DAQ_Pritz_UE06

May 16, 2023

1 DAQ UE06

Author: Sebastian Pritz

Aufwand: 7h

1.1 Imports

[ ]:
```
import pandas as pd
import numpy as np
from os import listdir
from os.path import exists
import re
from sklearn.preprocessing import KBinsDiscretizer
import matplotlib.pyplot as plt
```

1.2 Code

[ ]:
```
folderpath = "bestsellerdata"

total_data = pd.DataFrame()
sheets = dict()
for path in listdir(folderpath):
    if path.endswith(".xlsx"):
        filedata = pd.read_excel(f"{folderpath}/{path}", sheet_name=None)

        weekMatch = re.search("(\d{2}-\d{2}-\d{2})", path)
        monthMatch = re.search("(\d{2}-\d{2})", path)
        if weekMatch:
            week = weekMatch.group(1)
            combined_data = pd.DataFrame()
            for sheetname in filedata.keys():
                df = filedata[sheetname]
                df["Category"] = sheetname

                # depending on the field, act differently
                if "Information" in df.columns.values:
                    df["Info"] = df["Information"]
```

```
                    df.drop(["Information"], axis=1, inplace=True)
                if "Info" in df.columns.values:
                    df[[f"Title", f"AuthorPrice", f"ISBN"]] = df["Info"].str.
↪split('|', expand=True)
                    df[[f"Author", f"Price"]] = df["AuthorPrice"].str.
↪split('$', expand=True)
                    df.drop(["Info", "AuthorPrice"], axis=1, inplace=True)
                if "Last Week/ Weeks on List" in df.columns.values:
                    df[[f"Previous Rank", f"Weeks on List"]] = df["Last Week/␣
↪Weeks on List"].str.split('/', expand=True)
                    df.drop(["Last Week/ Weeks on List"], axis=1, inplace=True)

                # remove ISBN: prefix from field and add week info and add rows
                df.replace(r'ISBN:', "", regex=True, inplace=True)
                df["Week"] = [week] * df.shape[0]
                total_data = pd.concat([total_data, df], axis=0,␣
↪ignore_index=True)
        elif monthMatch:
            for sheetname in filedata.keys():
                df = filedata[sheetname]

                # Remove prefixes and save as separate columns (with levels)
                df.columns = df.columns.str.split(" - ", expand=True)

                # bring level 1 indices (categories) back to level 0 and rename␣
↪it
                df = df.stack(level=0).reset_index(level=1).reset_index()
                df["Rank"] = df["index"].apply(lambda x: x+1)
                df["Category"] = df["level_1"]
                df.drop(["index", "level_1"],axis=1, inplace=True)

                # Add week data and add to final dataframe
                df["Week"] = [sheetname] * df.shape[0]
                total_data = pd.concat([total_data, df], axis=0)


# Convert week to datetime, set index and sort
total_data["Week"] = pd.to_datetime(total_data["Week"], format="%m-%d-%y",␣
 ↪errors="coerce")
total_data["Price"] = total_data["Price"].astype(float)
total_data.set_index("Week", inplace=True)
total_data.sort_values(inplace=True, by=["Week","Category","Rank"])
total_data
```

[ ]:

Week	Author	ISBN	Previous Rank	Price	Rank	\
2018-01-03	R.J. Palacio	9781524720193.0	NaN	16.99	1	

```
2018-01-03  Madeleine L'Engle  9781250153272.0           NaN   8.99     2
2018-01-03       R.J. Palacio  9781101934852.0           NaN  16.99     3
2018-01-03     Kelly Barnhill  9781616205676.0           NaN  16.95     4
2018-01-03  Victoria Jamieson  9780525429999.0           NaN  12.99     5
…                         …              …             …    …   …
2018-02-28         Nic Stone     9781101939499           NaN  17.99    11
2018-02-28        John Green     9780142424179           NaN  12.99    12
2018-02-28          Marie Lu     9780399547966           NaN  18.99    13
2018-02-28       Holly Black     9780316310277           NaN  18.99    14
2018-02-28       Nicola Yoon     9780553496680           NaN  18.99    15

                                Title Weeks on List        Category
Week
2018-01-03                       Wonder           NaN  Early & Middle
2018-01-03           A Wrinkle in Time           NaN  Early & Middle
2018-01-03                 Auggie & Me           NaN  Early & Middle
2018-01-03  The Girl Who Drank the Moon           NaN  Early & Middle
2018-01-03  All's Faire in Middle School           NaN  Early & Middle
…                                   …           …              …
2018-02-28                Dear Martin            NaN     Young Adult
2018-02-28      The Fault in Our Stars           NaN     Young Adult
2018-02-28                   Warcross            NaN     Young Adult
2018-02-28           The Cruel Prince            NaN     Young Adult
2018-02-28      The Sun Is Also a Star           NaN     Young Adult

[902 rows x 8 columns]
```

1.2.1 Liste aller Bestseller von Jänner und Februar (Wochennummer, Autor, Titel, Preis, ISBN, Rang, Kategorie) (out_0.csv)

Über den vorher erstellten Datensatz können mit wenig Aufwand die wichtigsten Daten gefiltert werden.

