Pattern

Patent

Northwestern

Let’s Pattern USPTO Patents

Are you thinking filing patents on your own?

Is your company looking for a patent agent/attorney?

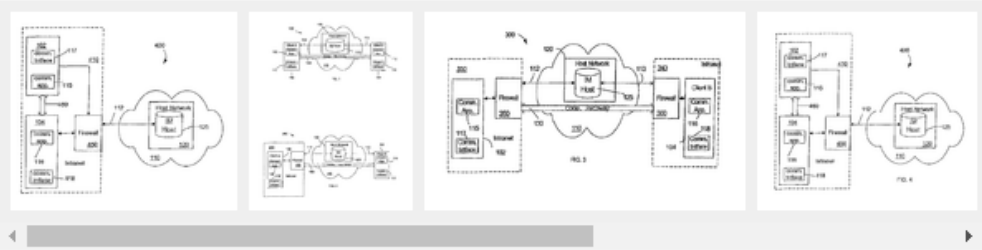
- *Would you afford the time cost?*
- *Is your case likely to be patented?*
- *Which attorney are you talking to?*
- *Would you want to live around other inventors?*
- *Do you know the examiner's comfort zone?*

Implicit population of access control lists

Abstract

Communication applications may include lists of users with which a user of the application communicates. If two users of a communications application each include the other user on their user lists, an implicit trust may be established between the users. For example, if user A includes user B in her list and user B includes user A in his list, then it may be determined that each user knows and/or trusts the other user. As a result, a connection or communications pathway may be automatically created between the client devices of the users to facilitate communications between the users based on the implicit trust.

Images (7)



Classifications

H04L63/0272 Virtual private networks

USRE45254E1
United States

Download PDF | Find Prior Art | Similar

Inventor: James A. Roskind

Current Assignee : Facebook Inc

Worldwide applications

2002 • US 2007 • US 2009 • US 2013 • US 2014 • US

Application US13/907,761 events

2002-12-31 • Priority to US10/334,142

2013-05-31 • Application filed by Facebook Inc

2014-11-18 • Publication of USRE45254E1

2014-11-18 • Application granted

Status • Active

2022-12-31 • Anticipated expiration

https://patents.google.com/patent/USRE45254

Description

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation application of and claims priority to U.S. application Ser. No. 11/782,461, now U.S. Pat. No. 7,490,238, filed Jul. 24, 2007, which is a continuation of U.S. application Ser. No. 10/334,142, now U.S. Pat. No. 7,263,614, filed on Dec. 31, 2002, the entire contents all of which are hereby incorporated by reference.

TECHNICAL FIELD

The following description relates to network communications.

BACKGROUND

With the rapid proliferation and affordability of computers, the Internet has become the communications medium of choice for many users. Although the Internet is a public medium, techniques have been developed for using the Internet to enable private communications between networks. One such private communications technique is used to enable instant messaging.

Instant messaging allows users to rapidly communicate with other users of a communications network. Generally, client messaging software runs on a client A device **102** and provides a communications interface for entry of a message. The intended message recipient may be entered manually or may be selected from a user list, such as a Buddy List™ from America Online, Inc. Instant messaging may be used to communicate text messages, images, and sounds or voice.

SUMMARY

In one general aspect, messaging applications, systems, and methods may be used to automatically configure a communications pathway based on an implicit trust between users. Each user of a communications application may have a user list that identifies other users to which a message may be sent. If two users of the communications application each include the other user on their user lists, an implicit trust may be inferred between the users. For example, if user A includes user B in her user list and user B includes user A in his. user list, then it may be inferred or determined that each user knows and/or implicitly trusts the other user. As a result, a connection or communications pathway may be automatically created and/or configured between the client devices of the users to facilitate communications between the users based on the implicit trust.

Claims (33)

Show Dependent

What is claimed is:

1. A method comprising:

logging, by a server, a first client device into a the server;

logging, by the server, a second client device into the server;

accessing, using by the server, a first user list associated with a first user of the first client device;

accessing, using by the server, a second user list associated with a second user of the second client device, wherein the first user list and the second user list are maintained separately from one another;

analyzing, using by the server, the accessed first user list to determine whether an identifier of the second user is included in the first user list;

analyzing, using by the server, the accessed second user list to determine whether an identifier of the first user is included in the second user list; and

regulating, using by the server, a communications pathway communications between the first client device and the second client device based on both the determination of whether the identifier of the first user is included on the second user list and the determination of whether the identifier of the second user is included on the first user list.

...

...

...

...

...

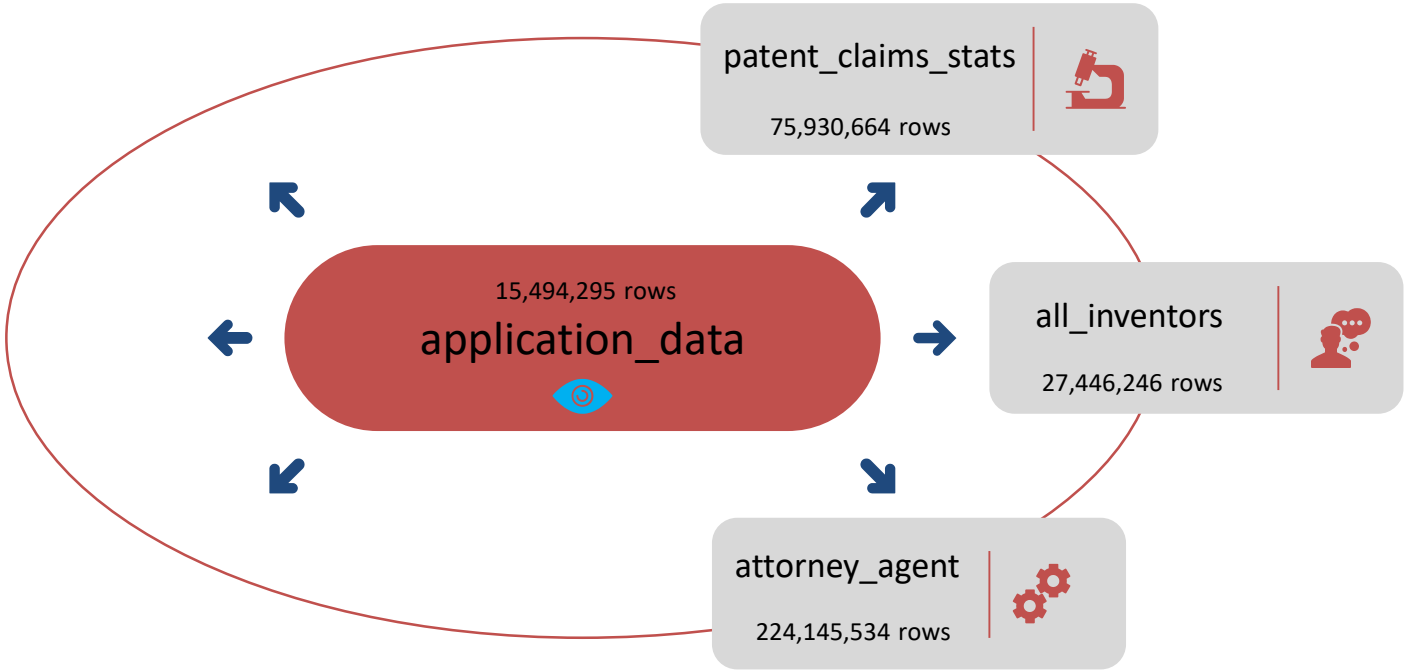
...

...

9. A system comprising:

https://patents.google.com/patent/USRE45254

.csv used in this project – Raw Data



Data Source

https://www.uspto.gov/learning-and-resources/electronic-data-products/patent-examination-research-dataset-public-pair
https://www.uspto.gov/learning-and-resources/electronic-data-products/patent-claims-research-dataset

application_data

Study days between patent issuing and filing

patent_issue_date: Latest 2020-04-21 – good data set!

