Early Access

OK, Python

NEWLY EXPANDED

Starring: Python, Pandas and NumPy

Automate repetitive tasks in Excel, let Python do the work for you and simplify your life.

Excerpt from book OK, Python

Examples you can just copy and use.

Lubomír Husar
Founder of LovelyData.cz

Have your say about the contents of this book and write to me your suggestions for other examples. Primarily, I am interested in those you deal with at work daily. Contact me at lubomir@lovelydata.cz.

OK, Python.

From Excel to Python: The practical guide

Early access, version 1.0

Cover design and graphic layout by www.grafikli.cz.

Icons in the *Test data* and *Solution* sections have been designed by means of sources at Flaticon.com.

About the author

Lubomír "Lubo" Husar

Founder of LovelyData.cz.

Lubomír has years of experience in the field of data. He helped clients, mostly large multinational companies, to succeed in many critical projects. In addition to IT, he is also active in other fields. He lived and worked in various European countries, such as Belgium, Denmark, the Netherlands, Sweden and France. He has extensive experience in managerial and technical roles.

Acknowledgements

I would like to thank you for making a clever decision and investing in a book which will make your life easier.

Since I believe that computers should make people's lives easier not harder, I wrote this book which I have carried around in my head for a long time. And Python is to blame.

Just realize that only a few years ago it was almost impossible to process data simply in various formats, analyze them, clean, visualize and save them in various formats again. For such tasks, you needed commercial software and an open space full of consultants who could cope with it.

Fortunately, all this is changing very quickly, and Python is playing a major role in this change. Actually, is there any other tool able to automate work for sure in a few lines and for free at that? This is an offer you can't turn down!

LH

Contents

Introduction

This book is a manual for all who work with Excel and want to learn to use Python.
This means for all those who are tired of excessive mouse clicking, *copy* and *paste* operations and all that manual work they have to do every day.

Instructions in the book show how to solve these tasks quickly and elegantly, by means of several code lines. And all of that thanks to the Python programming language and Pandas and NumPy libraries.

Numerous examples of Python, Pandas and NumPy can be found on the Internet. They mostly describe a single specific problem and its solution. What I personally miss, however, is a comprehensive set of examples which can be used in practice, examples which you can easily modify and adapt, examples which you can easily understand although you are not a programmer.

And this is the very objective of this manual. I would be pleased if you wrote to me at lubomir@lovelydata.cz and let me know how I succeeded.

Python conquers the world

Python has literally stormed the world. Just have a look at public inquiries and ratings measuring the popularity of programming languages. You can bet that Python will be ranked high.

And it is no wonder. Over the last several years, Python has established itself as the language for data. Whether you need to analyze, clean and visualize data or use them to train artificial intelligence – you can find Python everywhere.

Knowledge in Python has thus become a must not only for programmers and analysts, but in fact for all who want to facilitate their work with data.

There is a good reason why Python is so popular. It is very friendly to beginners, who can take their first steps in it quickly. And the more advanced, again, can make use of the enormous ecosystem of libraries which significantly extend Python's potentials.

For whom this book is intended

The book will most of all benefit those with at least essential knowledge in Python. If they have come across the Pandas library, it is a plus. If not, never mind. The examples will lead them in the right direction.

Practical instructions in the book, however, can also be used by those who work with Python and Pandas already. But perhaps not so efficiently as they could.

The input and output format to be used is Excel (.xlsx format). The reason is that data in Excel can be found in every company. This format is so flexible as to enable its users to be enormously creative. Sometimes, however, this creativity must be tamed, and it is exactly in this way that Python will help you.

The book assumes that you have Python 3 installed on your PC together with several libraries listed below.

If you want to make the best of the book, I recommend that you transcribe the examples manually, line by line. Thus you will understand them better. No one will stop you, however, from merely copying, pasting and starting up the examples.

For whom this book is not intended

This book is not suitable for absolute beginners with no previous experience in Python. To those I can recommend the on-line course Python for analysts on the website LovelyData.cz.

Organization of individual chapters

Every example is based on tasks which can be solved by means of Excel. The aim is to show how Python and Pandas can be used for these tasks. The result is an easily comprehensible code, mostly consisting of a few lines only, which you can easily adapt to your specific needs.

Test data

This book does not contain test data files on purpose. You can generate data in the `XLSX` format yourself by means of NumPy and Pandas libraries. The Excel file created is then used as the input file in the *Solution* section.

Creation of test data often includes use of the NumPy library. This library works efficiently with memory and is quick even at a high number of lines.

Beginners can copy and run the code in this part with an easy conscience. The more advanced can find in it alternative solutions to tasks for which they may use for loops in Python.

Solution

The given task is solved in a few lines, easily comprehensible even to beginners. Each part includes commentary for better understanding of what is going on. Thanks to the Pandas library, data processing is quick even at a large data volume.

Directory structure

The examples assume that there is directory `data` in the current directory.