[ ]:
```
sales_janfeb = total_data[total_data.index.month <= 2]
sales_janfeb = total_data.loc[:,["Author", "Title", "Price", "ISBN", "Rank",␣
 ↪"Category"]]
sales_janfeb["Week"] = sales_janfeb.index.isocalendar().week
sales_janfeb.set_index("Week", drop=True, inplace=True)
sales_janfeb.to_csv("out_0.csv")
sales_janfeb
```

[ ]:
```
                Author                         Title  Price             ISBN
Week
1         R.J. Palacio                        Wonder  16.99  9781524720193.0  \
1    Madeleine L'Engle            A Wrinkle in Time   8.99  9781250153272.0
1         R.J. Palacio                  Auggie & Me  16.99  9781101934852.0
1       Kelly Barnhill  The Girl Who Drank the Moon  16.95  9781616205676.0
```

```
1     Victoria Jamieson  All's Faire in Middle School  12.99  9780525429999.0
…                   …                             …    …              …
9            Nic Stone                  Dear Martin   17.99    9781101939499
9           John Green       The Fault in Our Stars   12.99    9780142424179
9             Marie Lu                    Warcross   18.99    9780399547966
9          Holly Black            The Cruel Prince   18.99    9780316310277
9          Nicola Yoon       The Sun Is Also a Star   18.99    9780553496680

      Rank        Category
Week
1        1  Early & Middle
1        2  Early & Middle
1        3  Early & Middle
1        4  Early & Middle
1        5  Early & Middle
…    …             …
9       11     Young Adult
9       12     Young Adult
9       13     Young Adult
9       14     Young Adult
9       15     Young Adult

[902 rows x 6 columns]
```

1.2.2 Liste aller Bücher, welche den gleichen Buch Namen wie Filmtitel, sowie den gleichen Autor haben (out_1.csv)

Der Aufbau des Files war etwas verwirrend, bzw. gab es beim Joinen über die "Book Title" Spalte keine Matches (da diese ja auch sehr spärlich befüllt ist). Deswegen wurde die "Movie Title" Spalte zum Joinen verwendet.

[ ]:
```
filmtitles = pd.read_csv("bonusdata/Books into Movies.csv", encoding="ANSI")
filmtitles["Movie Author"] = filmtitles["Author"]
filmtitles.drop(["Author"], inplace=True, axis=1)
data_with_movies = pd.merge(total_data, filmtitles, how="inner",␣
 ↪left_on="Title", right_on="Movie Title")
data_with_movies
same_author = data_with_movies.loc[(data_with_movies["Author"]) ==␣
 ↪(data_with_movies["Movie Author"])]

filtered = same_author.loc[:,["Author","Title","Price","ISBN","Category"]].
 ↪drop_duplicates()

filtered.to_csv("out_1.csv")
filtered
```

[ ]:
```
            Author      Title  Price             ISBN
0     R.J. Palacio     Wonder  16.99  9781524720193.0  \
5     Doug Stanton  12 Strong   9.99  9781501179952.0
7     Doug Stanton  12 Strong  18.00  9781501178511.0
16  David Levithan  Every Day   9.99  9780307931894.0

                     Category
0               Early & Middle
5                  Mass Market
7   Trade Paperback Nonfiction
16                 Young Adult
```

1.2.3 Einteilung der Bücher in gleich große Preiskategorien (out_2.csv)

[ ]:
```
# Distribute to 3 bins using equal_width method
books = total_data.loc[:,["Author","Title","Price","ISBN","Category"]].
 ↪drop_duplicates()
est = KBinsDiscretizer(n_bins=3, encode='ordinal', strategy="uniform")
est.fit(books["Price"].values.reshape(-1,1))
books["PriceCategory"] = est.transform(books["Price"].values.reshape(-1,1)).
 ↪astype(np.uint8)
books.sort_values(["Price", "PriceCategory"], inplace=True)
books.to_csv("out_2.csv")
books
```