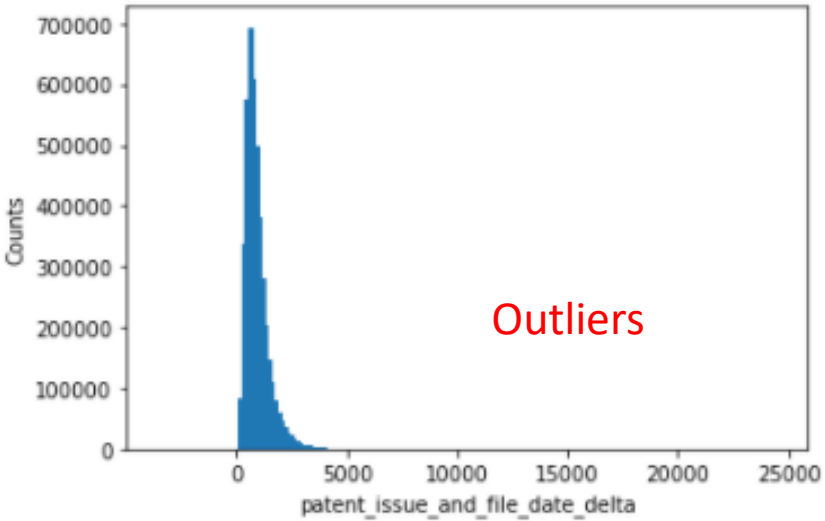
application_cleaned_nona
(4,964,813 rows)

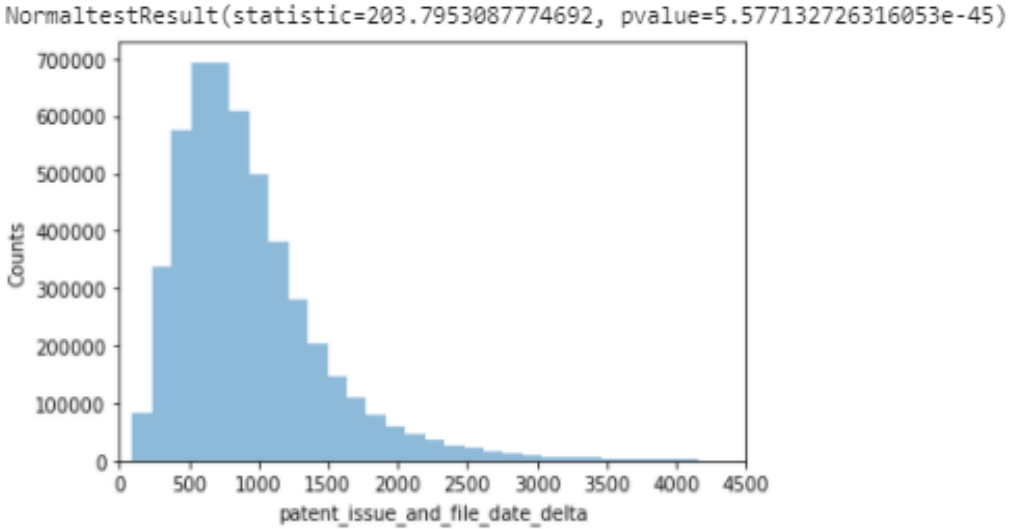
```
(application_cleaned["patent_issue_date"].map(pd.isna) == False)
 & (application_cleaned["filing_date"].map(pd.isna) == False)
 & (application_cleaned["uspc_class"].map(pd.isna) == False)
 & (application_cleaned["appl_status_desc"] == "Patented Case"), :]


ng_date
na['patent_issue_date'], errors = 'coerce') - pd.to_datetime(application_cleaned_nona['filing_date'], errors = 'coerce')
e_delta'] = df_date_ser.dt.days

.loc[(application_cleaned_nona['patent_issue_and_file_date_delta']) > 0, :]
```

"patent_issue_and_file_date_delta"

```
count    4964813.000000
mean         930.683743
std          546.752235
min            1.000000
25%          564.000000
50%          813.000000
75%         1153.000000
max        24479.000000
Name: patent_issue_and_file_date_delta, dtype: object
```

2.55 years





Reject the null hypothesis. Data doesn't seem to be normal. Right skew.

I trust USPTO and my own data analysis

on average how many years to get a patent

All | News | Videos | Maps | Shopping | More | Settings | Tools

About 546,000,000 results (0.75 seconds)

If you're wondering how long it takes to obtain a patent, there's a short answer and a long answer. The short answer for a design patent is between one and **two years** from the filing date. The short answer for a utility patent is between one and **five years** from the filing date. Aug 8, 2018

https://www.bestlawyers.com › article › how-long-does-it-...

How Long Does It Take to Obtain a Patent? | Best Lawyers

About featured snippets • Feedback

People also ask

How long does it take to obtain a patent?

about 25 months

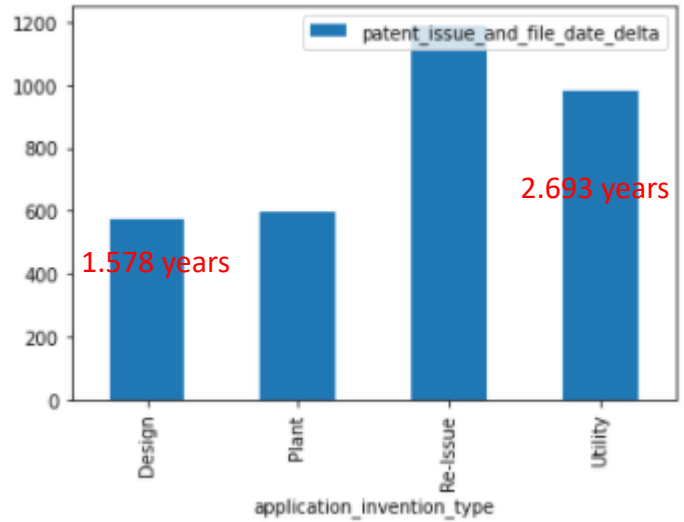
According to the United States Patent and Trademark Office (USPTO), the average time it takes to get a patent is **about 25 months**. If you want to expedite the process you can pay an extra fee ($1000-$4000) to the USPTO to get prioritized examination utility patents you can cut the time down to **6 to 12 months**.

https://milleripl.com › blogs › how-long-does-it-take-to...

How long & how much to get a patent? – Miller IP Law

```
application_cleaned_nona['application_invention_type'].unique()
```

```
array(['Utility', nan, 're-Issue', 'Plant', 'Design'], dtype=object)
```

