

Projekt Natalie Borter

April 8, 2021

1 Zusammenfassung

Ziel dieser Arbeit ist es zu untersuchen, von welchen Faktoren die Anzahl der gemeldeten Todesfälle an einem Tag im Kanton Basel-Stadt abhängt. Zunächst soll deskriptiv festgestellt werden, ob die Anzahl der Todesfälle zwischen 2005 und 2020 angestiegen ist. Ein solcher Anstieg kann durch den schweizweiten Bevölkerungsanstieg seit 1900 erwartet werden. Allerdings dürfte dieser Effekt insbesondere die ältere Bevölkerung betreffen, da durch den medizinischen Fortschritt und diverse Vorsichtsmassnahmen (Strassenverkehr, Massnahmen gegen plötzlichen Kindstod..) weniger junge Menschen sterben, die Lebenserwartung also in den letzten Jahren angestiegen ist. Die Lebenserwartung hängt aber auch vom Geschlecht ab, wobei diese für Frauen höher ausfällt. Dieser Unterschied in der Lebenserwartung wird auch damit erklärt, dass Männer öfter relativ jung sterben, weil sie ein risikoreicheres Leben führen (Extremsport, schnell Autofahren, Berufe mit erhöhter Unfallgefahr..). Entsprechend kann die Vermutung aufgestellt werden, dass bei den jüngeren Personen (in dieser Arbeit unter 65 Jahre alt) mehr Männer als Frauen sterben und bei den älteren Personen (in dieser Arbeit 65+) mehr Frauen als Männer sterben. Des Weiteren soll aufgezeigt werden, ob es bestimmte Monate mit besonders vielen gemeldeten Todesfällen gibt und ob an gewissen Wochentagen mehr gestorben wird als an anderen.

Beim untersuchten Datensatz handelt es sich um öffentlich zugängliche Daten des Kantons Basel Stadt (<https://data.bs.ch/pages/home/>). Der Datensatz zeigt die Anzahl der verstorbenen Personen nach Altersklasse (unter 65 versus über 65) und Geschlecht für jeden Tag seit 1.1.2005.

Wegen etwaigen Nachmeldungen werden nur Todesfälle berücksichtigt, welche mindestens 15 Tage zurückliegen. Wie im UML in Abbildung 1 ersichtlich, wird für jeden Tag die Anzahl der verstorbenen Männer mit einem Alter bis und mit 64, die Anzahl der verstorbenen Männer mit einem Alter ab 65 und die Anzahl der verstorbenen Frauen mit einem Alter bis und mit 64 sowie ab 65 berichtet. Zusätzlich wird die Summe aller an diesem Tag verstorbenen Personen in der Variable “anz_total” berichtet. Neben den verschiedenen Angaben zu der Anzahl verstorbener Personen, finden sich auch diverse Angaben zum Sterbedatum: Sterbedatum, Datum des Wochenstarts des Sterbedatums, Jahr, Monat, Woche und Wochentag des Sterbedatums.

Todesfälle Basel-Stadt

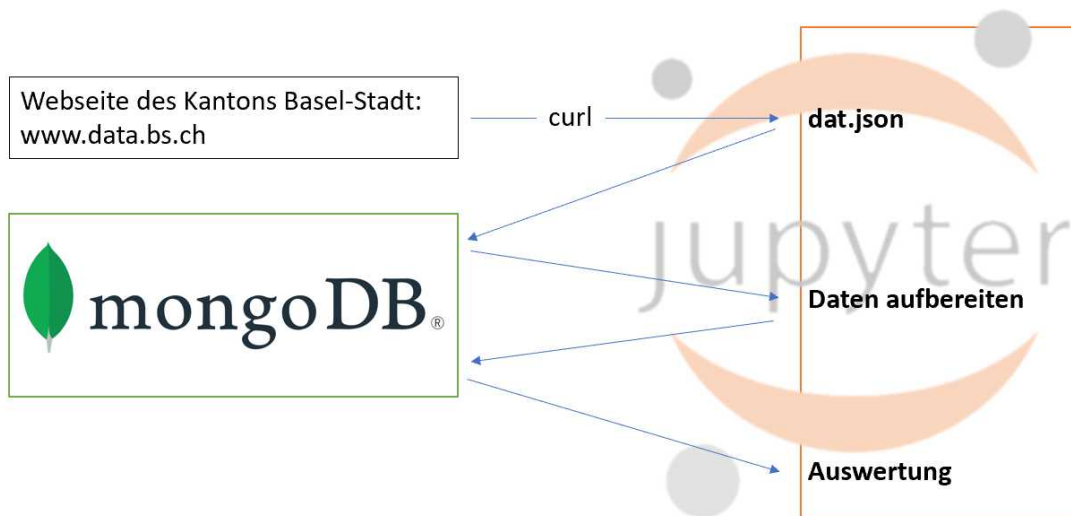
UML



Abbildung 1:

Diese Struktur macht die Daten unübersichtlich, weshalb wir die datumsbezogenen Angaben später in ein embedded field: “Datum” verschieben werden. Zuerst aber eine Übersicht der Schritte zur Auswertung der Daten:

Abbildung 2:



Die Daten werden mittels curl von der Webseite des Kantons Basel Stadt als .json geholt und als dat.json abgespeichert. Dann werden die Daten mittels pymongo in die mongoDB verschoben. Die Daten in mongoDB werden mittels jupyter aufbereitet und die aufbereiteten Daten werden als

neue Kollektion (collection) in der MongoDB abgelegt. Im letzten Schritt werden die aufbereiteten Daten dann in jupyter ausgewertet.

