Early Access

OK, Python

V hlavních rolích: Python, Pandas a NumPy

Praktický manuál, jak přejít z Excelu na Python.

Ukázka knihy OK, Python

Příklady, které stačí jen zkopírovat a použít.

Autor: Lubomír Husar
Zakladatel LovelyData.cz

Ovlivněte obsah této knihy a napište mi náměty na další příklady.
Nejvíc mě zajímají ty, které denně řešíte v práci.
Jsem na lubomir@lovelydata.cz.

OK, Python.

Praktický manuál, jak přejít z Excelu na Python.

Early access, verze 0.6

Návrh obálky a grafická úprava www.grafikli.cz.

Pro informace o objednávkách kontaktujte www.lovelydata.cz.

Ikony v sekcích *Testovací data* a *Řešení* byly navrženy pomocí zdrojů z Flaticon.com.

O autorovi

Lubomír "Lubo" Husar

Zakladatel LovelyData.cz.

Lubomír má dlouholeté zkušenosti v oblasti dat. Pomáhal zákazníkům, většinou velkým nadnárodním společnostem, uspět v mnoha kritických projektech. Kromě IT je aktivní i v dalších oblastech. Pracoval a žil v různých evropských zemích, jako například v Belgii, Dánsku, Holandsku, Švédsku nebo Francii. Má bohaté zkušenosti z manažerských i technických rolí.

Poděkování

Chci vám poděkovat, že jste udělali chytré rozhodnutí a investovali do knížky, která Vám usnadní život.

Věřím totiž, že počítače mají lidem život usnadňovat a ne jim ho ztrpčovat. Proto jsem napsal tuhle knížku, kterou jsem nosil v hlavě už dlouho. A za všechno může Python.

Stačí si jen uvědomit, že ještě před pár lety bylo téměř nemožné jednoduše pracovat s daty v různých formátech, analyzovat je, čistit, vizualizovat a zase je do různých formátů ukládat. K takovým úkolům jste potřebovali komerční software a plný open space konzultantů, kteří s ním uměli zacházet.

Tohle všechno se naštěstí velmi rychle mění a Python v téhle změně hraje hlavní roli. Vždyť který jiný nástroj dovede automatizovat práci doslova na pár řádcích a ještě k tomu zdarma? To je nabídka, která se nedá odmítnout!

LH

Obsah

Úvod 8
Python dobývá svět 8
Pro koho je tato kniha 9
Pro koho tato kniha není 9
Organizace jednotlivých kapitol 10
Použitý software 11
Jak nahradit text a čísla 12
Testovací data 12
Řešení 14
Rozděl emailové adresy (podle zavináče) 17
Testovací data 17
Řešení 19
Vyber z textu jen čísla 20
Testovací data 20
Řešení 22
Jak zjistit minimum a maximum 23
Testovací data 23
Řešení 24
Jak používat KDYŽ (IF) 25
Testovací data 25
Řešení 26
Jak na VLOOKUP 29
Testovací data 29
Řešení 31

Označ duplicity 32
Testovací data 32
Řešení 34
Rozdělení jednoho listu do více listů 35
Testovací data 35
Řešení 37
Sloučení více listů do jednoho listu 38
Testovací data 38
Řešení 39
Vygeneruj kalendář 40
Testovací data 40
Řešení 41
Porovnej listy a najdi rozdíly 43
Testovací data 43
Řešení 45
Závěr 47

Úvod

Tato kniha je manuálem pro všechny, kteří pracují s Excelem a chtějí se naučit používat Python.
Tedy pro všechny, které už nebaví spousta klikání myší, operací *zkopíruj* a *vlož* a všechna ta manuální práce, kterou musí provádět každý den.

Návody v knížce ukazují, jak tyto úkoly řešit rychle a elegantně, pomocí pár řádků kódu. A to všechno díky programovacímu jazyku Python a knihovnám Pandas a NumPy.

Příkladů na Python, Pandas a NumPy najdete na internetu spoustu. Co mě osobně ale chybí, je ucelená sbírka příkladů, které můžete použít v praxi. Příklady, které si můžete snadno upravit a přizpůsobit. Příklady, které snadno pochopíte, i když nejste programátor.

A přesně to je cílem tohoto manuálu. Budu rád, když mi napíšete na lubomir@lovelydata.cz, jak se mi to povedlo.