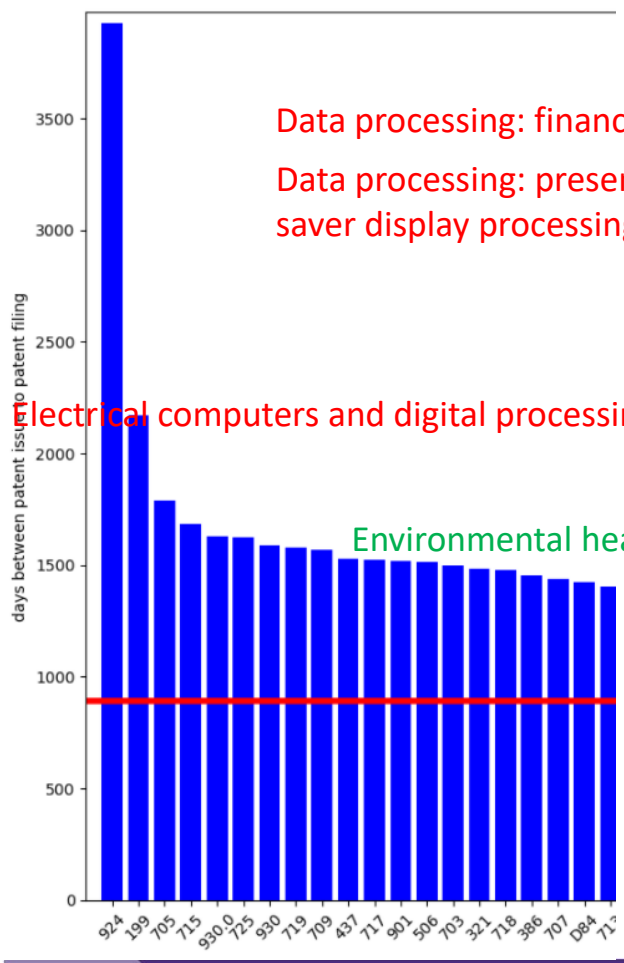


application_data.csv

How about “patent_issue_and_file_date_delta” under each uspc_class?

uspc_class	patented_cases_per_class	total_delta	average_days	average_years
924	1	3927	3927.000000	10.758904
199	2	4343	2171.500000	5.949315
705	30761	55097663	1791.153181	4.907269
715	17129	28831270	1683.184658	4.611465
930.0	2	3259	1629.500000	4.464384
...	...	...	...	...
025	1	504	504.000000	1.380822
814	2	906	453.000000	1.241096
803	1	420	420.000000	1.150685
259	1	335	335.000000	0.917808
520	1	249	249.000000	0.682192

```
count     995.000000
mean      894.148763
std       217.331800
min       249.000000
25%       772.299478
50%       867.779215
75%       989.022585
max      3927.000000
Name: average_days, dtype: float64
```



Type casting

Data processing: financial, business practice, management, or cost/price determination

Data processing: presentation processing of document, operator interface processing, and screen saver display processing

Interactive video distribution systems

Peptide or protein sequence

Electrical computers and digital processing systems: interprogram communication or interprocess communication (ipc)

Environmental heating and cooling; fluid handling and sanitary equipment

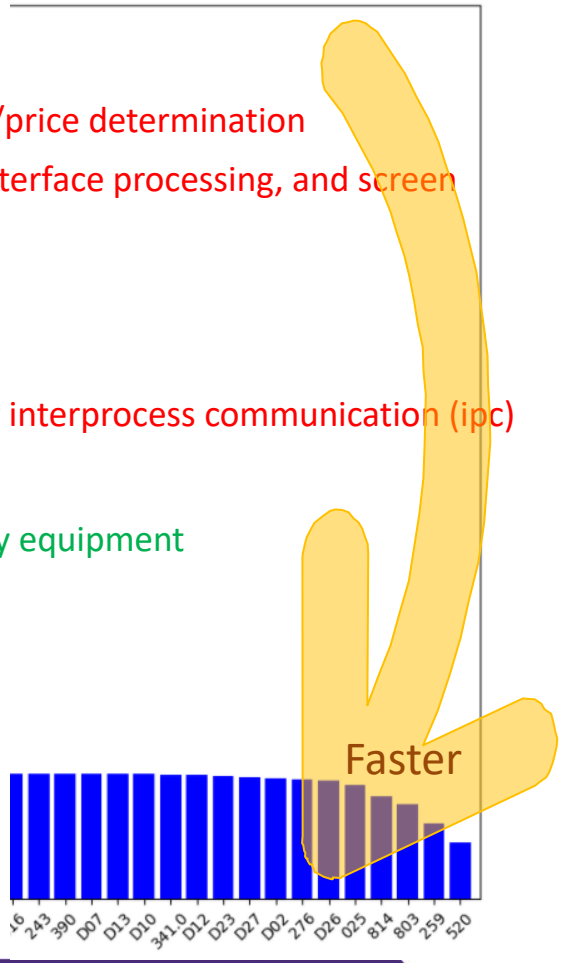
Tobacco and smokers' supplies

Apparel and haberdashery

Typesetting

Lighting

Synthetic resins or natural rubbers



Study patented rate

	uspc_class	application_number_all
0	0	57317
1	1	9496
2	2	6217
3	4	3596
4	5	4638
...	...	...
1010	D96	1
1011	D99	2099
1012	ENG	1
1013	PLT	28001
1014	XXX	46

	uspc_class	application_number_patented
0	2	1858
1	4	1628
2	5	1897
3	7	79
4	8	816
...	...	...
993	D96	1
994	D99	2017
995	ENG	1
996	PLT	26454
997	XXX	36

	uspc_class	application_number_all	application_number_patented	patented_rate
773	390	2	2	100.0000%
426	903	1	1	100.0000%
935	808	2	2	100.0000%

```
.groupby('uspc_class').sum()
.agg({"patented_cases_per_class": "sum"})
```

Lessons and learn:

- ***Use small dataset to test on.***
- ***Manually calculate on one uspc_class and compare.***
- ***Two incoming DataFrames are from "application_cleaned ", only filtered by different criteria, so data is accurate.***

```
uspc_class.sort_values(["patented_rate"], ascending = False)
```

