handson_full

December 19, 2020

0.1 This is the hands-on for "Introduction to Data Analytics" workshop

Prepared by Dr. Tan Wooi Haw, Multimedia University

[1]:
```
# Initial settings
%matplotlib inline
from warnings import filterwarnings
filterwarnings('ignore')
```

Hands-on 1: Find the unique alphabets and their frequency of occurence

[2]:
```
# Hands-on 1
s1 = 'A quick brown fox jumps over the lazy dog.'

t1 = [c for c in s1.lower() if c.isalpha()]
d1 = {}
for c in sorted(t1):
    d1[c] = d1.get(c, 0) + 1
print(d1)

# Can you try with the following sentence?
s2 = 'Amazingly few discotheques provide jukeboxes.'
```

```
{'a': 2, 'b': 1, 'c': 1, 'd': 1, 'e': 2, 'f': 1, 'g': 1, 'h': 1, 'i': 1, 'j': 1,
'k': 1, 'l': 1, 'm': 1, 'n': 1, 'o': 4, 'p': 1, 'q': 1, 'r': 2, 's': 1, 't': 1,
'u': 2, 'v': 1, 'w': 1, 'x': 1, 'y': 1, 'z': 1}
```

Hands-on 2: Find the unique words and their frequency of occurence

[3]:
```
# Hands-on 2
s3 = '''Peter Piper picked a peck of pickled peppers.
    A peck of pickled peppers Peter Piper picked.
    If Peter Piper picked a peck of pickled peppers,
    where is the peck of pickled peppers Peter Piper picked?'''

t3 = [c if c.isalpha() else ' ' for c in s3.lower()]
w3 = ''.join(t3).split()
d3 = {}
for c in sorted(w3):
```

```
    d3[c] = d3.get(c, 0) + 1
print(d3)

# Using Counter from the collections module
from collections import Counter
count3 = Counter(w3)
print(count3)
print(count3.most_common(3))

# Can you try with the following sentences?
s4 = '''She sells seashells on the sea shore.
    The shells she sells are seashells, I am sure.
    And if she sells seashells on the sea shore,
    then I am sure she sells seashore shells.'''
```

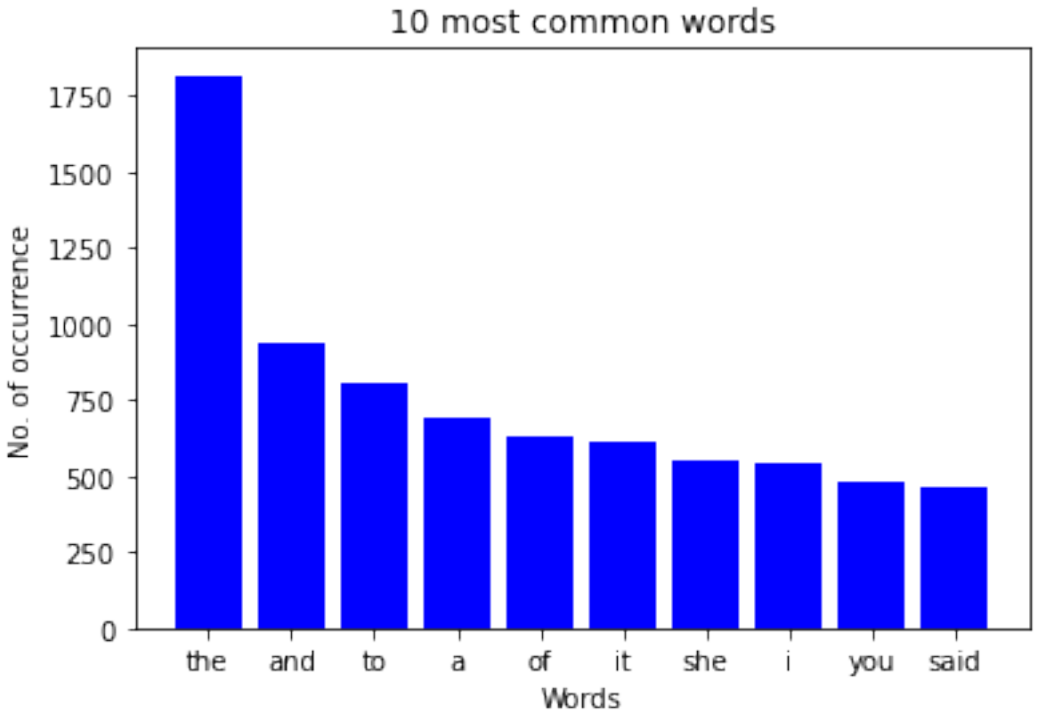
```
{'a': 3, 'if': 1, 'is': 1, 'of': 4, 'peck': 4, 'peppers': 4, 'peter': 4,
'picked': 4, 'pickled': 4, 'piper': 4, 'the': 1, 'where': 1}
Counter({'peter': 4, 'piper': 4, 'picked': 4, 'peck': 4, 'of': 4, 'pickled': 4,
'peppers': 4, 'a': 3, 'if': 1, 'where': 1, 'is': 1, 'the': 1})
[('peter', 4), ('piper', 4), ('picked', 4)]
```

