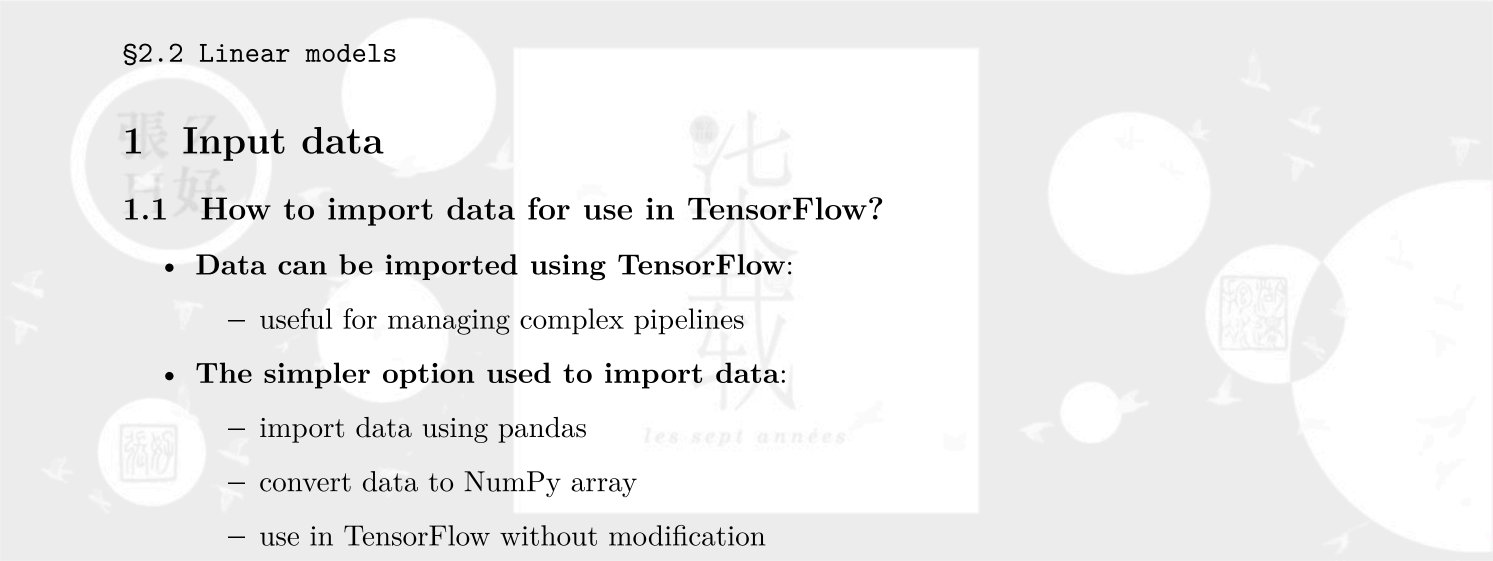
Input data

Puteaux, Fall/Winter 2020-2021

```
################################
##                            ##
##  Deep Learning in Python   ##
##                            ##
################################
```

```
§2 Introduction to TensorFlow in Python
```



```
§2.2 Linear models
```

1 Input data

1.1 How to import data for use in TensorFlow?

- **Data can be imported using TensorFlow**:
  - useful for managing complex pipelines
- **The simpler option used to import data**:
  - import data using pandas
  - convert data to NumPy array
  - use in TensorFlow without modification
- Pandas also has methods for handling data in other formats:
  - e.g., `read_json()` , `read_html()` , `read_excel()`

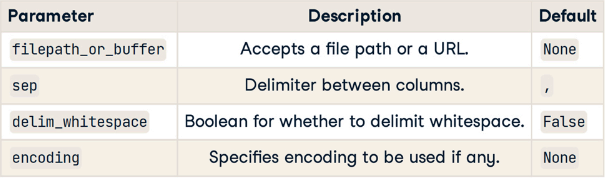
1.2 Code of how to import and convert data:

