Image source: https://www.forbes.com/billionaires/

Data source: https://corgis-edu.github.io/corgis/json/billionaires/

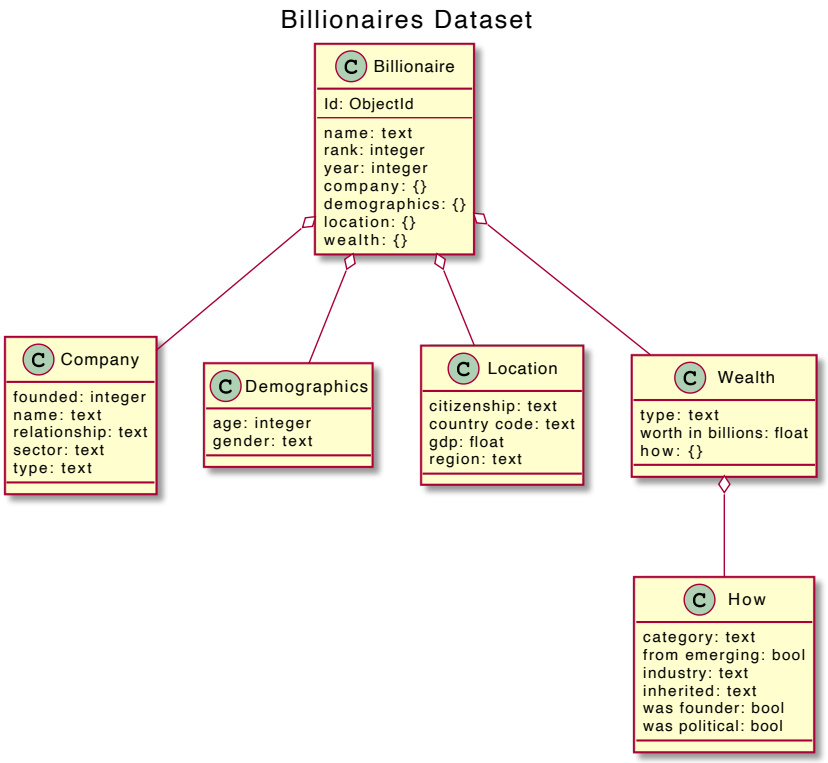
Course: Big Data Lab 2
Author: Ferris Storke
22 December 2021

Who Are The Richest People In The World?

Table of Contents

1 Overview

If you ever wanted to know more about the richest people in the world, you came to the right place: I found, processed, and analysed a dataset on the amount they own, where it all came from, and much more. The data was fetched via CORGIS' API and it is represented in a Unified Modelling Language (UML) Class Diagram below:



2 Requirements & Configuration

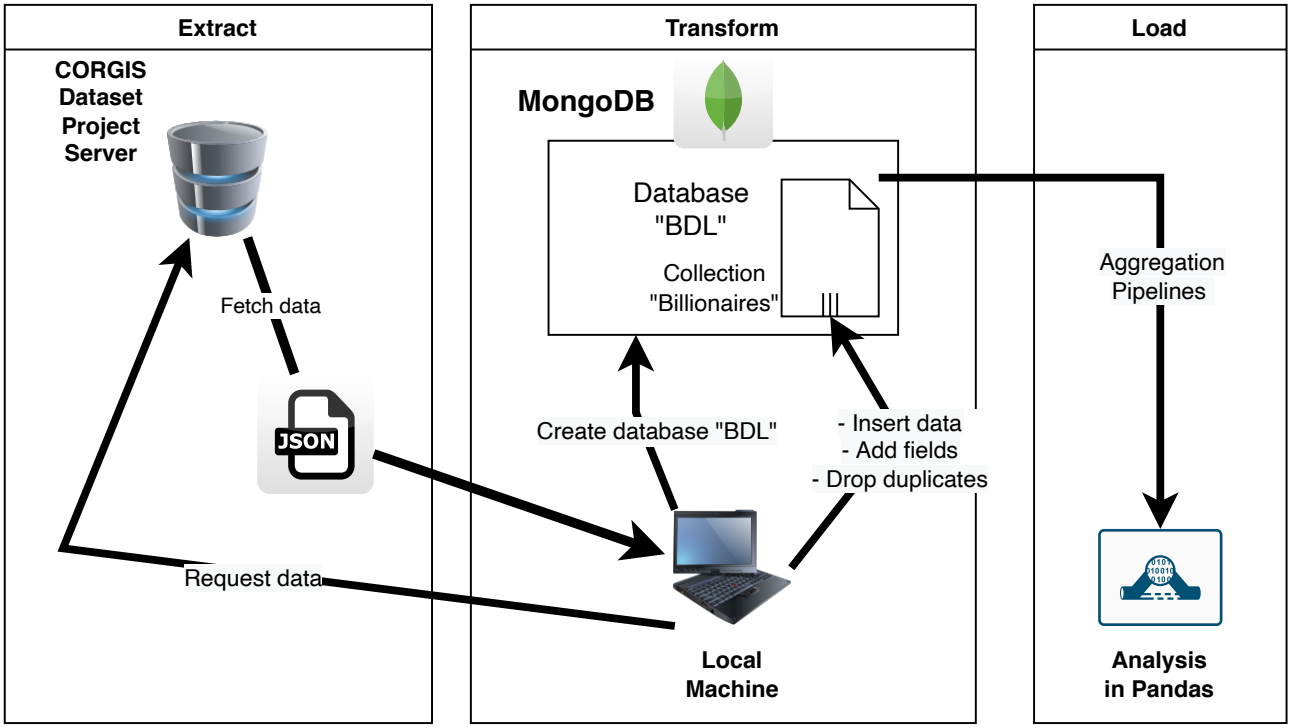
```
import pymongo
from pprint import pprint
import pandas as pd
import requests
import json
import numpy as np
import matplotlib.pyplot as plt
```

```
# MongoDB connection and Database name
CNX_STR = ("mongodb+srv://shiba:4487@cluster0.66u5q.mongodb.net"
           "/myFirstDatabase?retryWrites=true&w=majority")
DB_NAME = "BDL"
```

```
# Connection to MongoDB
client = pymongo.MongoClient(CNX_STR)
db = client[DB_NAME]
billionaires = db["billionaires"]
```

```
# API to fetch databaset
BILLIONAIRES_API_URL = ("https://corgis-edu.github.io/corgis/"
                        "datasets/json/billionaires/billionaires.json")
```