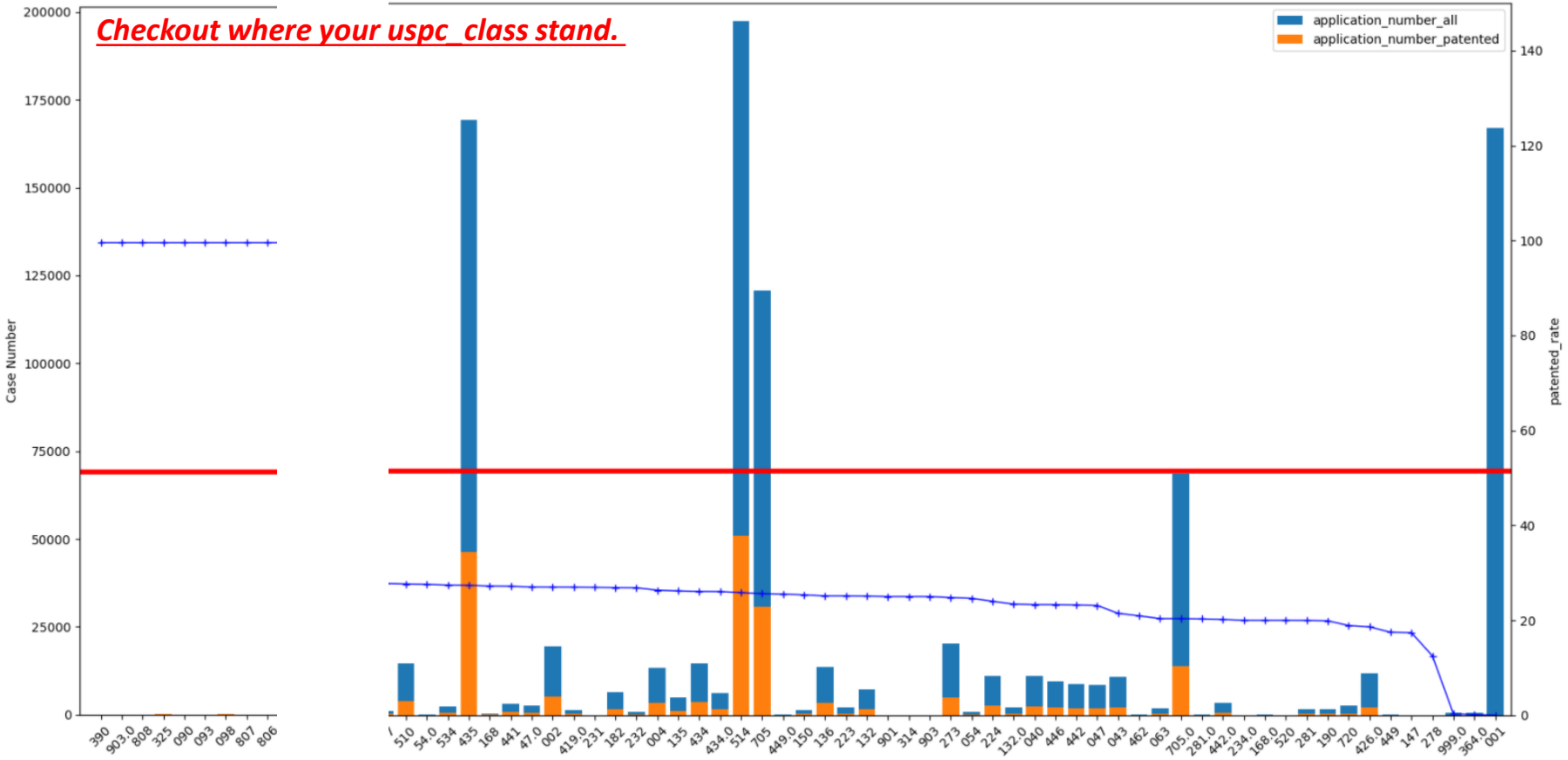
application_number_all

patented_rate

application_number_all stacked with application_number_patented

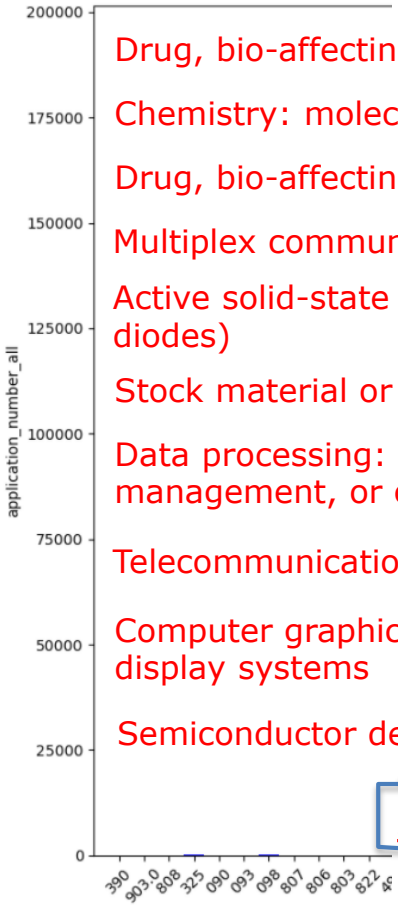
combined

Checkout where your uspc_class stand.



application_data.csv

total application numbers by uspc_class



Drug, bio-affecting and body treating compositions

Chemistry: molecular biology and microbiology

Drug, bio-affecting and body treating compositions

Multiplex communications

Active solid-state devices (e.g., transistors, solid-state diodes)

Stock material or miscellaneous articles

Data processing: financial, business practice, management, or cost/price determination → *This one also takes long time as we just reviewed!*

Telecommunications

Computer graphics processing and selective visual display systems

Semiconductor device manufacturing: process

MOST

Needle and pin making

Hybrid electric vehicles (hevs)

Coopering

Whips and whip apparatus

Type casting

LEAST

application_data.csv

```
.sort_values(["application_number_all", "patented_rate"], ascending = [False, False])
```

	uspc_class	application_number_all	application_number_patented	patented_rate
863	514	196500	50752	25.8280
815	435	168540	46133	27.3721
804	424	168511	52691	31.2686
431	001	166150	7	0.0042
755	370	150331	78732	52.3724
656	257	146513	79510	54.2682
808	428	126593	46650	36.8504
909	705	120092	30762	25.6154
833	455	111943	56758	50.7026
730	345	109589	51559	47.0476
818	438	103545	54882	53.0030
913	709	98728	46468	47.0667
911	707	81713	40059	49.0240
266	370.0	81640	46832	57.3640
185	257.0	80148	52064	64.9598
968	D14	75143	72799	96.8806
726	340	74416	32703	43.9462
767	382	72508	37231	51.3474
897	600	72074	28795	39.9520
310	424.0	71836	22740	31.6554
733	348	69532	33945	48.8192
363	514.0	68348	23710	34.6901
405	705.0	68299	13899	20.3502
497	073	68151	30294	44.4513
760	375	68085	35239	51.7574

Drug, bio-affecting and body treating compositions

Chemistry: molecular biology and microbiology

Multiplex communications

Active solid-state devices (e.g., transistors, solid-state diodes)

Telecommunications

Semiconductor device manufacturing: process

Multiplex communications

Active solid-state devices (e.g., transistors, solid-state diodes)

Recording, communication, or information retrieval equipment

Image analysis

Pulse or digital communications

My Interests

D25	Building units and construction elements
D12	Transportation
074	Machine element or mechanism
052	Static structures (e.g., buildings)
405	Hydraulic and earth engineering

My Interests

	uspc_class	application_number_all	application_number_patented	patented_rate	
979	D25	12023	11668	97.0473	Building units and construction elements
966	D12	43858	42512	96.9310	Transportation
498	074	25947	10678	41.1531	Machine element or mechanism
785	405	14319	5401	37.7191	Hydraulic and earth engineering
477	052	42476	14532	34.2123	Static structures (e.g., buildings)

```
# Calculate Independent (Two Sample) t-test
sts.ttest_ind(D25, D12, equal_var=False)
```

```
Ttest_indResult(statistic=10.96380215526102, pvalue=7.027939464538541e-28)
```

D25 and D12 have close patented rate,
check on their property: patent_issue_and_file_date_delta with Independent t-tests

application_data.csv

One Step Further:

Make an App (SQL, PowerApp)

Are you applying for patent? Take your risk!

Under interested class:

Would you afford the time cost?
Is your case likely to be patented?

Hamlet App

Merge filing period DataFrame with patented rate DataFrame

	uspc_class	application_number_all	application_number_patented	patented_rate	patented_cases_per_class	average_years
186	D25	12023	11668	97.0473	11668	1.594788
44	D12	43858	42512	96.9310	42508	1.506665
93	074	25947	10678	41.1531	10674	2.514654
159	405	14319	5401	37.7191	5400	2.232857
45	052	42476	14532	34.2123	14522	2.513783

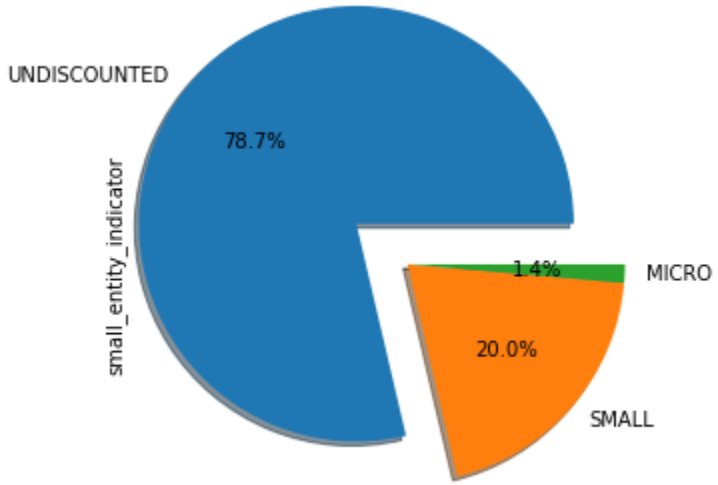
Are you a small entity applying for patent?

