

DEI Jobs Step 4 - Analysis- JP level3.0 - Public and Private

November 22, 2025

1 Intro

This code create the final sample

```
[1]: import re
import os
import pandas as pd
import numpy as np
from tqdm import tqdm
import swifter
import time
pd.set_option('display.max_colwidth', 100)
pd.set_option('display.max_rows', 2000)
pd.set_option('display.max_columns', 2000)
```

2 Raw Data Processing

```
[2]: df1_private = pd.read_pickle("Raw_Data/20230226 private_JP_chosen_full.pkl")
df1_private["d_public"] = 0

[3]: df1_private = df1_private[df1_private["created"]>="2019-06-30"]

[4]: df1_temp = df1_private.copy()
df1_temp.loc[df1_temp["delete_date"].isna(), "delete_date"] = "2099-12-31 09:20:
↪00+00:00"
df1_temp["delete_date2"] = df1_temp["delete_date"].apply(lambda x:x[0:10])
df1_temp['delete_date2'] = pd.to_datetime(df1_temp['delete_date2'])
df1_temp['created_date2'] = pd.to_datetime(df1_temp['created'])
df1_temp['filling_days'] = df1_temp['delete_date2'] - df1_temp['created_date2']
df1_temp['filling_days'] = df1_temp['filling_days'].dt.days
df1_temp_clean = df1_temp[["hash", "the_key", "delete_date2", "created_date2",
↪"filling_days"]]
df1_temp_clean["filled"] = 1
# Filter "delete_date" = "2099-12-31 09:20:00+00:00" values
df1_temp_clean.loc[df1_temp_clean["filling_days"]>20000, "filled"] = 0
df1_temp_clean.loc[df1_temp_clean["filling_days"]>20000, "filling_days"] = None
```

```
df1_temp_clean.to_pickle("Processed_Data/private_chosen_filling_time.pkl")
```

C:\Users\dxl902\AppData\Local\conda\conda\envs\an_env\lib\site-packages\ipykernel_launcher.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
# Remove the CWD from sys.path while we load stuff.
```

C:\Users\dxl902\AppData\Local\conda\conda\envs\an_env\lib\site-packages\pandas\core\indexing.py:1817: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
self._setitem_single_column(loc, value, pi)
```

```
[5]: df1_private.to_pickle("Processed_Data/private_JP_chosen_final.pkl")
```

```
[6]: df1_public = pd.read_csv("Processed_Data/jp_dimensions_medium_simple_new.csv")  
df1_public["d_public"] = 1
```

```
[7]: df1 = df1_public.append(df1_private)
```

3 Merge RavenPack Data

```
[8]: df_rplinkup = pd.read_pickle("pd_sum_clean.pkl")
```

```
[9]: df_rplinkup = df_rplinkup.reset_index()
```

```
[10]: df2_rp = df1.merge(df_rplinkup, left_on = ["hash"], right_on =  
↳ ["PROVIDER_DOCUMENT_CHAIN_ID"], how = "inner")
```

```
[11]: df2_rp2 = df2_rp.rename(columns={"Diversity, Equity and Inclusion": "dei",  
↳ "Equal Employment Opportunity": "eeo"})
```

```
[12]: df2_rp2.to_pickle("Processed_Data/jp_rp_sum_new.pkl")
```

4 Main

```
[1]: import re  
import os  
import pandas as pd  
import numpy as np  
from tqdm import tqdm
```

```

from fake_useragent import UserAgent
import tarfile
import pandas_read_xml as pdx
import swifter
import time
pd.set_option('display.max_colwidth', 100)
pd.set_option('display.max_rows', 2000)
pd.set_option('display.max_columns', 2000)

```

```
[2]: df2_rp2 = pd.read_pickle("Processed_Data/jp_rp_sum_new.pkl")
```

```
[3]: df2 = df2_rp2
```

```

[4]: df2["d_dei"] = 0
df2.loc[df2["dei"] > 0, "d_dei"] = 1

df2["d_eeo"] = 0
df2.loc[df2["eeo"] > 0, "d_eeo"] = 1

```

4.0.1 Merge filling time data

```
[5]: df_fillingtime = pd.read_pickle("Processed_Data/
↳20220909US_Job_Record_FillingTime_Public_Clean.pkl")
```

```
[6]: df1_temp_clean = pd.read_pickle("Processed_Data/private_chosen_filling_time.pkl")
```

```

[7]: df_fillingtime = df_fillingtime.append(df1_temp_clean)
df_fillingtime

```

```

[7]:

```

	hash	the_key	delete_date2	\
9967766	0000003cf0ed2539cc4950a95f420261	015633	2008-08-11	
9967770	00000090d1133192dd83e2ed9dc33b1c	015314	2016-02-19	
9967774	000000be62247aa211a827915bbd1ad1	028877	2020-09-16	
9967776	000000c11b16e4556fb058b041e5f630	065417	2019-11-10	
9967782	000001125c8bde7c446643f50211a129	006829	2018-03-17	
...	...	...	...	
2095306	e2448570b1c578f67efb2ca160719245	fskey_07SP06-E	2020-12-04	
2095307	ec932c5608b2ffd7e77db0ec195effcf	fskey_07SP06-E	2022-01-19	
2095308	ee4e67eba2947dc51e342b9376b1cc20	fskey_07SP06-E	2022-05-05	
2095309	f9c3e9dced242d37f3c6bc213c4509a6	fskey_07SP06-E	2021-09-23	
2095310	feab2b72d8f3855b5221d36530035988	fskey_07SP06-E	2021-01-09	

	created_date2	filling_days	filled
9967766	2008-08-02	9.0	1
9967770	2016-02-05	14.0	1
9967774	2020-07-30	48.0	1
9967776	2019-11-03	7.0	1

9967782	2017-12-18	89.0	1
...	...	...	...
2095306	2020-03-04	275.0	1
2095307	2021-01-09	375.0	1
2095308	2020-03-04	792.0	1
2095309	2021-07-20	65.0	1
2095310	2020-12-04	36.0	1

[65657844 rows x 6 columns]

```
[8]: df2_fillingtime = df2.merge(df_fillingtime, on = ["hash"], how = "left")
```

```
[9]: df2_fillingtime["d_public"] = 1
df2_fillingtime.loc[df2_fillingtime["gvkey"].isna(), "d_public"] = 0
```

```
[10]: df2_fillingtime = df2_fillingtime.drop(columns = ['zip', 'country', 'last_checked', 'last_updated',
    ↪ 'delete_date', 'unmapped_location', 'onet_occupation_code', 'url',
    'company_id_updated', 'onet_occupation_code_NEW_x', 'factset_entity_id',
    'cusip', 'cusip_6', "gvkey"])
```

```
[11]: df2_fillingtime.to_pickle("Processed_Data/20230227_DEI_data.pkl")
df2_fillingtime.to_csv("Processed_Data/20230227_DEI_data.csv", index = False)
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```