3 ETL



3.1 Remove All Existing Documents

```
# Remove all existing documents from the collection and count remaining ones
billionaires.drop()
print("Number of remaining documents in billionaires collection: ",
      billionaires.count_documents({}))
```

```
Number of remaining documents in billionaires collection:  0
```

3.2 Fetch Data

```
# Fetch JSON from BILLIONAIRES_API_URL
billionaires_data = requests.get(BILLIONAIRES_API_URL).json()
```

3.3 Insert Into MongoDB

```
# Insert the data into MongoDB "billionaires" collection
billionaires.insert_many(billionaires_data);
```

```
# Count number of documents inserted into the "billionaires" collection
document_count = billionaires.count_documents({})
print("Number of documents in billionaires collection: ", document_count)
```

```
Number of documents in billionaires collection:  2614
```

```
# Show first entry to better understand structure of the data
pprint(billionaires.find_one({}))
```

```
{'_id': ObjectId('61c34973cea5efe7f8021a3d'),
 'company': {'founded': 1975,
             'name': 'Microsoft',
             'relationship': 'founder',
             'sector': ' Software',
             'type': 'new'},
 'demographics': {'age': 40, 'gender': 'male'},
 'location': {'citizenship': 'United States',
              'country code': 'USA',
              'gdp': 8100000000000.0,
              'region': 'North America'},
 'name': 'Bill Gates',
 'rank': 1,
 'wealth': {'how': {'category': 'New Sectors',
                    'from emerging': True,
                    'industry': 'Technology-Computer',
                    'inherited': 'not inherited',
                    'was founder': True,
                    'was political': True},
            'type': 'founder non-finance',
            'worth in billions': 18.5},
 'year': 1996}
```

3.4 Transform

First, it is important to gain a better understanding of the contained data. As a starter, it would be good to know, for which years data is available.

```
billionaires.distinct("year")
```

```
[1996, 2001, 2014]
```

So we have data for three years: 1996, 2001 and 2014. It's a shame we don't have more up-to-date information, but let's try to make the best out of it. Since there are different years, it could be possible that the same person appears twice. Since this would create bias in some of my analyses, I therefore counted the number of times each name occurs in the database.

```
match = {"$match": {"year": {"$exists": "True"}}}

# Group by name of billionaire
group1 = {"$group": {"_id": "$name", "count": {"$sum": 1}}}

# Group by count of the same name
group2 = {"$group" : { "_id" : '$count',
                       "Count" : {"$sum" : 1}}}

project = {"$project": {"Count": 1, "year": 1, "name": 1}}

# Group by descending number of occurences
sort = {"$sort": {"Count": -1}}

pipeline = [match, group1, group2, sort]
count_billionaires = billionaires.aggregate(pipeline)
pd.DataFrame(count_billionaires).set_index('_id').rename_axis('Number of Occurences')
```

Number of Occurences	Count
1	1674
2	269
3	134

Among the 2614 billionaires in the dataset, 269 appear twice and 134 three times. This will have to be taken into consideration for some of the following analyses as for some of them, it would not make sense to include the same person more than once.

Since this transformation is only needed for a few analyses in the end, we will start with the first part of the data analysis now and perform the transformation later.

4 Data Analysis

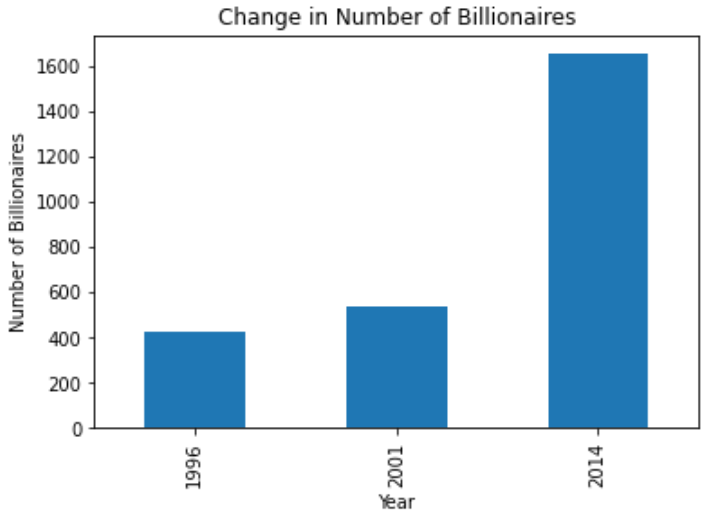
4.1 Number of Billionaires, Total Wealth and Average Wealth

Let's dive straight into how many billionaires there were in each year and how their wealth changed over time.

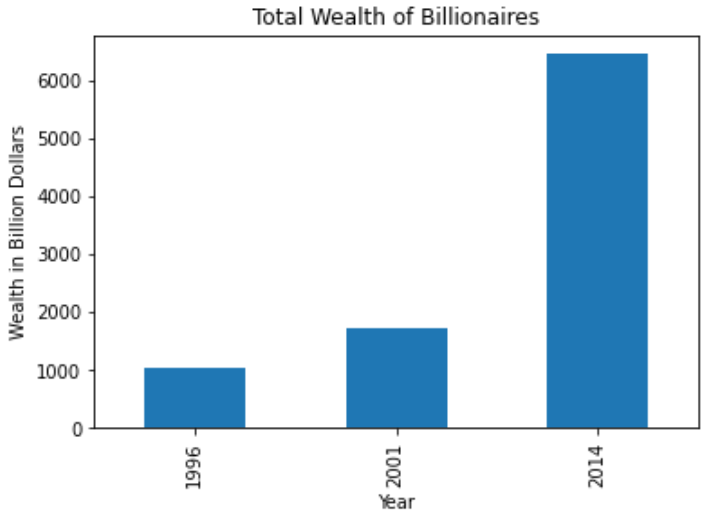
```
# Select year and wealth
project = {"$project": {"_id": 0, "year": 1, "wealth.worth in billions": 1}}

# Group by year and calculate number of billionaires, as well as their total and average wealth
group = {"$group": {"_id": "$year",
                    "Total Billionaires": {"$sum": 1},
                    "Total Wealth": {"$sum": "$wealth.worth in billions"},
                    "Average Wealth": {"$avg": "$wealth.worth in billions"}}}

# Sort by decreasing wealth
sort = {"$sort": {"Total Wealth": -1}}

pipeline = [project, group, sort]

wealth_years = billionaires.aggregate(pipeline)

df = pd.DataFrame(wealth_years).set_index('_id').rename_axis('Year')
df["Total Number Change"] = df["Total Billionaires"].pct_change(-1)
df["Total Wealth Change"] = df["Total Wealth"].pct_change(-1)
df
```

