Introduction to Web Science

Assignment 6

Prof. Dr. Steffen Staab
staab@uni-koblenz.de

René Pickhardt
rpickhardt@uni-koblenz.de

Korok Sengupta
koroksengupta@uni-koblenz.de

Institute of Web Science and Technologies
Department of Computer Science
University of Koblenz-Landau

Submission until: December 6, 2016, 10:00 a.m.
Tutorial on: December 9, 2016, 12:00 p.m.

Please look at the lessons 1) **Simple descriptive text models** & 2) **Advanced descriptive text models**

For all the assignment questions that require you to write code, make sure to include the code in the answer sheet, along with a separate python file. Where screen shots are required, please add them in the answers directly and not as separate files.

Team Name: Golf

Members: Atique Baig, Mtarji Adam, Deepak Garg

1 Digging deeper into Norms (10 points)

You have been introduced to the concept of a norm and have seen that the uniform norm $||\cdot||_\infty$ fullfills all three axioms of a norm which are:

1. Positiv definite
2. Homogeneous
3. Triangle inequality

Recall that for a function $f : M \longrightarrow \mathbb{R}$ with M being a finite set[1] we have defined the L_1-norm of f as:

$$||f||_1 := \sum_{x \in M} |f(x)| \tag{1}$$

In this exercise you should

1. calculate $||f-g||_1$ and $||f-g||_\infty$ for the functions f and g that are defined as
   - $f(0) = 2, f(1) = -4, f(2) = 8, f(3) = -4$ and
   - $g(0) = 5, g(1) = 1, g(2) = 7, g(3) = -3$

   Answer:

$$\begin{aligned} ||f-g||_1 &= \sum_{x \in M} |f(x) - g(x)| \\ &= |f(0) - g(0)| + |f(1) - g(1)| + |f(2) - g(2)| + |f(3) - g(3)| \\ &= |-3| + |-5| + |1| + |-1| \\ &= 10 \end{aligned} \tag{2}$$

$$\begin{aligned} ||f-g||_\infty &= max\{|f(x) - g(x)| : x = 0, 1, 2, 3\} \\ &= max\{|f(0) - g(0)|, |f(1) - g(1)|, |f(2) - g(2)|, |f(3) - g(3)|\} \\ &= max\{|-3|, |-5|, |1|, |-1|\} \\ &= 5 \end{aligned} \tag{3}$$

2. proof that all three axioms for norms hold for the L_1-norm.

   Answer:
   Let us start with the positive definite axiom. ($||f||_1 = 0 \implies f = 0$)

[1] You could for example think of the function measuring the frequency of a word depening on its rank.

Proof:

$$\begin{aligned} ||f||_1 = 0 &\iff \sum_{x \in M} |f(x)| = 0 \\ &\Rightarrow |f(x)| = 0 \forall x \in M \\ &\Rightarrow f(x) = 0 \forall x \in M \\ &\Rightarrow f = 0 \end{aligned} \tag{4}$$

Next is the Homogeneous axiom. ($||\alpha f||_1 = \alpha ||f||_1, \alpha \in \mathbb{R} : \alpha > 0$)
Proof:

$$\begin{aligned} ||\alpha f||_1 &= \sum_{x \in M} |\alpha f(x)| \\ &= |\alpha| \sum_{x \in M} |f(x)| \\ &= \alpha ||f||_1 \end{aligned} \tag{5}$$

Last we prove the triangle inequality axiom. ($||f + g||_1 \leqslant ||f||_1 + ||g||_1$)
Proof:

$$\begin{aligned} ||f + g||_1 &= \sum_{x \in M} |f(x) + g(x)| \\ &\leqslant \sum_{x \in M} |f(x)| + \sum_{x \in M} |g(x)| \\ &\leqslant ||f||_1 + ||g||_1 \end{aligned} \tag{6}$$

1.1 Hints:

1. The proofs work in a very similar fashion to those from the uniform norm that was depicted in the videos.
2. You can expect that the proofs for each property also will be ”three-liners”.
3. Both parts of this exercise are meant to practice proper and clean mathematical notation as this is very helpfull when reading and understanding research papers. Discuss in your study group not only the logics of the calculation and the proof (before submission) but try to emphasize on the question whether your submission is able to communicate exactly what you are doing.

