

Workforce Diversity Private Firms

November 22, 2025

```
[1]: import pandas as pd
import swifter
pd.set_option('display.max_columns', 500)
pd.set_option('display.max_rows', 500)
```

1 Read files

```
[2]: df_position = pd.read_pickle("Processed_Data_PrivateFirms/df_positions.pkl")
```

```
[3]: df_position.head()
```

```
[3]:
```

	user_id	position_id	\
0	user_id_434913968	position_id_-6119937108085565443	
1	user_id_437023895	position_id_-2560430386434105047	
2	user_id_566408117	position_id_-584339158170326001	
3	user_id_864111882	position_id_-286706079450774703	
4	user_id_114293100	position_id_1072074487094150251	

	company_cleaned	region	country	\
0	food vending	Northern America	United States	
1	goodwill north georgia	Northern America	United States	
2	mario sinacola and sons excavating inc	Northern America	United States	
3	sinacola	empty	empty	
4	empire cat	Northern America	United States	

	state	msa	startdate	\
0	NY	New York-Northern New Jersey-Long Island NY-NJ...	2017-02-01	
1	GA	empty	1995-09-01	
2	TX	Dallas-Fort Worth-Arlington TX MSA	2008-01-01	
3	empty	empty	2016-03-01	
4	AZ	Phoenix-Mesa-Scottsdale AZ MSA	2012-06-01	

	enddate	jobtitle_raw	salary	seniority	rn	\
0	2020-03-01	Self Employed	16204.951	2	5	
1	NaN	Retired	61663.747	5	1.0	
2	2016-05-01	Accounts Payable Associate	33135.598	1	2.0	
3	2017-07-01	Refrigeration Engineer	36952.229	2	3.0	

```
4         NaN   General Maneger Empire Transport   86024.104           5   1.0
```

```
      rcid   onet_code factset_entity_id
0  rcid_100855  41-9022.00          07X448-E
1  rcid_383520  39-4011.00          07P7V4-E
2    rcid_6755  43-3031.00          07PR9S-E
3    rcid_6755  17-2071.00          07PR9S-E
4   rcid_31841  53-3033.00          05KXXK-E
```

```
[4]: private_firm_match_table = df_position[["rcid", "factset_entity_id"]].
      ↪drop_duplicates("rcid")
      private_firm_match_table.rename(columns={'factset_entity_id': 'the_key'},
      ↪inplace=True)
```

```
[5]: private_firm_match_table.to_pickle("Processed_Data/private_firm_match_table.
      ↪pk1")
```

```
[6]: df_position_US = df_position[df_position["country"] == "United States"]
```

```
[7]: df_position_US.head()
```

```
[7]:      user_id      position_id \
0  user_id_434913968  position_id_-6119937108085565443
1  user_id_437023895  position_id_-2560430386434105047
2  user_id_566408117  position_id_-584339158170326001
4  user_id_114293100  position_id_1072074487094150251
5  user_id_116548952  position_id_3431516321617879048
```

```
      company_cleaned      region      country \
0      food vending  Northern America  United States
1      goodwill north georgia  Northern America  United States
2  mario sinacola and sons excavating inc  Northern America  United States
4      empire cat  Northern America  United States
5      empire cat  Northern America  United States
```

```
      state      msa      startdate \
0  NY  New York-Northern New Jersey-Long Island NY-NJ...  2017-02-01
1  GA      empty  1995-09-01
2  TX      Dallas-Fort Worth-Arlington TX MSA  2008-01-01
4  AZ      Phoenix-Mesa-Scottsdale AZ MSA  2012-06-01
5  AZ      Phoenix-Mesa-Scottsdale AZ MSA  2017-12-01
```

```
      enddate      jobtitle_raw      salary seniority \
0  2020-03-01  Self Employed  16204.951      2
1      NaN      Retired  61663.747      5
2  2016-05-01  Accounts Payable Associate  33135.598      1
4      NaN  General Maneger Empire Transport  86024.104      5
```

	rn	rcid	onet_code	factset_entity_id
0	5	rcid_100855	41-9022.00	07X448-E
1	1.0	rcid_383520	39-4011.00	07P7V4-E
2	2.0	rcid_6755	43-3031.00	07PR9S-E
4	1.0	rcid_31841	53-3033.00	05KXXK-E
5	7.0	rcid_31841	17-2112.00	05KXXK-E

```
[8]: df_position_US = df_position_US[["user_id", "position_id", "state", "msa",
    ↪ "seniority", "factset_entity_id", "startdate", "enddate", "jobtitle_raw"]]
```

```
[9]: df_position_US.head()
```

```
[9]:
```