Year	Total Billionaires	Total Wealth	Average Wealth	Total Number Change	Total Wealth Change
2014	1653	6454.4	3.904658	2.072491	2.733889
2001	538	1728.6	3.213011	0.271868	0.647070
1996	423	1049.5	2.481087	NaN	NaN

```
# Plot change in number of billionaires
df = df.sort_values(by=["Total Billionaires"])
df["Total Billionaires"].plot(kind = "bar",
                              ylabel = "Number of Billionaires",
                              title = "Change in Number of Billionaires");
```



```
# Plot wealth change
df = df.sort_values(by=["Total Wealth"])
df["Total Wealth"].plot(kind = "bar",
                        ylabel = "Wealth in Billion Dollars",
                        title = "Total Wealth of Billionaires");
```



Both the number of billionaires and their overall wealth increased throughout the years. Between 1996 and 2001, the total number of billionaires rose by 27 percent from 423 to 538. At the same time, their overall wealth increased by around 65 percent.

This trend further accelerated between 2011 and 2014: the total number of billionaires more than doubled to 1653, and their total wealth rose even further by an astonishing 273 percent from 1.7 to almost 6.5 trillion dollars.

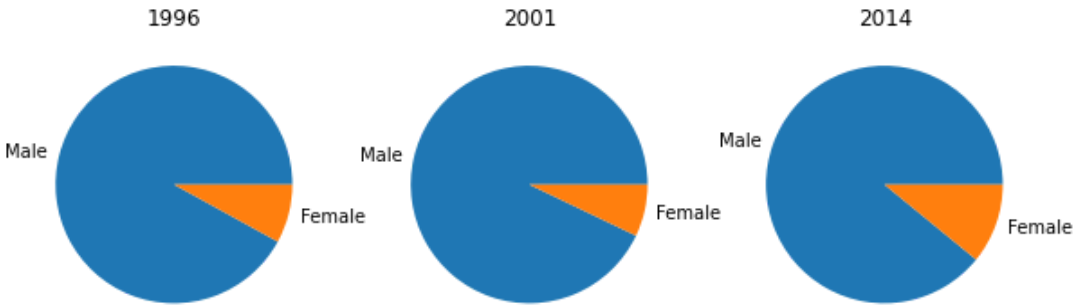
4.2 Gender

Now let's have a look at how many of these people are men and women and how this has changed over the years.

```
# Keep only males and females
match = {"$match": {"demographics.gender": {"$in": ["female", "male"]}}}

# Select year and gender of billionaires
project1 = {"$project": {
    "year": 1,
    "Male": {"$cond": [{"$eq": ["$demographics.gender", "male"]}, 1, 0]},
    "Female": {"$cond": [{"$eq": ["$demographics.gender", "female"]}, 1, 0]}}}

# Group by year and compute total of men and women
group = {"$group": { "_id": "$year",
                     "Male": {"$sum": "$Male"},
                     "Female": {"$sum": "$Female"},
                     "Total": {"$sum": 1}}}

# Project the percentage of men and women by year
project2 = {"$project": {"year": 1, "Male": 1, "Female": 1, "Total": 1,
                         "Male %": {"$divide": ["$Male", "$Total"]},
                         "Female %": {"$divide": ["$Female", "$Total"]}}}

# Sort by increasing year
sort = {"$sort": {
        "_id": 1}}

pipeline = [match, project1, group, project2, sort]

gender_billionaires = billionaires.aggregate(pipeline);
df = pd.DataFrame(gender_billionaires).set_index('_id').rename_axis('Year')
df
```

Year	Male	Female	Total	Male %	Female %
1996	357	31	388	0.920103	0.079897
2001	498	38	536	0.929104	0.070896
2014	1473	180	1653	0.891107	0.108893

```
# Plotting
fig = plt.figure(figsize=(10,10))

for i, (name, row) in enumerate(df[["Male", "Female"]].iterrows()):
    ax = plt.subplot(2,3, i+1)
    ax.set_title(row.name)
    ax.set_aspect('equal')
    ax.pie(row, labels = row.index)

plt.show()
```



Throughout the years, the overwhelming majority of billionaires were men. After their share had even further increased slightly from 92 to 93 percent between 1996 and 2001, by 2014 it had dropped again, but still only 11 percent of billionaires were women.

Yet, over the years, the total number of female billionaires increased almost six-fold from 31 to 180.

4.3 Highest Share of Female Billionaires by Country

I went a little bit deeper and looked into where these few females came from.

```
# Keep only records from 1996 for which gender is either male or female
match = {"$match": {"demographics.gender": {"$in": ["female", "male"]},
                    "year": {"$eq": 1996}}}

# Group by nationality and count the total for each gender
group = {"$group": {"_id": "$location.citizenship",
                    "Male Billionaires 1996": {
                        "$sum": {
                            "$cond": [
                              {  "$eq": ['$demographics.gender', 'male']},
                                1, 0]}},
                    "Female Billionaires 1996": {
                        "$sum": {
                            "$cond": [
                              {  "$eq": ['$demographics.gender', 'female']},
                                1, 0]}}}}

project = {"$project": {"Male Billionaires 1996": 1, "Female Billionaires 1996": 1,
                        "Share Female Billionaires 1996": {"$cond": [ {"$or": [{
                            "$eq": ["$Female Billionaires 1996", 0]},
                                    {"$eq": ["$Male Billionaires 1996", 0]}]},
                                  0, {"$divide": ["$Female Billionaires 1996", {
                                      "$add": ["$Male Billionaires 1996",
                                               "$Female Billionaires 1996"]}]}]}}}

# Sort by decreasing share of females
sort = {"$sort": {"Share Female Billionaires 1996": -1}}

# Show only ten countries with highest share of females
limit = {"$limit": 10}

pipeline = [match, group, project, sort, limit]

female_billionaires = billionaires.aggregate(pipeline);
pd.DataFrame(female_billionaires).set_index('_id').rename_axis('Country')
```