Software used

The Python 3.9 virtual environment and miniconda installation have been used for examples in this book. Jupyter Notebook has been used as the editor.

Naturally, you can use any IDE, text editor or even the command line. The code will work equally everywhere.

To prepare the environment, I ran the following commands in the command line:

```
conda create -n okpython python=3.9
conda activate okpython
conda install pandas
conda install jupyter
conda install openpyxl
```

Specifically, I used these versions:

```
Name       Version
--------   ------
python     3.9.1
jupyter    1.0.0
pandas     1.2.2
numpy      1.20.1
openpyxl   3.0.6
```

To install Python and the virtual environment, you can naturally utilize `pip` and `venv` as well.

1. How to replace text and numbers

Search and replacement of text is a frequent task, which can easily be done manually in Excel. But how to automate this trivial activity by means of Python?

The answer is the replace function, which can be used for the individual columns or even for the entire DataFrame.

Test data

Excel will have 100 lines and 3 columns.

- The `number` column will contain integral numbers between 100–1000.
- The `text` column will contain 10 random A-Z characters.
- The `text` and `number` column will contain merged records of both columns.

```
import numpy as np
import pandas as pd


row_count = 100
char_count = 10

# Random numbers 100-1000
numbers = np.random.randint(low=100, high=1001, size=row_count)

# Random characters A-Z
chars = np.random.randint(low=65, high=91,
                          size=row_count*char_count,
                          dtype='int32'
                          ).view(f"U{char_count}")

# Concatenate characters and numbers
chars_numbers = np.char.add(chars, np.char.mod('-%d', numbers))
```

```
19
20 # Create a DataFrame
21 df = pd.DataFrame(data={'number': numbers,
22                         'text': chars,
23                         'text and number': chars_numbers })
24
25 # Save to Excel
26 df.to_excel('data/test.xlsx', index=False)
```

Show the first 5 lines

```
 number  text        text and number
-------  ----------  --------------
    939  VGATVDKYSB  VGATVDKYSB-939
    170  WAQFUCYCIB  WAQFUCYCIB-170
    708  VINRRVAONK  VINRRVAONK-708
    550  ZPSQYDYOYM  ZPSQYDYOYM-550
    573  PUYNJUIFPO  PUYNJUIFPO-573
```

Solution

Pandas offers a lot of options for data replacement. We will use the `replace` method, working on both individual columns and the entire DataFrame.

1. Replacement of text in the text column

In the `text` column, replace the `A`, `B` or `C` letter with an asterisk `*`.

```
import pandas as pd


# Read Excel
df = pd.read_excel('data/test.xlsx',
                   usecols=['number', 'text'])

# Make a copy of original text
df['original text'] = df['text']

# Use Regular Expression (regex) on text column.
df['text'] = df['text'].str.replace('A|B|C', '*', regex=True)

# Save to Excel
df.to_excel('data/output.xlsx', index=False)
```

Show the first 5 lines

```
 number  text        original text
-------  ----------  --------------
    939  VG*TVDKYS*  VGATVDKYSB
    170  W*QFU*Y*I*  WAQFUCYCIB
    708  VINRRV*ONK  VINRRVAONK
    550  ZPSQYDYOYM  ZPSQYDYOYM
    573  PUYNJUIFPO  PUYNJUIFPO
```

2. Replacement of numbers in numerical column

In the `number` column, replace numbers `1-5` with the number `0`.

```
import pandas as pd


# Read Excel
df = pd.read_excel('data/test.xlsx', usecols='A')

# Make a copy of original text
df['original number'] = df['number']

# Replace is a string method
df['number'] = df['number'].astype('str').str.replace('[0-5]',
                                                      '0',
                                                      regex=True)

# Save to Excel
df.to_excel('data/output.xlsx', index=False)
```

Show 5 random lines.

```
 number  original number
-------  ---------------
    090              290
    078              478
    800              802
    000              551
    009              339
```

3. Replacement of text in the entire file

Letters `A`, `B` or `C` will be replaced with an asterisk `*` in the entire file.

```
import pandas as pd


# Read Excel
df = pd.read_excel('data/test.xlsx')

# Use Regular Expression (regex) across the whole DataFrame.
df.replace('A|B|C', '*', regex=True, inplace=True)

# Save to Excel
df.to_excel('data/output.xlsx', index=False)
```

Show the first 5 lines

```
 number  text        text and number
-------  ----------  --------------
    939  VG*TVDKYS*  VG*TVDKYS*-939
    170  W*QFU*Y*I*  W*QFU*Y*I*-170
    708  VINRRV*ONK  VINRRV*ONK-708
    550  ZPSQYDYOYM  ZPSQYDYOYM-550
    573  PUYNJUIFPO  PUYNJUIFPO-573
```

This is an excerpt from the book OK, Python.

The complete book includes all the examples that will help you switching from Excel to Python.

If you've enjoyed the sample chapter, you can purchase the full version at www.ok-python.com.