



PYTHON FOR MATLAB USERS

# Python dictionaries

Justin Kiggins  
Product Manager



# What is a dictionary?

- Similar to a MATLAB structure array
- Collection of key:value pairs
- Ordering is not guaranteed



# Retrieving data from a dictionary

```
type(definitions)
```

```
dict
```

```
print(definitions['aardvark'])
```

```
A nocturnal burrowing mammal with long ears, a tubular snout, and a long  
extensible tongue, feeding on ants and termites. Aardvarks are native to  
Africa and have no close relatives.
```



# Creating dictionaries

```
# Create a dictionary  
dog = {'name': 'Toby', 'breed': 'Basset Hound'}
```



# Adding data to a dictionary

```
# Add more key:value pairs
dog['weight (lbs)'] = 552.3
dog['birthdate'] = "2016-06-26"
print(dog)
```

```
{'name': 'Toby',
 'breed': 'Basset Hound',
 'weight (lbs)': 552.3,
 'birthdate': '2016-06-26'}
```



# Updating dictionaries

```
# Update weight value
dog['weight (lbs)'] = 52.3

print(dog)
```

```
{'name': 'Toby',
 'breed': 'Basset Hound',
 'weight (lbs)': 52.3,
 'birthdate': '2016-06-26'}
```



# Removing data from a dictionary

```
dog.pop('birthdate')  
print(dog)  
  
{'name': 'Toby',  
 'breed': 'Basset Hound',  
 'weight (lbs)': 52.3}
```



PYTHON FOR MATLAB USERS

**Let's practice!**





PYTHON FOR MATLAB USERS

# Introduction to DataFrames

**Justin Kiggins**  
Product Manager



# DataFrames

- `pandas` package
- Tabular data
- `pandas DataFrame` = MATLAB table
- Store data where each observation has mixed types:
  - floats
  - integers
  - Booleans
  - strings



# DataFrames

```
state      rank  num_households  frac_dog_owners  frac_cat_owners
Alabama    17      1828000      0.441             0.274
Arizona    18      2515000      0.401             0.296
Arkansas    6       1148000      0.479             0.306
California  40     12974000      0.328             0.283
Colorado   13      1986000      0.425             0.323
Connecticut 33      1337000      0.283             0.319
Delaware   25       334000      0.337             0.337
District of Columbia 48    287000      0.131             0.116
Florida    32      7609000      0.357             0.273
Georgia    31      3798000      0.401             0.273
```



# .head() method

```
pets.head()
```

state	rank	num_households	frac_dog_owners	frac_cat_owners
Alabama	17	1828000	0.441	0.274
Arizona	18	2515000	0.401	0.296
Arkansas	6	1148000	0.479	0.306
California	40	12974000	0.328	0.283
Colorado	13	1986000	0.425	0.323



# .columns attribute

```
pets.columns
```

```
Index(['rank', 'num_households', 'frac_dog_owners', 'frac_cat_owners'],  
      dtype='object')
```

# .index attribute

```
pets.index
```

```
Index(['Alabama', 'Arizona', 'Arkansas', 'California', 'Colorado',  
      'Connecticut', 'Delaware', 'District of Columbia', 'Florida',  
      'Georgia', 'Idaho', 'Illinois', 'Indiana', 'Iowa', 'Kansas',  
      'Kentucky', 'Louisiana', 'Maine', 'Maryland', 'Massachusetts',  
      'Michigan', 'Minnesota', 'Mississippi', 'Missouri', 'Montana',  
      'Nebraska', 'Nevada', 'New Hampshire', 'New Jersey', 'New Mexico',  
      'New York', 'North Carolina', 'North Dakota', 'Ohio', 'Oklahoma',  
      'Oregon', 'Pennsylvania', 'Rhode Island', 'South Carolina',  
      'South Dakota', 'Tennessee', 'Texas', 'Utah', 'Vermont', 'Virginia',  
      'Washington', 'West Virginia', 'Wisconsin', 'Wyoming'],  
      dtype='object', name='state')
```



