IM-UH 1511 **Introduction to Digital Humanities**

HOMEWORK 7a ¶

Best Korean Movies in IMDb: Descriptive Statistics

25 points totally

In [1]:

```
import pandas as pd
import numpy as np
import networkx as nx
import pygraphviz
from networkx.drawing.nx_agraph import graphviz_layout
from networkx.drawing.nx_agraph import to_agraph
import matplotlib.pyplot as plt
import matplotlib as mpl
from pylab import hist
import random
from collections import Counter
import operator
import itertools
from wordcloud import WordCloud
import warnings
warnings.filterwarnings("ignore", category=RuntimeWarning)
warnings.filterwarnings("ignore", category=UserWarning)
warnings.simplefilter('ignore')
```

In [2]:

```
allfilms = pd.read_csv('300KoreanFilms.csv', sep=',', encoding="utf-8")
films=allfilms.sample(100)
films = films.reset_index(drop=True)
print(len(films))
films
```

```
100
```

Out[2]:

	TITLE	YEAR	DIRECTOR	STARS	GENRE
0	Parasite	2019	Joon-ho Bong (D)	Kang-ho Song, Sun-kyun Lee, Yeo-jeong Jo	Comedy, Drama, Thriller
1	Tazza: The High Rollers	2006	Dong-hoon Choi	Seung-woo Cho, Yun-shik Baek, Hye-su Kim, Hae-...	Comedy, Crime
2	Hwasango	2001	Tae-gyun Kim	Hyuk Jang, Min-a Shin, Su-ro Kim, Sang-Woo Kwon	Action, Comedy, Fantasy
3	As One	2012	Hyeon-seong Moon	Ji-won Ha, Doona Bae, Han Yeri, Yoon-young Choi	Drama, Sport
4	Perfect Number	2012	Eun-jin Pang (D)	Seung-bum Ryoo, Yo-won Lee, Jin-woong Cho, Yoo...	Drama, Thriller
...	...	...	...	...	...
95	Lost Flower Eo Woo-dong	2015	Soo Sung Lee	Do-bin Baek, Eun-pi Kang, Wook Han Yeo, Kyung-...	Drama

In [3]:

```
titles=films.TITLE.tolist()
print(len(titles),len(set(titles)))
```

```
100 99
```

In [4]:

```
year=films.YEAR.tolist()
year=list(set(year))
print(len(year),min(year),max(year))
sorted(year)
```

```
21 1998 2019
```

Out[4]:

```
[1998,
 1999,
 2000,
 2001,
 2002,
 2003,
 2004,
 2005,
 2006,
 2007,
 2008,
 2009,
 2010,
 2011,
 2012,
 2013,
 2014,
 2015,
 2016,
 2017,
 2019]
```

In [5]:

```
director=films.DIRECTOR.tolist()
director=sorted(list(set(director)))
print(len(director))
director
```

```
76
```

In [6]:

```
s_films=pd.DataFrame()
for i in range(len(films)):
    d=dict(films.loc[i])
    sl= films.STARS[i].split(",")
    gl=films.GENRE[i].split(", ")
    for s in sl:
        for t in gl:
            d["ACTOR"]=s.strip()
            d['UGENRE']=t
            s_films=s_films.append(d,ignore_index=True)
s_films['YEAR'] = s_films['YEAR'].astype(int)
s_films.rename(columns={'GENRE':'MGENRE','UGENRE':'GENRE'}, inplace=True)
s_films=s_films[["YEAR","TITLE","DIRECTOR","ACTOR","GENRE"]]
df=s_films
print(len(df))
df.sort_values(by="YEAR").head(20)
```

```
920
```

Out[6]:

	YEAR	TITLE	DIRECTOR	ACTOR	GENRE
728	1998	Christmas in August	Jin-ho Hur	Suk-kyu Han	Drama
735	1998	Christmas in August	Jin-ho Hur	Ji-hye Oh	Romance
734	1998	Christmas in August	Jin-ho Hur	Ji-hye Oh	Drama
733	1998	Christmas in August	Jin-ho Hur	Goo Shin	Romance
732	1998	Christmas in August	Jin-ho Hur	Goo Shin	Drama
730	1998	Christmas in August	Jin-ho Hur	Eun-ha Shim	Drama
729	1998	Christmas in August	Jin-ho Hur	Suk-kyu Han	Romance
731	1998	Christmas in August	Jin-ho Hur	Eun-ha Shim	Romance
217	1999	Memento Mori	Tae-yong Kim	Yeong-jin Lee	Drama
361	1999	Memento Mori	Kyu-dong Min	Yeong-jin Lee	Drama
360	1999	Memento Mori	Kyu-dong Min	Yeong-jin Lee	Romance

In [7]:

```
df.to_csv('Random100KoreanFilms.csv',encoding='utf-8')
```

In [8]:

```
actor=df.ACTOR.tolist()
actor=sorted(list(set(actor)))
print(len(actor))
actor
```

```
291
```

In [9]:

```
unique_genre=df.GENRE.tolist()
unique_genre=sorted(list(set(unique_genre)))
print(len(unique_genre))
unique_genre
```

```
17
```

Out[9]:

```
['Action',
 'Adventure',
 'Animation',
 'Biography',
 'Comedy',
 'Crime',
 'Drama',
 'Family',
 'Fantasy',
 'History',
 'Horror',
 'Music',
 'Mystery',
 'Romance',
 'Sci-Fi',
 'Sport',
 'Thriller']
```

```
In [10]: title=df.TITLE.tolist()
         title=sorted(list(set(title)))
         print(len(title))
         title
```

```
99
```