	user_id	position_id	state	\
0	user_id_434913968	position_id_-6119937108085565443	NY	
1	user_id_437023895	position_id_-2560430386434105047	GA	
2	user_id_566408117	position_id_-584339158170326001	TX	
4	user_id_114293100	position_id_1072074487094150251	AZ	
5	user_id_116548952	position_id_3431516321617879048	AZ	

	msa	seniority	\
0	New York-Northern New Jersey-Long Island NY-NJ...	2	
1	empty	5	
2	Dallas-Fort Worth-Arlington TX MSA	1	
4	Phoenix-Mesa-Scottsdale AZ MSA	5	
5	Phoenix-Mesa-Scottsdale AZ MSA	3	

	factset_entity_id	startdate	enddate	\
0	07X448-E	2017-02-01	2020-03-01	
1	07P7V4-E	1995-09-01	NaN	
2	07PR9S-E	2008-01-01	2016-05-01	
4	05KXXK-E	2012-06-01	NaN	
5	05KXXK-E	2017-12-01	2019-06-01	

	jobtitle_raw
0	Self Employed
1	Retired
2	Accounts Payable Associate
4	General Maneger Empire Transport
5	Continuous Improvement Project Manager

```
[23]: # Removing time restriction to allow unmatched private firms considered
df_position_US = df_position_US.sort_values(["user_id", "startdate"])
df_position_US['factset_entity_id_lag'] = df_position_US.
    ↪groupby('user_id')['factset_entity_id'].shift(1)
```

```
df_position_US['enddate_lag'] = df_position_US.groupby('user_id')['enddate'].
    ↪shift(1)
df_position_US['d_poaching'] = (
    (df_position_US['factset_entity_id'] !=
    ↪df_position_US['factset_entity_id_lag']) &
    (df_position_US['factset_entity_id_lag'].notna())
).astype(int)
```

```
[26]: df_position_US = df_position_US[["user_id", "position_id", "state", "msa",
    ↪"seniority", "startdate", "enddate", "factset_entity_id", "d_poaching"]]
```

```
[27]: df_users = pd.read_pickle("Processed_Data_PrivateFirms/df_users_clean.pkl")
```

```
[29]: df_users = df_users[["user_id", "sex_predicted", "ethnicity_predicted"]]
```

```
[30]: df_users.head()
```

```
[30]:
```

	user_id	sex_predicted	ethnicity_predicted
538	user_id_100354593	Male	White
907	user_id_100574741	Female	White
1415	user_id_100912887	Female	White
2164	user_id_101377091	Male	White
2277	user_id_101435107	Male	White

2 Merge everything

```
[31]: df_sum = df_position_US.merge(df_users, on = "user_id", how = "left")
```

```
[33]: df_sum.to_pickle("Processed_Data_PrivateFirms/df_level_demographics.pkl")
```

3 Create aggregated measure

```
[34]: import pandas as pd
import swifter
pd.set_option('display.max_columns', 500)
pd.set_option('display.max_rows', 500)
import os
os.chdir(r"D:\User.Data\Dan.Li\Revelio")
```

```
[2]: df_sum = pd.read_pickle("Processed_Data_PrivateFirms/df_level_demographics.pkl")
```

```
[35]: df_sum = df_sum[["seniority", "startdate", "enddate", "factset_entity_id",
    ↪"sex_predicted", "ethnicity_predicted", "d_poaching"]]
```

```
[36]: df_sum.head()
```

```
[36]: seniority    startdate    enddate factset_entity_id sex_predicted \
0         2  1981-01-01  1984-01-01          05LJMQ-E      Female
1         2  2018-08-01         NaN          07Q1J2-E      Female
2         4  2012-06-01  2015-09-01          07DNRW-E      Female
3        2.0  2019-02-01  2020-05-01          002CS2-E        Male
4         2  2009-01-01  2010-01-01          09GS4F-E      Female

ethnicity_predicted  d_poaching
0             White           0
1          Hispanic           0
2             White           0
3          Hispanic           0
4             White           0
```

```
[39]: df_sum["d_male"] = df_sum["sex_predicted"].apply(lambda x:1 if x == "Male" else 0)
df_sum["d_female"] = df_sum["sex_predicted"].apply(lambda x:1 if x == "Female" else 0)
df_sum["d_api"] = df_sum["ethnicity_predicted"].apply(lambda x:1 if x == "API" else 0)
df_sum["d_white"] = df_sum["ethnicity_predicted"].apply(lambda x:1 if x == "White" else 0)
df_sum["d_hispanic"] = df_sum["ethnicity_predicted"].apply(lambda x:1 if x == "Hispanic" else 0)
df_sum["d_black"] = df_sum["ethnicity_predicted"].apply(lambda x:1 if x == "Black" else 0)
```

```
[40]: df_sum['enddate'] = df_sum['enddate'].fillna(pd.to_datetime('2099-12-31'))
```

```
[41]: df_sum['startdate'] = pd.to_datetime(df_sum['startdate']).dt.date
df_sum['enddate'] = pd.to_datetime(df_sum['enddate']).dt.date
```

```
[42]: df_sum = df_sum[df_sum["startdate"].notna()]
```

```
[43]: df_sum.head()
```

```
[43]: seniority    startdate    enddate factset_entity_id sex_predicted \
0         2  1981-01-01  1984-01-01          05LJMQ-E      Female
1         2  2018-08-01  2099-12-31          07Q1J2-E      Female
2         4  2012-06-01  2015-09-01          07DNRW-E      Female
3        2.0  2019-02-01  2020-05-01          002CS2-E        Male
4         2  2009-01-01  2010-01-01          09GS4F-E      Female

ethnicity_predicted  d_poaching  d_male  d_female  d_api  d_white \
0             White           0         0         1         0         1
1          Hispanic           0         0         1         0         0
2             White           0         0         1         0         1
```