Small entity

- typically a nonprofit organization or a company with fewer than 500 employees
- entitles the applicant to a 50 percent discount on most fee payments to USPTO.

micro entity

- Qualify as a USPTO-defined small entity,
- Not be named on more than four previously filed applications,
- Not have a gross income more than three times the median household income from the previous year
- Not be under any obligation to assign, grant, or convey a license or other ownership to another entity that does not meet the micro entity requirements above entitles the applicant to a 75 percent discount



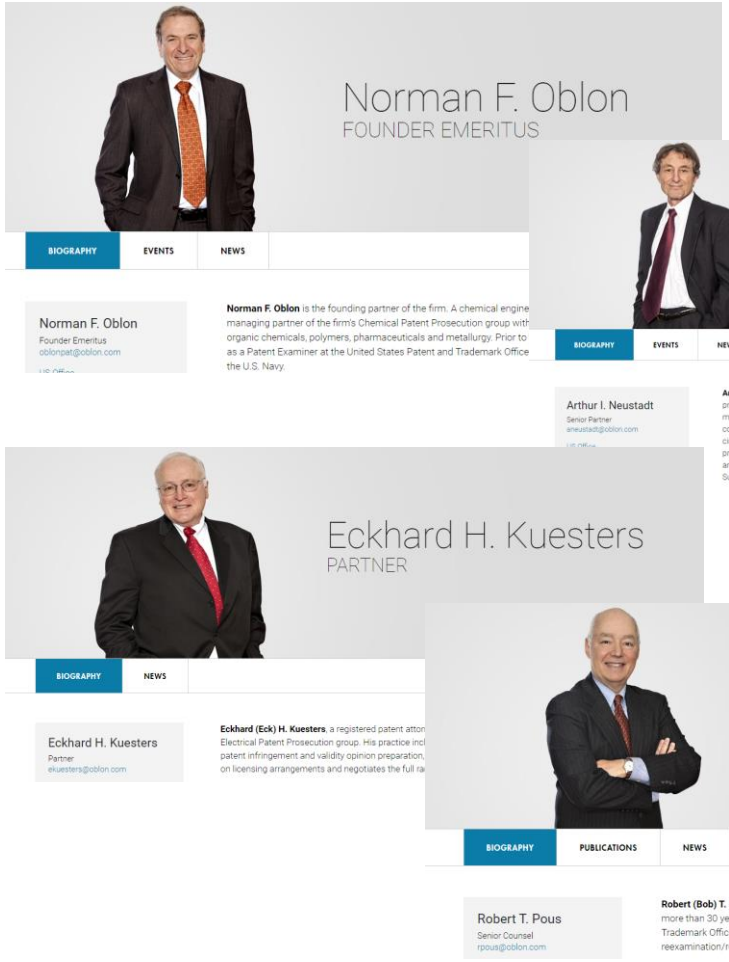
https://www.uspto.gov/learning-and-resources/fees-and-payment/uspto-fee-schedule

attorney_agent

Meet the top richest attorneys

atty_registration_number	application_number	atty_name_last	atty_name_first	atty_phone_number	atty_practice_category
24618	172959	172959	172959	172959	172959
24854	166309	166309	166309	166309	166309
28870	161830	161830	161830	161830	161830
29099	160826	160826	160826	160826	160826
26395	160687	160687	160687	160687	160687
53868	157286	157286	157286	157286	157286
34423	155165	155165	155165	155165	155165
40294	154471	154471	154471	154471	154471
43922	154159	154159	154159	154159	154159
35126	154006	154006	154006	154006	154006

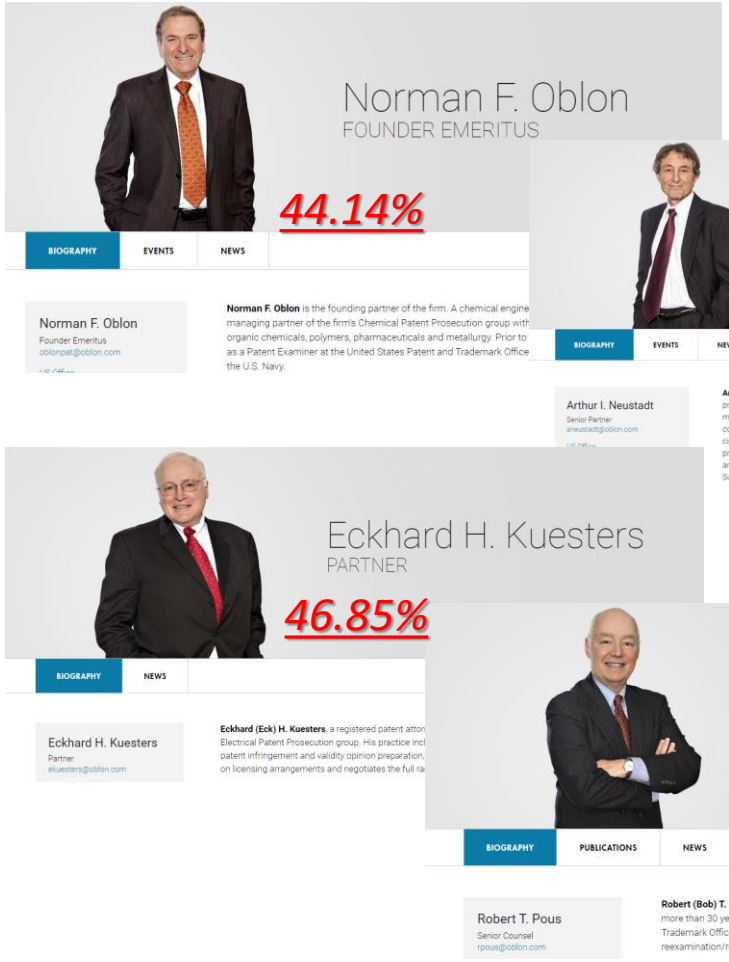
```
.groupby(['atty_registration_number']).count().sort_values(['application_number'], ascending = False).head(10)
```



Find any common features of the top richest attorneys across all USPTO ??

How about their successful rate?

Picture source-https://www.oblon.com/



Find any common features of the top richest attorneys across all USPTO ??

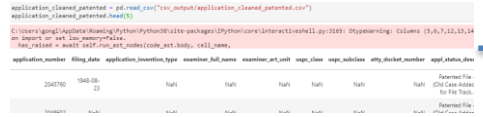
How about their successful rate?

Picture source-https://www.oblon.com/

How to search the attorney's success rate?

	atty_registration_number	application_number_patented	application_number_all	success_rate	atty_name_last	atty_name_first	atty_practice_category
23733	26395	16830	35669	47.1838	Gholz	Charles	Attorney
23851	29099	16773	35659	47.0372	Pous	Robert	Attorney
24000	28870	16832	35928	46.8493	Kuesters	Eckhard	Attorney
24034	24854	19322	41287	46.7992	Neustadt	Arthur	Attorney
26428	24618	20700	46898	44.1383	Oblon	Norman	Attorney

.csv generated from application_data.csv

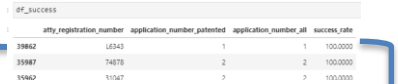


on=["application_number"]

attorney_agent.csv

```
df_patented_att.head()
```

	atty_registration_number	application_number_patented
0	24618	20700
1	24854	10222



on=["application_number"]

.csv generated from application_data.csv

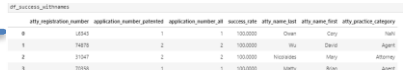
```
application_cleaned = pd.read_csv("csv_output/application_cleaned.csv")
application_cleaned.head(5)
```

```
C:\Users\gongl\AppData\Roaming\Python\Python38\site-packages\IPython\core\interactiveshell.py:31
s.Specify dtype option on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

application_number	filing_date	application_invention_type	examiner_full_name	examiner_art_unit	uspc_class	usp
2000161	NaN	Utility	NaN	NaN	NaN	
2022046	NaN	Utility	NaN	NaN	NaN	