2 Coming up with a research hypothesis (12 points)

You can find all the text of the articles from Simple English Wikipedia at http://141.26.208.82/simple-20160801-1-article-per-line.zip each line contains one single article.

In this task we want you to be creative and do some research on this data set. The ultimate goal for this exercise is to practice the way of coming up with a research hypothesis and testable predictions.

In order to do this please **shortly**[2] answer the following questions:

1. What are some obervations about the data set that you can make? State at least three obervations.

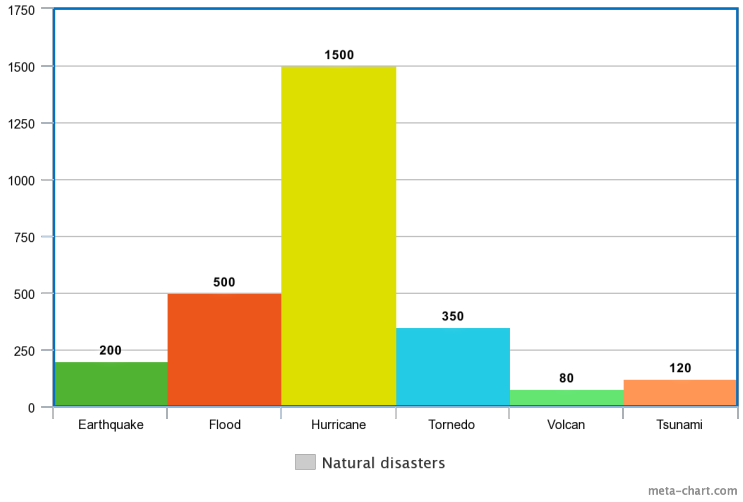
   Answer:
   Following are some of our observations:

   a) A number of these articles are about historical figures and famous people coming from different regions, particularly the USA

   b) We notice a number of articles about high impact natural disasters like hurricanes, floods, volcanic eruptions and earthquakes, and that hurricanes are mentioned a lot more then expected.
   After doing some exploration using looking for specific key words, we made a draft to describe our observation:

[2] Depending on the question shortly could mean one or two sentences or up to a thousand characters. We don't want to give a harsh limit because we trust in you to be reasonable.

Figure 1: Frequency of articles about some natural disasters



c) We notice that articles about movies and TV shows are more frequent then novels and books.

d) As a more general observations,a lot of these articles describe events and personalities that happened/lived in the US.

2. Which of these observations make you curious and awaken your interest? Ask a question about why this pattern could occur.

Answer:
We are more curious about the first observation, there seems to be a high number of famous people from the USA. Could this pattern be the result of the American culture promoting movie and TV show stars and bringing them to the spotlight? or is the wikipedia contributors focus more on personalities coming from USA?

3. Formulate up to three potential research hypothesis.

Answer:
Following are our hypothesis:

a) The largest portion of the people and historical figures mentioned in the articles are from the USA, and the largest portion of them are movie or TV stars.
By studying the data, we could easily fall into a trap and start counting fictional characters, since many articles concerns movies that are more or less about recent events in human history, and since the US was involved in most of them, it is the country of origin for many fictional characters too. This could inflate the numbers by a lot, giving us a false illustration.

b) The majority of the recorded natural disasters on the wikipedia happened in the United States.
We assume this mainly because hurricanes are known to hit North America more often then other countries, and by doing keyword search we find a large number of Hurricanes mentioned, and lot of articles about hurricanes contains lot more words then other articles, which could make our predictions inflated.

c) There are two times more articles about movies and TV shows then books and novels.
Using key word search we could get an inflated prediction due to a large number of movie stars who have their own articles, which means we will have to eliminate many duplicates (a movie can be mentioned in its own article and in the stars article). This means that our predictions might be inflated due to a large number of duplicates.

4. Take the most promising hypothesis and develop testable predictions.

   Answer:
   We choose the first hypothesis, We expect the largest portion to be around 50% from the USA, and from them we expect around 60% movie or TV stars

5. Explain how you would like to use the data set to test the prediction by means of descriptive statistics. Also explain how you would expect your outcome.

   (If you realize that the last two steps would not lead anywhere repeat with one of your other research hypothesis.)