Hands-on 3: Find the 10 most frequent words in a text file

[4]:
```
# Hands-on 3
from collections import Counter
import matplotlib.pyplot as plt

with open('data/alice.txt', 'r') as f:
    s5 = f.read()
print(s5[:400])

t5 = [c if c.isalpha() else ' ' for c in s5.lower()]
w5 = ''.join(t5).split()
count5 = Counter(w5)
print(count5.most_common(10))
print(f'The word "alice" appears {count5["alice"]} times.')

# Plot bar chart for 10 most common words
if 'count5' in globals():
    freq10 = count5.most_common(10)
    fw, ff = [i for i, j in freq10], [j for i, j in freq10]
    plt.bar(fw, ff, color='blue')
    plt.title('10 most common words')
    plt.xlabel('Words')
    plt.ylabel('No. of occurrence')
    plt.show()
```

```
ï»¿Project Gutenberg's Alice's Adventures in Wonderland, by Lewis Carroll
```

```
This eBook is for the use of anyone anywhere at no cost and with
almost no restrictions whatsoever.  You may copy it, give it away or
re-use it under the terms of the Project Gutenberg License included
with this eBook or online at www.gutenberg.org


Title: Alice's Adventures in Wonderland

Author: Lewis Carroll

Posting Da
[('the', 1818), ('and', 940), ('to', 809), ('a', 690), ('of', 631), ('it', 610),
('she', 553), ('i', 545), ('you', 481), ('said', 462)]
The word "alice" appears 403 times.
```



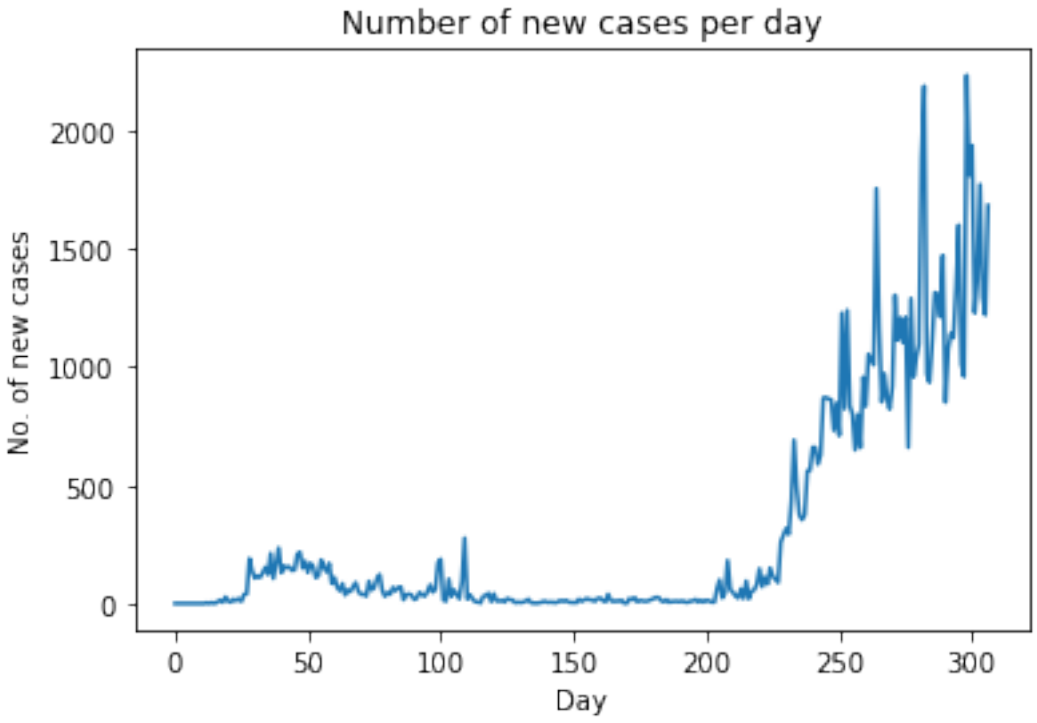
Hands-on 4: Analyze the Covid-19 data for Malaysia from 16 Feb 2020 to 18 Dec 2020