Country	Male Billionaires 1996	Female Billionaires 1996	Share Female Billionaires 1996
Netherlands	1	1	0.500000
Spain	2	1	0.333333
Israel	2	1	0.333333
United States	115	20	0.148148
Brazil	8	1	0.111111
Germany	29	3	0.093750
France	10	1	0.090909
Mexico	13	1	0.071429
Hong Kong	17	1	0.055556
Colombia	3	0	0.000000

In 1996, there were only nine countries with female billionaires and only two with more than two: the United States (20) and Germany (3).

```
# Keep only records from 2001 for which gender is either male or female
match = {"$match": {"demographics.gender": {"$in": ["female", "male"]},
                    "year": {"$eq": 2001}}}

# Group by nationality and count the total for each gender
group = {"$group": {"_id": "$location.citizenship",
                    "Male Billionaires 2001": {
                        "$sum": {
                            "$cond": [
                              {  "$eq": ['$demographics.gender', 'male']}, 1, 0]}},
                    "Female Billionaires 2001": {
                        "$sum": {
                            "$cond": [
                              {  "$eq": ['$demographics.gender', 'female']}, 1, 0]}}}}

project = {"$project": {"Male Billionaires 2001": 1,  "Female Billionaires 2001": 1,
                        "Share Female Billionaires 2001": {"$cond": [
                            {"$or": [{"$eq": ["$Female Billionaires 2001", 0]},
                                     {"$eq": ["$Male Billionaires 2001", 0]}]},
                                   0, {"$divide": ["$Female Billionaires 2001", {
                                       "$add": ["$Male Billionaires 2001",
                                                "$Female Billionaires 2001"]}]}]}}}

# Sort by decreasing share of females
sort = {"$sort": {"Share Female Billionaires 2001": -1}}

# Show only ten countries with highest share of females
limit = {"$limit": 10}

pipeline = [match, group, project, sort, limit]

female_billionaires = billionaires.aggregate(pipeline);
pd.DataFrame(female_billionaires).set_index('_id').rename_axis('Country')
```

Country	Male Billionaires 2001	Female Billionaires 2001	Share Female Billionaires 2001
Argentina	2	2	0.500000
Sweden	3	2	0.400000
Spain	6	2	0.250000
Israel	3	1	0.250000
United States	244	25	0.092937
Mexico	12	1	0.076923
Hong Kong	13	1	0.071429
Germany	26	2	0.071429
France	14	1	0.066667
Italy	16	1	0.058824

By 2001, the share of female billionaires in the US had decreased from 15 to 9 percent.

```
# Keep only records from 2014 for which gender is either male or female
match = {"$match": {"demographics.gender": {"$in": ["female", "male"]},
                    "year": {"$eq": 2014}}}

# Group by nationality and count the total for each gender
group = {"$group": {"_id": "$location.citizenship",
                    "Male Billionaires 2014": {
                        "$sum": {
                            "$cond": [
                              {  "$eq": ['$demographics.gender', 'male']}, 1, 0]}},
                    "Female Billionaires 2014": {
                        "$sum": {
                            "$cond": [
                              {  "$eq": ['$demographics.gender', 'female']}, 1, 0]}}}}

project = {"$project": {"Male Billionaires 2014": 1,
                        "Female Billionaires 2014": 1,
                        "Share Female Billionaires 2014": {"$cond": [
                            {"$or": [{"$eq": ["$Female Billionaires 2014", 0]},
                                     {"$eq": ["$Male Billionaires 2014", 0]}]},
                                    0, {"$divide": ["$Female Billionaires 2014", {
                                        "$add": ["$Male Billionaires 2014",
                                                 "$Female Billionaires 2014"]}]}]}}}

# Sort by decreasing share of females
sort = {"$sort": {"Share Female Billionaires 2014": -1}}

# Show only twenty countries with highest share of females
limit = {"$limit": 10}

pipeline = [match, group, project, sort, limit]

female_billionaires = billionaires.aggregate(pipeline);
pd.DataFrame(female_billionaires).set_index('_id').rename_axis('Country')
```

Country	Male Billionaires 2014	Female Billionaires 2014	Share Female Billionaires 2014
Macau	1	1	0.500000
Switzerland	14	8	0.363636
Monaco	2	1	0.333333
Chile	8	4	0.333333
Netherlands	5	2	0.285714
Finland	3	1	0.250000
Nigeria	3	1	0.250000
Peru	6	2	0.250000
Brazil	51	14	0.215385
Sweden	15	4	0.210526

In 2014, the share of female billionaires in the United States had increased to 13 percent again. Switzerland ranked second in highest share of female billionaires with more than a third of all billionaires being women.

4.4 Age

Let's now look into how old these people are.

```
# Keep only males and females and drop documents with age less than 1
match = {"$match": {"demographics.gender": {"$in": ["female", "male"]},
                    "demographics.age": {"$gte": 1}}}

# Project year and age of billionaires, as well as age by gender
project1 = {"$project": {"year": 1,
                         "Age": "$demographics.age",
                         "Age Male": {"$cond": [{"$eq": ["$demographics.gender", "male"]},
                                                "$demographics.age", ""]},
                         "Age Female": {"$cond": [{"$eq": ["$demographics.gender", "female"]},
                                                  "$demographics.age", ""]}}}

# Group by year and compute max and min age, as well as average for both genders
group = {"$group": { "_id": "$year",
                    "Max Age": {"$max": "$Age"},
                    "Min Age": {"$min": "$Age"},
                    "Avg Age Male": {"$avg": "$Age Male"},
                    "Avg Age Female": {"$avg": "$Age Female"}}}

# Project the percentage of men and women by year
project2 = {"$project": {"year": 1, "Max Age": 1, "Min Age": 1,
                         "Avg Age Male": 1, "Avg Age Female": 1}}

# Sort by increasing year
sort = {"$sort": {
        "_id": 1}}

pipeline = [match, project1, group, project2, sort]

age_billionaires = billionaires.aggregate(pipeline);
df = pd.DataFrame(age_billionaires).set_index('_id').rename_axis('Year')
df
```

Year	Max Age	Min Age	Avg Age Male	Avg Age Female
1996	87	12	59.078049	58.000000
2001	92	29	61.640103	60.592593
2014	98	24	63.427669	62.608434

While the average billionaire is quite old, the age has only risen slightly over the years from around 58 for women and 59 for men to around 62 and 63 respectively.