# Getting one column out

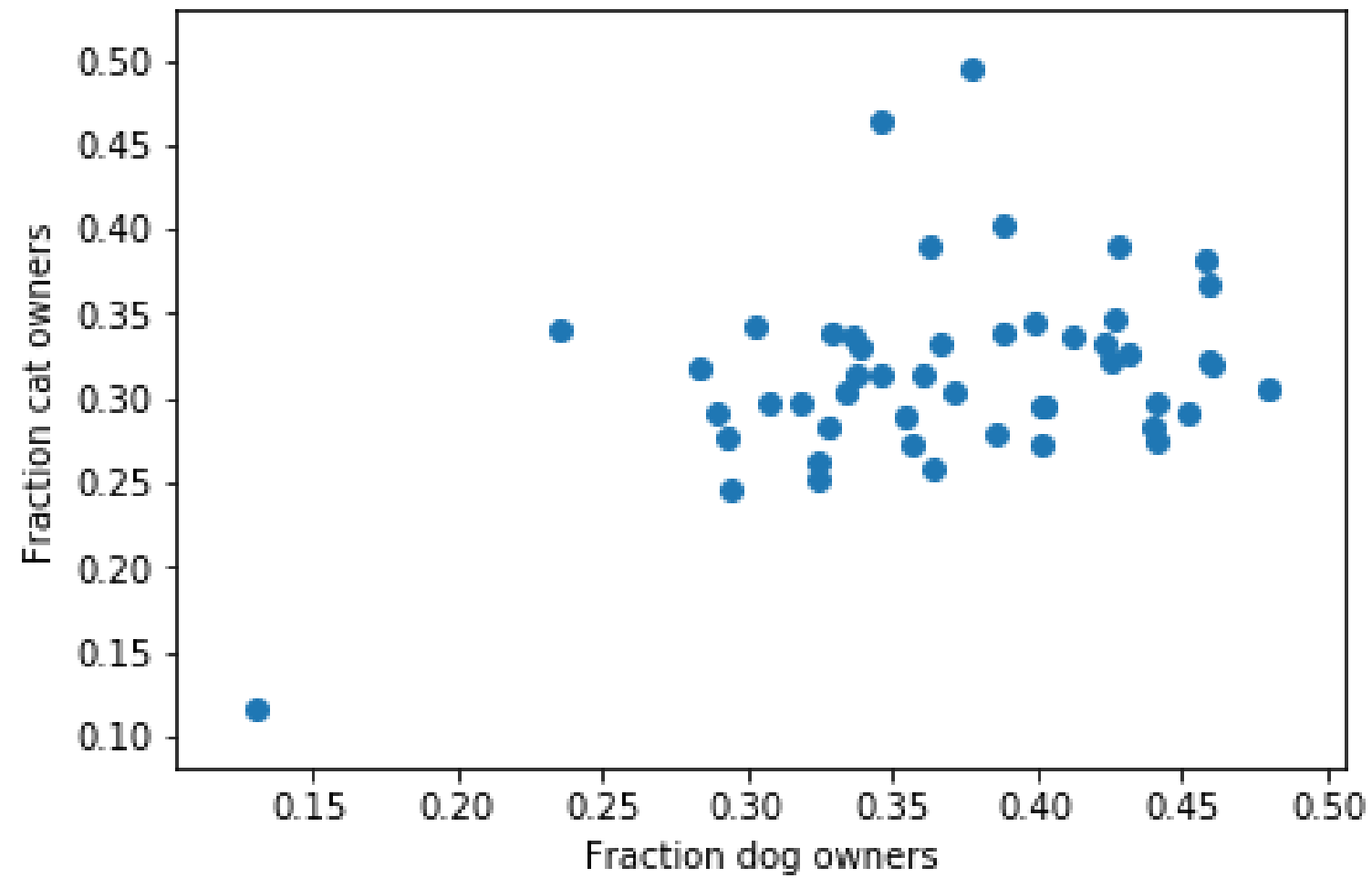
```
pets['rank']
```

```
state
Alabama      17
Arizona      18
Arkansas       6
California    40
Colorado      13
Connecticut   33
Delaware      25
District of Columbia  48
...
Tennessee    16
Texas        20
Utah         44
Vermont       0
Virginia      37
Washington    5
West Virginia  7
Wisconsin     21
Wyoming       9
Name: rank, dtype: int64
```