- Find out how many days with more than 1000 cases
- Find out how many days with more than 10 death

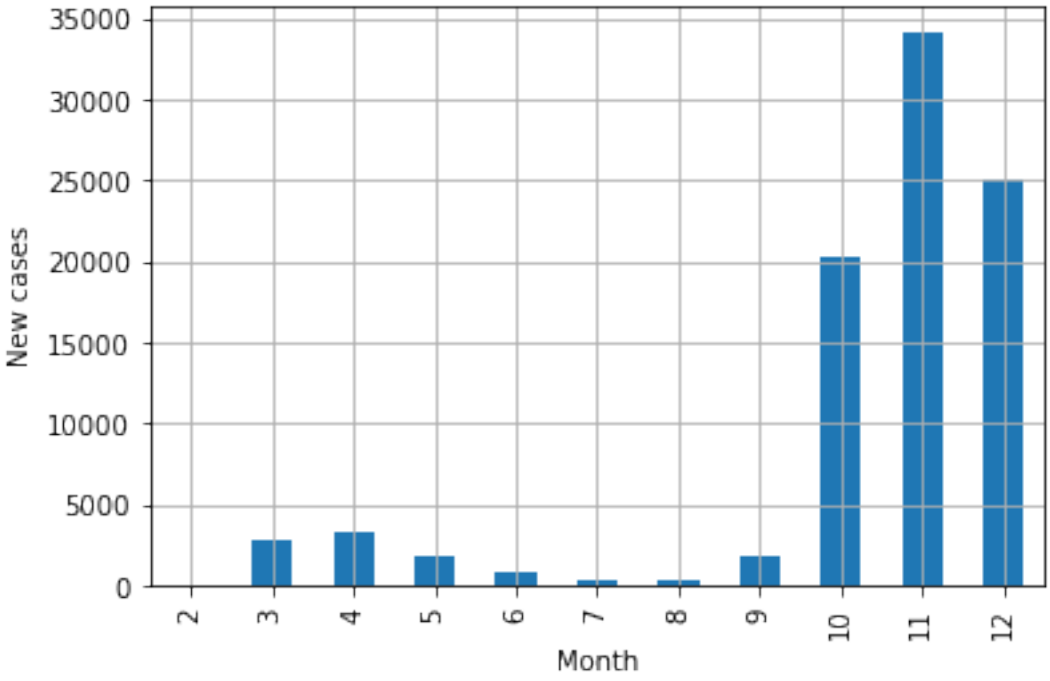
[5]:
```
# Hands-on 4
import pandas as pd
import matplotlib.pyplot as plt
```