At the same time, the age of the oldest billionaire steadily increased from 87 to 98, and the youngest billionaire's age had risen from 12 to 29 in 2001 before falling back to 24 in 2014.

4.5 Youngest Billionaires

How could someone at the young age of 12 or even 24 already be a billionaire? Let's find out!

```
# Keep only billionaires between the ages of 1 and 30
match = {"$match": {"demographics.age": {"$gte": 1, "$lte": 30}}}

# Project their name, year of appearance, age, wealth, whether they
# inherited their money or not, and their company and relationship to it.
project = {"$project": {"_id": "$name",
                        "Year of Appearance": "$year",
                        "Age": "$demographics.age",
                        "Wealth": "$wealth.worth in billions",
                        "Inherited": "$wealth.how.inherited",
                        "Company": "$company.name",
                        "Relationship to Company": "$company.relationship"}}

# Sort them by ascending age
sort = {"$sort": {"Age": 1}}

pipeline = [match, project, sort]
youngest_billionaires = billionaires.aggregate(pipeline)

df = pd.DataFrame(youngest_billionaires).drop_duplicates("_id")
df.set_index('_id').rename_axis('Name')
```

Name	Year of Appearance	Age	Wealth	Inherited	Company	Relationship to Company
Albert von Thurn und Taxis	1996	12	1.5	5th generation or longer	von Thurn and Taxis family	relation
Robert Ziff	1996	21	1.0	3rd generation	Ziff Davis Inc	relation
Daniel Ziff	1996	24	1.0	3rd generation	Ziff Davis Inc	relation
Perenna Kei	2014	24	1.3	father	Logan Property Holdings	relation
Kumar Birla	1996	28	2.1	4th generation	Aditya Birla Group	relation
Emilio Azcarraga Jean	1996	28	2.0	father	Grupo Televisa	relation
Mark Zuckerberg	2014	29	28.5	not inherited	Facebook	founder
Dustin Moskovitz	2014	29	6.8	not inherited	Facebook	founder
Anton Kathrein, Jr.	2014	29	1.4	3rd generation	Katherine Werke AG	relation
Ernesto Bertarelli	1996	30	2.5	3rd generation	Serono	relation
Lorenzo Mendoza	1996	30	1.3	father	Empresas Polar	relation
Drew Houston	2014	30	1.2	not inherited	Dropbox	founder

The youngest billionaire, Albert von Thurn und Taxis, inherited his wealth from his parents and so did most of the under 30s on the list. The two founders of Facebook, Mark Zuckerberg and Dustin Moskovitz, are the youngest self-made billionaires on the list with 28.5 and 6.8 billion dollars on their bank account. The third youngest selfmade billionaire was Drew Houston, who founded Dropbox, and was worth 1.2 billion dollars in 2014.

4.6 Nationality

Another interesting question is which countries produce most billionaires and how this changed over the years.

```
# Keep only records from 1996 for which nationality is provided
match = {"$match": {"location.citizenship": {"$exists": "true"},
                    "year": {"$eq": 1996}}}

# Group by nationality and count the total for each country
group = {"$group": { "_id": "$location.citizenship",
                    "Billionaires in 1996": {"$sum": 1}}}

# Sort by decreasing count
sort = {"$sort": {"Billionaires in 1996": -1}}

# Limit to top 10
limit = {"$limit": 10}

pipeline = [match, group, sort, limit]

nationality_billionaires_1996 = billionaires.aggregate(pipeline);
df = pd.DataFrame(nationality_billionaires_1996)
df.set_index('_id').rename_axis('Country')
```

Country	Billionaires in 1996
United States	135
Germany	47
Japan	40
Hong Kong	18
Mexico	15
France	14
Switzerland	13
Malaysia	11
Thailand	10
Brazil	10

```
# Keep only records from 2001 for which nationality is provided
match = {"$match": {"location.citizenship": {"$exists": "true"},
                    "year": {"$eq": 2001}}}


# Group by nationality and count the total for each country
group = {"$group": { "_id": "$location.citizenship",
                    "Billionaires in 2001": {"$sum": 1}}}

# Sort by decreasing count
sort = {"$sort": {"Billionaires in 2001": -1}}

# Limit to top 10
limit = {"$limit": 10}

pipeline = [match, group, sort, limit]

nationality_billionaires_2001 = billionaires.aggregate(pipeline);
pd.DataFrame(nationality_billionaires_2001).set_index('_id').rename_axis('Country')
```

Country	Billionaires in 2001
United States	269
Japan	29
Germany	28
Italy	17
Switzerland	16
Canada	16
France	15
Hong Kong	14
Mexico	13
United Kingdom	12

```
# Keep only records from 2014 for which nationality is provided
match = {"$match": {"location.citizenship": {"$exists": "true"},
                    "year": {"$eq": 2014}}}


# Group by nationality and count the total for each country
group = {"$group": { "_id": "$location.citizenship",
                    "Billionaires in 2014": {"$sum": 1}}}

# Sort by decreasing count
sort = {"$sort": {"Billionaires in 2014": -1}}

# Limit to top 10
limit = {"$limit": 10}

pipeline = [match, group, sort, limit]

nationality_billionaires_2014 = billionaires.aggregate(pipeline);
df = pd.DataFrame(nationality_billionaires_2014)
df.set_index('_id').rename_axis('Country')
```

Country	Billionaires in 2014
United States	499
China	152
Russia	111
Germany	85
Brazil	65
India	56
United Kingdom	47
Hong Kong	45
France	43
Italy	35

Since 1996, most billionaires were from the United States. Germany lost its second place from 1996, sliding down to third in 2001 and fourth in 2014. Switzerland was in the top ten in 1996 and 2001, but not anymore in 2014.

A phenomenal rise was seen in China and Russia, which were not in the top 10 until 2014, where they took second and third place with 152 and 111 billionaires respectively.