2 Installationen, Voraussetzungen

```
[36]: # Im ersten Schritt soll sichergestellt werden, dass die benötigten Packages
# installiert sind
! echo $VIRTUAL_ENV
! pip3 list | grep -E 'pymongo|dnspython|pandas'
```

```
/home/edu-013/venv/bd103-1-jpy-3.8
dnspython      2.1.0
pandas         1.2.3
pymongo        3.11.3
```

```
[37]: # Benötigte Packages importieren
import pandas as pd
import pymongo
from pprint import pprint
from collections import OrderedDict
```

3 Konfiguration

```
[38]: # Hier werden diverse Variablen definiert, damit wir später darauf
# zurückgreifen können. Auf diese Weise müssen wir etwaige skript-
# übergreifende Veränderungen nur an dieser Stelle vornehmen.

# Optionen pandas
pd.set_option('precision', 2)
pd.set_option('max_rows', 30)
pd.set_option('max_colwidth', 50)

# Optionen für die Arbeit mit MongoDB
HOST_mongo = 'localhost'
OPTIONS_mongo = ''
USER_mongo = ""
PASS_mongo = ""
if USER_mongo:
    credentials=f"{USER_mongo}:{PASS_mongo}@"
else:
    credentials=""

# Optionen für die Datenbank
DBL_project="DBL_project"
URL_API= "https://data.bs.ch/api/v2/catalog/datasets/100079/exports/json"
COLL_data="COLL_data"
```

```
COLL_new="COLL_new"
```

4 Verbindung

```
[39]: # Hier wird der client und der Name der Datenbank definiert.
client = pymongo.MongoClient(f"mongodb://
    ↳{credentials}{HOST_mongo}{OPTIONS_mongo}")
DBL = client[DBL_project]
```

4.1 Alle existierenden Kollektionen löschen

```
[40]: # Löschen der vorhandenen Kollektionen und Ausgabe der Kollektionen.
DBL[COLL_data].drop()
DBL[COLL_new].drop()

c = DBL.list_collections()
pd.DataFrame(c)
```

```
[40]: Empty DataFrame
Columns: []
Index: []
```

5 ETL: Extraktion, Transformation, Laden

5.1 Originaldaten laden / Extraktion

```
[41]: %%bash
curl -sL https://data.bs.ch/api/v2/catalog/datasets/100079/exports/json > dat.
    ↳json
    mongoimport --db DBL_project --jsonArray --drop --collection COLL_data dat.
    ↳json

# Mittels bash laden wir die json-Daten von der Webseite herunter und speichern
# sie unter "dat.json" ab. "dat.json" wird mittels mongoimport in die
# Datenbank geladen. Die Kollektion nennen wir COLL_data.
```

```
2021-04-08T16:38:25.825+0200    connected to: mongodb://localhost/
2021-04-08T16:38:25.827+0200    dropping: DBL_project.COLL_data
2021-04-08T16:38:26.286+0200    6010 document(s) imported successfully. 0
document(s) failed to import.
```

```
[42]: # Beispieldatensatz ausgeben (ein Dokument).
# Zusätzlich lassen wir uns den Datentypen des Beispiels ausgeben.
example=DBL.COLL_data.find_one()
```

```
print(example)
print(type(example))

# Alles ist wie es sein soll
```

```
{'_id': ObjectId('606f156150db224c07e9a4b5'), 'anz_maenner_0_64': 0,
'anz_frauen_65_plus': 2, 'sterbedatum': '2010-09-01', 'jahr': '2010',
'anz_frauen_0_64': 1, 'woche_in_jahr': 35, 'tag_in_jahr': 244,
'anz_maenner_65_plus': 2, 'wochentag': 'Mi', 'monat': 9,
'datum_wochensstart_ssterbedatum': '2010-08-30', 'anz_total': 5}
<class 'dict'>
```

5.2 Transformation und Laden

```
[43]: # Um die Daten übersichtlicher gestalten zu können, sollen alle datumsbezogenen
# Informationen in ein "embedded field" mit dem Namen "Datum" verschoben werden.
# All diese Angaben liessen sich aus dem Datum erschliessen.
# Zuerst sollen die Namen aller Felder ausgegeben werden.
fields=list(example.keys())
print(fields)
```

```
['_id', 'anz_maenner_0_64', 'anz_frauen_65_plus', 'sterbedatum', 'jahr',
'anz_frauen_0_64', 'woche_in_jahr', 'tag_in_jahr', 'anz_maenner_65_plus',
'wochentag', 'monat', 'datum_wochensstart_ssterbedatum', 'anz_total']
```