Python dobývá svět

Python vzal svět doslova útokem. Stačí se jen podívat na ankety a žebříčky, které měří popularitu programovacích jazyků. Můžete se vsadit, že Python bude na čelních místech.

A není se co divit. Během posledních několika let se Python prosadil jako *ten* jazyk pro data. Ať už potřebujete data analyzovat, čistit, vizualizovat nebo je používat k trénování umělé inteligence - Python najdete všude.

Znalost Pythonu se tak stala nutností nejen pro programátory a analytiky, ale vlastně pro všechny, kteří si chtějí práci s daty usnadnit.

Existuje dobrý důvod, proč je Python tak populární. Je totiž velmi přátelský k začátečníkům, kteří v něm první krůčky zvládnou rychle. A ti pokročilejší zase mohou využít obrovský ekosystém knihoven, které možnosti Pythonu výrazně rozšiřují.

Pro koho je tato kniha

Nejvíce budou mít z knihy všichni, kteří mají alespoň základní znalosti Pythonu. Plusem je, pokud se už setkali s knihovnou Pandas. Pokud ne, nevadí. Příklady je navedou správným směrem.

Praktické návody v knížce ale využijí i ti, kteří s Pythonem a Pandas už pracují. Jen možná ne tak efektivně, jak by mohli.

Jako vstupní a výstupní formát budeme používat Excel (formát xlsx). Je to proto, že data v Excelu najdete v každé firmě.

Kniha předpokládá, že máte na svém PC nainstalovaný Python 3 a několik knihoven, které uvádím dále.

Pokud chcete mít z knihy co nejvíc, doporučuji si příklady přepisovat ručně, řádek po řádku. Lépe je tak pochopíte. Nikdo vám ale nebrání, abyste příklady jen zkopírovali, vložili a spustili.

Pro koho tato kniha není

Tato kniha není vhodná pro úplné začátečníky, kteří nemají s Pythonem žádné předchozí zkušenosti. Těm mohu doporučit on-line kurz Python pro analytiky na stránkách LovelyData.cz.

Organizace jednotlivých kapitol

Každý příklad vychází z úkolů, které se dají řešit pomocí Excelu. Cílem je ukázat, jak pro tyto úkoly použít Python a Pandas. Výsledkem je snadno pochopitelný kód, který má většinou jen pár řádků a který si můžete snadno přizpůsobit pro svoje konkrétní potřeby.

Testovací data

Součástí knihy záměrně nejsou soubory s testovacími daty. Data si totiž sami vygenerujete ve formátu `XLSX` pomocí knihoven NumPy a Pandas. Vytvořený excelovský soubor je následně použit jako vstupní soubor v sekci *Řešení*.

Vytvoření testovacích dat často zahrnuje použití knihovny NumPy. Tato knihovna pracuje s pamětí efektivně a je rychlá i při velkém množství řádků.

Začátečníci mohou kód v této části s klidným svědomím zkopírovat a spustit. Ti pokročilejší v něm najdou alternativní řešení pro úkoly, na které možná v Pythonu používají `for` smyčky.

Řešení

Zadaný úkol je vyřešen na pár řádcích, které jsou snadno pochopitelné i pro začínající uživatele. Každá část je okomentována pro lepší porozumění toho, co se právě děje. Díky knihovně Pandas je zpracování dat rychlé i při velkém objemu dat.

Adresářová struktura

Příklady počítají s tím, že v aktuálním adresáři existuje adresář `data`.

Použitý software

Pro příklady v knize jsem použil virtuální prostředí Pythonu 3.9 a instalaci miniconda. Jako editor jsem použil Jupyter Notebook.

Vy samozřejmě můžete používat libovolné IDE, textový editor nebo klidně i příkazovou řádku. Kód bude fungovat všude stejně.

Pro přípravu prostředí jsem na příkazové řádce spustil následující příkazy:

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

Konkrétně jsem používal tyto verze:

```
Jméno      Verze
--------   ------
python     3.9.1
jupyter    1.0.0
pandas     1.2.2
numpy      1.20.1
openpyxl   3.0.6
```

K instalaci Pythonu a virtuálního prostředí můžete samozřejmě využít i `pip` a `venv`.