Grouping per Year

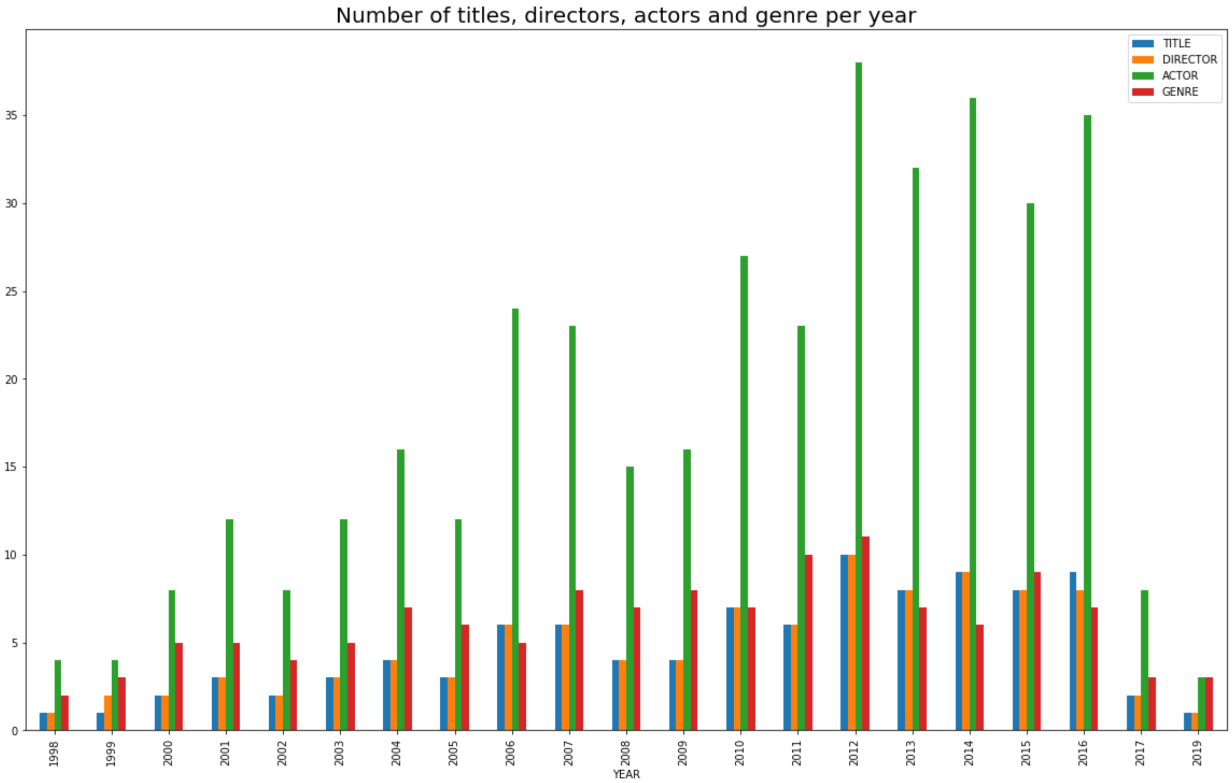
In [11]:

```
gdf=df.groupby("YEAR").nunique()[["TITLE","DIRECTOR","ACTOR","GENRE"]]
gdf = gdf.reset_index()
gdf
```

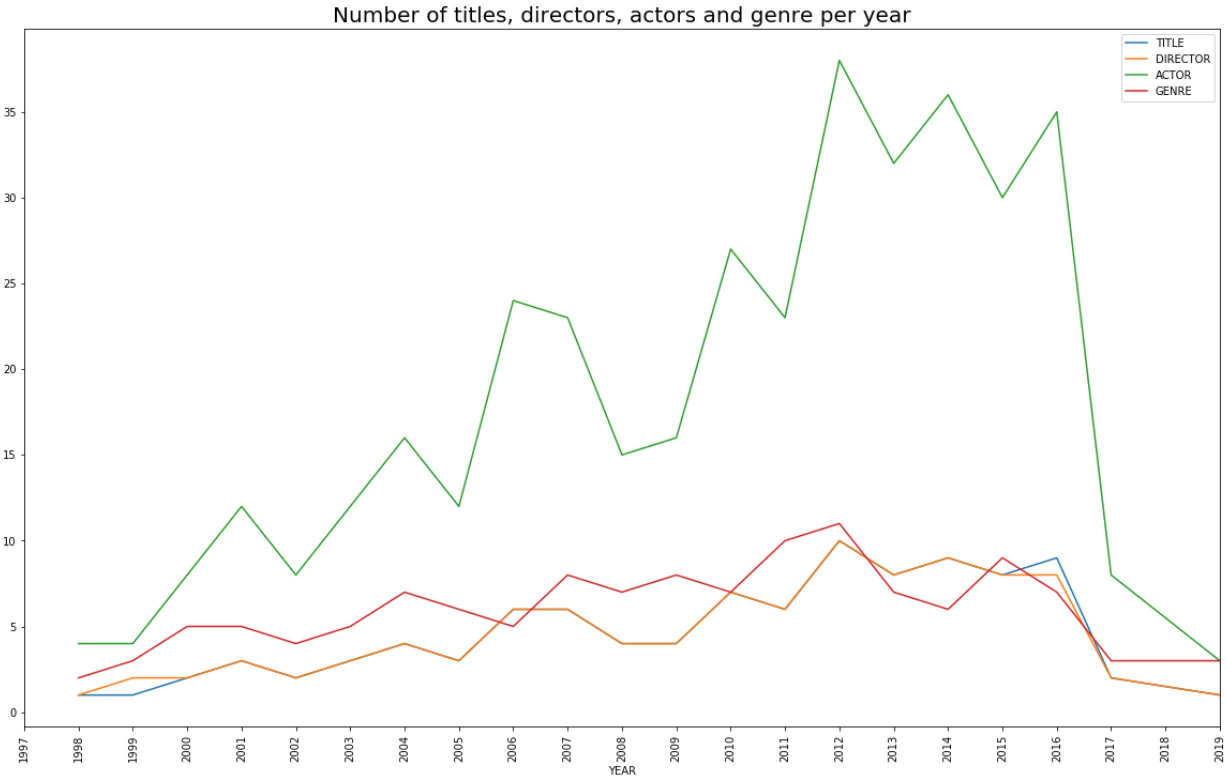
Out[11]:

	YEAR	TITLE	DIRECTOR	ACTOR	GENRE
0	1998	1	1	4	2
1	1999	1	2	4	3
2	2000	2	2	8	5
3	2001	3	3	12	5
4	2002	2	2	8	4
5	2003	3	3	12	5
6	2004	4	4	16	7
7	2005	3	3	12	6
8	2006	6	6	24	5
9	2007	6	6	23	8
10	2008	4	4	15	7
11	2009	4	4	16	8
12	2010	7	7	27	7
13	2011	6	6	23	10
14	2012	10	10	38	11
15	2013	8	8	32	7
16	2014	9	9	36	6
17	2015	8	8	30	9
18	2016	9	8	35	7
19	2017	2	2	8	3
20	2019	1	1	3	3

```
In [12]: ax=gdf.plot(x='YEAR', y=["TITLE", "DIRECTOR", "ACTOR", "GENRE"], kind="bar"
         ax.set_title('Number of titles, directors, actors and genre per year', font
```



In [13]:
```
ax=gdf.plot(x='YEAR', y=["TITLE", "DIRECTOR", "ACTOR", "GENRE"], kind="line
plt.xticks(np.arange(1997, 2020, step=1),rotation='vertical');
ax.set_title('Number of titles, directors, actors and genre per year', font
```



Grouping per Genre

In [14]:

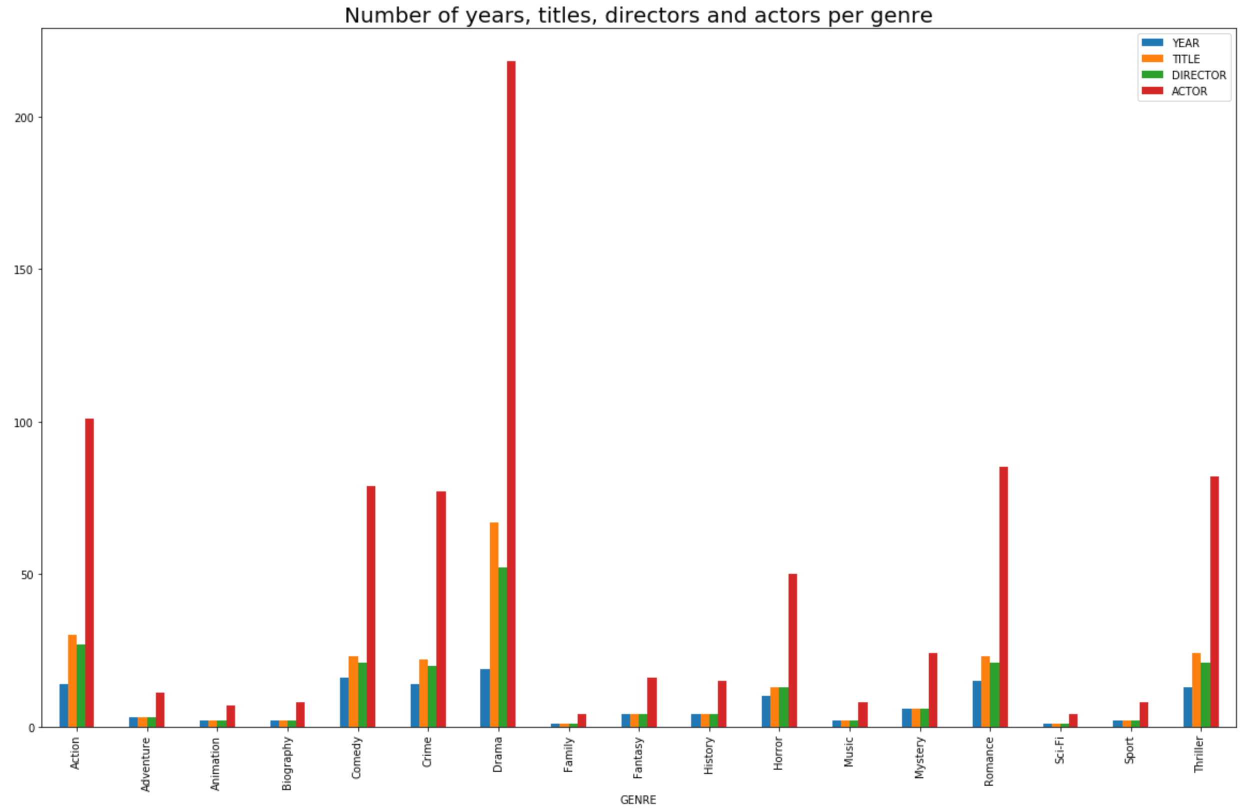
```
ggdf=df.groupby("GENRE").nunique()[["YEAR","TITLE","DIRECTOR","ACTOR"]]
ggdf = ggdf.reset_index()
ggdf
```

Out[14]:

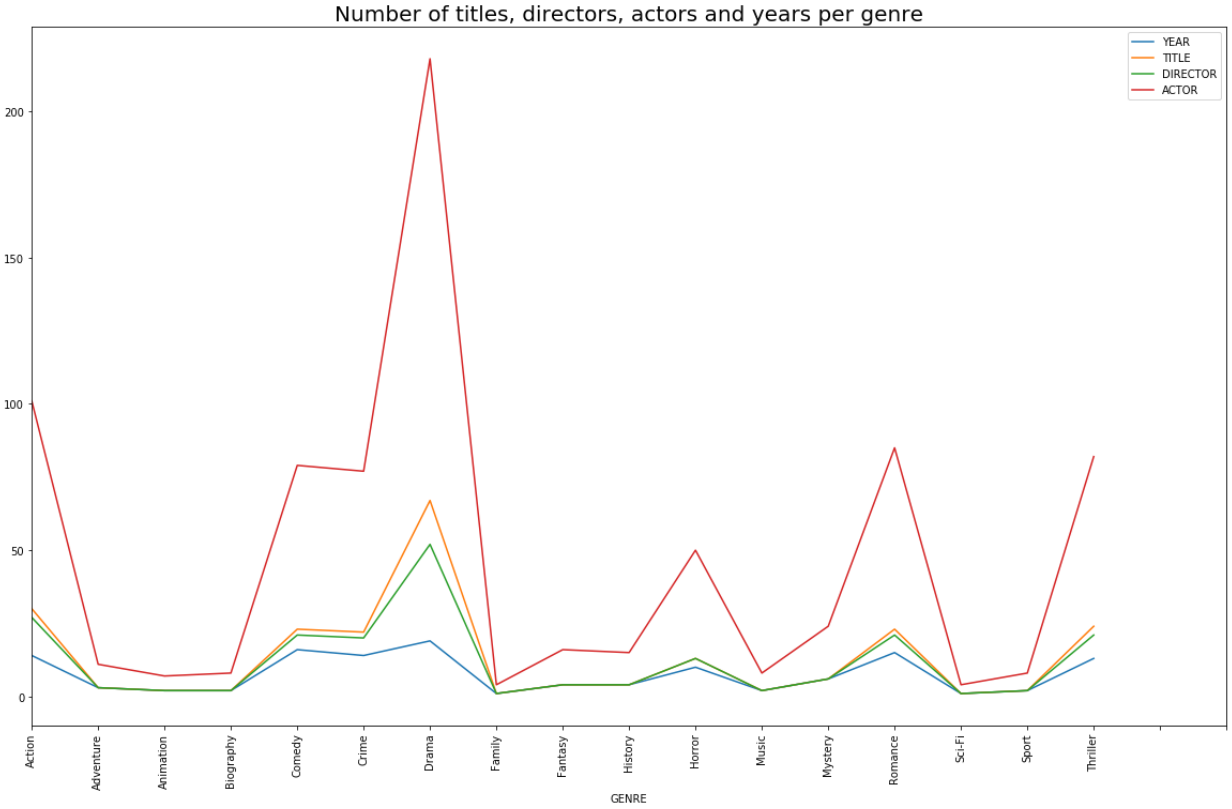
	GENRE	YEAR	TITLE	DIRECTOR	ACTOR
0	Action	14	30	27	101
1	Adventure	3	3	3	11
2	Animation	2	2	2	7
3	Biography	2	2	2	8
4	Comedy	16	23	21	79
5	Crime	14	22	20	77
6	Drama	19	67	52	218
7	Family	1	1	1	4
8	Fantasy	4	4	4	16
9	History	4	4	4	15
10	Horror	10	13	13	50
11	Music	2	2	2	8
12	Mystery	6	6	6	24
13	Romance	15	23	21	85
14	Sci-Fi	1	1	1	4
15	Sport	2	2	2	8
16	Thriller	13	24	21	82

