Introduction to Big Data

- Developed by Dr. Keungoui KIM
- https://awekim.github.io/portfolio/

Lecture 4. Data Manipulation with Pandas II

```
from google.colab import drive
drive.mount('/content/drive')
```

```
import pandas as pd
import numpy as np
sample_1 = pd.read_csv('/content/drive/MyDrive/[Lecture]/IntBigData/BigData_Python/04_DataManipulation/IB
employee = pd.read_csv('/content/drive/MyDrive/[Lecture]/IntBigData/BigData_Python/04_DataManipulation/IB
```

DataFrame Check

```
sample_1.head()
```

```
sample_1.shape
```

```
sample_1_row, sample_1_col = sample_1.shape
print(sample_1_row)
print(sample_1_col)
```

```
sample_1.size
```

```
len(sample_1)
```

```
sample_1.info()
```

```
sample_1.describe()
```

```
sample_1.sort_values(by=['Name'], ascending = True)
```

```
sample_1.sort_values(by=['score'], ascending = False)
```

DataFrame Manipulation

```
sample_1.info()
```

```
sample_1.select_dtypes(include=['object'])
```

```
sample_1.select_dtypes(exclude=['object','int64']).columns
```

```
sample_1_sample = sample_1
sample_1_sample['YearofBorn'] = 2023 - sample_1_sample.age
sample_1_sample.head()
```

```
sample_1_sample = sample_1
sample_1_sample.assign(YearofBorn = 2023 - sample_1_sample.age,
                       ageNext = sample_1_sample.age + 1)
```

```
sample_1_sample
```

```
sample_1_sample.transpose()
```

```
sample_1_sample.columns
```

```
sample_1_sample.columns[[0,1,4,2,3]]
```

```
sample_1_sample.columns = sample_1_sample.columns[[0,1,4,5,2,3]]
```

```
sample_1_sample.head()
```

Checking Missing Values

```
sample_1
```

```
sample_1.isnull()
```

```
sample_1.notnull()
```

```
sample_1.isnull().any()
```

```
sample_1.isnull().any().any()
```

Checking Frequency of Missing Values

```
1 == True
```

```
0 == False
```

```
sample_1.isnull()
```

```
sample_1.isnull().sum()
```

```
type(sample_1.isnull())
```

```
type(sample_1.isnull().sum())
```

```
sample_1.notnull().sum()
```

```
sample_1.notnull().sum().sort_values(ascending=True)
```

```
sample_1.notnull().sum().sort_values(ascending=False)
```

Finding Missing Values

```
sample_1
```

```
pd.Series([True,False]*5)
```

```
sample_1[pd.Series([True,False]*5)]
```

```
sample_1
```

```
sample_1['score'].isnull()
```

```
sample_1[sample_1['score'].isnull()]
```

```
sample_1[sample_1['score'].notnull()]
```

```
sample_1['grade']
```

```
sample_1['grade']=='A'
```

```
sample_1[sample_1['grade']=='A']
```

```
sample_1.loc[sample_1['grade']=='A']
```

```
sample_1['score']>80
```

```
sample_1[sample_1['score']>80]
```

```
sample_1.loc[sample_1['score']>80]
```

```
sample_1.iloc[sample_1['score']>80]
```

Handling Missing Values

.dropna()

```
sample_1.dropna()
```

```
sample_1.dropna(axis=0)
```

```
sample_1.dropna(axis=1)
```

```
sample_1.dropna(how='any')
```

```
sample_1.dropna(how='all')
```

```
sample_1.dropna(inplace=False)
sample_1
```

.fillna()

```
sample_1.fillna(0)
```

```
sample_1.fillna(method='backfill')
```

```
sample_1.fillna(method='bfill')
```

```
sample_1.fillna(method='pad')
```

```
sample_1.fillna(method='ffill')
```