4.7 Net Worth By Country

In addition to where billionaires are from, it is interesting to see, where they are the richest.

```
# Keep only records from 1996 for which wealth is provided
match = {"$match": {"wealth.worth in billions": {"$gt": 0},
                    "year": {"$eq": 1996}}}

# Group by nationality and compute the total and average for each country
group = {"$group": {"_id": "$location.citizenship",
                    "Total Wealth in 1996": {"$sum":
                                             "$wealth.worth in billions"},
                    "Average Wealth in 1996": {"$avg":
                                               "$wealth.worth in billions"}}}

# Sort by decreasing total wealth
sort = {"$sort": {"Total Wealth in 1996": -1}}

# Show only ten countries with highest total wealth
limit = {"$limit": 10}

pipeline = [match, group, sort, limit]

total_wealth_1996 = billionaires.aggregate(pipeline);
pd.DataFrame(total_wealth_1996).set_index('_id').rename_axis('Country')
```

Country	Total Wealth in 1996	Average Wealth in 1996
United States	312.1	2.311852
Germany	129.6	2.757447
Japan	90.0	2.250000
Hong Kong	73.9	4.105556
Switzerland	38.9	2.992308
France	35.7	2.550000
Indonesia	29.8	2.980000
Malaysia	26.0	2.363636
Mexico	25.6	1.706667
Taiwan	24.9	3.557143

```
# Keep only records from 2001 for which wealth is provided
match = {"$match": {"wealth.worth in billions": {"$gt": 0},
                    "year": {"$eq": 2001}}}

# Group by nationality and compute the total and average for each country
group = {"$group": {"_id": "$location.citizenship",
                    "Total Wealth in 2001": {"$sum":
                                             "$wealth.worth in billions"},
                    "Average Wealth in 2001": {"$avg":
                                               "$wealth.worth in billions"}}}

# Sort by decreasing total wealth
sort = {"$sort": {"Total Wealth in 2001": -1}}

# Show only ten countries with highest total wealth
limit = {"$limit": 10}

pipeline = [match, group, sort, limit]

total_wealth_2001 = billionaires.aggregate(pipeline);
pd.DataFrame(total_wealth_2001).set_index('_id').rename_axis('Country')
```

Country	Total Wealth in 2001	Average Wealth in 2001
United States	907.6	3.373978
Germany	140.0	5.000000
Japan	89.3	3.079310
France	65.9	4.393333
Hong Kong	50.6	3.614286
Switzerland	49.5	3.093750
Canada	46.7	2.918750
Italy	46.0	2.705882
Saudi Arabia	40.8	5.100000
Sweden	36.0	7.200000

```
# Keep only records from 2014 for which wealth is provided
match = {"$match": {"wealth.worth in billions": {"$gt": 0},
                    "year": {"$eq": 2014}}}

# Group by nationality and compute the total and average for each country
group = {"$group": {"_id": "$location.citizenship",
                    "Total Wealth in 2014": {"$sum":
                                             "$wealth.worth in billions"},
                    "Average Wealth in 2014": {"$avg":
                                               "$wealth.worth in billions"}}}

# Sort by decreasing total wealth
sort = {"$sort": {"Total Wealth in 2014": -1}}

# Show only ten countries with highest total wealth
limit = {"$limit": 10}

pipeline = [match, group, sort, limit]

total_wealth_2014 = billionaires.aggregate(pipeline);
pd.DataFrame(total_wealth_2014).set_index('_id').rename_axis('Country')
```

Country	Total Wealth in 2014	Average Wealth in 2014
United States	2322.4	4.654108
Russia	422.5	3.806306
Germany	401.4	4.722353
China	375.8	2.472368
France	235.3	5.472093
Hong Kong	213.7	4.748889
Brazil	192.2	2.956923
India	191.9	3.426786
Italy	158.1	4.517143
United Kingdom	152.0	3.234043

Similarly to the total number of billionaires, the United States also lead the ranking in total wealth of billionaires from 1996 through 2014. After tripling between 1996 and 2001, this number had again more than doubled to over 2.3 trillion dollars in 2014.

The most notable change is that China and Russia catapulted themselves to second and fourth position in 2014, exceeding the total wealth of American billionaires in 1996.

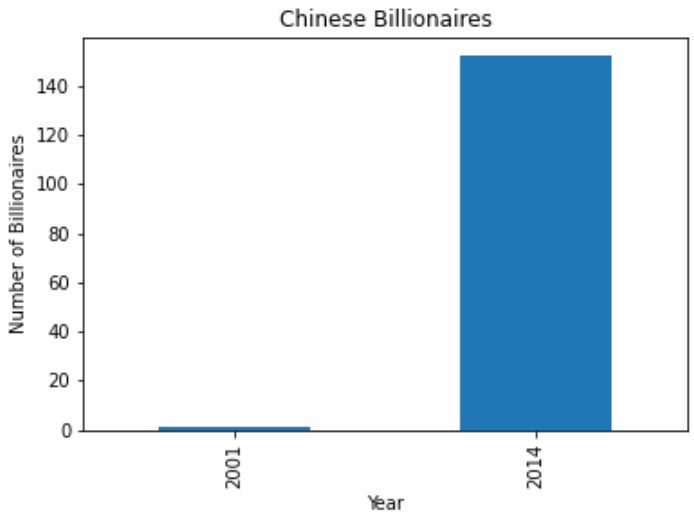
Among the ten countriest with highest billionaire wealth, in 1996, Hong Kong had on average the richest billionaires who owned over 4 billion dollars. However, the country was kicked off the throne by Sweden in 2001 with an average wealth of over 7 billion dollars. In 2014, the country with the highest average wealth was France with almost 5.5 billion per billionaire.

4.8 Chinese Billionaires

Since the astonishing rise of Chinese billionaires deserves some further study, I looked more into them and their gender.

```
# Keep only Chinese billionaires
match = {"$match": {"location.citizenship": {"$eq": "China"}}}

# Group by year, count the total and compute gender share
group = {"$group": { "_id": "$year", "Chinese Billionaires": {"$sum": 1},
                     "Thereof Males": {"$sum": {"$cond": [{
                         "$eq": ['$demographics.gender', 'male']}, 1, 0]}},
                     "Thereof Females": {"$sum": {"$cond": [{
                         "$eq": ['$demographics.gender', 'female']}, 1, 0]}}}}

# Sort by decreasing count
sort = {"$sort": {"Count": 1}}

pipeline = [match, group, sort]

chinese_billionaires = billionaires.aggregate(pipeline);
df = pd.DataFrame(chinese_billionaires)
df = df.set_index('_id').rename_axis('Year')
```

```
df = df.sort_values(by=["Year"])
df["Chinese Billionaires"].plot(kind = "bar",
                                ylabel = "Number of Billionaires",
                                title = "Chinese Billionaires");
```



While there were actually no Chinese billionaires in 1996, and only one in 2001, a breathtaking 152, ten of which were women, made it onto the Forbes list.

