



COGS 108 – DATA SCIENCE IN PRACTICE

(DR. BRAD VOYTEK)

ENLIN WEI

SECTIONS A01 & A04

(MONDAYS & FRIDAYS 2PM-3PM)

OH FRIDAYS 12-2PM @ PC ROOM 4

SERIES.APPLY(FUNCTION)

- Allows us to call a function on all values of a Series
- Make sure to save the output back to the Series!
 - `df['year'] = df['year'].apply(standardize_year)`

ALTERNATIVE: LAMBDA FUNCTION

- Creating a new column with the squares of a column of your dataframe:
 - `df['squared'] = df['values'].apply(lambda n: n**2)`

SERIES.VALUE_COUNTS()

- Returns a DataFrame where the indices are the unique entries and the counts are numbers of each entry, in order of highest to lowest counts

SERIES.VALUE_COUNTS()

fav_boba
Almond
Honey
Honey
Wintermelon
Brown sugar
Wintermelon
Honey
Jackfruit
Honey
Wintermelon

```
val_ct = df['fav_boba'].value_counts()
```

val_ct

Honey	4
Wintermelon	3
Almond	1
Brown sugar	1
Jackfruit	1

val_ct.index[2]

val_ct[0]

SCIPY.STATS.NORMALTEST()

- Pass in numpy array (df['column'].values)
- Outputs a p-value and a standard deviation

```
if p-val < alpha-val:  
    print("NOT a normal distribution!")  
else:  
    print("A normal distribution!")
```

DATAFRAME.PIVOT_TABLE()

- Arguments: value & index (as lists)
- Allows us to compute averages of “values” given two “indices”
- See example I posted on GitHub