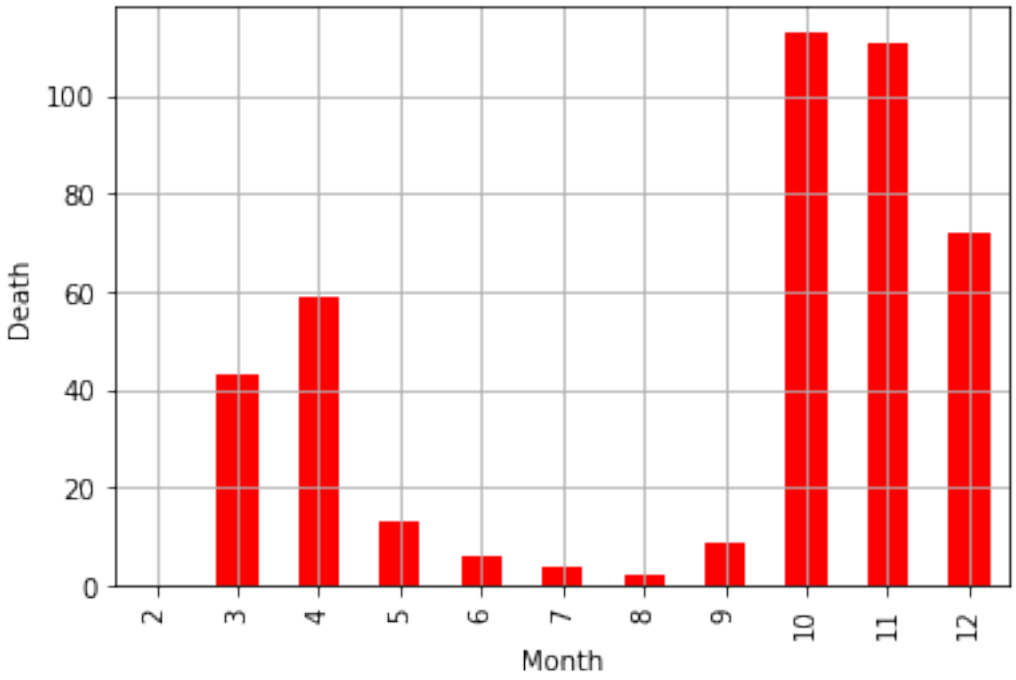
```
df = pd.read_csv('data/covid19_malaysia.csv')
df['date'] = pd.to_datetime(df['date'])

x = df.values[:, 0]
y = df.values[:, 1]

plt.plot(range(len(x)), y)
plt.xlabel('Day')
plt.ylabel('No. of new cases')
plt.title('Number of new cases per day')
plt.show()

max_cases = df['new_cases'] == df['new_cases'].max()
print(df[max_cases])

max_death = df['death'] == df['death'].max()
print(df[max_death])

cases_1000 = df['new_cases'] > 1000
print(f'Number of days with more than 1000 cases: {len(df[cases_1000])}')

death_10 = df['death'] > 10
print(f'Number of days with more than 10 deaths: {len(df[death_10])}')

# Plot bar chart for new cases per month
groups = df.groupby(df['date'].dt.month)
groups['new_cases'].sum().plot.bar()
plt.grid(True)
plt.xlabel('Month')
plt.ylabel('New cases')
plt.show()

# Plot bar chart for death per month
groups['death'].sum().plot.bar(color='red')
plt.grid(True)
plt.xlabel('Month')
plt.ylabel('Death')
plt.show()

# Plot bar chart for new cases for each day of week
groups = df.groupby(df['date'].dt.day_name())
groups['new_cases'].sum().plot.bar()
plt.grid(True)
plt.xlabel('Month')
plt.ylabel('New cases')
plt.show()
```