   Answer:
   Using the data (article per line), we will first extract all articles about famous people by looking for the key word ”born in”, once an article is identified, we will look for the first country name after the key word, using a predefined list of all countries and states. We also need to look for any of these key words (movie, TV, television, film, actor, actress), to identify if its an article about a star or other. Using the collected data, we expect to see a Choropleth map (or thematic map) of the world, with colored bin representing the density of famous people per country of origin. The second diagram will be a pie chart representing the proportion of each country of origin, and a third diagram will be a pie chart representing the percentage of movie/Tv stars from the USA data.

Figure 2: Draft of the 2nd diagram, representing the proportion of the USA as a country of origin

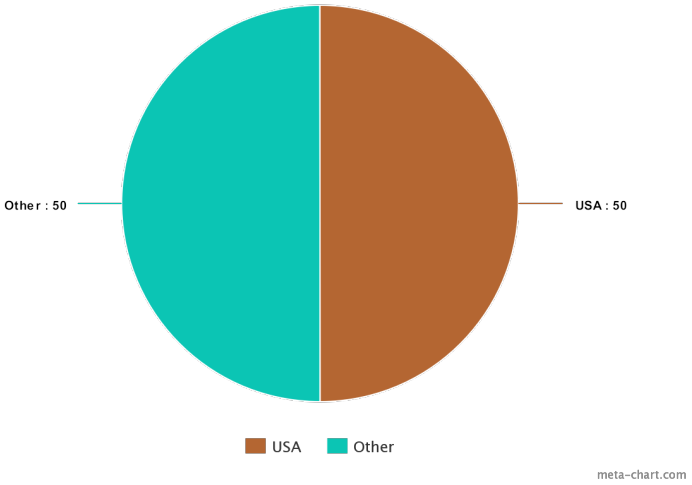
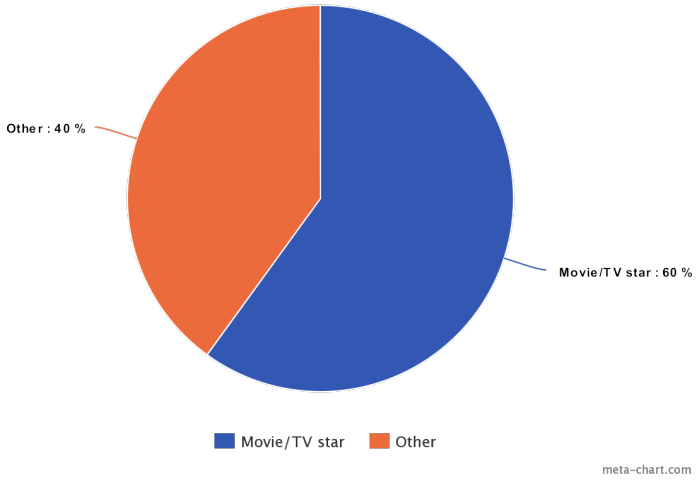


Figure 3: figure
Draft of the 3rd diagram, representing the proportion of movie/TV stars from other personalities



2.1 Hints:

- The first question could already include some diagrams (from the lecture or ones that you did yourselves).
- In step 3 explain how each of your hypothesis is falsifiable.
- In the fifth step you could state something like: "We expect to see two diagrams.

The first one has ... on the x-axis and ... on the y-axis. The image should look like a ... The second diagram ...". You could even draw a sketch of the diagram and explain how this would support or reject your testable hypothesis.

3 Statistical Validity (8 points)

In the above question, you were asked to formulate your hypothesis. In this one, you should follow your own defined roadmap from task 2 validate (or reject) your hypothesis.

3.1 Hints:

- In case feel uncomfortable to test one of the predictions from task 2 you can "steal" one of the many hypothesis (and with them imlicitly associated testable predictions) or diagrams depicted from the lecture and reproduce it. However in that case you cannot expect to get the total amount of points for task 3.

Answer:

Hypothesis:
The largest portion of the famous people and historical figures mentioned in the articles are from the USA, and the largest portion of them are movie or TV stars.