4.9 Top 5 Richest Billionaires

Before looking into the overall distribution of billionaires by world region, let's have a look at the five richest people for the years 1996, 2001, and 2014.

```
# Keep only records from 1996 for which rank is provided
match = {"$match": {"rank": {"$exists": "True"}, "year": {"$eq": 1996}}}

# Select rank, name, and wealth fields
project = {"$project": {"_id": "$rank", "Name": "$name",
                        "Wealth": "$wealth.worth in billions"}}

# Sort by decreasing rank
sort = {"$sort": {"_id": 1}}

# Limit to top 5
limit = {"$limit": 5}

pipeline = [match, project, sort, limit]

top5_1996 = billionaires.aggregate(pipeline);
pd.DataFrame(top5_1996).set_index('_id').rename_axis('Rank')
```

Rank	Name	Wealth
1	Bill Gates	18.5
2	Warren Buffett	15.0
3	Oeri Hoffman and Sacher	13.1
4	Lee Shau Kee	12.7
5	Tsai Wan-lin	12.2

```
# Keep only records from 1996 for which rank is provided
match = {"$match": {"rank": {"$exists": "True"}, "year": {"$eq": 2001}}}

# Select rank, name, and wealth fields
project = {"$project": {"_id": "$rank", "Name": "$name",
                        "Wealth": "$wealth.worth in billions"}}

# Sort by decreasing rank
sort = {"$sort": {"_id": 1}}

# Limit to top 5
limit = {"$limit": 5}

pipeline = [match, project, sort, limit]

top5_2001 = billionaires.aggregate(pipeline);
pd.DataFrame(top5_2001).set_index('_id').rename_axis('Rank')
```

Rank	Name	Wealth
1	Bill Gates	58.7
2	Warren Buffett	32.3
3	Paul Allen	30.4
4	Larry Ellison	26.0
5	Theo and Karl Albrecht	25.0

```
# Keep only records from 1996 for which rank is provided
match = {"$match": {"rank": {"$exists": "True"}, "year": {"$eq": 2014}}}

# Select rank, name, and wealth fields
project = {"$project": {"_id": "$rank", "Name": "$name",
                        "Wealth": "$wealth.worth in billions"}}

# Sort by decreasing rank
sort = {"$sort": {"_id": 1}}

# Limit to top 5
limit = {"$limit": 5}

pipeline = [match, project, sort, limit]

top5_2014 = billionaires.aggregate(pipeline);
pd.DataFrame(top5_2014).set_index('_id').rename_axis('Rank')
```

Rank	Name	Wealth
1	Bill Gates	76.0
2	Carlos Slim Helu	72.0
3	Amancio Ortega	64.0
4	Warren Buffett	58.2
5	Larry Ellison	48.0

Throughout the years, Bill Gates, the founder of Microsoft, lead this ranking. From 1996 to 2001, his wealth grew from 18.5 to 58.7 billion dollars (a 317% increase). By 2014, it had risen another 30% to a total of 76 billion dollars. Warren Buffett was the only other billionaire who was in the top 5 in all three years.

4.10 Geographic Distribution by World Region

Since the last two analyses are not focused on changes over time, but the overall distribution by region, as well as the more specific role of industries in creating billionaires, we will drop all billionaires who appear more than once in the ranking as this would otherwise create biased results.

```
# Group billionaires by name and create list with the year they appear and the corresponding id
group = {"$group": {
         "_id": "$name",
         "years": {"$addToSet": "$year"},
         "Ids": {"$addToSet": "$_id"},
         "count": {"$sum": 1}}}

# Keep only those who appear more than once
match = {"$match": {
         "count": {"$gt": 1}}}

# Sort them by number of appearance
sort = {"$sort": {
         "count": -1}}

id_billionaires = billionaires.aggregate([group, match, sort]);
df = pd.DataFrame(id_billionaires)
df.head(5)
```

	_id	years	Ids	count
0	David Koch	[2014, 2001, 1996]	[61c34973cea5efe7f8021a4f, 61c34973cea5efe7f80...	3
1	Riley Bechtel	[2014, 1996, 2001]	[61c34973cea5efe7f8021eeb, 61c34973cea5efe7f80...	3
2	John Mars	[2001, 1996, 2014]	[61c34973cea5efe7f8021a9a, 61c34973cea5efe7f80...	3
3	Dennis Washington	[2001, 1996, 2014]	[61c34973cea5efe7f8021deb, 61c34973cea5efe7f80...	3
4	James Jannard	[1996, 2014, 2001]	[61c34973cea5efe7f802200d, 61c34973cea5efe7f80...	3

```
# Write years and corresponding ids to lists
years_list = df.years.tolist()
ids_list = df.Ids.tolist()

# Find index of latest year and save to list
index_list = []
for entry in years_list:
    max_value = max(entry)
    max_index = entry.index(max_value)
    index_list.append(max_index)

# Delete id corresponding to latest year from list of ids to be able
# to drop the remaining ones, so only the latest entry remains
for i in range(len(ids_list)):
    ids_list[i].pop(index_list[i])

# Convert ObjectIDs to string and store in new list
ids_to_drop = []
for entry in ids_list:
    for object_id in entry:
        ids_to_drop.append((object_id))
```

```
# Remove IDs corresponding to duplicates where latest year is kept
billionaires.delete_many({"_id": {"$in": ids_to_drop}})

# Show number of dropped documents and count number remaining ones
print("Number of documents dropped: ",
      document_count - billionaires.count_documents({}))
print("Number of documents remaining in billionaires collection: ",
      billionaires.count_documents({}))
```