In [15]:

```
ax=ggdf.plot(x='GENRE', y=["YEAR","TITLE", "DIRECTOR", "ACTOR"], kind="bar"
ax.set_title('Number of years, titles, directors and actors per genre', fon
```



In [16]:
```
ax=ggdf.plot(x='GENRE', y=["YEAR", "TITLE", "DIRECTOR", "ACTOR"], kind="lin
ax.set_xticks(np.arange(0, 19, step=1))
ax.set_xticklabels(unique_genre,rotation='vertical')
ax.set_title('Number of titles, directors, actors and years per genre', fon
```



Grouping per Director

In [17]:
```
dgdf=df.groupby("DIRECTOR").nunique()[["YEAR","TITLE","ACTOR",'GENRE']]
dgdf = dgdf.reset_index()
dgdf.sort_values(by="TITLE",ascending=False)
```

Out[17]:

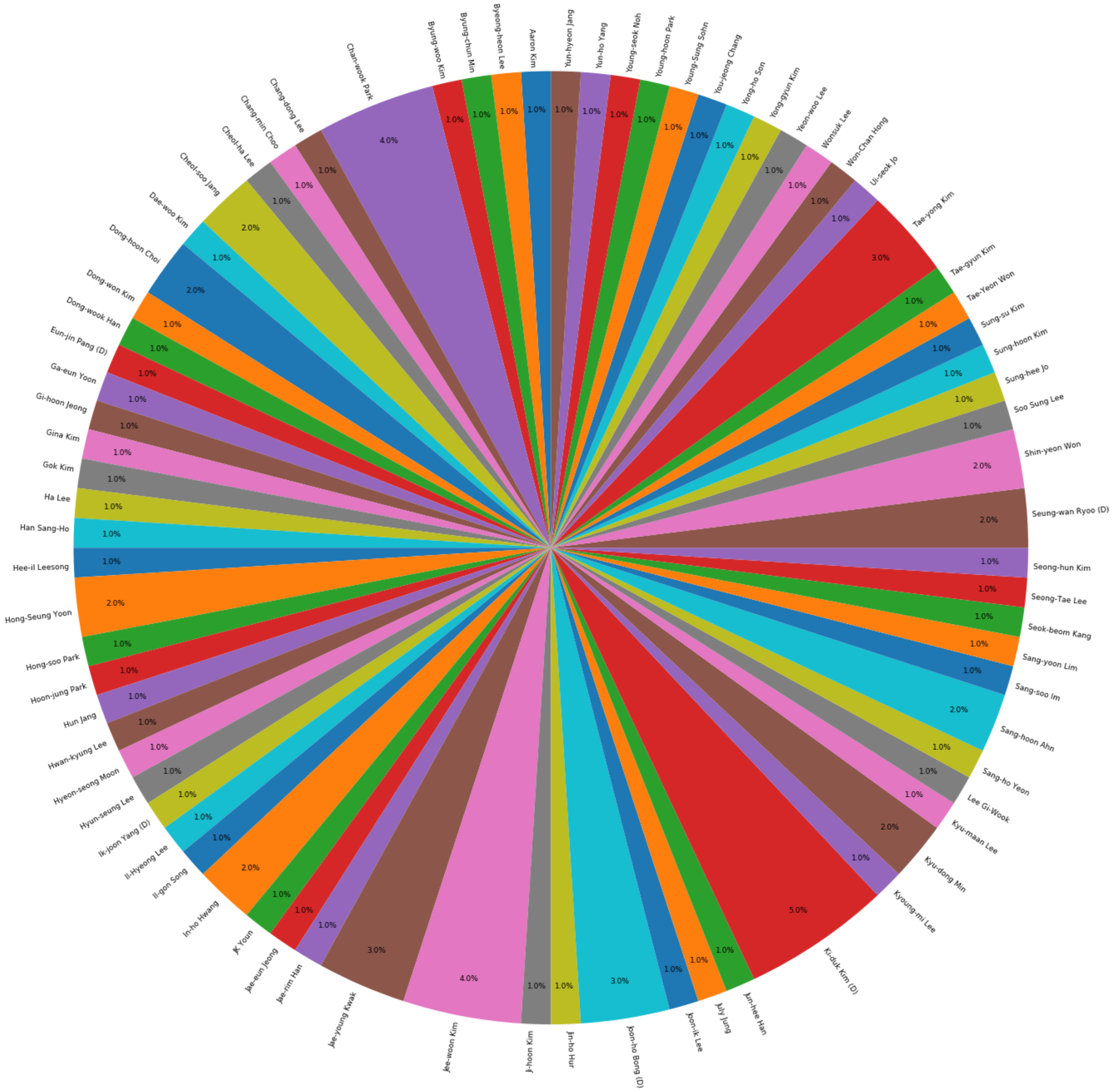
	DIRECTOR	YEAR	TITLE	ACTOR	GENRE
43	Ki-duk Kim (D)	5	5	20	5
4	Chan-wook Park	3	4	15	7
36	Jee-woon Kim	4	4	14	8
63	Tae-yong Kim	3	3	12	3
39	Joon-ho Bong (D)	3	3	10	5
...	...	...	...	...	...
33	Jae-eun Jeong	1	1	4	1
34	Jae-rim Han	1	1	4	3
37	Ji-hoon Kim	1	1	4	3
1	Byeong-heon Lee	1	1	4	3
75	Yun-hyeon Jang	1	1	4	3

76 rows × 5 columns

In [18]:

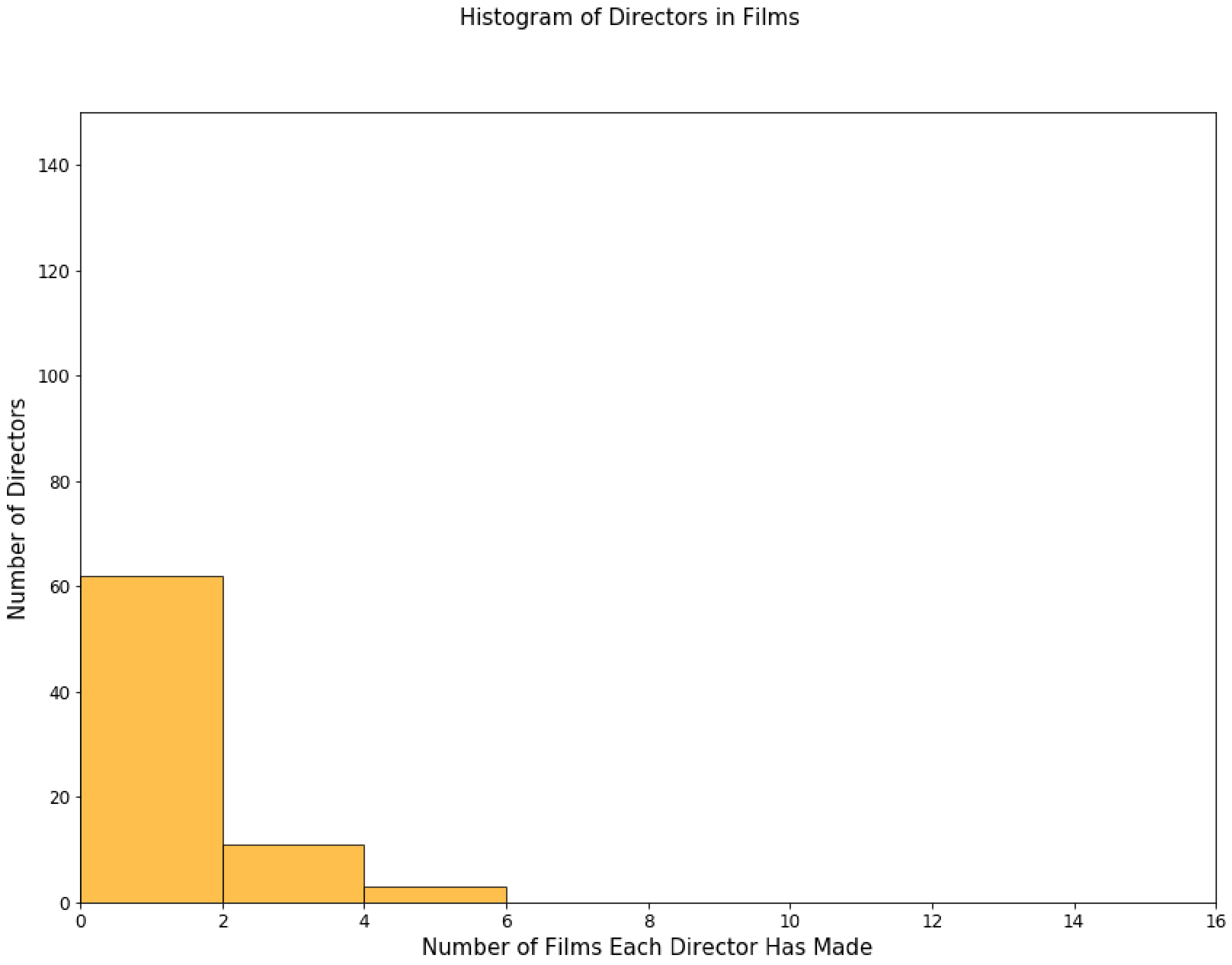
```
close = dgdf
mpl.rcParams['font.size'] = 12.0
fig = plt.figure(figsize=(20,20))
plt.pie(
    close['TITLE'],
    labels=close['DIRECTOR'],
    shadow=False,
    startangle=90,
    # with the percent listed as a fraction
    autopct='%1.1f%%',
    rotatelabels=True,labeldistance=1.01,textprops={'fontsize':9},pctdistan
    )

# View the plot drop above
plt.axis('equal')

# View the plot
ss="Number of Titles per Director"
plt.suptitle(ss,fontsize=25)
plt.tight_layout(rect=[0, 0, 1, 0.93])
plt.show()
```

Number of Titles per Director



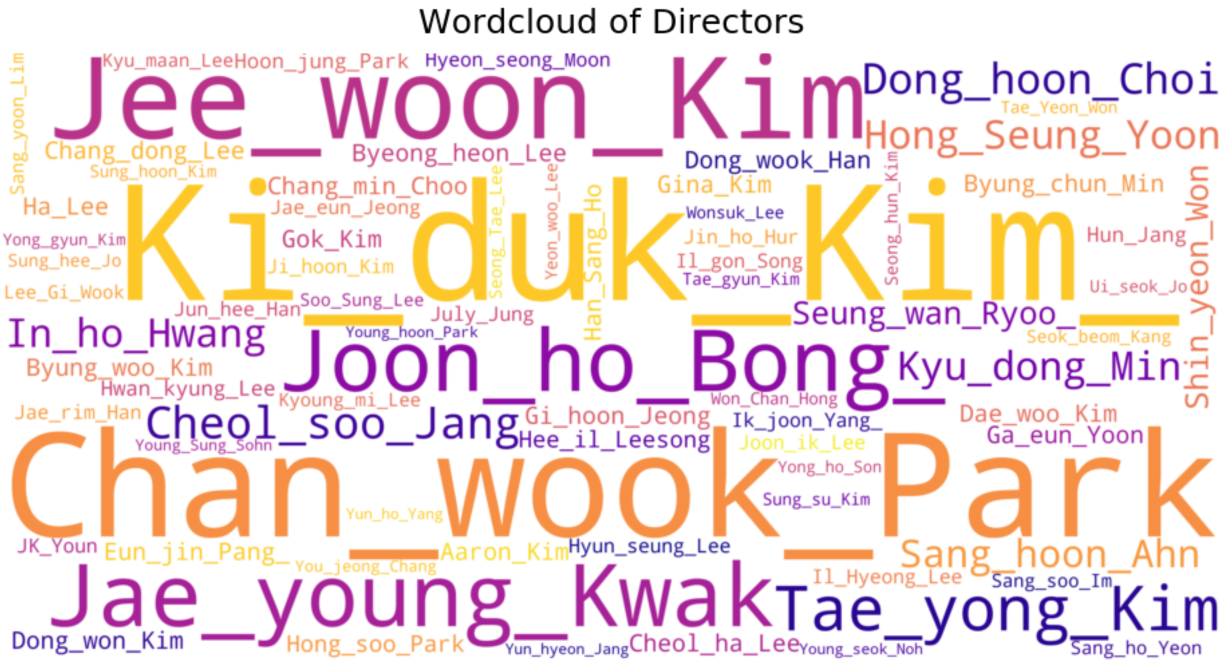
In [19]:

```
x=dgdf.TITLE.tolist()
fig = plt.figure(figsize=(14,10))
plt.hist(x,bins = [0,2,4,6,8,10,12,14,16],edgecolor='black',color="orange",
plt.axis([0, 16, 0, 150])
plt.xlabel("Number of Films Each Director Has Made",fontsize=15)
plt.ylabel("Number of Directors",fontsize=15)
plt.xticks([0,2,4,6,8,10,12,14,16])
plt.suptitle("Histogram of Directors in Films",fontsize=15)
plt.show()
```



In [20]:

```
subsetd = dgdf[['DIRECTOR', 'TITLE']]
tuplesd = [tuple(x) for x in subsetd.values]

t=[]
for (i,j) in tuplesd:
    for k in range(j):
#           print(i.replace(" ","_").replace("-","_"))
        t.append(i.replace(" ","_").replace("-","_"))
ttd=' '.join(t)

wordcloud = WordCloud(collocations=False,background_color="white",colormap=
fig = plt.figure(figsize=(13,13))
default_colors = wordcloud.to_array()
plt.imshow(default_colors, interpolation="bilinear")
plt.axis("off")
ss="Wordcloud of Directors"
plt.suptitle(ss,fontsize=25)
plt.tight_layout(rect=[0, 0, 1, 1.4])
plt.show()
```