5.2.1 Datumsbezogene Informationen in ein embedded field verschieben

```
[44]: # Das embedded field heisst "Datum".
DBL.COLL_data.update_many(
    {}),
[{"$set" :{"Datum.sterbedatum" : "$sterbedatum", "Datum.jahr" : "$jahr", "Datum.
↪woche_in_jahr" : "$woche_in_jahr", "Datum.tag_in_jahr" : "$tag_in_jahr",
↪"Datum.wochentag" : "$wochentag", "Datum.datum_wochensstart_ssterbedatum" :
↪"$datum_wochensstart_ssterbedatum", "Datum.monat" : "$monat"}}])
```

```
[44]: <pymongo.results.UpdateResult at 0x7fb971ca85c0>
```

```
[45]: # Löschen der Felder, ausserhalb vom "embedded field".
DBL.COLL_data.update_many({}, {'$unset': {'sterbedatum': "", 'jahr':
↪"", "woche_in_jahr": "", "tag_in_jahr": "", "wochentag":
↪"", "datum_wochensstart_ssterbedatum": "", "monat": ""}})
```

```
[45]: <pymongo.results.UpdateResult at 0x7fb971c93700>
```

```
[46]: # Ein Beispieldatensatz mit der neuen Struktur
c = DBL[COLL_data].aggregate([
    {"$limit": 1},
```

```

])

for doc in c:
    pprint(f"{doc}":["500])

```

```

({'_id': ObjectId('606f156150db224c07e9a4b5'), 'anz_maenner_0_64': 0, "
  "'anz_frauen_65_plus': 2, 'anz_frauen_0_64': 1, 'anz_maenner_65_plus': 2, "
  "'anz_total': 5, 'Datum': {'sterbedatum': '2010-09-01', 'jahr': '2010', "
  "'woche_in_jahr': 35, 'tag_in_jahr': 244, 'wochentag': 'Mi', "
  "'datum_wochenstart_sterbedatum': '2010-08-30', 'monat': 9}}")

```

```

[47]: # Die überarbeitete Struktur lassen wir uns hübsch dargestellt ausgeben.
# Auf einen Blick können wir die relevanten Informationen sehen.
c = DBL[COLL_data].aggregate([
    {"$limit": 3},
])

pd.DataFrame(c)

```

```

[47]:
      _id  anz_maenner_0_64  anz_frauen_65_plus  \
0  606f156150db224c07e9a4b5          0          2
1  606f156150db224c07e9a4b6          1          4
2  606f156150db224c07e9a4b7          1          1

      anz_frauen_0_64  anz_maenner_65_plus  anz_total  \
0          1          2          5
1          1          1          7
2          0          2          4

      Datum
0  {'sterbedatum': '2010-09-01', 'jahr': '2010', ...
1  {'sterbedatum': '2007-03-18', 'jahr': '2007', ...
2  {'sterbedatum': '2019-11-01', 'jahr': '2019', ...

```

Todesfälle Basel-Stadt

UML

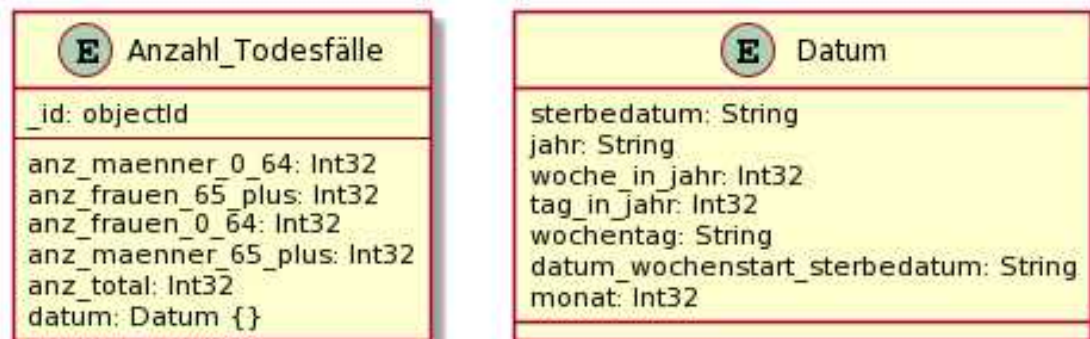


Abbildung 3:

Die neue Datenstruktur ist im UML in Abbildung 3 dargestellt.

5.2.2 Felder validieren

```
[48]: # Zunächst wird untersucht, ob jedes Dokument genau 7 Attribute hat
a = DBL.COLL_data.aggregate([
    {"$match": {"fields": {"$exists" : False}}},
])
number_allfields=len(list(a))
number_total=client.DBL_project.COLL_data.count_documents({})
print("number of documents with 7 fields:", number_allfields, ", toatal number_
→of docuemnts:", number_total)
# jedes Dokument hat 7 Attribute
```