Jak nahradit text a čísla

Vyhledání a nahrazení textu je častým úkolem, který se dá v Excelu snadno provést manuálně. Jak ale tuto triviální činnost zautomatizovat pomocí Pythonu?

Odpovědí je funkce `replace`, kterou můžeme použít na jednotlivé sloupce nebo klidně na celý DataFrame.

Testovací data

Excel bude mít 100 řádků a 3 sloupce.

- Sloupec `čísla` bude obsahovat celá čísla mezi 100-1000.
- Sloupec `text` bude obsahovat 10 náhodných znaků A-Z.
- Sloupec `text a číslo` bude obsahovat spojené záznamy z obou sloupců.

```
import numpy as np
import pandas as pd


pocet_radku = 100
pocet_znaku = 10

# Náhodná čísla 100-1000
cisla = np.random.randint(low=100, high=1001, size=pocet_radku)

# Náhodné znaky A-Z
znaky = np.random.randint(low=65, high=91,
                          size=pocet_radku*pocet_znaku,
                          dtype='int32'
                          ).view(f"U{pocet_znaku}")

# Spoj znaky a čísla
znaky_cisla = np.char.add(znaky, np.char.mod('-%d', cisla))
```

```
19
20 # Vytvoř DataFrame
21 df = pd.DataFrame(data={'číslo': cisla,
22                         'text': znaky,
23                         'text a číslo': znaky_cisla })
24
25 # Ulož do Excelu
26 df.to_excel('data/test.xlsx', index=False)
```

Zobraz prvních 5 řádků

```
  číslo  text        text a číslo
-------  ----------  --------------
    939  VGATVDKYSB  VGATVDKYSB-939
    170  WAQFUCYCIB  WAQFUCYCIB-170
    708  VINRRVAONK  VINRRVAONK-708
    550  ZPSQYDYOYM  ZPSQYDYOYM-550
    573  PUYNJUIFPO  PUYNJUIFPO-573
```

Řešení

Pandas nabízí spoustu možností, jak nahrazovat data. My využijeme metodu `replace`, která funguje jak na jednotlivé sloupce, tak i na celý DataFrame.

1. Nahrazení textu v textovém sloupci

Ve sloupci `text` nahradíme písmena `A`, `B` nebo `C` hvězdičkou `*`.

```
1  import pandas as pd
2
3
4  # Načti Excel
5  df = pd.read_excel('data/test.xlsx',
6                     usecols=['číslo', 'text'])
7
8  # Zkopíruj původní text pro pozdější kontrolu
9  df['původní text'] = df['text']
10
11 # Použijeme regulární výraz (regex) nad sloupcem text.
12 df['text'] = df['text'].str.replace('A|B|C', '*', regex=True)
13
14 # Ulož do Excelu
15 df.to_excel('data/zmena-text.xlsx', index=False)
```

Zobraz prvních 5 řádků

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

2. Nahrazení čísel v číselném sloupci

Ve sloupci `čísla` nahradíme čísla `1-5` číslem `0`.

```
import pandas as pd


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

# Zkopírujeme původní text pro pozdější kontrolu
df['původní číslo'] = df['číslo']

# Replace je metoda, která pracuje nad stringy.
df['číslo'] = df['číslo'].astype('str').str.replace('[0-5]',
                                                    '0',
                                                    regex=True)

# Ulož do Excelu
df.to_excel('data/zmena-cisla.xlsx', index=False)
```

Zobraz 5 náhodných řádků

```
  číslo    původní číslo
-------  ---------------
    090              290
    078              478
    800              802
    000              551
    009              339
```

3. Nahrazení textu v celém souboru

Písmena `A`, `B` nebo `C` budou nahrazena hvězdičkou `*` v celém souboru.

```
import pandas as pd


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

# Použijeme regulární výraz (regex) pro celý DataFrame.
df.replace('A|B|C', '*', regex=True, inplace=True)

# Ulož do Excelu
df.to_excel('data/zmena-vse.xlsx', index=False)
```

Zobraz prvních 5 řádků

```
  číslo  text        text a číslo
-------  ----------  --------------
    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
```

Toto je ukázka z knihy OK, Python.

Kompletní kniha zahrnuje všechny příklady, které vám pomohu přejít z Excelu na Python.

Pokud Vás ukázková kapitola bavila, můžete si zakoupit plnou verzi na www.lovelydata.cz/mooc/kurz/LDO029-ok-python/.