

### Useful Shortcuts

Control Q	access documentation
Control D	duplicate code block
dir()	see the list of methods
Tab (in Git bash)	to toggle the files in the directory

### Useful OS commands

os.getcwd()	get current directory
os.chdir('path')	change directory
os.listdir('folder_path')	list directory
os.path.join('path1', 'path2')	join file path

### Set up Env

conda env create --prefix P:\env\myenv --file=P:\filepath\env.yml

conda create env\_name create a new env

pip install -r "P:\filepath\requirements.txt"

pip install 'P:\filepath\package.whl' --force-reinstall reinstall whl file stall

### Pandas Display

pd.set\_option('display.max\_colwidth', 10\_000)

pd.set\_option('display.max\_rows', 1000)

ps.set\_option('display.max\_columns', 100)

### Neat Functions

[function(i) for i in list if logic\_expression] list comprehension

lambda arguments: a lambda function is a small anonymous function

(e.g.) x = lambda a, b : a \* b

print(x(5, 6))

### Explore data

df.col.describe()

df.col.value\_counts() normalize=True

### Reshape dataframe

df.groupby('col1').col2.count() count / sum / min / max / mean

df.groupby(['col1', 'col2']).['col3', 'col4'].count()

df1 = df.groupby('col').agg({'value1':['min','max'], 'value2':['mean']})

df.pivot\_table(index='col1', columns='col2', values=['col3', 'col4'], aggfunc={'col3':np.mean, 'col4':[min, max]})

df1.columns = ['\_'.join(col) for col in df1.columns] to flatten multi index df after groupby / pivot

df1.merge(df2, left\_on='l\_key', right\_on='r\_key', how='left') how: {'left', 'right', 'outer', 'inner', 'cross'}, default 'inner'

pd.concat([df1, df2]) union 2 dataframes

### Statements

#### For Loop

```
for var in range:
    statements
```

#### While Loop

```
while expression:
    statements
```

#### If Statement

```
if expression:
    statements
elif expression:
    statements
else:
    statements
```

#### Exception Handling

```
try:
    print(x)
except NameError:
    print("variable x is not defined")
except:
    print("sth else went wrong")
else:
    print("nth went wrong")
finally:
    print("the 'try except' is finished")
```