Grouping per Actor

In [21]:

```
agdf=df.groupby("ACTOR").nunique()[["YEAR","DIRECTOR","TITLE",'GENRE']]
agdf = agdf.reset_index()
agdf.sort_values(by="TITLE",ascending=False)
```

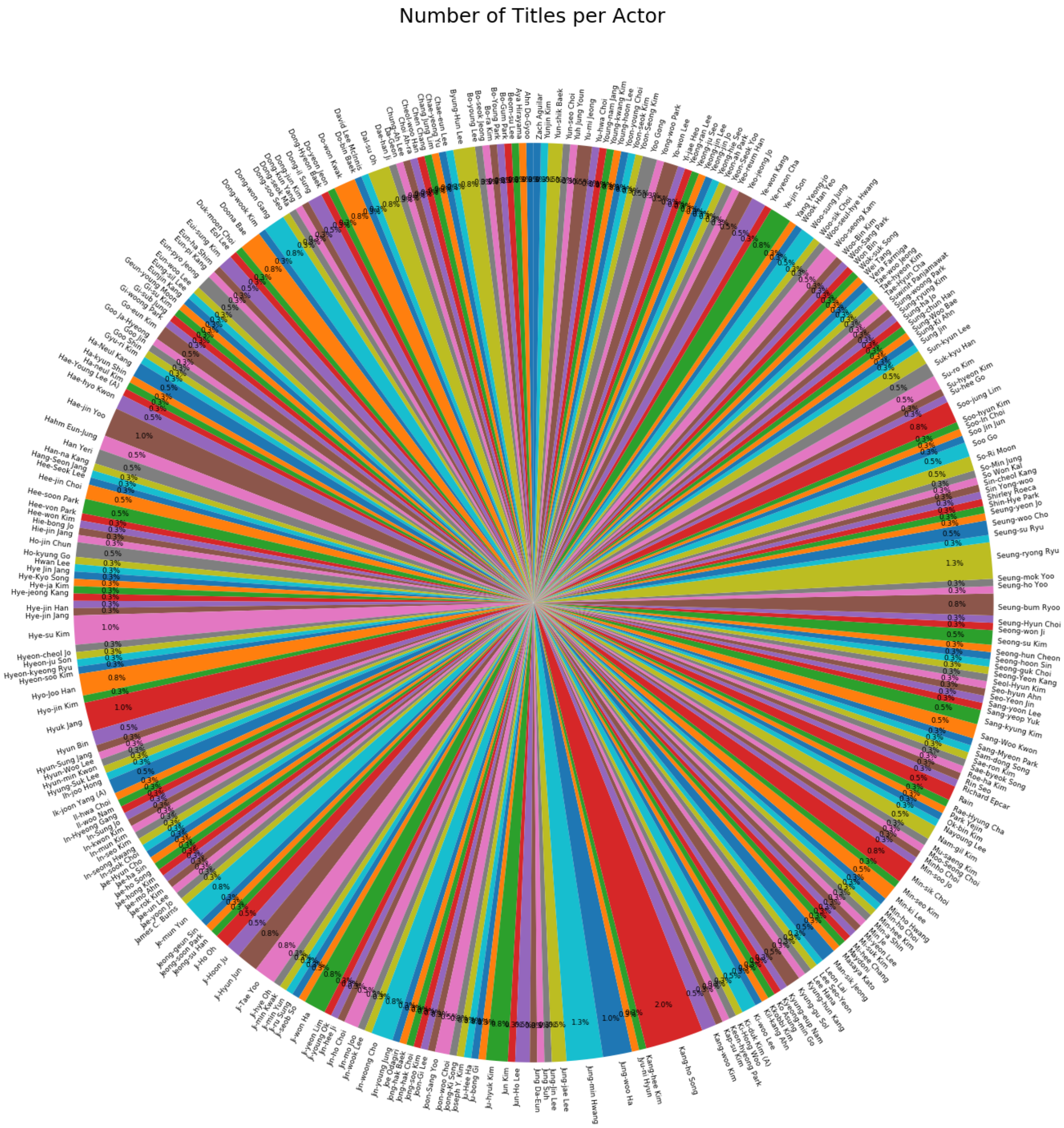
Out[21]:

	ACTOR	YEAR	DIRECTOR	TITLE	GENRE
153	Kang-ho Song	7	6	8	8
149	Jung-min Hwang	4	5	5	4
218	Seung-ryong Ryu	5	5	5	7
55	Hae-jin Yoo	4	3	4	5
76	Hye-su Kim	4	4	4	6
...	...	...	...	...	...
111	Jeong-soon Park	1	1	1	2
112	Jeong-su Han	1	1	1	2
117	Ji-hye Oh	1	1	1	2
118	Ji-min Kwak	1	1	1	1
290	Zach Aguilar	1	1	1	3