[1]:
```
# Import numpy and pandas
import numpy as np
import pandas as pd

# Load data from csv
housing = pd.read_csv('ref1. King county house sales.csv')

# Convert to numpy array
housing = np.array(housing)
```

```
print(housing)
```

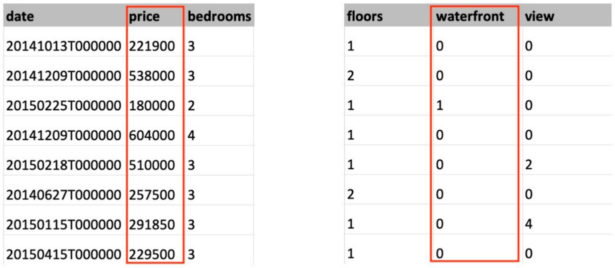
```
[[7129300520 '20141013T000000' 221900.0 … -122.257 1340 5650]
 [6414100192 '20141209T000000' 538000.0 … -122.319 1690 7639]
 [5631500400 '20150225T000000' 180000.0 … -122.23299999999999 2720 8062]
 …
 [1523300141 '20140623T000000' 402101.0 … -122.29899999999999 1020 2007]
 [291310100 '20150116T000000' 400000.0 … -122.069 1410 1287]
 [1523300157 '20141015T000000' 325000.0 … -122.29899999999999 1020 1357]]
```

1.3 What are the parameters of read_csv()?



Parameter	Description	Default
filepath_or_buffer	Accepts a file path or a URL.	None
sep	Delimiter between columns.	,
delim_whitespace	Boolean for whether to delimit whitespace.	False
encoding	Specifies encoding to be used if any.	None

1.4 How to use mixed type datasets?



date	price	bedrooms
20141013T000000	221900	3
20141209T000000	538000	3
20150225T000000	180000	2
20141209T000000	604000	4
20150218T000000	510000	3
20140627T000000	257500	3
20150115T000000	291850	3
20150415T000000	229500	3

floors	waterfront	view
1	0	0
2	0	0
1	1	0
1	0	0
1	0	2
2	0	0
1	0	4
1	0	0

1.5 Code of setting the data type:

[2]:
```
# Load KC dataset
housing = pd.read_csv('ref1. King county house sales.csv')

# Convert price column to float32
price = np.array(housing['price'], np.float32)

# Convert waterfront column to Boolean
waterfront = np.array(housing['waterfront'], np.bool)

print(price)
print(waterfront)
```

