Data collection, Modelling and Compilation ¶

Import required packages

In [1]:
```
import numpy as np
import pandas as pd
import sqlite3
```

Creating a dataframe from hardcoded data (using dictionaries)

In [2]:
```
my_dict = {'name': ['a', 'b', 'c', 'd', 'e', 'f', 'g'],
           'age' : [20, 27, 35, 55, 18, 21, 35],
           'designation' : ['VP', 'CEO', 'CFO', 'VP', 'VP', 'CEO', 'MD']}

df = pd.DataFrame(my_dict)
df
```

Out[2]:

	name	age	designation
0	a	20	VP
1	b	27	CEO
2	c	35	CFO
3	d	55	VP
4	e	18	VP
5	f	21	CEO
6	g	35	MD

Saving a dataframe to a CSV file

In [3]:
```
df.to_csv("output_files/Practical 1.csv")
```

	A	B	C	D	E
1		name	age	designation	
2	0	a	20	VP	
3	1	b	27	CEO	
4	2	c	35	CFO	
5	3	d	55	VP	
6	4	e	18	VP	
7	5	f	21	CEO	
8	6	g	35	MD	
9					
10					

Loading CSV file as a dataframe

In [4]:
```
df_csv = pd.read_csv('output_files/Practical 1.csv')
df_csv
```

Out[4]:

	Unnamed: 0	name	age	designation
0	0	a	20	VP
1	1	b	27	CEO
2	2	c	35	CFO
3	3	d	55	VP
4	4	e	18	VP
5	5	f	21	CEO
6	6	g	35	MD

Loading data from a CSV file without headers

In [5]:
```
df_student_mat = pd.read_csv('data_sets/student-mat.csv', header = None)
df_student_mat.head()
```

Out[5]:

	0	1	2	3	4	5	6	7	8	9	...	23	24
0	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime
1	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3
2	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3
3	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3
4	GP	F	15	U	GT3	T	4	2	health	services	...	3	2

5 rows × 33 columns

Creating a dataframe using hardcoded data (multiple lists)

In [6]:
```
names = ['Bob', 'Jessica', 'Mary', 'John', 'Mel']
grades = [76, 95, 77, 78, 99]
bscdegrees = [1, 1, 0, 0, 1]
mscdegrees = [2, 1, 0, 0, 0]
phddegrees = [0, 1, 0, 0, 0]
Degrees = zip(names, grades, bscdegrees, mscdegrees, phddegrees)
columns = ['Names', 'Grades', 'BS', 'MS', 'PhD']
df_multi_lists = pd.DataFrame(data = Degrees, columns = columns)
df_multi_lists
```

Out[6]:

	Names	Grades	BS	MS	PhD
0	Bob	76	1	2	0
1	Jessica	95	1	1	1
2	Mary	77	0	0	0
3	John	78	0	0	0
4	Mel	99	1	0	0

Loading data from Excel files into dataframes

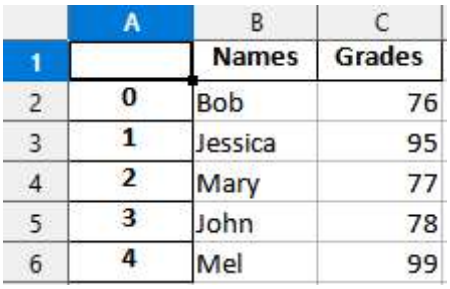
In [7]:
```
df_xlsx_read = pd.read_excel('data_sets/gradedata.xlsx')
df_xlsx_read.columns = ['first', 'last', 'sex', 'age', 'exer', 'hrs', 'grd', 'add
df_xlsx_read.head()
```

Out[7]:

	first	last	sex	age	exer	hrs	grd	addr
0	Marcia	Pugh	female	17	3	10	82.4	7379 Highland Rd. , Dublin, GA 31021
1	Kadeem	Morrison	male	18	4	4	78.2	8 Bayport St. , Honolulu, HI 96815
2	Nash	Powell	male	18	5	9	79.3	Encino, CA 91316, 3 Lilac Street
3	Noelani	Wagner	female	14	2	7	83.2	Riverview, FL 33569, 9998 North Smith Dr.
4	Noelani	Cherry	female	18	4	15	87.4	97 SE. Ocean Street , Bethlehem, PA 18015

Saving a dataframe to a Excel file

In [8]:
```
names = ['Bob', 'Jessica', 'Mary', 'John', 'Mel']
grades = [76, 95, 77, 78, 99]
grade_list = zip(names, grades)
df = pd.DataFrame(data = grade_list, columns = ['Names', 'Grades'])
writer = pd.ExcelWriter('output_files/Practical 1.xlsx', engine = "xlsxwriter")
df.to_excel(writer, sheet_name = "Sheet 1")
writer.save()
```



	A	B	C
1		Names	Grades
2	0	Bob	76
3	1	Jessica	95
4	2	Mary	77
5	3	John	78
6	4	Mel	99