Predictions:
We expect the largest portion to be around 50% from the USA, and from them we expect around 60% movie or TV stars

Required data (input):

- The simple wikipedia 1 article per line file
- A list of all countries and states
- a shapefile representing the world map (used for ploting a thematic map)

Processing and collecting data :

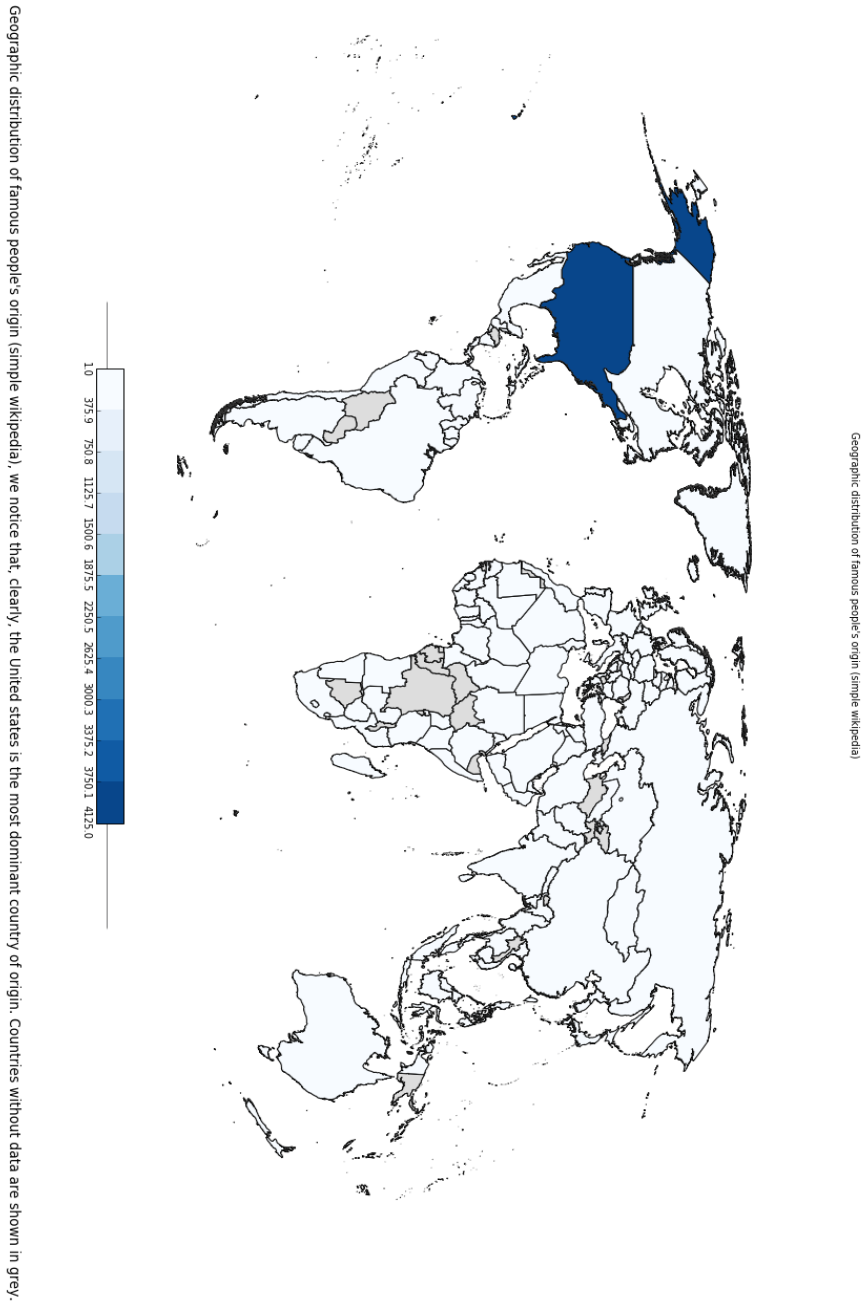
We will collect the data of each article and constitute a csv file :

Figure 4: figure
Dataframe created using the collected data stored in a csv file

code	count	name	star
LKA	6	Sri Lanka	3
ZAF	23	South Africa	5
LTU	5	Lithuania	1
ESP	120	Spain	16
ZMB	1	Zambia	0
BGR	12	Bulgaria	2
SGP	2	Singapore	0
DEU	195	Germany	51
GIN	1	Guinea	0
COL	4	Colombia	2
BMU	1	Bermuda	1
ARG	32	Argentina	14
JEY	1	Jersey	1
IMN	3	Isle of Man	0
LVA	3	Latvia	1
MRT	1	Mauritania	0
JOR	4	Jordan	2
CHN	30	China	2
VEN	7	Venezuela	0
ROU	12	Romania	6
GBR	42	United Kingdom	8