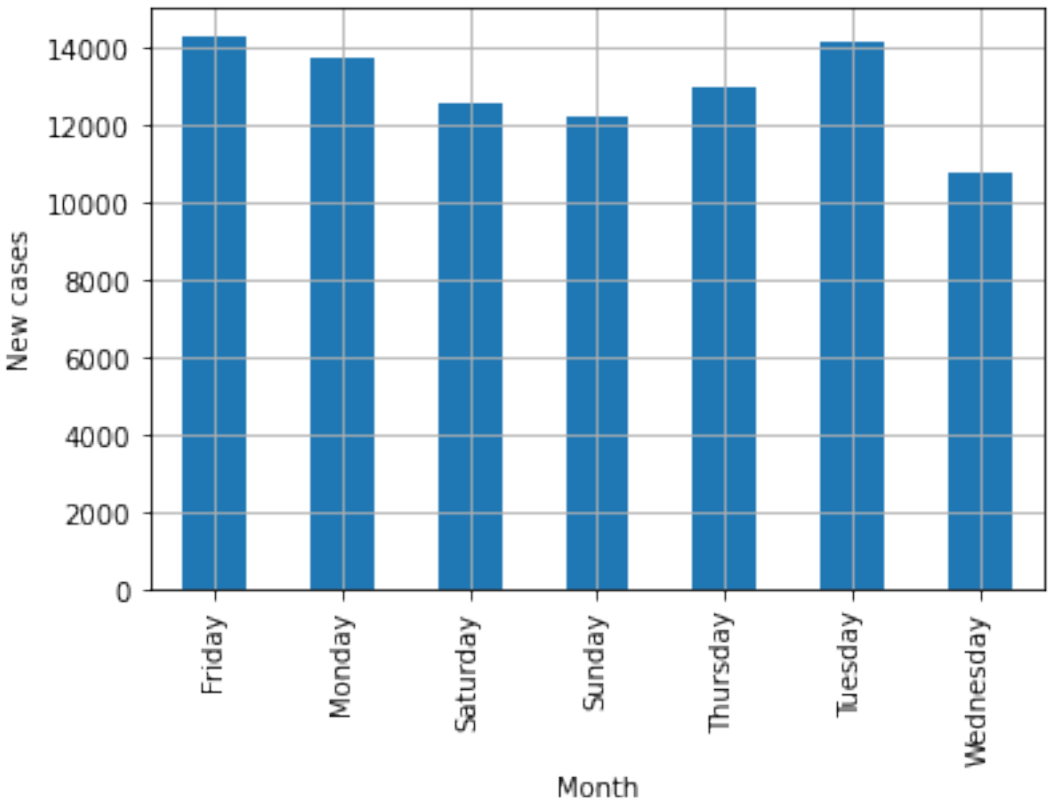
```
# Plot bar chart for death for each day of week
groups['death'].sum().plot.bar(color='red')
plt.grid(True)
plt.xlabel('Month')
plt.ylabel('Death')
plt.show()
```

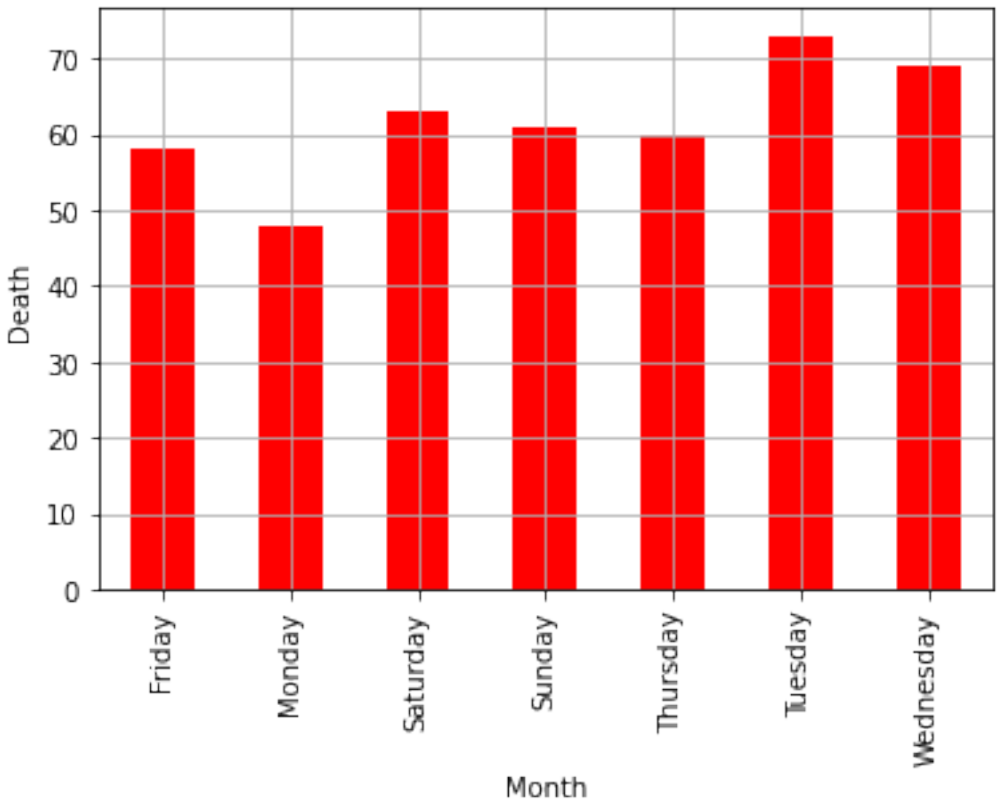



```
          date  new_cases  death
298 2020-12-10       2234      3
          date  new_cases  death
261 2020-11-03       1054     12
Number of days with more than 1000 cases: 37
Number of days with more than 10 deaths: 2
```