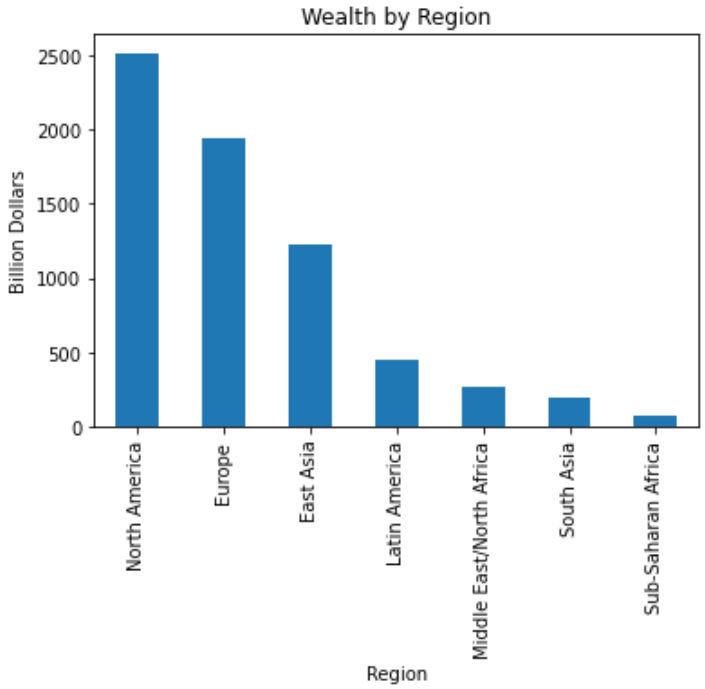
```
Number of documents dropped:  537
Number of documents remaining in billionaires collection:  2077
```

We now managed to drop all documents for billionaires with multiple entries, keeping only the latest one, which means we can now proceed with the analysis of the data.

```
# Keep only records for which region has a valid value
match = {"$match": {"location.region": {"$nin": ["", "0"]}}}

# Group by region and compute the count, as well as total and average wealth for each region
group = {"$group": {"_id": "$location.region", "Count": {"$sum": 1},
                    "Total Wealth": {"$sum":
                                     "$wealth.worth in billions"},
                    "Average Wealth": {"$avg":
                                       "$wealth.worth in billions"}}}

# Sort by decreasing total number of billionaires
sort = {"$sort": {"Count": -1}}

pipeline = [match, group, sort]

region = billionaires.aggregate(pipeline);
df = pd.DataFrame(region).set_index('_id').rename_axis('Region')
df
```

Region	Count	Total Wealth	Average Wealth
North America	735	2517.3	3.424898
Europe	574	1945.9	3.390070
East Asia	446	1222.2	2.740359
Latin America	142	449.2	3.163380
Middle East/North Africa	99	259.1	2.617172
South Asia	63	197.2	3.130159
Sub-Saharan Africa	17	67.6	3.976471

```
df = df["Total Wealth"].sort_values(ascending = False)
df.plot(kind = "bar",
        ylabel = "Billion Dollars",
        title = "Wealth by Region");
```



With 734 billionaires, who own almost 2.5 trillion dollars overall, North America leads this ranking. However, the region with the highest average wealth per billionaire is actually Sub-Saharan Africa, which comes in before Europe at almost 4 billion dollars.

4.11 Industries

```
# Keep only records for which industry has a valid value
match = {"$match": {"wealth.how.industry": {"$nin": ["", "0"]}}}

# Group by industry and compute the count, as well as
# total and average wealth for each industry
group = {"$group": {"_id": "$wealth.how.industry",
                    "Count": {"$sum": 1},
                    "Total Wealth": {"$sum":
                                     "$wealth.worth in billions"},
                    "Average Wealth": {"$avg":
                                       "$wealth.worth in billions"}}}

# Sort by decreasing number of billionaires
sort = {"$sort": {"Count": -1}}

pipeline = [match, group, sort]

industry = billionaires.aggregate(pipeline);
pd.DataFrame(industry).set_index('_id').rename_axis('Industry')
```

Industry	Count	Total Wealth	Average Wealth
Consumer	370	1183.7	3.199189
Real Estate	227	601.3	2.648899
Retail, Restaurant	215	808.4	3.760000
Money Management	191	517.1	2.707330
Technology-Computer	159	617.0	3.880503
Media	154	571.7	3.712338
Diversified financial	141	586.6	4.160284
Energy	105	344.3	3.279048
Non-consumer industrial	98	313.2	3.195918
Technology-Medical	93	256.0	2.752688
Constrution	82	196.8	2.400000
Mining and metals	76	227.6	2.994737
Other	69	162.0	2.347826
Hedge funds	54	165.1	3.057407
Private equity/leveraged buyout	20	74.7	3.735000
Venture Capital	6	9.3	1.550000
banking	1	1.3	1.300000
services	1	1.2	1.200000

370 billionaires made their money from the consumer industry, amassing almost 1.2 trillion dollars in total, followed by the real estate industry with 227 billionaires being worth a total of more than 600 billion dollars.

While tech companies are only ranked fourth in the overall wealth they created for billionaires, they created the highest average wealth per billionaires with almost 4 billion dollars and the third highest overall wealth with 617 billion dollars.

5 Conclusions ¶

This database allowed some pretty interesting insights into who the world's richest people are. While the latest data dates back seven years and especially entries for 1996 are scarce and there are lots of missing values, it still provides a good understanding of the trends of the past two decades.

First, Bill Gates was consistently the richest man in the world, but what is even more interesting is that his wealth virtually exploded, especially between 1996 and 2001, which is a time period in which the dot com bubble falls. Similarly, the number of billionaires around the world almost quadrupled over the time period and their total wealth rose more than six-fold. Most of this increase occurred in the 2000s.

My analysis also revealed a strong gender imbalance among billionaires with only 11 percent being women in 2014.

Another interesting finding was the astronomical rise of China: whereas there had not been a single billionaire in 1996 and only one in 2001, there were over 150 in 2014 and their wealth amounted to around 376 billion dollars.

Since the United States ranked number one for most billionaires and highest wealth of billionaires, it is not surprising that North America also produced most billionaires and the highest overall wealth. However, on average, billionaires are richest in Sub-Saharan Africa.

Lastly, the analysis has shown that the industries that produced most billionaires were the consumer industry and real estate, the first of which also produced the most wealth. However, on average, the richest billionaires are from the tech sector.

In summary, the database reveals that there has been a remarkable proliferation in billionaires and their wealth over the past decades, which, given increasing inequality in many western societies, could be a troubling trend.

6 Learnings

This was a very interesting project as it allowed me to apply what we had learned in class about databases, how to model data and use pymongo to extract information with a combination of different aggregators. What was most challenging for me was to find the right combination of conditions to undertake more complicated calculations directly in pymongo. For example, when getting the share of females by year, I had to account for divisions by zero throwing errors for countries were there were no female billionaires.

Overall, this was a fun experience and a new dataset would provide the opportunity to use and learn about lots of new operators that were not required for this analysis.