291 rows × 5 columns

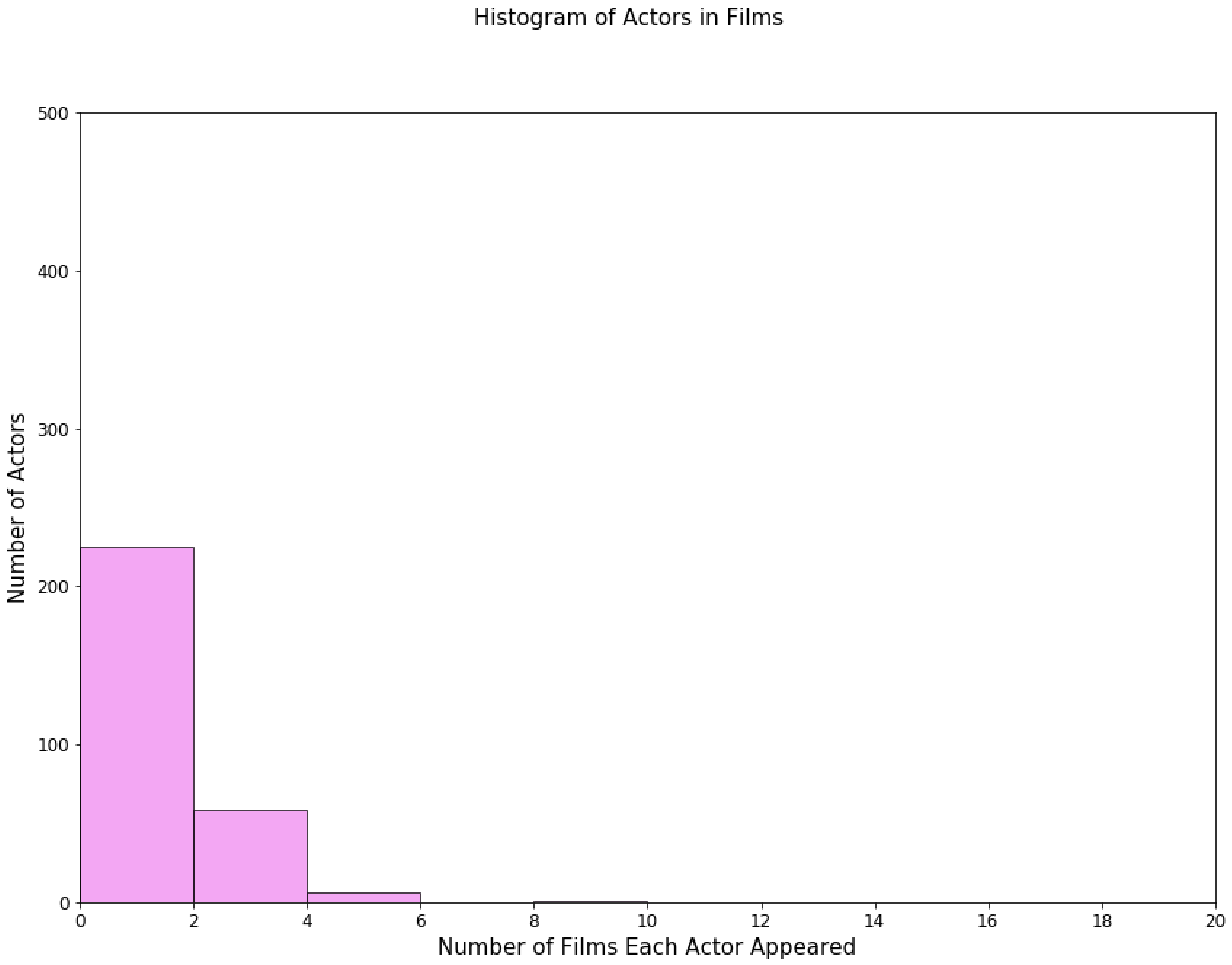
In [22]:

```
aclose = agdf
mpl.rcParams['font.size'] = 12.0
fig = plt.figure(figsize=(20,20))
plt.pie(
    aclose['TITLE'],
    labels=aclose['ACTOR'],
    shadow=False,
    startangle=90,
    # with the percent listed as a fraction
    autopct='%1.1f%%',
    rotatelabels=True,labeldistance=1.01,textprops={'fontsize':9},pctdistan
    )

# View the plot drop above
plt.axis('equal')

# View the plot
ss="Number of Titles per Actor"
plt.suptitle(ss,fontsize=25)
plt.tight_layout(rect=[0, 0, 1, 0.93])
plt.show()
```