.replace()

```
sample_1.replace(to_replace ='Kim',
                 value = 'Kimmy')
```

```
sample_1.replace(to_replace ={'kim':'kimmy',
                              'Park':'Ppark'})
```

```
sample_1.replace(to_replace ='A',
                 value = 'A+')
```

```
np.nan
```

```
np.nan + 10
```

```
np.nan * 10
```

```
sample_1.replace(to_replace =np.nan,
                 value = 0)
```

.interpolate()

```
sample_1.interpolate(method ='linear')
```

DataFrame Reshape

Reshape with DataFrame (pandas)

```
sample_1_wide = sample_1
sample_1_wide
```

```
sample_1_wide.melt(id_vars='Name',
                   value_vars=['age','major',
                               'score','grade'])
```

```
sample_1_wide.melt(id_vars='Name',
                   value_vars=['age','major','score','grade'],
                   var_name='VariableType',
                   value_name='Amount').head()
```

```
sample_1_long = sample_1_wide.melt(id_vars='Name',value_vars=['age','major','score','grade'])
sample_1_long.rename(columns={'variable':'attribute','value':'amount'}, inplace=True)
sample_1_long.sort_values('Name').head()
```

```
sample_1_long.pivot(index='Name',
                    columns='attribute',
                    values='amount')
```

```
sample_1_long_wide = sample_1_long.pivot(index='Name',
                                         columns='attribute',
                                         values='amount')
sample_1_long_wide
```

```
sample_1_long_wide.columns
```

```
sample_1.columns
```

```
sample_1_long_wide.columns.values
```

```
sample_1_long_wide.columns = sample_1_long_wide.columns.values
sample_1_long_wide
```

```
sample_1_long_wide.index
```

```
sample_1_long.pivot(index='Name',
                    columns='attribute',
                    values='amount').reset_index()
```

Review

1. Import employee.csv to variable called employee
2. What are the names of columns?
3. How many number of rows and columns are there?
4. Is there any missing value?
5. If so, which column?
6. Find the rows where the value of RACE is missing.
7. Convert this table into long table.

```
import pandas as pd
employee = pd.read_csv('/content/drive/MyDrive/[Lecture]/IntBigData/BigData_Python/04_DataManipulation/IBD
```

```
employee
```

```
employee.columns
```

```
employee.shape
```

```
employee.isnull().any().any()
```

```
employee.isnull().any()
```

```
employee[employee.RACE.isnull()]
```

```
employee.melt(id_vars='UNIQUE_ID',
              value_vars=['POSITION_TITLE', 'DEPARTMENT', 'BASE_SALARY', 'RACE', 'EMPLOYMENT_TYPE',
                          'GENDER', 'EMPLOYMENT_STATUS', 'HIRE_DATE','JOB_DATE'],
              var_name='VariableType',
              value_name='Value').head()
```

```
employee = pd.read_csv('/content/drive/MyDrive/[Lecture]/IntBigData/BigData_Python/04_DataManipulation/IBD
employee.columns
employee.shape
employee.isnull().any().any()
employee.isnull().any()
employee[employee.RACE.isnull()]
var_list = ['POSITION_TITLE', 'DEPARTMENT', 'BASE_SALARY', 'RACE',
            'EMPLOYMENT_TYPE', 'GENDER', 'EMPLOYMENT_STATUS',
            'HIRE_DATE','JOB_DATE']
employee.melt(id_vars='UNIQUE_ID',
              value_vars=var_list,
              var_name='VariableType',
              value_name='Value').head()
```

Variable Conversion

```
import pandas as pd
import numpy as np
sample_1 = pd.read_csv('/content/drive/MyDrive/[Lecture]/IntBigData/BigData_Python/04_DataManipulation/sar
sample_1['age'] = [10, 29, 33, 42, 52, 53, 62, 90, 34, 25]
sample_1
```

```
bins = [0, 19, 29, 49, 69, 89]
labels = ['Kids', '20s', 'AZ', '50s/60s','70s/80s']
sample_1['age_group'] = pd.cut(sample_1['age'],
                               bins=bins,
                               labels=labels)
sample_1
```