```
df_all_att.head()
```

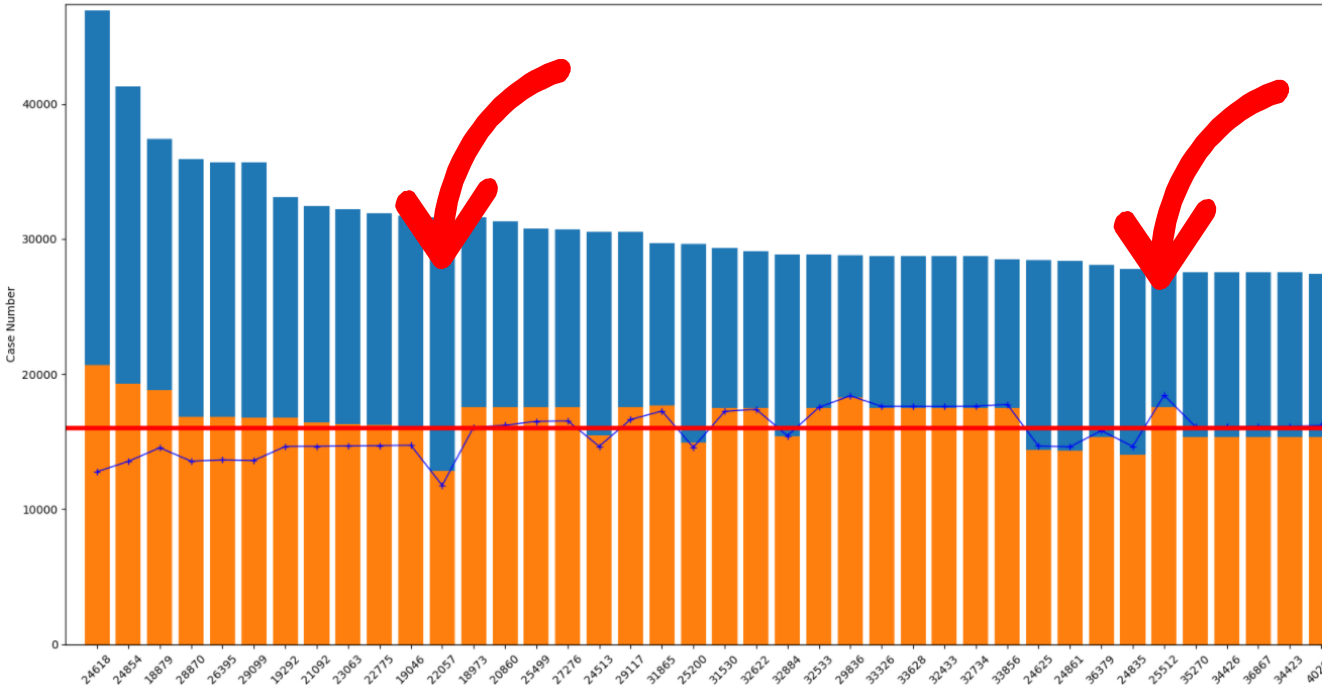
	atty_registration_number	application_number_all
0	24618	46898
1	24854	41287

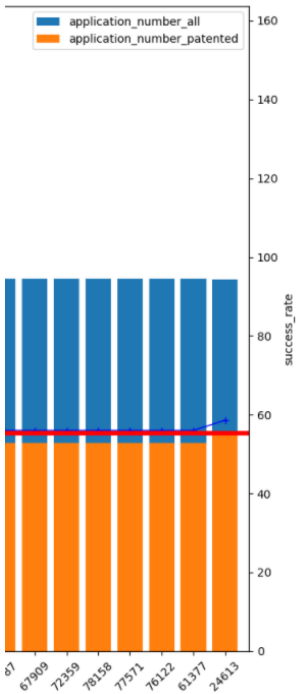


Road map to DataFrame: df_success_withnames

Which attorney would you want to pay to and deal with next few years?

Check out the success_rate for Top 100 attorneys who files most cases





One Step Further:

Make an App

Are you looking for patent attorney? Take your risk!

Under interested class, which attorney has highest success_rate?

df_patented

```
df_patented_groupbyclassandatt_focus.sort_values(['application_number_patented'], ascending = False).head(10)
```

	uspc_class	atty_registration_number	application_number_patented
1855915	430	18879	1763
1855945	430	19046	1548
1855995	430	19292	1510

df_all