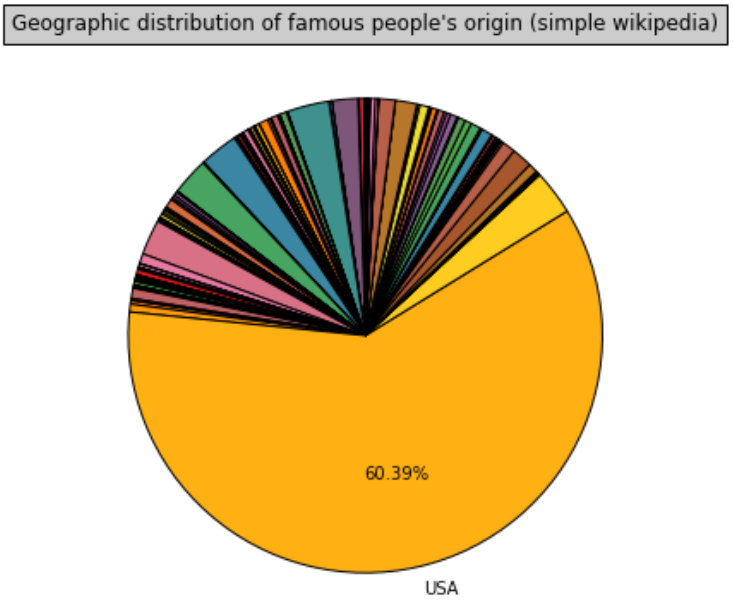
Ploting the results (Output):
We import a shapefile representing the countries of the world and their borders, we map our data with the shapefile data using the iso3 country code as key. We create a degradation of color saturation from dark blue (high density) to light blue, the difference in color saturation represents different density from one country to another.
In the following diagram, we clearly see that a very high number of people are from the US.
NOTE : Countries with no data are represented with a gray color

Figure 5: figure
Geographic distribution of famous people's origin (simple wikipedia)



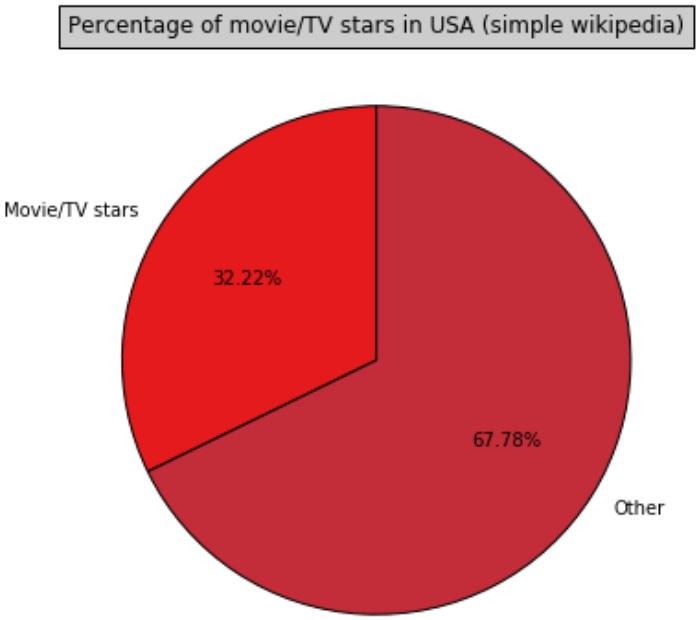
The next diagram shows that USA is the country of origin of over 60% of the people mentioned in the articles.

Figure 6: figure
Proportion of each country of origin



The last diagram shows that a third of the famous people with USA as a country of origin are movie/TV stars.

Figure 7: figure
Proportion of movie/TV stars



Analysis and final thoughts:
We expected a proportion of 50% of the famous personalities to be from the USA, our findings shows around 63%, which is in the same direction as our hypothesis. On the other side, we expected to have 60% movie/TV stars out of all the famous Americans, but our findings shows around 32.22% or about a third. while stars are not as common as we expected, a third is still quite high, and if the other 70% are distributed among many disciplines and fields, movie/TV stars could very well still be the most common category of famous people in the USA. To reach further conclusions, this will require more data exploration and more accurate algorithm to avoid traps like fictional characters born in the US.

Some curious data exploration:
If we exclude the USA, how would the distribution be illustrated?
We notice in the next diagrams that the distribution is much closer between the other countries.