number of documents with 7 fields: 6010 , toatal number of docuemnts: 6010

```
[49]: # Gibt es für jedes Jahr entweder 365 oder 366 Dokumente?
# Oder anders gesagt hat es für jeden Tag im Jahr
# genau einen Eintrag (ein Dokument)?
# Das Jahr 2021 ist nicht aussagekräftig, weil es noch nicht zu Ende ist.

docs=client.DBL_project.COLL_data.aggregate([
    { "$group": {
        "_id": {"Jahr": "$Datum.jahr"},
        "distinctCount": { "$sum": 1 }},
    {"$sort":OrderedDict([("_id",-1)])}
])

for doc in docs:
    pprint(doc)

# Die Anzahl der erfassten Dokumenten pro Jahr weicht in den meisten
```

```
# Fällen von den Anzahl Tagen pro Jahr ab.
```

```
{'_id': {'Jahr': '2021'}, 'distinctCount': 82}  
{'_id': {'Jahr': '2020'}, 'distinctCount': 363}  
{'_id': {'Jahr': '2019'}, 'distinctCount': 374}  
{'_id': {'Jahr': '2018'}, 'distinctCount': 368}  
{'_id': {'Jahr': '2017'}, 'distinctCount': 371}  
{'_id': {'Jahr': '2016'}, 'distinctCount': 372}  
{'_id': {'Jahr': '2015'}, 'distinctCount': 373}  
{'_id': {'Jahr': '2014'}, 'distinctCount': 383}  
{'_id': {'Jahr': '2013'}, 'distinctCount': 366}  
{'_id': {'Jahr': '2012'}, 'distinctCount': 374}  
{'_id': {'Jahr': '2011'}, 'distinctCount': 385}  
{'_id': {'Jahr': '2010'}, 'distinctCount': 368}  
{'_id': {'Jahr': '2009'}, 'distinctCount': 366}  
{'_id': {'Jahr': '2008'}, 'distinctCount': 367}  
{'_id': {'Jahr': '2007'}, 'distinctCount': 364}  
{'_id': {'Jahr': '2006'}, 'distinctCount': 363}  
{'_id': {'Jahr': '2005'}, 'distinctCount': 371}
```

```
[50]: # Weil die Anzahl der Dokumente nicht mit der Anzahl  
# Tage übereinstimmt, wir überprüft, ob es für manche  
# Sterbedaten mehrere Einträge gibt.
```

```
a=DBL.COLL_data.distinct("Datum.sterbedatum")  
b=DBL.COLL_data.distinct("_id")  
print(len(a))  
print(len(b))  
  
# Tatsächlich gibt es mehr verschiedene IDs als Sterbedaten.  
# Gewisse Daten kommen also mehrfach vor.
```

5906

6010

5.2.3 Datensatz auf ein Dokument pro Tag reduzieren

```
[51]: # Wie viele Dokumente gibt es maximal pro Tag?  
c = DBL.COLL_data.aggregate([  
    {"$project": {"_id":0,"Datum.sterbedatum": 1}},  
    {"$sortByCount": "$Datum.sterbedatum"}  
)  
  
pd.DataFrame(c)  
  
# Maximal gibt es zwei Einträge pro Tag und das kommt mehrfach vor.
```



```
[51]:
```

	_id	count
0	2015-12-24	2
1	2009-02-08	2
2	2017-12-09	2
3	2012-02-24	2
4	2015-11-09	2
...
5901	2020-12-27	1
5902	2005-11-10	1
5903	2020-07-24	1
5904	2013-05-22	1
5905	2011-07-20	1

[5906 rows x 2 columns]

```
[52]: # Um die Statistik nicht zu verfälschen, wird pro Datum jeweils nur der erste
# Eintrag (das erste Dokument) verwendet.
# Kommt ein Datum doppelt vor, dann wird eines der beiden Dokumente gelöscht.
# Die so entstandene Kollektion wird unter COLL_new abgespeichert.
DBL.COLL_data.aggregate([{"$group":{"_id": "$Datum.sterbedatum", "doc" :
→{"$first": "$$ROOT"}}},
→{"$replaceRoot": { "newRoot":
→{"$doc"}}}, {"$out": COLL_new}]])
```

```
[52]: <pymongo.command_cursor.CommandCursor at 0x7fb9743c7970>
```

```
[53]: # Jetzt stimmen also Anzahl_IDS und Anzahl Sterbedaten überein.
# Es gibt so viele verschiedene Sterbedaten wie es IDs gibt.
a=DBL.COLL_new.distinct("Datum.sterbedatum")
b=DBL.COLL_new.distinct("_id")
print(len(a))
print(len(b))
```

5906

5906

```
[54]: # Erneut wird überprüft, ob es pro Jahr entweder 365 oder 366 Einträge gibt
# (abgesehen von 2021)
docs=client.DBL_project.COLL_new.aggregate([
    { "$group": {
        "_id": {"Jahr": "$Datum.jahr"},
        "distinctCount": { "$sum": 1 }},
    {"$sort":OrderedDict([("_id",-1)])}
])

for doc in docs:
    pprint(doc)
```

```
# Tatsächlich gibt es jetzt mehr Jahre mit entweder 365 oder 366 Einträgen.
# Allerdings gibt es auch mehrere Jahre mit weniger Einträgen.
```

```
{'_id': {'Jahr': '2021'}, 'distinctCount': 82}
{'_id': {'Jahr': '2020'}, 'distinctCount': 363}
{'_id': {'Jahr': '2019'}, 'distinctCount': 365}
{'_id': {'Jahr': '2018'}, 'distinctCount': 361}
{'_id': {'Jahr': '2017'}, 'distinctCount': 365}
{'_id': {'Jahr': '2016'}, 'distinctCount': 366}
{'_id': {'Jahr': '2015'}, 'distinctCount': 365}
{'_id': {'Jahr': '2014'}, 'distinctCount': 365}
{'_id': {'Jahr': '2013'}, 'distinctCount': 363}
{'_id': {'Jahr': '2012'}, 'distinctCount': 364}
{'_id': {'Jahr': '2011'}, 'distinctCount': 362}
{'_id': {'Jahr': '2010'}, 'distinctCount': 365}
{'_id': {'Jahr': '2009'}, 'distinctCount': 364}
{'_id': {'Jahr': '2008'}, 'distinctCount': 366}
{'_id': {'Jahr': '2007'}, 'distinctCount': 363}
{'_id': {'Jahr': '2006'}, 'distinctCount': 363}
{'_id': {'Jahr': '2005'}, 'distinctCount': 364}
```