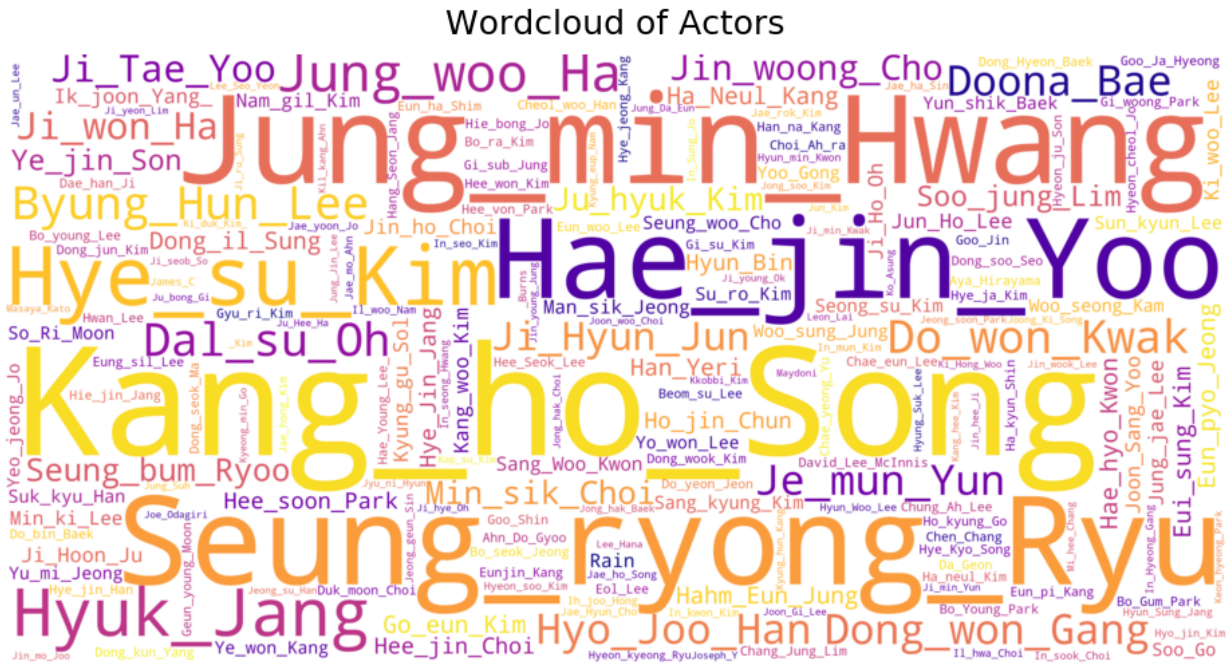

Number of Titles per Actor
Ahn Do-Gyoo
Aya Hirayama
Beom-su Lee
Bo-Gum Park
Bo-Young Park
Bo-ra Kim
Bo-seok Jeong
Bo-young Lee
Byung-Hun Lee
Chae-eun Lee
Chae-yeong Yu
Chang Jung Lim
Chen Chang
Cheol-woo Han
Choi Ah-ra
Chung-Ah Lee
Da-Geon
Dae-han Ji
Dal-su Oh
David Lee McInnis
Do-bin Baek
Do-won Kwak
Do-yeon Jeon
Dong-Hyeon Baek
Dong-il Sung
Dong-jun Kim
Dong-kun Yang
Dong-seok Ma
Dong-soo Seo
Dong-won Gang
Dong-wook Kim
Doona Bae
Duk-moon Choi
Eol Lee
Eui-sung Kim
Eun-ha Shim
Eun-pi Kang
Eun-pyo Jeong
Eun-woo Lee
Eung-sil Lee
Eunjin Kang
Geun-young Moon
Gi-su Kim
Gi-sub Jung
Gi-woong Park
Go-eun Kim
Goo Ja-Hyeong
Goo Jin
Goo Shin
Gyu-ri Kim
Ha-Neul Kang
Ha-kyun Shin
Ha-neul Kim
Hae-Young Lee (A)
Hae-hyo Kwon
Hae-jin Yoo
Hahm Eun-Jung
Han Yeri
Han-na Kang
Hang-Seon Jang
Hee-Seok Lee
Hee-jin Choi
Hee-soon Park
Hee-von Park
Hee-won Kim
Hie-bong Jo
Hie-jin Jang
Ho-jin Chun
Ho-kyung Go
Hwan Lee
Hye Jin Jang
Hye-Kyo Song
Hye-ja Kim
Hye-jeong Kang
Hye-jin Han
Hye-jin Jang
Hye-su Kim
Hyeon-cheol Jo
Hyeon-ju Son
Hyeon-kyeong Ryu
Hyeon-soo Kim
Hyo-Joo Han
Hyo-jin Kim
Hyuk Jang
Hyun Bin
Hyun-Sung Jang
Hyun-Woo Lee
Hyun-min Kwon
Hyung-Suk Lee
Ih-joo Hong
Ik-joon Yang (A)
Il-hwa Choi
Il-woo Nam
In-Hyeong Gang
In-Sung Jo
In-kwon Kim
In-mun Kim
In-seo Kim
In-seong Hwang
In-sook Choi
Jae-Hyun Cho
Jae-ha Sin
Jae-ho Song
Jae-hong Kim
Jae-mo Ahn
Jae-rok Kim
Jae-un Lee
Jae-yoon Jo
James C. Burns
Je-mun Yun
Jeong-geun Sin
Jeong-soon Park
Jeong-su Han
Ji-Ho Oh
Ji-Hoon Ju
Ji-Hyun Jun
Ji-Tae Yoo
Ji-hye Oh
Ji-min Kwak
Ji-min Yun
Ji-ru Sung
Ji-seob So
Ji-won Ha
Ji-yeon Lim
Ji-young Ok
Jin-hee Ji
Jin-ho Choi
Jin-mo Joo
Jin-wook Lee
Jin-woong Cho
Jin-young Jung
Joe Odagiri
Jong-hak Baek
Jong-hak Choi
Jong-soo Kim
Joon-Gi Lee
Joon-Sang Yoo
Joon-woo Choi
Joong-Ki Song
Joseph Y. Kim
Ju-Hee Ha
Ju-bong Gi
Ju-hyuk Kim
Jun Kim
Jun-Ho Lee
Jung Da-Eun
Jung Suh
Jung-Jin Lee
Jung-jae Lee
Jung-min Hwang
Jung-woo Ha
Jyu-ni Hyun
Kang-hee Kim
Kang-ho Song
Kang-woo Kim
Kap-su Kim
Keon-hyeong Park
Ki-Hong Woo
Ki-duk Kim (A)
Ki-woo Lee
Kil-kang Ahn
Kkobbi Kim
Ko Asung
Kyeong-min Go
Kyung-eup Nam
Kyung-gu Sol
Kyung-hun Kang
Lee Hana
Lee Seo-Yeon
Leon Lai
Man-sik Jeong
Masaya Kato
Maydoni
Mi-hee Chang
Mi-suk Kim
Mi-yeon Lee
Min Je
Min-a Shin
Min-hee Kim
Min-ho Choi
Min-ho Hwang
Min-ki Lee
Min-seo Kim
Min-sik Choi
Min-soo Jo
Minho Choi
Moo-Seong Choi
Mu-saeng Kim
Nam-gil Kim
Nayoung Lee
Ok-bin Kim
Park Yejin
Rae-Hyung Cha
Rain
Richard Epcar
Rin Seo
Roe-ha Kim
Sae-byeok Song
Sae-ron Kim
Sam-dong Song
Sang-Myeon Park
Sang-Woo Kwon
Sang-kyung Kim
Sang-yeop Yuk
Sang-yoon Lee
Seo-Yeon Jin
Seo-hyun Ahn
Seol-Hyun Kim
Seong-Yeon Kang
Seong-guk Choi
Seong-hoon Sin
Seong-hun Cheon
Seong-su Kim
Seong-won Ji
Seung-Hyun Choi
Seung-bum Ryoo
Seung-ho Yoo
Seung-mok Yoo
Seung-ryong Ryu
Seung-su Ryu
Seung-woo Cho
Seung-yeon Jo
Shin-Hye Park
Shirley Roeca
Sin Yong-woo
Sin-cheol Kang
So Won Kal
So-Min Jung
So-Ri Moon
Soo Go
Soo Jin Jun
Soo-In Choi
Soo-hyun Kim
Soo-jung Lim
Su-hee Go
Su-hyeon Kim
Su-ro Kim
Suk-kyu Han
Sun-kyun Lee
Sung Jin
Sung-Ki Ahn
Sung-Woo Bae
Sung-chun Han
Sung-ha Jo
Sung-ryung Kim
Sung-woong Park
Suwinit Panjamawat
Tae-Hyun Cha
Tae-hyeon Kim
Tae-woo Jeong
Vera Farmiga
Wei Tang
WoK-suk Song
Won Bin
Won-Sang Park
Woo-Bin Kim
Woo-seong Kam
Woo-seul-hye Hwang
Woo-sik Choi
Woo-sung Jung
Wook Han Yeo
Yang Yeong-jo
Ye-jin Son
Ye-ryeon Cha
Ye-won Kang
Yeo-jeong Jo
Yeo-reum Han
Yeon-Seok Yoo
Yeon-ah Park
Yeong-hie Seo
Yeong-jin Jo
Yeong-jin Lee
Yeong-ju Seo
Yeong-ran Lee
Yi-jae Heo
Yo-won Lee
Yong-woo Park
Yoo Gong
Yoon-Seong Kim
Yoon-seok Kim
Yoon-young Choi
Young-hoon Lee
Young-kwang Kim
Young-nam Jang
Yu-hwa Choi
Yu-mi Jeong
Yuh Jung Youn
Yun-seo Choi
Yun-shik Baek
Yunjin Kim
Zach Aguilar
2.0%
1.3%
1.0%
0.8%
0.5%
0.3%

In [23]:

```
x=agdf.TITLE.tolist()
fig = plt.figure(figsize=(14,10))
plt.hist(x,bins = [0,2,4,6,8,10,12,14,16,18,20],edgecolor='black',color="vi
plt.axis([0, 20, 0, 500])
plt.xlabel("Number of Films Each Actor Appeared",fontsize=15)
plt.ylabel("Number of Actors",fontsize=15)
plt.xticks([0,2,4,6,8,10,12,14,16,18,20])
plt.suptitle("Histogram of Actors in Films",fontsize=15)
plt.show()
```



```
In [24]: subset = agdf[['ACTOR', 'TITLE']]
         tuples = [tuple(x) for x in subset.values]

         t=[]
         for (i,j) in tuples:
             for k in range(j):
         #         print(i.replace(" ","_").replace("-","_"))
                 t.append(i.replace(" ","_").replace("-","_"))
         tt=' '.join(t)

         wordcloud = WordCloud(collocations=False,background_color="white",colormap=
         fig = plt.figure(figsize=(13,13))
         default_colors = wordcloud.to_array()
         plt.imshow(default_colors, interpolation="bilinear")
         plt.axis("off")
         ss="Wordcloud of Actors"
         plt.suptitle(ss,fontsize=25)
         plt.tight_layout(rect=[0, 0, 1, 1.4])
         plt.show()
```



```
In [ ]:
```