DataFrame Summarise

```
import pandas as pd
import numpy as np
```

```
hack_sal = pd.read_csv('/content/drive/MyDrive/[Lecture]/IntBigData/BigData_Python/04_DataManipulation/II
```

```
hack_sal.head()
```

Grouping DataFrame

```
hack_sal['job_title_category'].unique()
```

```
hack_sal['job_title_category'].nunique()
```

```
type(hack_sal)
```

```
hack_sal_group = (
    hack_sal.
    groupby('job_title_category')
)
type(hack_sal_group)
```

```
hack_sal_group
```

```
hack_sal_group.groups
```

```
hack_sal_group.ngroups
```

```
hack_sal_group.size()
```

```
hack_sal_group.get_group('Data')
```

```
hack_sal_group2 = (
    hack_sal.groupby(
        ['job_title_category',
         'total_experience_years_d'])
)
type(hack_sal_group2)
```

```
hack_sal_group2.groups
```

```
hack_sal_group2.ngroups
```

```
hack_sal_group2.size()
```

```
hack_sal_group2.first()
# hack_sal_group2.last()
```

```
(
    hack_sal_group2.
 get_group(['Data', 'overDecade'])
)
```

```
(hack_sal_group2.
 get_group(('Data', 'overDecade'))
)
```

Aggregating DataFrame

```
import pandas as pd
hack_sal = pd.read_csv('/content/drive/MyDrive/[Lecture]/IntBigData/BigData_Python/04_DataManipulation/II
hack_sal_group = (
    hack_sal.
    groupby('job_title_category')
)
hack_sal_group2 = (
    hack_sal.groupby(
        ['job_title_category',
         'total_experience_years_d'])
)
```

```
hack_sal_group.head(1)
```

```
hack_sal_group2.groups
```

```
hack_sal.iloc[[342, 388, 611, 627, 1444, 1643],]
```

```
hack_sal_group.annual_base_pay.max()
```

```
hack_sal_group[['annual_base_pay','signing_bonus']].max()
```

```
hack_sal_group2.annual_base_pay.max()
```

```
(
    hack_sal_group2.annual_base_pay.
 agg(['max','min','count','median','mean'])
)
```

```
standardization = lambda x: (x - x.mean()) / x.std()
```

```
hack_sal_group2.annual_base_pay.apply(standardization).head()
```

```
hack_sal_group2.signing_bonus.apply(standardization)
```

```
hack_sal_group2.head()
```

```
(
    hack_sal_group2['annual_base_pay','signing_bonus','annual_bonus'].
 apply(standardization).head()
)
```

```
hack_sal.head()
```

Apply exercise

```
import pandas as pd
df_sample = {'type':['a','a','b','b'],'var_1':[1,2,3,4],
             'var_2':[1,1,1,1]}
df_sample = pd.DataFrame(df_sample)
df_sample
```

```
df_sample['var_1_mean'] = df_sample['var_1'].mean()
df_sample['var_2_mean'] = df_sample['var_2'].mean()
df_sample
```

```
len(df_sample.type)
```

```
df_sample.type.apply(len)
```

```
df_sample.var_1.mean()
```

```
df_sample.var_1.apply("mean")
```

```
df_sample_gr = (
    df_sample.groupby('type')#, group_keys=True)
)
```

```
import numpy as np
df_sample['var_1'].apply(np.mean)
```

```
df_sample_gr['var_1'].apply(np.mean)
```

```
df_sample.var_1.transform(lambda x: x - x.mean())
```

```
df_sample.var_1.apply(lambda x: x - df_sample['var_1'].mean())
```

```
df_sample_gr.var_1.apply(lambda x: x - x.mean()).reset_index()
```

```
df_sample_gr.var_1.transform(lambda x: x - x.mean()).reset_index()
```