5.2.4 Behandlung von Tagen ohne Todesfälle

```
[55]: # Möglicherweise werden Tage ohne Todesfälle in der Statistik nicht aufgeführt?
# Deshalb vergleichen wir die Anzahl der Tage mit genau einem und
# keinem Todesfall miteinander.
```

```
a= DBL.COLL_new.aggregate([
    {"$project": {"_id":0,"Datum.sterbedatum": 1,"anz_total":1}},
    {"$match": {"anz_total": 0}}])
c=pd.DataFrame(a)
e=len(c)

b= DBL.COLL_new.aggregate([
    {"$project": {"_id":0,"Datum.sterbedatum": 1,"anz_total":1}},
    {"$match": {"anz_total": 1}}])
d=pd.DataFrame(b)
f=len(d)

print("An so vielen Tagen ist keine Person verstorben:", e, "; An so vielen_
↳Tagen ist eine Person verstorben:", f)

# Es gibt keinen einzigen Tag mit 0 Toten aber über 100 Tage mit einer
# / einem Toten. Damit ist es statistisch sehr wahrscheinlich,
```

```
# dass Tage mit 0 Toten nicht aufgeführt werden.  
# Das erklärt weshalb nicht jedes Jahr genau 365 oder 366 Einträge existieren.  
# Das verfälscht unsere Statistik nicht und stellt damit kein Problem dar.
```

An so vielen Tagen ist keine Person verstorben: 0 ; An so vielen Tagen ist eine Person verstorben: 143

5.2.5 Beschreibung des aufbereiteten Datensatzes

```
[56]: %%bash -s  
      mongoeeye --db DBL_project --col COLL_new --sample all
```

MongoEYE v0.4 - MongoDB exploration tool

Connecting: ...OK

Analyzing: ...OK

	KEY	COUNT	%
all documents		5906	
analyzed documents		5906	100.0
_id	objectId	5906	100.0
anz_frauen_0_64	int	5906	100.0
anz_frauen_65_plus	int	5906	100.0
anz_maenner_0_64	int	5906	100.0
anz_maenner_65_plus	int	5906	100.0
anz_total	int	5906	100.0
Datum	object	5906	100.0
datum_wochenstart_sterbedatum	string	5906	100.0
jahr	string	5906	100.0
monat	int	5906	100.0
sterbedatum	string	5906	100.0
tag_in_jahr	int	5906	100.0
woche_in_jahr	int	5906	100.0
wochentag	string	5906	100.0

OK 0.068s (local analysis)
5906/5906 docs (100.0%)
14 fields, depth 2

6 Datenanalyse

Im nächsten Schritt sollten die folgenden Fragestellungen untersucht werden: 1. Steigt die Zahl der gemeldeten Todesfälle im Kanton Basel-Stadt seit 2005 in der Tendenz an? 2. Ist dieser Anstieg in

der älteren Gruppe (65+) stärker als in der jüngeren Gruppe (<65)? 3. Hängt die Gesamtzahl der Todesfälle in der älteren und jüngeren Gruppe vom Geschlecht ab? 4. Verteilt sich die Anzahl der Todesfälle gleichmässig über die Monate? 5. Verteilt sich die Anzahl der Todesfälle gleichmässig über die Wochentage?

6.1 Umgang mit Schaltjahren

```
[57]: # Nicht alle Jahre weisen gleich viele Tage auf, weil die Schaltjahre
# einen zusätzlichen Tag haben. Bei den Jahren 2008, 2012, 2016 und 2020
# handelt es sich um Schaltjahre. Im nächsten Schritt untersuchen wir,
# wie viele Personen jeweils am zusätzlichen Tag (den 29. Februar) starben.

c = client.DBL_project.COLL_new.aggregate([
    {"$project": {"_id":0,"Datum.sterbedatum":1,"Datum.tag_in_jahr":1,"tot":
    ↳"$anz_total","Datum.jahr":1}},
    {"$match": {"Datum.sterbedatum": {"$regex" : "-02-29"},"Datum.jahr" :
    ↳{"$ne" : "2021"}}},
    {"$sort":OrderedDict([("Datum.jahr",1)])},
])

pd.DataFrame(c)

# In allen Fällen verändert sich das Total der Todesfälle durch den
# 29. Februar nur geringfügig. Ausserdem sind die Schaltjahre
# regelmässig verteilt und sollten damit nicht zu einem
# systematischen Bias führen.
# Deshalb werden die Schaltjahre nicht weiter beachtet.
```

```
[57]:
```