Figure 8: figure
Geographic distribution of famous people's origin (simple wikipedia) excluding USA

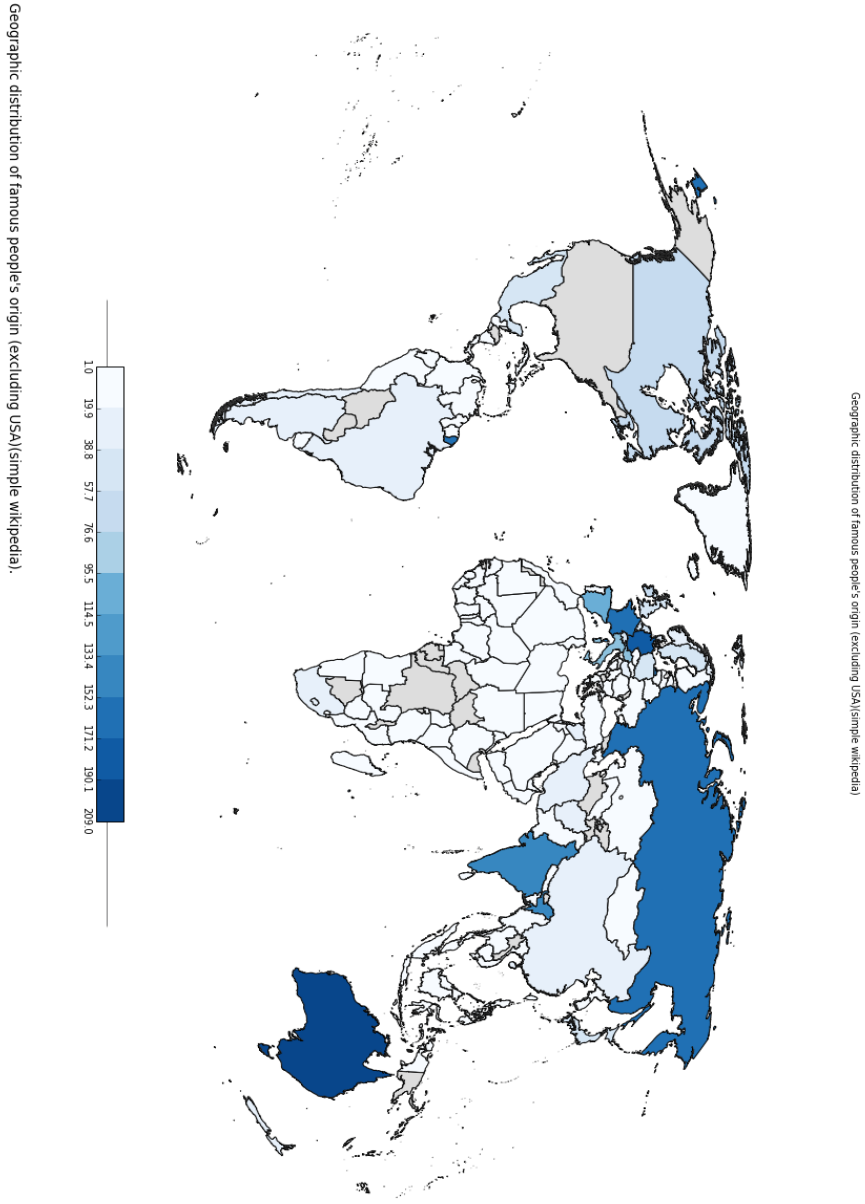
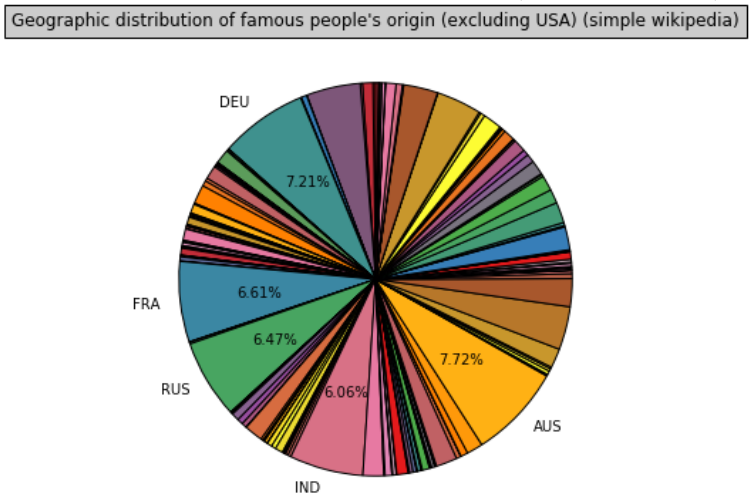


