1. Treehouse - Python course

Python course [Treehouse]

Learning resources

1. Treehouse - Python course
2. Intro to Comp Sci ... in Python at MIT

Python course [Treehouse]

Learning resources

1. Treehouse - Python course
2. Intro to Comp Sci ... in Python at MIT
3. Harvard CS50 course

Python course [Treehouse]

Learning resources

1. Treehouse - Python course
2. Intro to Comp Sci ... in Python at MIT
3. Harvard CS50 course
4. Clean Code - An essential book on good practice

Python course [Treehouse]

Learning resources

1. Treehouse - Python course
2. Intro to Comp Sci ... in Python at MIT
3. Harvard CS50 course
4. Clean Code - An essential book on good practice
5. Python for data analysis - A good reference book

Python course [Treehouse]

Learning resources

1. Treehouse - Python course
2. Intro to Comp Sci ... in Python at MIT
3. Harvard CS50 course
4. Clean Code - An essential book on good practice
5. Python for data analysis - A good reference book
6. Stack Overflow - Q&A site

Python course [Treehouse]

Learning resources

1. Treehouse - Python course
2. Intro to Comp Sci ... in Python at MIT
3. Harvard CS50 course
4. Clean Code - An essential book on good practice
5. Python for data analysis - A good reference book
6. Stack Overflow - Q&A site
7. sentdex or other YouTube channels

Python course [Treehouse]

Learning resources

1. Treehouse - Python course
2. Intro to Comp Sci ... in Python at MIT
3. Harvard CS50 course
4. Clean Code - An essential book on good practice
5. Python for data analysis - A good reference book
6. Stack Overflow - Q&A site
7. sentdex or other YouTube channels
8. Udemy - Online courses

Python course [Treehouse]

Recommended packages and tools

Python packages:

- os provides operating system interfaces

Tools:

Recommended packages and tools

Python packages:

- os provides operating system interfaces
- re provides expression matching operations

Tools:

Recommended packages and tools

Python packages:

- os provides operating system interfaces
- re provides expression matching operations
- pandas is a powerful data analysis library that provides the DataFrame object

Tools:

Recommended packages and tools

Python packages:

- os provides operating system interfaces
- re provides expression matching operations
- pandas is a powerful data analysis library that provides the DataFrame object
- numpy provides a powerful N-dimensional array object

Tools:

Recommended packages and tools

Python packages:

- os provides operating system interfaces
- re provides expression matching operations
- pandas is a powerful data analysis library that provides the DataFrame object
- numpy provides a powerful N-dimensional array object
- matplotlib.pyplot provides a matlab like plotting framework

Tools:

Recommended packages and tools

Python packages:

- os provides operating system interfaces
- re provides expression matching operations
- pandas is a powerful data analysis library that provides the DataFrame object
- numpy provides a powerful N-dimensional array object
- matplotlib.pyplot provides a matlab like plotting framework

Tools:

- PyCharm or Visual Studio Code for editing

Recommended packages and tools

Python packages:

- os provides operating system interfaces
- re provides expression matching operations
- pandas is a powerful data analysis library that provides the DataFrame object
- numpy provides a powerful N-dimensional array object
- matplotlib.pyplot provides a matlab like plotting framework

Tools:

- PyCharm or Visual Studio Code for editing
- git for version control

Recommended packages and tools

Python packages:

- os provides operating system interfaces
- re provides expression matching operations
- pandas is a powerful data analysis library that provides the DataFrame object
- numpy provides a powerful N-dimensional array object
- matplotlib.pyplot provides a matlab like plotting framework

Tools:

- PyCharm or Visual Studio Code for editing
- git for version control
- MS Visio to create diagrams to aid Systems Engineering

Typical data analysis procedure

1. Parse a text file

Typical data analysis procedure

1. Parse a text file
2. Manipulate the data

Typical data analysis procedure

1. Parse a text file
2. Manipulate the data
3. Display graphs

Weaknesses to be aware of

- Slower processing than lower level languages

Weaknesses to be aware of

- Slower processing than lower level languages
- Parallelisation possible but not native and not easy

Weaknesses to be aware of

- Slower processing than lower level languages
- Parallelisation possible but not native and not easy
- Backward compatibility not guaranteed

Conclusion

- There is a strong case for learning and using Python

Conclusion

- There is a strong case for learning and using Python
- Links provided to various Python learning resources

Conclusion

- There is a strong case for learning and using Python
- Links provided to various Python learning resources
- A number of key packages can make typical data analysis tasks very easy

Conclusion

- There is a strong case for learning and using Python
- Links provided to various Python learning resources
- A number of key packages can make typical data analysis tasks very easy
- Potential to expand beyond 'just data analysis' to more powerful Machine Learning applications

Conclusion

- There is a strong case for learning and using Python
- Links provided to various Python learning resources
- A number of key packages can make typical data analysis tasks very easy
- Potential to expand beyond 'just data analysis' to more powerful Machine Learning applications
- If you need help, please feel free to get in touch!

References

Pierre Carbonnelle. Popularity of programming language. http://pypl.github.io/PYPL.html, 2016.

Treehouse. Treehouse. https://teamtreehouse.com/.