		Datum	tot
0	{'sterbedatum': '2008-02-29', 'jahr': '2008', ...		5
1	{'sterbedatum': '2012-02-29', 'jahr': '2012', ...		5
2	{'sterbedatum': '2016-02-29', 'jahr': '2016', ...		7
3	{'sterbedatum': '2020-02-29', 'jahr': '2020', ...		4

6.2 Fragestellung 1: Anstieg der gemeldeten Todesfälle über den untersuchten Zeitraum

```
[58]: # Ist die Anzahl der Todesfälle über die Jahre gestiegen?
# Das Jahr 2021 wird nicht betrachtet.

from collections import OrderedDict
list(DBL.COLL_new.aggregate([
    {"$match": {"Datum.jahr" : {"$ne" : "2021"}}},
    {"$project": {"_id":0,"Datum.jahr" :1,"tot" :"$anz_total"}},
    {"$group":{"_id": "$Datum.jahr","total_jahr":{"$sum" : "$tot"}}},
    {"$sort":OrderedDict([("_id",-1)])},
]))
```

```
# Hier zeigt sich keine klare Tendenz
```

```
[58]: [{'_id': '2020', 'total_jahr': 2145},
      {'_id': '2019', 'total_jahr': 2064},
      {'_id': '2018', 'total_jahr': 2077},
      {'_id': '2017', 'total_jahr': 2124},
      {'_id': '2016', 'total_jahr': 1993},
      {'_id': '2015', 'total_jahr': 2139},
      {'_id': '2014', 'total_jahr': 2113},
      {'_id': '2013', 'total_jahr': 2066},
      {'_id': '2012', 'total_jahr': 2106},
      {'_id': '2011', 'total_jahr': 2006},
      {'_id': '2010', 'total_jahr': 2173},
      {'_id': '2009', 'total_jahr': 2181},
      {'_id': '2008', 'total_jahr': 2171},
      {'_id': '2007', 'total_jahr': 2018},
      {'_id': '2006', 'total_jahr': 2175},
      {'_id': '2005', 'total_jahr': 2156}]
```

6.3 Fragestellung 2: Anstieg der gemeldeten Todesfälle stärker für 65+

```
[60]: # Ist die Anzahl der Todesfälle bei den unter 65 jährigen  
# über die Jahre angestiegen?
```

```
from collections import OrderedDict  
list(DBL.COLL_new.aggregate([  
    {"$match": {"Datum.jahr" : {"$ne" : "2021"}}},  
    {"$project": {"_id":0,"Datum.jahr" :1,"tot" :{"$add":  
→["$anz_frauen_0_64","$anz_maenner_0_64"]}},  
    {"$group":{"_id": "$Datum.jahr","total_0_64":{"$sum" : "$tot"}}},  
    {"$sort":OrderedDict([("_id",-1)])},  
]))
```

```
# Hier zeigt sich eine leicht abnehmende Tendenz
```

```
[60]: [{'_id': '2020', 'total_0_64': 229},
      {'_id': '2019', 'total_0_64': 232},
      {'_id': '2018', 'total_0_64': 239},
      {'_id': '2017', 'total_0_64': 229},
      {'_id': '2016', 'total_0_64': 216},
      {'_id': '2015', 'total_0_64': 235},
      {'_id': '2014', 'total_0_64': 278},
      {'_id': '2013', 'total_0_64': 261},
      {'_id': '2012', 'total_0_64': 298},
      {'_id': '2011', 'total_0_64': 240},
      {'_id': '2010', 'total_0_64': 284},
```

```
{'_id': '2009', 'total_0_64': 292},
{'_id': '2008', 'total_0_64': 298},
{'_id': '2007', 'total_0_64': 290},
{'_id': '2006', 'total_0_64': 290},
{'_id': '2005', 'total_0_64': 310}]
```

[61]: *# Ist die Anzahl der Todesfälle bei den 65+ über die Jahre angestiegen?*

```
from collections import OrderedDict
list(DBL.COLL_new.aggregate([
    {"$match": {"Datum.jahr" : {"$ne" : "2021"}}},
    {"$project": {"_id":0,"Datum.jahr" :1,"tot" :{"$add":
→["$anz_frauen_65_plus","$anz_maenner_65_plus"]}}},
    {"$group":{"_id": "$Datum.jahr","total_65_plus":{"$sum" : "$tot"}}},
    {"$sort":OrderedDict([("_id",-1)])},
]))

# Hier zeigt sich keine Tendenz.

# Die Anzahl der Todesfälle ist also im
# Kanton Basel Stadt weder bei den 0 bis und mit 64 jährigen noch
# bei den 65+ angestiegen.
```