Figure 9: figure
Proportion of each country of origin (excluding USA)



Scripts:

extractData:

```
"""
Introduction to Web Science
Assignment 5
Question 3
Team : golf

Script used to extract data from the article-per-line file and process it
to finaly write it in a csv file
"""
import pandas as pd
from geonamescache import GeonamesCache
from geonamescache.mappers import country


gc = GeonamesCache() # we use the GeonamesCache to get the name of countries

# creating a mapper between the iso3 code and the country name
mapper = country(from_key='name', to_key='iso3')
countries = list(gc.get_dataset_by_key(gc.get_countries(), 'name',).keys())
# for the US we are going to use the states
states = list(gc.get_us_states_by_names())
```

```
#print(countries)
# any of these key words could indicate that we are reading about a star
key_words=['movie','film','TV','television','actor','actress']
articles=[]
dataset={}

with open('article-per-line.txt','r',encoding="utf8") as f:
    articles=f.read().splitlines()


for a in articles:
    dec=a.split('born in',1)
    proceed=True
    # we still need to optimize and factorize our code for this part
    if len(dec) > 1:
        for s in states:
            #print("Looking for %s"%s)
            if(s in dec[1]):
                star=0
                proceed=False
                for k in key_words:
                        if(k in a):
                            star=1
                            break
                country_info=dataset.get('USA')

                if(country_info):
                    country_info['count']=country_info['count']+1
                    country_info['star']=country_info['star']+star
                else:
                    dataset['USA']={'name':'United States',
                    'count':1,'star':star}
                break
        if proceed:
            for c in countries:
                #print("Looking for %s"%c)
                if(c in dec[1]):
                    star=0
                    for k in key_words:
                            if(k in a):
                                star=1
                                break
                    country_info=dataset.get(mapper(c))
```

```
                    if(country_info):
                        country_info['count']=country_info['count']+1
                        country_info['star']=country_info['star']+star
                    else:
                        dataset[mapper(c)]={'name':c,'count':1,'star':star}
                    break

country_ids=[]
frames=[]
for country_iso,d in dataset.items():
    frames.append(pd.DataFrame.from_dict(dict([['code',[country_iso]],
                                               ['name',d['name']],
                                               ['count',[d['count']]]
                                               ,['star',[d['star']]]]))))

df =pd.concat(frames)
df=df.set_index('code')
df.to_csv('dataset.csv', sep=',', encoding='utf-8')
print(df)
```

plotMap:

```
"""
Introduction to Web Science
Assignment 5
Question 3
Team : golf

Script used to read the csv file and plot it in a thematic map
"""
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from geonamescache import GeonamesCache #used to import country names
from matplotlib.patches import Polygon
from matplotlib.collections import PatchCollection
from mpl_toolkits.basemap import Basemap

filename = 'dataset.csv'
# importing shp file containing the country borders
shapefile = 'shp/countries/ne_10m_admin_0_countries'
num_colors = 12
year = '2012'
```

```
cols = ['code','count','star','name']
title = "Geographic distribution of famous people's origin (simple wikipedia)"
imgfile = 'img/{}.png'.format("foast")

descripton = "Geographic distribution of famous people's \
origin (simple wikipedia), we notice that, clearly, \
the United states is the most dominant country of origin. \
Countries without data are shown in grey."


gc = GeonamesCache()
iso3_codes = list(gc.get_dataset_by_key(gc.get_countries(), 'iso3').keys())

df = pd.read_csv(filename,  usecols=cols)
print(df)
df.set_index('code', inplace=True)
df = df.ix[iso3_codes].dropna() # Filter out non-countries and missing values.
#df.drop('USA', inplace=True)
values = df['count']
cm = plt.get_cmap('Blues')
scheme = [cm(i / num_colors) for i in range(num_colors)]
bins = np.linspace(values.min(), values.max(), num_colors)
df['bin'] = np.digitize(values, bins) - 1
df.sort_values('bin', ascending=False).head(10)

fig = plt.figure(figsize=(22, 12))

ax = fig.add_subplot(111, axisbg='w', frame_on=False)
fig.suptitle(title)

m = Basemap(lon_0=0, projection='robin')
m.drawmapboundary(color='w')

m.readshapefile(shapefile, 'units', color='#444444', linewidth=.2)
for info, shape in zip(m.units_info, m.units):
    iso3 = info['ADM0_A3']
    if iso3 not in df.index:
        color = '#dddddd'
    else:
        color = scheme[df.ix[iso3]['bin']]

    patches = [Polygon(np.array(shape), True)]
    pc = PatchCollection(patches)
    pc.set_facecolor(color)
```

```
    ax.add_collection(pc)

# Cover up Antarctica so legend can be placed over it.
ax.axhspan(0, 1000 * 1800, facecolor='w', edgecolor='w', zorder=2)

# Draw color legend.
ax_legend = fig.add_axes([0.35, 0.14, 0.3, 0.03], zorder=3)
cmap = mpl.colors.ListedColormap(scheme)
cb = mpl.colorbar.ColorbarBase(ax_legend, cmap=cmap, ticks=bins,
                               boundaries=bins, orientation='horizontal')
cb.ax.set_xticklabels([str(round(i, 1)) for i in bins])

# Set the map footer.
plt.annotate(descripton, xy=(-.8, -3.2), size=14, xycoords='axes fraction')

plt.savefig('map.png')
```