Reading data from a SQLite database

In [9]:
```
import sqlite3

con = sqlite3.connect('data_sets/portal_mammals.sqlite')

cur = con.cursor()

for row in cur.execute('select * from species;'):
    print(row)


con.close()
```

```
('AB', 'Amphispiza', 'bilineata', 'Bird')
('AH', 'Ammospermophilus', 'harrisi', 'Rodent')
('AS', 'Ammodramus', 'savannarum', 'Bird')
('BA', 'Baiomys', 'taylori', 'Rodent')
('CB', 'Campylorhynchus', 'brunneicapillus', 'Bird')
('CM', 'Calamospiza', 'melanocorys', 'Bird')
('CQ', 'Callipepla', 'squamata', 'Bird')
('CS', 'Crotalus', 'scutalatus', 'Reptile')
('CT', 'Cnemidophorus', 'tigris', 'Reptile')
('CU', 'Cnemidophorus', 'uniparens', 'Reptile')
('CV', 'Crotalus', 'viridis', 'Reptile')
('DM', 'Dipodomys', 'merriami', 'Rodent')
('DO', 'Dipodomys', 'ordii', 'Rodent')
('DS', 'Dipodomys', 'spectabilis', 'Rodent')
('DX', 'Dipodomys', 'sp.', 'Rodent')
('EO', 'Eumeces', 'obsoletus', 'Reptile')
('GS', 'Gambelia', 'silus', 'Reptile')
('NL', 'Neotoma', 'albigula', 'Rodent')
('NX', 'Neotoma', 'sp.', 'Rodent')
('OL', 'Onychomys', 'leucogaster', 'Rodent')
('OT', 'Onychomys', 'torridus', 'Rodent')
('OX', 'Onychomys', 'sp.', 'Rodent')
('PB', 'Chaetodipus', 'baileyi', 'Rodent')
('PC', 'Pipilo', 'chlorurus', 'Bird')
('PE', 'Peromyscus', 'eremicus', 'Rodent')
('PF', 'Perognathus', 'flavus', 'Rodent')
('PG', 'Pooecetes', 'gramineus', 'Bird')
('PH', 'Perognathus', 'hispidus', 'Rodent')
('PI', 'Chaetodipus', 'intermedius', 'Rodent')
('PL', 'Peromyscus', 'leucopus', 'Rodent')
('PM', 'Peromyscus', 'maniculatus', 'Rodent')
('PP', 'Chaetodipus', 'penicillatus', 'Rodent')
('PU', 'Pipilo', 'fuscus', 'Bird')
('PX', 'Chaetodipus', 'sp.', 'Rodent')
('RF', 'Reithrodontomys', 'fulvescens', 'Rodent')
('RM', 'Reithrodontomys', 'megalotis', 'Rodent')
('RO', 'Reithrodontomys', 'montanus', 'Rodent')
('RX', 'Reithrodontomys', 'sp.', 'Rodent')
('SA', 'Sylvilagus', 'audubonii', 'Rabbit')
('SB', 'Spizella', 'breweri', 'Bird')
('SC', 'Sceloporus', 'clarki', 'Reptile')
('SF', 'Sigmodon', 'fulviventer', 'Rodent')
('SH', 'Sigmodon', 'hispidus', 'Rodent')
('SO', 'Sigmodon', 'ochrognathus', 'Rodent')
('SS', 'Spermophilus', 'spilosoma', 'Rodent')
```

```
('ST', 'Spermophilus', 'tereticaudus', 'Rodent')
('SU', 'Sceloporus', 'undulatus', 'Reptile')
('SX', 'Sigmodon', 'sp.', 'Rodent')
('UL', 'Lizard', 'sp.', 'Reptile')
('UP', 'Pipilo', 'sp.', 'Bird')
('UR', 'Rodent', 'sp.', 'Rodent')
('US', 'Sparrow', 'sp.', 'Bird')
('ZL', 'Zonotrichia', 'leucophrys', 'Bird')
('ZM', 'Zenaida', 'macroura', 'Bird')
```

Obtaining one or all enteries from an SQLite database

In [10]:
```
import sqlite3

con = sqlite3.connect('data_sets/portal_mammals.sqlite')

cur = con.cursor()

cur.execute('select plot_id from plots where plot_type = "Control"')
print(cur.fetchall())

cur.execute('select species from species where taxa = "Bird"')
print(cur.fetchone())

con.close()
```

```
[(2,), (4,), (8,), (11,), (12,), (14,), (17,), (22,)]
('bilineata',)
```

Writing a DataFrame to a SQLite database

In [11]:
```
conn = sqlite3.connect('output_files/Practical 1.db')
cur = conn.cursor()

cars = {'Brand' : ['Honda Civic', 'Toyota Corolla', 'Ford Focus', 'Audi A4'],
        'Price' : [22000, 25000, 27000, 35000]}
df = pd.DataFrame(cars, columns = ['Brand', 'Price'])
print(df)

cur.execute('drop table if exists cars1_fds')
cur.execute('create table cars1_fds(Brand text, Price number)')
conn.commit()

df.to_sql('cars1_fds', conn, if_exists = 'replace', index = False)
```

```
            Brand  Price
0     Honda Civic  22000
1  Toyota Corolla  25000
2      Ford Focus  27000
3         Audi A4  35000
```

Loading data from SQLite database into a dataframe

In [12]:
```
cur.execute('select Brand, max(Price) from cars1_fds')
df = pd.DataFrame(cur.fetchall(), columns = ['Brand', 'Price'])
df
```

Out[12]:

	Brand	Price
0	Audi A4	35000

In [ ]:
```

```