# NumPy & Matplotlib compatible

```
plt.scatter(pets['frac_dog_owners'], pets['frac_cat_owners'])
```







## PYTHON FOR MATLAB USERS

**Let's practice!**



PYTHON FOR MATLAB USERS

# Accessing pandas DataFrames

Justin Kiggins  
Product Manager



# Selecting multiple dataframe columns

```
pets[['frac_dog_owners', 'rank']]
```

	frac_dog_owners	rank
state		
Alabama	0.441	17
Arizona	0.401	18
Arkansas	0.479	6
California	0.328	40
Colorado	0.425	13
Connecticut	0.283	33
Delaware	0.337	25
District of Columbia	0.131	48
...		
Tennessee	0.441	16
Texas	0.440	20
Utah	0.294	44
Vermont	0.377	0
Virginia	0.354	37
Washington	0.363	5
West Virginia	0.458	7
Wisconsin	0.339	21
Wyoming	0.388	9



# Selecting rows

1. pandas indexing with `.loc`
2. Python indexing with `.iloc`



# Selecting rows using pandas indexing

```
pets.loc['Massachusetts']
```

```
rank          47.000  
num_households 2618000.000  
frac_dog_owners    0.236  
frac_cat_owners    0.341  
Name: Massachusetts, dtype: float64
```



# Selecting rows using Python indexing

```
pets.iloc[17]
```

```
rank          4.000  
num_households  548000.000  
frac_dog_owners  0.346  
frac_cat_owners  0.464  
Name: Maine, dtype: float64
```



## PYTHON FOR MATLAB USERS

**Let's practice!**



PYTHON FOR MATLAB USERS

# Creating pandas DataFrames

Justin Kiggins  
Product Manager





# From a CSV file

```
Date, AveragePrice, Total Volume, 4046, 4225, 4770
2015-12-27, 1.33, 64236.62, 1036.74, 54454.85, 48.16
2015-12-20, 1.35, 54876.98, 674.28, 44638.81, 58.33
2015-12-13, 0.93, 118220.22, 794.70, 109149.67, 130.50
2015-12-06, 1.08, 78992.15, 1132.00, 71976.41, 72.58
2015-11-29, 1.28, 51039.60, 941.48, 43838.39, 75.78
```

```
import pandas as pd
avocados = pd.read_csv('avocados.csv')
```

```
avocados.head()
```

	Date	AveragePrice	Total Volume	4046	4225	4770
0	2015-12-27	1.33	64236.62	1036.74	54454.85	48.16
1	2015-12-20	1.35	54876.98	674.28	44638.81	58.33
2	2015-12-13	0.93	118220.22	794.70	109149.67	130.50
3	2015-12-06	1.08	78992.15	1132.00	71976.41	72.58
4	2015-11-29	1.28	51039.60	941.48	43838.39	75.78

```
avocados = pd.read_csv('avocados.csv', index_col=0)
```



# From a dictionary of lists

```
import pandas as pd
pd.DataFrame()

forecast_raw = {
    'weekday': ['Mon', 'Tues', 'Wed', 'Thurs', 'Fri', 'Sat', 'Sun'],
    'rain': [True, False, False, False, True, True, False],
    'temp': [68, 72, 73, 75, 67, 68, 68]
}
```

```
forecast = pd.DataFrame(forecast_raw)
print(forecast)
```

	rain	temp	weekday
0	True	68	Mon
1	False	72	Tues
2	False	73	Wed
3	False	75	Thurs
4	True	67	Fri
5	True	68	Sat
6	False	68	Sun

# From a list of dictionaries

```
forecast_raw = [  
    {'rain': True, 'temp': 68, 'weekday': 'Mon'},  
    {'rain': False, 'temp': 72, 'weekday': 'Tues'},  
    {'rain': False, 'temp': 73, 'weekday': 'Wed'},  
    {'rain': False, 'temp': 75, 'weekday': 'Thurs'},  
    {'rain': True, 'temp': 67, 'weekday': 'Fri'},  
    {'rain': True, 'temp': 68, 'weekday': 'Sat'},  
    {'rain': False, 'temp': 68, 'weekday': 'Sun'},  
]
```

```
import pandas as pd  
forecast = pd.DataFrame(forecast_raw)  
print(forecast)
```

	rain	temp	weekday
0	True	68	Mon
1	False	72	Tues
2	False	73	Wed
3	False	75	Thurs
4	True	67	Fri
5	True	68	Sat
6	False	68	Sun



## PYTHON FOR MATLAB USERS

**Let's practice**