[61]:

```
{'_id': '2020', 'total_65_plus': 1916},
{'_id': '2019', 'total_65_plus': 1832},
{'_id': '2018', 'total_65_plus': 1838},
{'_id': '2017', 'total_65_plus': 1895},
{'_id': '2016', 'total_65_plus': 1777},
{'_id': '2015', 'total_65_plus': 1904},
{'_id': '2014', 'total_65_plus': 1835},
{'_id': '2013', 'total_65_plus': 1805},
{'_id': '2012', 'total_65_plus': 1808},
{'_id': '2011', 'total_65_plus': 1766},
{'_id': '2010', 'total_65_plus': 1889},
{'_id': '2009', 'total_65_plus': 1889},
{'_id': '2008', 'total_65_plus': 1873},
{'_id': '2007', 'total_65_plus': 1728},
{'_id': '2006', 'total_65_plus': 1885},
{'_id': '2005', 'total_65_plus': 1846}]
```

6.4 Fragestellung 3: Todesfälle in Abhängigkeit von Alter und Geschlecht

[62]:

```
list(DBL.COLL_new.aggregate([
    {"$match": {"jahr" : {"$ne" : "2021"}}},
    {"$project": {"_id":0,"jahr" :1,"tot1" :"$anz_maenner_65_plus","tot2" :
→"$anz_frauen_65_plus","tot3" :"$anz_maenner_0_64","tot4" :
→"$anz_frauen_0_64",}},
```

```

    {"$group":{"_id":0,"maenner 65+":{"$sum" : "$tot1"},"frauen 65+":{"$sum" :
↪"$tot2"},"maenner_0_64":{"$sum" : "$tot3"},"frauen_0_64":{"$sum" :
↪"$tot4"}}},
  ]))

```

```

# Bei den 65+ starben mehr Frauen als Männer.
# Bei den 0-64 starben mehr Männer als Frauen.
# Die Gesamtzahl der Todesfälle in der jüngeren und älteren
# Gruppe hängt also vom Geschlecht ab.

```

```

[62]: [{'_id': 0,
       'maenner 65+': 12445,
       'frauen 65+': 17477,
       'maenner_0_64': 2682,
       'frauen_0_64': 1595}]

```

6.5 Fragestellung 4: Anzahl Todesfälle abhängig vom Monat?

[63]: # Unterscheiden sich die Todesfälle über die verschiedenen Monate?

```

list(DBL.COLL_new.aggregate([
    {"$match": {"Datum.jahr" : {"$ne" : "2021"}}},
    {"$project": {"_id":0,"Datum.monat" :1,"tot" :"$anz_total"}},
    {"$group":{"_id": "$Datum.monat","total_monat":{"$sum" : "$tot"}}},
    {"$sort":OrderedDict([("$_id",1)])},
  ]))

```

```

# Tendenz: In den kalten Monaten (November, Dezember, Januar, Februar, März)
# sterben tendenziell mehr Menschen (>2800) als in den übrigen Monaten (<2800).
# Beachte: diese Tendenz zeigt sich, obwohl die Monate unterschiedlich viele
# Tage aufweisen. Im Februar sterben über 2800 Menschen, obwohl der Februar
# teilweise 3 Tage weniger aufweist als die Monate mit 31 Tagen.

```

```

[63]: [{'_id': 1, 'total_monat': 3172},
       {'_id': 2, 'total_monat': 2844},
       {'_id': 3, 'total_monat': 3152},
       {'_id': 4, 'total_monat': 2758},
       {'_id': 5, 'total_monat': 2658},
       {'_id': 6, 'total_monat': 2550},
       {'_id': 7, 'total_monat': 2689},
       {'_id': 8, 'total_monat': 2762},
       {'_id': 9, 'total_monat': 2580},
       {'_id': 10, 'total_monat': 2766},
       {'_id': 11, 'total_monat': 2838},
       {'_id': 12, 'total_monat': 2938}]

```

6.6 Fragestellung 5: Anzahl Todesfälle abhängig vom Wochentag?

```
[64]: list(DBL.COLL_new.aggregate([
    {"$match": {"Datum.jahr" : {"$ne" : "2021"}}},
    {"$project": {"_id":0,"Datum.wochentag" :1,"tot" :"$anz_total"}},
    {"$group":{"_id": "$Datum.wochentag","total_wochentag":{"$sum" : "$tot"}}},
    {"$sort":OrderedDict([("$total_wochentag",-1)])},
    ]))

# Tendenz zu weniger Todesfällen an Freitag, Samstag und Sonntag
# (alle unter 4800).
```