35000
30000
25000
20000
15000
10000
5000
0
New cases
2
3
4
5
6
7
8
9
10
11
12
Month


100
80
60
40
20
0
Death
2
3
4
5
6
7
8
9
10
11
12
Month


14000
12000
10000
8000
6000
4000
2000
0
New cases
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday
Month



Hands-on 5: Analyze the data of 10000 persons

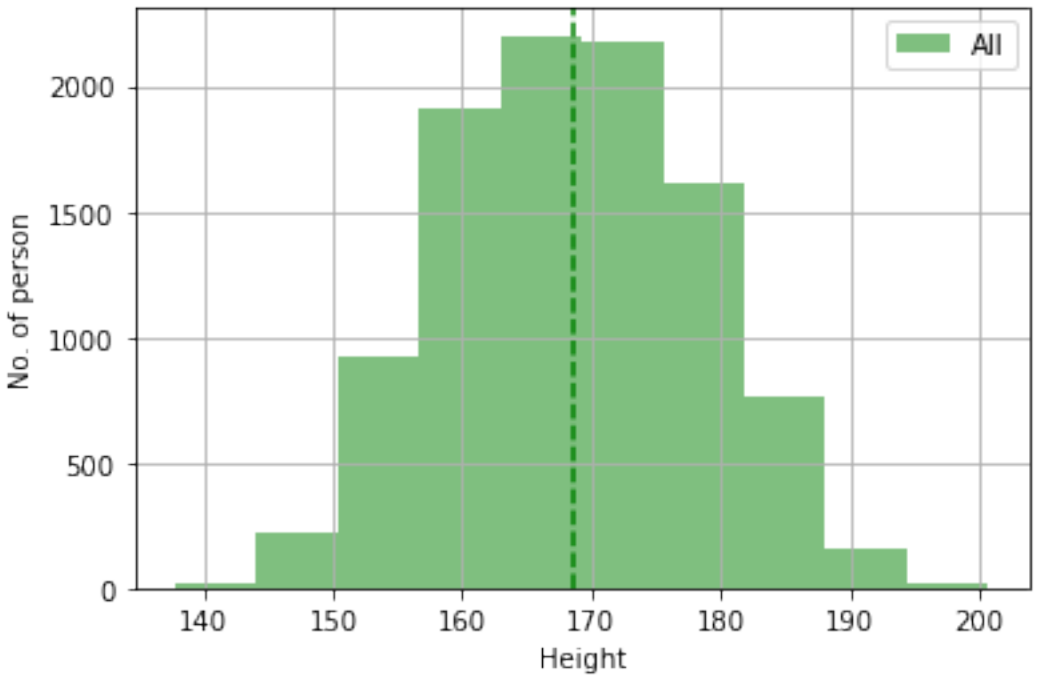
- Find the average, minimum and maximum height for all, males & females
- Find the average, minimum and maximum weight for all, males & females
- Find the number of males above average height
- Find the number of females below average weight