[ ]:
```
                              Author
Week
2018-02-21                 Jim McCann   \
2018-01-03        Old Farmer's Almanac
2018-01-10    Chimamanda Ngozi Adichie
2018-02-21   Chimamanda Ngozi Adichie
2018-01-03              Timothy Snyder
…                                 …
2018-02-21              Steven Pinker
2018-02-28              Steven Pinker
2018-01-03                 Ron Chernow
2018-02-07                Ron Chernow
2018-02-14                 Pete Souza

                                                      Title  Price
Week
2018-02-21          Marvel's Black Panther: The Junior Novel   6.99  \
2018-01-03                      The Old Farmer's Almanac 2018   7.95
2018-01-10                         We Should All Be Feminists   7.95
2018-02-21                       We Should All Be Feminists    7.95
2018-01-03                                        On Tyranny   7.99
…                                                        …    …
```

```
2018-02-21  Enlightenment Now: The Case for Reason, Scienc…  35.00
2018-02-28                                 Enlightenment Now   35.00
2018-01-03                                             Grant   40.00
2018-02-07                                             Grant   40.00
2018-02-14                      Obama: An Intimate Portrait   50.00

                      ISBN                    Category  PriceCategory
Week
2018-02-21    9780316413206             Early & Middle              0
2018-01-03  9781571987358.0  Trade Paperback Nonfiction              0
2018-01-10  9781101911761.0  Trade Paperback Nonfiction              0
2018-02-21    9781101911761  Trade Paperback Nonfiction              0
2018-01-03  9780804190114.0  Trade Paperback Nonfiction              0
…                      …                         …            …
2018-02-21    9780525427575       Hardcover Nonfiction              1
2018-02-28    9780525427575       Hardcover Nonfiction              1
2018-01-03  9781594204876.0       Hardcover Nonfiction              2
2018-02-07    9781594204876       Hardcover Nonfiction              2
2018-02-14    9780316512589       Hardcover Nonfiction              2

[316 rows x 6 columns]
```

1.2.4 Auswertung über alle Bücher, welche in den Bestenlisten sowie den Sales Daten vorkommen (out_3.csv + Grafiken?):

- Anzahl verkaufte Bücher
- Summe Umsätze pro verkauftes Buch
- Tag mit den größten Umsätzen
- Was dir sonst noch Spannendes einfällt

Im Endeffekt geht es hierbei nurmehr darum, über Joins die Überschneidungen zu finden und mit Group-By die gewünschten Berechnungen durchzuführen.

[ ]:
```
sales = pd.read_excel("bonusdata/MY SALES.xlsx")

filtered = pd.merge(books, sales, how="inner", on="Title")
filtered[["Transaction ID", "Date", "Author", "Title", "Price", "ISBN",␣
 ↪"Category", "Discount", "Sale Price"]] = filtered.loc[:,["Transaction ID",␣
 ↪"Date", "Author_x", "Title", "Price_x", "ISBN_x", "Category", "Discount",␣
 ↪"Sale Price"]]
filtered

# calculate revenue per day + best day per title
revenue_per_day_per_book = filtered.groupby(["Title", "Date"])["Sale Price"].
 ↪sum().reset_index()
idx = revenue_per_day_per_book.groupby(["Title"])["Sale Price"].idxmax()
best_days_per_book = revenue_per_day_per_book.loc[idx, ["Title", "Date"]]
best_days_per_book.set_index("Title")
```

```
# the other aggregations are trivial and can be done in one line
final_data = filtered["Title"].value_counts().reset_index()
final_data["Total Revenue"] = filtered.groupby("Title")["Sale Price"].sum().
 ↪reset_index()["Sale Price"]
final_data["Best Day"] = best_days_per_book.reset_index()["Date"]
final_data.to_csv("out_3.csv")
final_data
```

[ ]:

	Title	count	Total Revenue	Best Day
0	The Book of Dust: La Belle Sauvage	65	354.40	2018-02-02
1	The Book Thief	63	374.14	2018-02-06
2	The Giver	59	230.79	2018-02-02
3	The Cruel Prince	48	323.82	2018-02-04
4	The Sun Is Also a Star	46	451.77	2018-02-04
5	Every Day	36	131.88	2018-02-06
6	The Hazel Wood	36	814.47	2018-02-04
7	Warcross	24	1487.45	2018-02-06
8	Renegades	23	897.27	2018-02-06
9	Everything, Everything	21	190.95	2018-02-06
10	Dear Martin	20	579.42	2018-02-06
11	Turtles All the Way Down	20	197.89	2018-02-03
12	One of Us Is Lying	19	611.64	2018-02-06
13	The Fault in Our Stars	15	865.94	2018-02-06
14	Salt to the Sea	12	395.80	2018-02-06
15	The Hate U Give	11	455.76	2018-02-06