3	Hispanic	0	1	0	0	0
4	White	0	0	1	0	1

	d_hispanic	d_black
0	0	0
1	1	0
2	0	0
3	1	0
4	0	0

4 Create panel

```
[44]: id_list = df_sum.drop_duplicates("factset_entity_id")["factset_entity_id"].
      ↪to_list()
```

```
[46]: import pandas as pd
      from itertools import product

      # Generating year-month combinations from 1980-01 to 2022-12
      date_range = pd.date_range(start='1980-01-01', end='2022-12-01', freq='MS')
      year_month_combinations = [(date.year, date.month) for date in date_range]

      # Creating a list of dictionaries containing all combinations of
      ↪factset_entity_id, year, month, and date
      panel_data = []
      for entity_id in id_list:
          for year, month in year_month_combinations:
              date = pd.Timestamp(year=year, month=month, day=1)
              panel_data.append({
                  'factset_entity_id': entity_id,
                  'year': year,
                  'month': month,
                  'date': date
              })

      # Creating a DataFrame from the list of dictionaries
      panel_df = pd.DataFrame(panel_data)

      # Displaying the first few rows of the panel DataFrame
      print(panel_df.head())
```

	factset_entity_id	year	month	date
0	05LJMQ-E	1980	1	1980-01-01
1	05LJMQ-E	1980	2	1980-02-01
2	05LJMQ-E	1980	3	1980-03-01
3	05LJMQ-E	1980	4	1980-04-01

4 05LJMQ-E 1980 5 1980-05-01

```
[47]: panel_df2 = panel_df.merge(df_sum, on = "factset_entity_id")
```

```
[49]: panel_df2 = panel_df2[panel_df2["date"] >= panel_df2["startdate"]]
panel_df2 = panel_df2[panel_df2["date"] <= panel_df2["enddate"] ]
```

```
[50]: panel_df2.head()
```

```
[50]:
```

	factset_entity_id	year	month	date	seniority	startdate	\
335	05LJMQ-E	1980	1	1980-01-01	2.0	1978-01-01	
1169	05LJMQ-E	1980	1	1980-01-01	4.0	1974-01-01	
2107	05LJMQ-E	1980	1	1980-01-01	3	1968-04-01	
2376	05LJMQ-E	1980	1	1980-01-01	6	1979-10-01	
3307	05LJMQ-E	1980	1	1980-01-01	2.0	1980-01-01	

	enddate	sex_predicted	ethnicity_predicted	d_poaching	d_male	\
335	1981-01-01	Male	White	0	1	
1169	2099-12-31	Male	White	0	1	
2107	1991-03-01	Female	White	0	0	
2376	1986-07-01	Female	White	0	0	
3307	1983-05-01	Male	White	0	1	

	d_female	d_api	d_white	d_hispanic	d_black
335	0	0	1	0	0
1169	0	0	1	0	0
2107	1	0	1	0	0
2376	1	0	1	0	0
3307	0	0	1	0	0

```
[52]: panel_df2["d_senior"] = panel_df2["seniority"].apply(lambda x:1 if int(x)>=4
↳ else 0)
```

```
[54]: # Columns for interaction (replace with specific column names)
columns_for_interaction = ['d_male', 'd_female', 'd_api', 'd_white',
↳ 'd_hispanic', 'd_black']

for col in columns_for_interaction:

    panel_df2[col+'_senior'] = panel_df2['d_senior'] * panel_df2[col]
```

```
[56]: panel_df2.to_pickle("Processed_Data_PrivateFirms/df_position_panel.pkl")
```