```
df_all_groupbyclassandatt_focus.sort_values(['application_number_all'], ascending = False).head(10)
```

	uspc_class	atty_registration_number	application_number_all
2799353	430	18879	2842
2801564	430	27678	2566
2180135	358	18973	2558

on=['uspc_class', 'atty_registration_number']

df_groupbyclassandatt_focus

	uspc_class	atty_registration_number	application_number_patented	application_number_all	success_rate
0	2	24726	1	1	1.0
1	2	25124	1	2	0.5
2	2	25726	1	2	0.5
3	2	26510	1	1	1.0

Attorney info

df_matching_drop_duplicates

	atty_registration_number	atty_name_last	atty_name_first	atty_practice_category
0	17686	Wetzel	James	Attorney
1	28139	Manzo	Edward	Attorney
2	24435	Samuel	Richard	Attorney

df_groupbyclassandatt_focus_withnames

	uspc_class	atty_registration_number	application_number_patented	application_number_all	success_rate	atty_name_last	atty_name_first	atty_practice_category
0	2	24726	1	1	1.0	Greive	Edward	Attorney
1	336	24726	2	2	1.0	Greive	Edward	Attorney
2	442	24726	1	1	1.0	Greive	Edward	Attorney
3	522	24726	1	1	1.0	Greive	Edward	Attorney
4	704	24726	1	1	1.0	Greive	Edward	Attorney

HesJustNotThatIntoYou App

your_interest = "074", Machine element or mechanism

	uspc_class	atty_registration_number	application_number_patented	application_number_all	success_rate	atty_name_last	atty_name_first	atty_pract
266775	074	24618	149	288	0.517361	Oblon	Norman	
1769908	074	22115	163	266	0.612782	Schwartz	Arthur	
1770940	074	25479	153	261	0.586207	Schwaab	Richard	
265350	074	18879	108	253	0.426877	Mion	John	
267259	074	24854	134	252	0.531746	Neustadt	Arthur	
...	...	...	...	...	...	...	...	
471469	074	60045	1	1	1.000000	Punia	Surinder	Agent
471628	074	61723	1	1	1.000000	Stoop	Johan	Agent
471788	074	62992	1	1	1.000000	Bunn	Andrew	Agent
471949	074	63584	1	1	1.000000	Xu	Simon	Agent
773142	074	59856	1	1	1.000000	Tullis	Terry	Attorney

8118 rows × 8 columns

Which attorney would you want?

https://www.oblon.com/norman-f-oblon
http://laubscherlaw.com/our-firm/arthur-schwartz.html

all_inventors

Which world regions apply most to USPTO based on inventors?

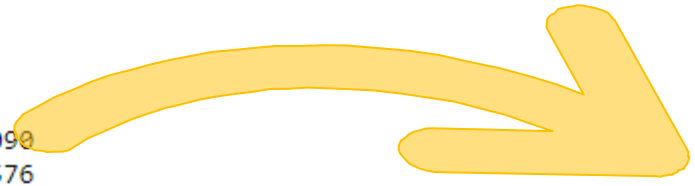
```
US    5890153
JP    1750736
DE     678604
KR     412890
TW     367756
GB     320880
CA     309812
FR     273009
CN     257047
IL     127099
CH     124187
IT     120554
NL     116948
SE     104032
IN      95258
AU      82293
BE      62876
FI      52145
AT      47127
DK      44379
```

	application_number	inventor_country_code	inventor_name_first	inventor_name_last	inventor_rank	inventor_city_name	inventor_region_code
0	4839798	US	4	4	4	4	4
1	4840815	US	1	1	1	1	1
2	5057868	DE	3	3	3	3	0
3	5154777	US	1	1	1	1	1
4	5163565	US	1	1	1	1	1
...	...	...	...	...	...	...	...
11573874	PCT/US19/42742	US	4	4	4	4	4
11573875	PCT/US19/42745	US	4	4	4	4	4
11573876	PCT/US19/45039	IL	3	3	3	3	0
11573877	PCT/US19/47795	IL	2	2	2	2	0
11573878	PCT/US19/47797	IL	4	4	4	4	0

11573879 rows × 7 columns

World Region Rank: US, Japan, Germnay, Korea, Taiwan, UK, Canada, France, China

Which US state apply most to USPTO based on inventors?



CA	1537090
NY	468576
TX	453458
MA	378110
NJ	317822
PA	288727
IL	282953
MI	262581
OH	248099
WA	232904
FL	220369
MN	200573
NC	163912
CT	158718
CO	143053
MD	131327
WI	130193
GA	123867
OR	120393
TN	119685

San Diego	268479
San Jose	251622
San Francisco	215371
Sunnyvale	116523
Palo Alto	96115
Mountain View	89531
SAN DIEGO	82261
Fremont	75752

	application_number	inventor_region_code	inventor_name_first	inventor_name_last	inventor_rank	inventor_city_name	inventor_country_code
0	4839798	PA	4	4	4	4	4
1	4840815	CA	1	1	1	1	1
2	5154777	CT	1	1	1	1	1
3	5163565	VA	1	1	1	1	1
4	5420999	VA	1	1	1	1	1
...	...	...	...	...	...	...	...
7077433	PCT/US19/34153	MS	2	2	2	2	2
7077434	PCT/US19/41346	TX	5	5	5	5	5
7077435	PCT/US19/41358	TX	5	5	5	5	5
7077436	PCT/US19/42742	NY	4	4	4	4	4
7077437	PCT/US19/42745	NY	4	4	4	4	4

7077438 rows × 7 columns

State Rank: CA, NY, TX, MA, NJ, PA, IL

API: from config import (census_key, gkey)

	State	Name	Population	Median Age	Household Income	Per Capita Income	Poverty Count	Poverty Rate	Unemployment Rate	application_per_state
0	01	Alabama	4876250.0	39.0	50536.0	27928.0	795989.0	16.323794	2.708946	30196
1	02	Alaska	737068.0	34.3	77640.0	36787.0	76933.0	10.437707	3.637114	3285

	Name	application_per_state
0	California	1537090
1	New York	468576

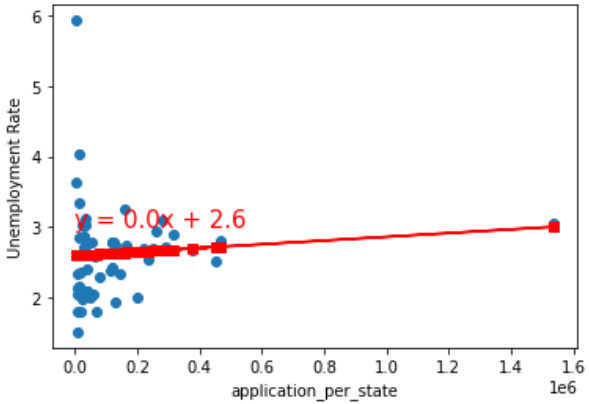
	Name	Latitude	Longitude
0	Alabama	32.7794	-86.8287
1	Alaska	64.0685	-152.2782

	State	Name	Population	Median Age	Household Income	Per Capita Income	Poverty Count	Poverty Rate	Unemployment Rate	application_per_state
0	01	Alabama	4876250.0	39.0	50536.0	27928.0	795989.0	16.323794	2.708946	30196
1	02	Alaska	737068.0	34.3	77640.0	36787.0	76933.0	10.437707	3.637114	3285

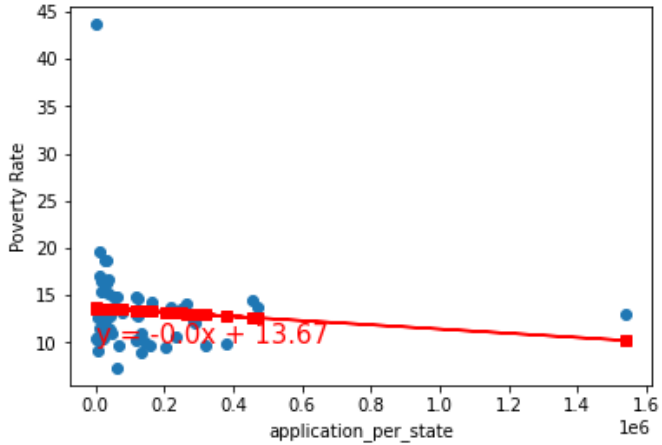
	State	Name	Population	Median Age	Household Income	Per Capita Income	Poverty Count	Poverty Rate	Unemployment Rate	application_per_state	Latitude	Longitude
0	01	Alabama	4876250.0	39.0	50536.0	27928.0	795989.0	16.323794	2.708946	30196	32.7794	-86.8287
1	02	Alaska	737068.0	34.3	77640.0	36787.0	76933.0	10.437707	3.637114	3285	64.0685	-152.2782

Correlation Checks

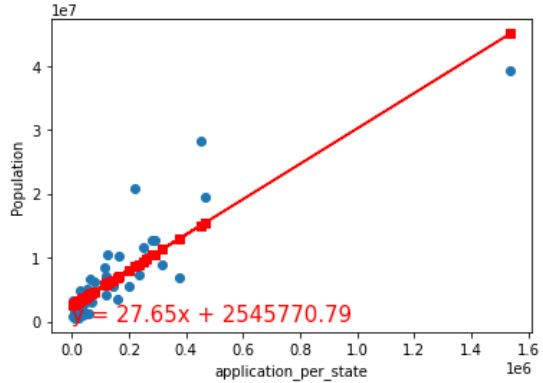
The r-squared is: 0.008113933112090124



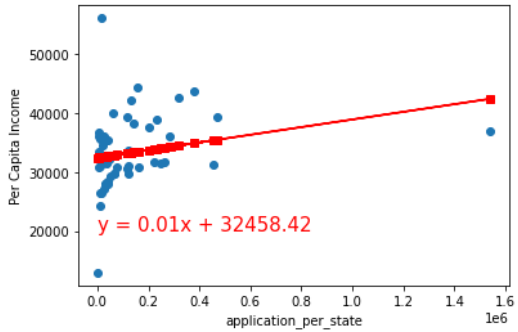
The r-squared is: 0.010296617369985257



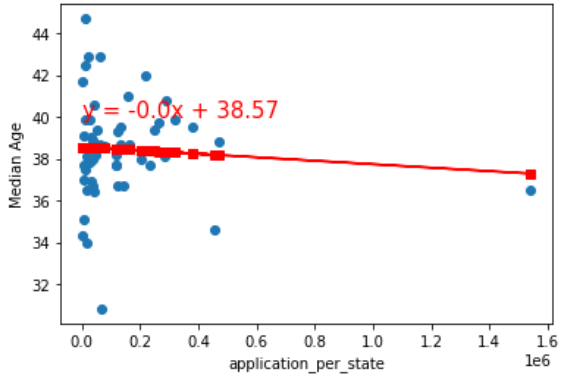
The r-squared is: 0.7830605231749171



The r-squared is: 0.05619503189259235 4



The r-squared is: 0.0062013326435580935



Heat Map for Inventors per State

Want to be surrounded by inventors?

Want to make more $ filing patents?

Move to California!

patent_claims_stats

application_data.csv

application_cleaned_patented.csv

on=["patent_number"]

patent_claims_stats.csv

patented_df

	patent_number	uspc_class	appl_status_desc	claim_no	word_ct	char_ct
0	RE30349	220	Patented Case	4	214	1364
1	RE30349	220	Patented Case	2	43	250
2	RE30349	220	Patented Case	5	24	145
3	RE30349	220	Patented Case	1	268	1873
4	RE30349	220	Patented Case	3	46	298

groupby(['patent_number'])