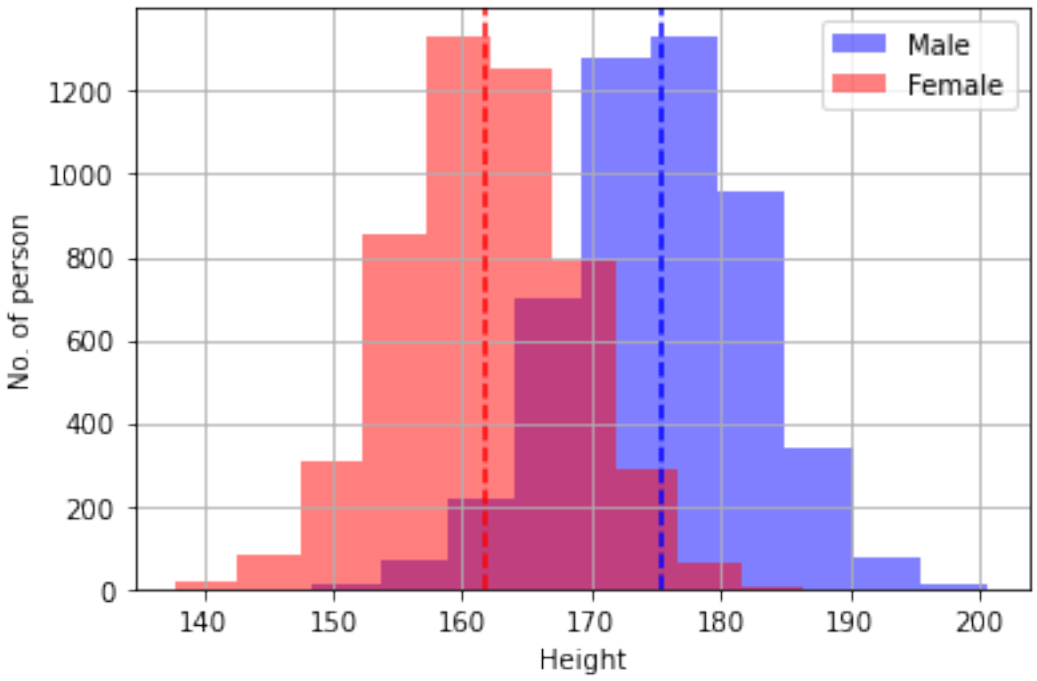
[6]:
```
# Hands-on 5
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_excel('data/genders_heights_weights.xlsx')
males = df[df['Gender']=='Male']
females = df[df['Gender']=='Female']

print(df.head())
print(df.describe())

groups = df.groupby('Gender')
print(groups.count())
print(groups.describe())
```