plotPies:

```
"""
Introduction to Web Science
Assignment 5
Question 3
Team : golf

Script used to read the csv file and plot pie charts representing the data
"""
import matplotlib.pyplot as plt
import pandas as pd
from matplotlib import cm
import numpy as np

filename = 'dataset.csv'
cols = ['code','count','star','name']
df = pd.read_csv(filename,  usecols=cols)
#print(df)
df.set_index('code', inplace=True)
#df.drop('USA', inplace=True)
total=sum(df['count'])
# make a square figure and axes
plt.figure(1, figsize=(6,6))
ax = plt.axes([0.1, 0.1, 0.8, 0.8])
cs=cm.Set1(np.arange(40)/40.)
# The slices will be ordered and plotted counter-clockwise.
```

```
labels = [n if (v/total >0.05) else '' for n, v in zip(df.index, df['count'])]
fracs = [(k/total) for k in df['count']]

def my_autopct(pct):
    return ('%.2f%%' % pct) if pct > 5 else ''

patches, texts, autotexts = ax.pie(fracs,  labels=labels,
                autopct=my_autopct,colors=cs,  startangle=90)

plt.title("Geographic distribution of famous people's origin\
 (simple wikipedia)", bbox={'facecolor':'0.8', 'pad':5})
plt.show()


# make a square figure and axes
plt.figure(2, figsize=(6,6))
ax = plt.axes([0.1, 0.1, 0.8, 0.8])

labels = ['Movie/TV stars','Other']
fracs = [df.at['USA','star']/(df.at['USA','count']+df.at['USA','star']),
         df.at['USA','count']/(df.at['USA','count']+df.at['USA','star'])]

patches, texts, autotexts = ax.pie(fracs,  labels=labels,
                autopct='%.2f%%',colors=cs,  startangle=90)
plt.title('Percentage of movie/TV stars in USA (simple wikipedia)',
          bbox={'facecolor':'0.8', 'pad':5})
plt.show()
```

Important Notes

Submission

- Solutions have to be checked into the github repository. Use the directory name `groupname/assignment6/` in your group's repository.
- The name of the group and the names of all participating students must be listed on each submission.
- Solution format: all solutions as *one* PDF document. Programming code has to be submitted as Python code to the github repository. Upload *all* `.py` files of your program! Use `UTF-8` as the file encoding. *Other encodings will not be taken into account!*
- Check that your code compiles without errors.
- Make sure your code is formatted to be easy to read.
  - Make sure you code has consistent indentation.
  - Make sure you comment and document your code adequately in English.
  - Choose consistent and intuitive names for your identifiers.
- Do *not* use any accents, spaces or special characters in your filenames.

Acknowledgment

This latex template was created by Lukas Schmelzeisen for the tutorials of "Web Information Retrieval".

LaTeX

Currently the code can only be build using LuaLaTeX, so make sure you have that installed. If on Overleaf, there's an error, go to settings and change the LaTeXengine to `LuaLaTeX`.