# Fact_Sheet Pandas

KI_4

2020/2021

Tobias Steiner

# Inhalt

## Install and Import:

`conda install pandas` → **mit diesem Command installiert man Panda**

`import pandas as pd` →**mit dem Command wird Pandas in Python Datei importiert**

## Series and DataFrames:

**Series** sind Spalten und **DataFrames** sind multi-dimensionale Tabellen.

## Creating DataFrames from scratch:

**Ein einfacher weg, ein DataFrame zu erstellen ist es, ein einfaches „dict“ zu verwenden.**
**Bsp:**

```
data = {
     ‘apples‘ : [3, 2 , 0 ,1]
     ‘orange‘ : [3, 2 , 0 ,1]
     }
```

**Nachdem man es erstellt hat, muss man es in den Konstruktor des pandas DataFrames geben**
**Bsp:**

`purchase = pd.DateFrame(data)`
`pruchase` → mit dem Befehl gibst du es dann aus

| | apples | oranges |
|---|---|---|
| **0** | 3 | 0 |
| **1** | 2 | 3 |
| **2** | 0 | 7 |
| **3** | 1 | 2 |

**Mit dem nächsten Befehl, können wir den Index umändern:**

```
purchases = pd.DataFrame(data, index= ['June', 'Robert', 'Lily',
'David'])
purchases
```

| | apples | oranges |
|---|---|---|
| **June** | 3 | 0 |
| **Robert** | 2 | 3 |
| **Lily** | 0 | 7 |
| **David** | 1 | 2 |

## Daten von einer CSV Datei lessen

`df = pd.read_csv(‘purchuases.csv’)`

`df` → **mit dem command gibt man es wieder aus**

| | Unnamed: 0 | apples | oranges |
|---|---|---|---|
| **0** | June | 3 | 0 |
| **1** | Robert | 2 | 3 |
| **2** | Lily | 0 | 7 |
| **3** | David | 1 | 2 |

**Als nächstes können wir unseren Index ändern:**

```
df = pd.read_csv('purchuases.csv', index_col = 0)

df
```

| | apples | oranges |
|---|---|---|
| June | 3 | 0 |
| Robert | 2 | 3 |
| Lily | 0 | 7 |
| David | 1 | 2 |

## Daten einlesen von JSON

```
df = pd.read_json('purchases.json')

df
```

| | apples | oranges |
|---|---|---|
| David | 1 | 2 |
| June | 3 | 0 |
| Lily | 0 | 7 |
| Robert | 2 | 3 |

**Manchmal funktioniert das einlesen der JSON Datei nicht und dazu muss man dann das Keyword „orient" benutzen.**

**Link für eine Erklärung: [https://pandas.pydata.org/pandas-docs/stable/generated/pandas.read_json.html](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.read_json.html)**

## Daten einlesen von SQL

**Zuerst müssen wir den Command:** `pip install pysqlite3`, **in unserem Terminal eingeben.**

**Oder wir lassen den Command in einer Zelle unseres Notebooks laufen**: `!pip install pysqlite3`

**Um das dann in Python verwenden zu können muss man folgendes eingeben:**

```
import sqlite3
```

`con = sqlite3.connect(„database.db")` → **Eine Verbindung mit der Datenbank wird hergestellt.**

```
df = pd.read_sql_query("SELECT * FROM pruchase", con)

df
```

| | index | apples | oranges |
|---|---|---|---|
| 0 | June | 3 | 0 |
| 1 | Robert | 2 | 3 |
| 2 | Lily | 0 | 7 |
| 3 | David | 1 | 2 |

**Mit dem folgenden command kann man den Index auf die Spalte festlegen:**

```
df = df.set_index('index')

df
```

| | apples | oranges |
|---|---|---|
| index | | |
| June | 3 | 0 |
| Robert | 2 | 3 |
| Lily | 0 | 7 |
| David | 1 | 2 |

## Zurück zu CSV, JSON oder SQL konvertieren

```
df.to_csv('new_purchases.csv')
df.to_json('new_purchases.json')
df.to_sql('new_purchases', con)
```

## Wichtigsten DataFrame Operationen

```
movies_df = pd.read_csv("IMDB-Movie-Data.csv", index_col="Title")
```

## Daten ansehen

**Der folgende Command gibt die ersten 5 Zeilen per default aus und als Parameter kann man auch andere Mengen festlegen, das funktioniert auch mit tail(das Ende)**

```
movies_df.head()
movies_df.tail()
```

## Info über die Datei bekommen

```
mavies_df.info()
```

Mit dem Befehl sieht man wie viele Spalten es gibt und ihre Eigenschaften (Name, Anzahl der Einträge, Eigenschaft, Datentyp)

```
<class 'pandas.core.frame.DataFrame'>
Index: 1000 entries, Guardians of the Galaxy to Nine Lives
Data columns (total 11 columns):
Rank                    1000 non-null int64
Genre                   1000 non-null object
Description             1000 non-null object
Director                1000 non-null object
Actors                  1000 non-null object
Year                    1000 non-null int64
Runtime (Minutes)       1000 non-null int64
Rating                  1000 non-null float64
Votes                   1000 non-null int64
Revenue (Millions)      872 non-null float64
Metascore               936 non-null float64
dtypes: float64(3), int64(4), object(4)
memory usage: 93.8+ KB
```

**Mit dem shape Command gibt man aus wie viele Reihen und Spalten es gibt.**

```
movies_df.sjape()
```

## Duplikate

**Mit dem nächsten Command duplizieren wir die Reihen:**

```
temp_df = movies_df.append(movies_df)
```

```
temp_df.shape
```

**Und mit dem Command dropen wir sie wieder:**

```
temp_df = temp_df.drop_duplicates()
```

```
temp_df.shape
```

**Jedoch geht es mit dem folgenden Command wieder leichter:**

```
temp_df.drop_duplicates(inplace = True)
```

**Es gibt dann für die drop Methode 3 verschiedene Zustände bei dem Keyword „keep“**

- First
  - Es werden alle Duplikate bis auf das erste gedropped
- last
  - Das letzte Duplikat wird gedropped
- False
  - alle Duplikate werden gedropped

## Column Cleanup

Mit dem Command: `movies_df.columns` kann man die einzelnen Spalten anzeigen.

**Mit dem Keyword „rename“ kann man es dann umbenennen.**

**Bsp:**
```
movies_df.rename(columns={
            'Runtime (Minutes)': 'Runtime',
            'Revenue (Millions)': 'Revenue_millions'
            }, inplace=True)
movies_df.columns
```

**Man kann es auch anders hinzufügen:**

```
movies_df.columns = ['rank', 'genre', 'description', 'director', 'actors',
'year', 'runtime', 'rating', 'votes', 'revenue_millions', 'metascore']
```

**Mit dem Command ändert man die Namen der Spalten zu Kleinbuchstaben:**

```
movies_df.columns = [col.lower() for col in movies_df]
```

**Mit dem Command:** `movie_df.isnull()` **kann man sich ausgeben lassen welche Zeilen keinen Wert beinhalten.**

**Das Gleiche kann man auch mit Spalten machen:** `movies_df.isnull().sum()`

**Wenn man die NullWerte droppen möchte benutzt man folgenden Command:**
`movies_df.dropna()`

**Mit** `inplace = true` **verändert man dann wirklich den DataFrame.**

**Wenn man eine ganze Spalte löschen will benutz man folgenden Command:**

```
movies_df.dropna(axis = 1)
```