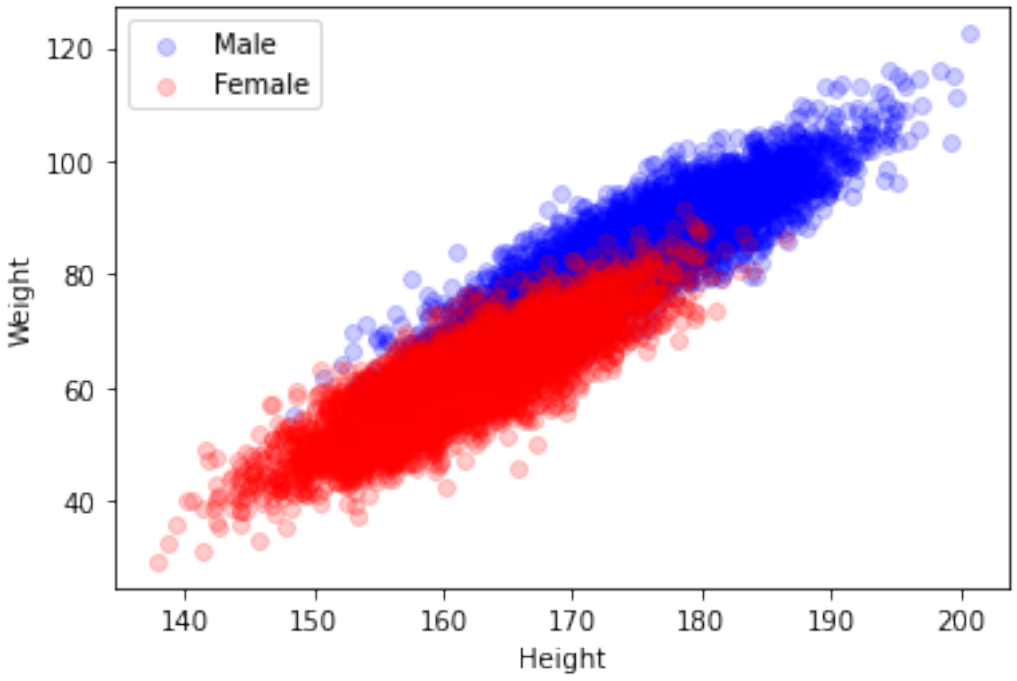
```
print(f'Number of males above average height: {sum(males["Height"] >␣
 ↪males["Height"].mean())}')
print(f'Number of females below average weight: {sum(males["Weight"] <␣
 ↪males["Weight"].mean())}')

# Plot histograms
df['Height'].hist(color='green', alpha=0.5, label='All')
plt.axvline(df['Height'].mean(), color='green', ls='--')
plt.xlabel('Height')
plt.ylabel('No. of person')
plt.legend(loc=0)
plt.show()

males['Height'].hist(color='blue', alpha=0.5, label='Male')
females['Height'].hist(color='red', alpha=0.5, label='Female')
plt.axvline(males['Height'].mean(), color='blue', ls='--')
plt.axvline(females['Height'].mean(), color='red', ls='--')
plt.xlabel('Height')
plt.ylabel('No. of person')
plt.legend(loc=1)
plt.show()

# Plot scatterplot
xm = males.values[:, 1]
ym = males.values[:, 2]
xf = females.values[:, 1]
yf = females.values[:, 2]
plt.scatter(xm, ym, c='blue', alpha=0.2, label='Male')
plt.scatter(xf, yf, c='red', alpha=0.2, label='Female')
plt.xlabel('Height')
plt.ylabel('Weight')
plt.legend(loc=0)
plt.show()
```

```
   Gender  Height  Weight
0  Female   162.5    67.3
1  Female   155.8    55.3
2  Female   168.7    58.7
3    Male   170.8    75.6
4  Female   159.8    59.7
             Height        Weight
count  10000.000000  10000.000000
mean     168.573940     73.228260
std        9.772842     14.563851
min      137.800000     29.300000
25%      161.300000     61.600000
```

```
50%      168.400000     73.100000
75%      175.700000     84.900000
max      200.700000    122.500000
        Height  Weight
Gender
Female    5000    5000
Male      5000    5000
        Height                                                    \
         count        mean       std    min    25%    50%    75%    max
Gender
Female  5000.0  161.82076  6.848886  137.8  157.2  161.9  166.5  186.4
Male    5000.0  175.32712  7.273214  148.4  170.6  175.3  180.3  200.7

        Weight
         count      mean      std   min   25%   50%   75%    max
Gender
Female  5000.0  61.62572  8.62890  29.3  55.8  61.7  67.5   91.7
Male    5000.0  84.83080  8.97242  51.2  78.9  84.8  90.9  122.5
Number of males above average height: 2493
Number of females below average weight: 2505
```




Male
Female
No. of person
Height
0
200
400
600
800
1000
1200
140
150
160
170
180
190
200


Male
Female
Weight
Height
40
60
80
100
120
140
150
160
170
180
190
200