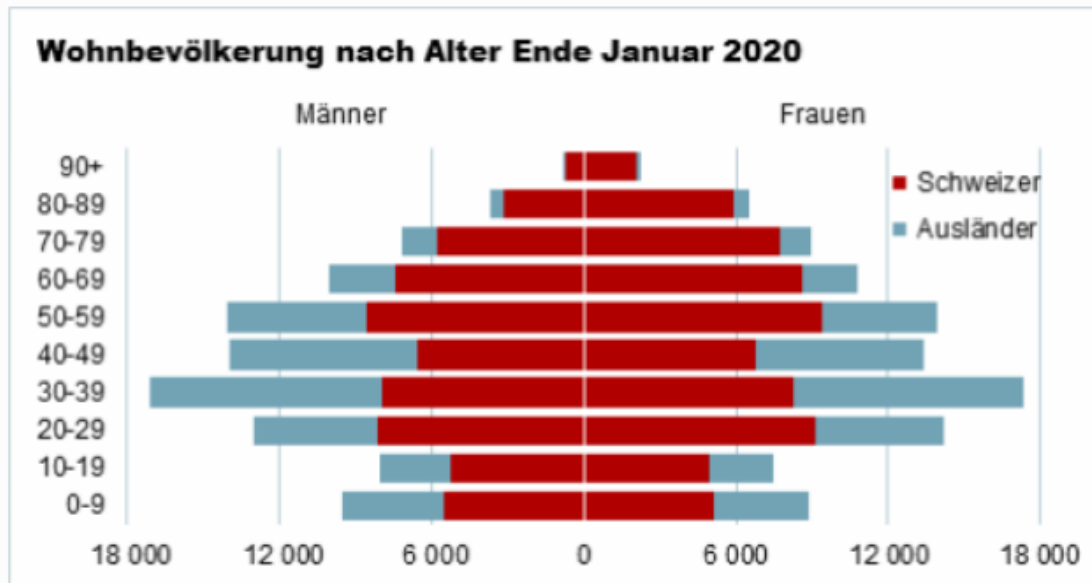
```
[64]: [{'_id': 'Di', 'total_wochentag': 4920},
      {'_id': 'Mi', 'total_wochentag': 4899},
      {'_id': 'Mo', 'total_wochentag': 4882},
      {'_id': 'Do', 'total_wochentag': 4824},
      {'_id': 'Sa', 'total_wochentag': 4783},
      {'_id': 'Fr', 'total_wochentag': 4742},
      {'_id': 'So', 'total_wochentag': 4657}]
```

7 Schlussfolgerung

Die Gesamtzahl der gemeldeten Todesfälle pro Jahr hat sich zwischen 2005 und 2020 nur sehr wenig verändert und es konnte kein klarer Trend festgestellt werden. Nur in den unter 65 jährigen zeigte sich ein leichter Rückgang der Todesfälle. Dieses Muster kann auf die immer besser werdende medizinischen Versorgung oder auf andere gesundheitsfördernde Massnahmen zurückgeführt werden, die dazu geführt haben, dass immer weniger “junge” Menschen sterben.

Dass sich kein Anstieg in der Anzahl der Todesfälle bei den über 65 jährigen zeigt, ist aber erstaunlich, wenn die Leute immer älter und immer mehr werden, müsste doch dieser Wert ansteigen. Das kann vielleicht auf die besonderen Gegebenheiten im Kanton Basel Stadt zurückgeführt werden. Der Kanton hat einen hohen Ausländeranteil, wobei diese Ausländer sehr oft mittleren Alters sind (siehe Abbildung 4). Somit ist also denkbar, dass die Anzahl der 65+ im Kanton Basel Stadt im letzten Jahrhundert eher abgenommen hat, weil ältere Personen eher auf dem Land leben. Hinzu kommt der medizinische Fortschritt, der theoretisch auch einen Einfluss haben kann, wenn viele der Personen mit niedrigerem Jahrgang (schlechterer medizinischen Versorgung) bereits vor 2005 verstorben sind und viele der Leute mit höherem Jahrgang (bessere medizinische Versorgung) noch leben. Es ist aber auch möglich, dass die 15 untersuchten Jahre eine zu kurze Zeit sind, um solche Trends zuverlässig aufzeigen zu können. Das sind aber reine Spekulationen. Eine komplexe Interaktion verschiedener Faktoren ist zu erwarten und könnte eine spannende Ausgangslange für weitere Untersuchungen sein.

Abbildung 4:



Quelle:

statistik.bs.ch

Bei den 65+ waren mehr Todesoper weiblich während bei den 0_64 mehr Todesopfer männlich waren. Das kann damit zu tun haben, dass Männer häufiger in jungen Jahren sterben, weil sie ein risikoreicheres Leben führen (Extremsport, schnell Autofahren...).

Sehr interessant zu sehen war, dass mehr Personen in den kälteren als in den wärmeren Monaten sterben. Möglicherweise hängt das damit zusammen, dass man im Winter viel öfter erkrankt und z.B. die Grippe, an der immer wieder ältere Menschen sterben, besonders in den kalten Monaten aktiv ist. Schwieriger zu erklären ist weshalb tendenziell am Wochenende (Fr, Sa, So) weniger gestorben wird. Vielleicht werden die Daten am Wochenende weniger genau berichtet oder sind ältere Personen am Wochenende weniger alleine, was zu einem niedrigeren Sterberisiko führt.

Für Folgearbeiten sollte weiterverfolgt werden, weshalb gewisse Daten im Datenset doppelt vorkommen. Ausserdem wäre es spannend einen grösseren Zeitraum (z.B. die Entwicklung seit 1960) zu betrachten. Die deskriptiven Analysen sollten statistisch überprüft werden.

ist, was ich mir erlaube, ist zu behaupten, dass ich mit der Zeit, die ich habe, das Beste aus MongoDB herausgeholt habe. Außerdem habe ich ziemlich viel gelernt und habe das Gefühl, MongoDB in Zukunft einsetzen zu können.