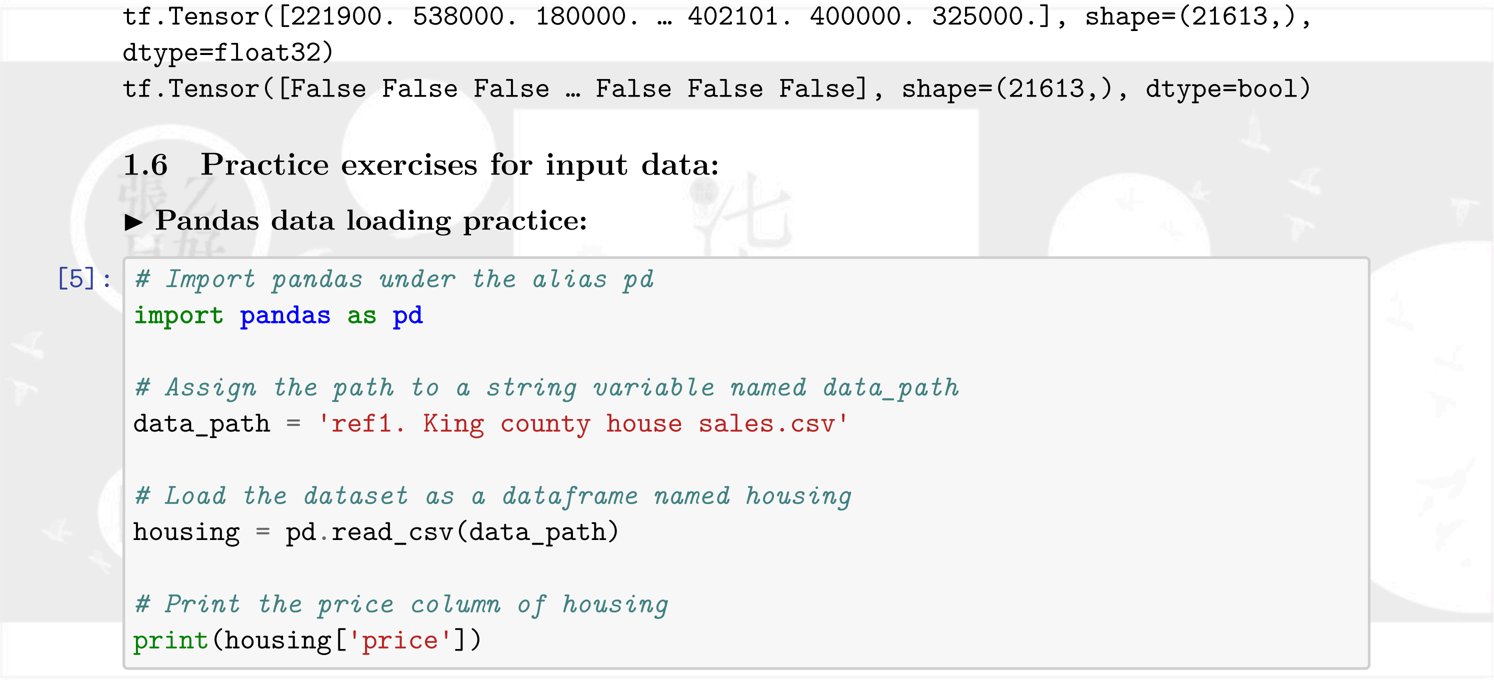
```
[221900. 538000. 180000. … 402101. 400000. 325000.]
[False False False … False False False]
```

[3]:

```
import tensorflow as tf
```

[4]:

```
# Load KC dataset
housing = pd.read_csv('ref1. King county house sales.csv')

# Convert price column to float32
price = tf.cast(housing['price'], tf.float32)

# Convert waterfront column to Boolean
waterfront = tf.cast(housing['waterfront'], tf.bool)

print(price)
print(waterfront)
```



```
tf.Tensor([221900. 538000. 180000. … 402101. 400000. 325000.], shape=(21613,),
dtype=float32)
tf.Tensor([False False False … False False False], shape=(21613,), dtype=bool)
```

1.6 Practice exercises for input data:

▶ **Pandas data loading practice:**

[5]:

```
# Import pandas under the alias pd
import pandas as pd

# Assign the path to a string variable named data_path
data_path = 'ref1. King county house sales.csv'

# Load the dataset as a dataframe named housing
housing = pd.read_csv(data_path)

# Print the price column of housing
print(housing['price'])
```

```
0        221900.0
1        538000.0
2        180000.0
3        604000.0
4        510000.0
           …
21608    360000.0
21609    400000.0
21610    402101.0
21611    400000.0
21612    325000.0
Name: price, Length: 21613, dtype: float64
```

▶ **Data type setting practice:**

[6]:

```
# Import numpy and tensorflow with their standard aliases
import numpy as np
import tensorflow as tf

# Use a numpy array to define price as a 32-bit float
price = np.array(housing['price'], np.float32)

# Define waterfront as a Boolean using cast
waterfront = tf.cast(housing['waterfront'], tf.bool)

# Print price and waterfront
print(price)
print(waterfront)
```

```
[221900. 538000. 180000. … 402101. 400000. 325000.]
tf.Tensor([False False False … False False False], shape=(21613,), dtype=bool)
```