5 Aggregate

```
[58]: df_aggregated = panel_df2.groupby(["factset_entity_id", "date", "year",  
    ↪ "month"]).sum().reset_index()

[59]: df_aggregated.to_pickle("Processed_Data_PrivateFirms/  
    ↪ df_position_panel_df_aggregated.pkl")

[60]: # Calculate gender_measurable and race_measurable
df_aggregated['gender_measurable'] = df_aggregated['d_male'] +  
    ↪ df_aggregated['d_female']
df_aggregated['race_measurable'] = df_aggregated['d_api'] +  
    ↪ df_aggregated['d_white'] + df_aggregated['d_hispanic'] +  
    ↪ df_aggregated['d_black']

# Calculate ratios
df_aggregated['male_ratio'] = df_aggregated['d_male'] /  
    ↪ df_aggregated['gender_measurable']
df_aggregated['female_ratio'] = df_aggregated['d_female'] /  
    ↪ df_aggregated['gender_measurable']

df_aggregated['api_ratio'] = df_aggregated['d_api'] /  
    ↪ df_aggregated['race_measurable']
df_aggregated['white_ratio'] = df_aggregated['d_white'] /  
    ↪ df_aggregated['race_measurable']
df_aggregated['hispanic_ratio'] = df_aggregated['d_hispanic'] /  
    ↪ df_aggregated['race_measurable']
df_aggregated['black_ratio'] = df_aggregated['d_black'] /  
    ↪ df_aggregated['race_measurable']

# Calculate gender_measurable_senior and race_measurable_senior
df_aggregated['gender_measurable_senior'] = df_aggregated['d_male_senior'] +  
    ↪ df_aggregated['d_female_senior']
df_aggregated['race_measurable_senior'] = df_aggregated['d_api_senior'] +  
    ↪ df_aggregated['d_white_senior'] + df_aggregated['d_hispanic_senior'] +  
    ↪ df_aggregated['d_black_senior']

# Calculate ratios for senior demographic variables
df_aggregated['male_senior_ratio'] = df_aggregated['d_male_senior'] /  
    ↪ df_aggregated['gender_measurable_senior']
df_aggregated['female_senior_ratio'] = df_aggregated['d_female_senior'] /  
    ↪ df_aggregated['gender_measurable_senior']

df_aggregated['api_senior_ratio'] = df_aggregated['d_api_senior'] /  
    ↪ df_aggregated['race_measurable_senior']
```

```
df_aggregated['white_senior_ratio'] = df_aggregated['d_white_senior'] /   
    ↪ df_aggregated['race_measurable_senior']  
df_aggregated['hispanic_senior_ratio'] = df_aggregated['d_hispanic_senior'] /   
    ↪ df_aggregated['race_measurable_senior']  
df_aggregated['black_senior_ratio'] = df_aggregated['d_black_senior'] /   
    ↪ df_aggregated['race_measurable_senior']
```

```
[63]: columns_to_prefix = [  
    'd_male', 'd_female', 'd_api', 'd_white', 'd_hispanic', 'd_black',  
    'gender_measurable', 'race_measurable', 'male_ratio', 'female_ratio',  
    'api_ratio', 'white_ratio', 'hispanic_ratio', 'black_ratio',  
    'd_male_senior', 'd_female_senior', 'd_api_senior', 'd_white_senior',   
    ↪ 'd_hispanic_senior', 'd_black_senior',  
    'gender_measurable_senior', 'race_measurable_senior', 'male_senior_ratio',  
    'female_senior_ratio', 'api_senior_ratio', 'white_senior_ratio',   
    ↪ 'hispanic_senior_ratio', 'black_senior_ratio',  
]  
  
for column in columns_to_prefix:  
    df_aggregated.rename(columns={column: 'l_' + column}, inplace=True)
```

```
[66]: df_aggregated.to_csv("Processed_Data_PrivateFirms/  
    ↪ 20240921df_level_aggre_private.csv")
```

```
[ ]:
```