```
p.count()
```

patent_number	uspc_class	appl_status_desc	claim_no	word_ct	char_ct
8899829.0	9	9	9	9	9
8907138.0	7	7	7	7	7
8914910.0	16	16	16	16	16
8918269.0	10	10	10	10	10
8922265.0	1	1	1	1	1
...	...	...	...	...	...
RE45249	33	33	33	33	33
RE45250	18	18	18	18	18
RE45251	37	37	37	37	37
RE45252	57	57	57	57	57
RE45254	33	33	33	33	33

p_word_ct = p.agg({"word_ct": "sum"})

p_char_ct = p.agg({"char_ct": "sum"})

claims_summary_df

Road map to DataFrame: claims_summary_df

claims_summary_df

For each distinct patent number:

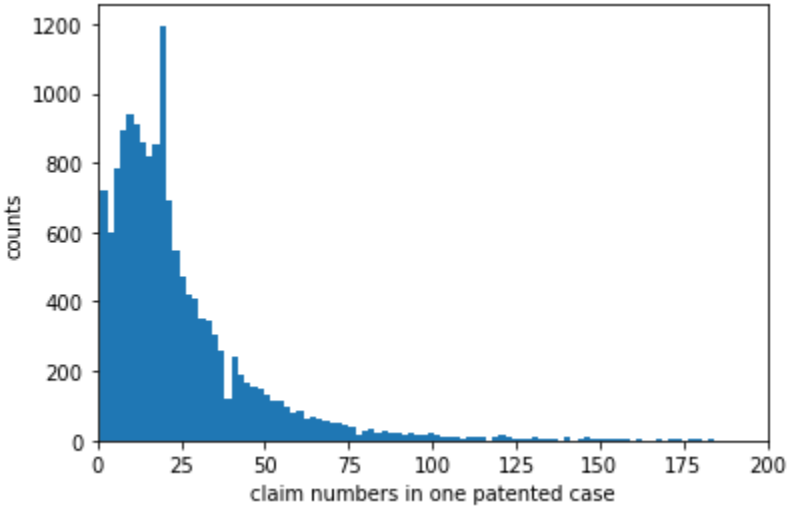
patent_number	claim_no	total_word_ct	total_char_ct	average_word_ct_eachclaim	average_char_ct_eachclaim
8899829.0	9	378	2469	42	274
8907138.0	7	167	1102	24	157
8914910.0	16	1450	8901	91	556
8918269.0	10	1790	11805	179	1180
8922265.0	1	850	4503	850	4503
...	...	...	...	...	...
RE45249	33	1665	10245	50	310
RE45250	18	1138	7417	63	412
RE45251	37	2154	13356	58	361
RE45252	57	4597	29228	81	513
RE45254	33	2825	17752	86	538

14945 rows × 5 columns

How many claims in one patent?

```
claims_summary_df['claim_no'].describe()
```

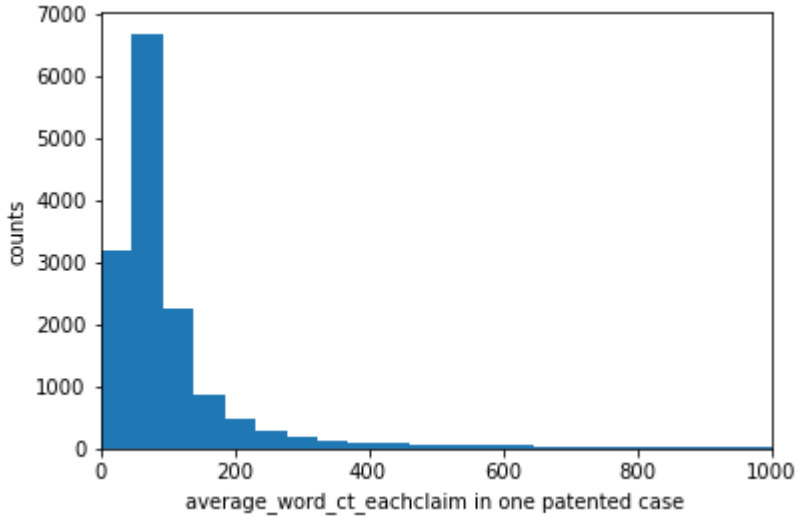
```
count    14945.000000
mean        24.608230
std         24.231829
min          1.000000
25%         10.000000
50%         19.000000
75%         31.000000
max        391.000000
Name: claim_no, dtype: float64
```



How words in one claim?

```
claims_summary_df['average_word_ct_eachclaim'].describe()
```

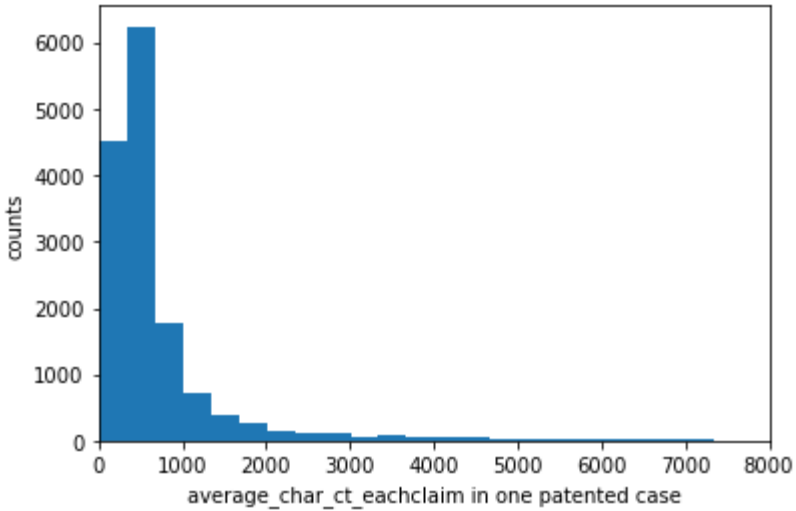
```
count    14945.000000
mean       135.302844
std        298.790267
min          0.000000
25%         49.000000
50%         70.000000
75%        114.000000
max       9244.000000
Name: average_word_ct_eachclaim, dtype: float64
```



How many characters in one claim?

```
claims_summary_df['average_char_ct_eachclaim'].describe()
```

```
count    14945.000000
mean       872.580796
std       1972.672801
min          0.000000
25%        309.000000
50%        444.000000
75%        726.000000
max      66801.000000
Name: average_char_ct_eachclaim, dtype: float64
```



What is the examiner's comfort zone?

Golden Rule

25 claims in one patent

135 words in one claim

873 characters in one claim

Summary

- *Would you afford the time cost?*

  Check out Hamlet App on Page 20

- *Is your case likely to be patented?*

  Check out Hamlet App on Page 20

- *Which attorney are you talking to?*

  Check out HesJustNotThatIntoYou App on Page 31

- *Would you want to live around other inventors?*

  If yes, move to CA, Ivy Leauge, RTP areas

- *Do you know the examiner's comfort zone?*

  25, 135, 837 rules

Copyright and Trademark Notice

Ready to buy Hamlet App and HesJustNotThatIntoYou App?

Questions?

Thank You