

R Programming - Assignment 03 Solution

```
In [1]: import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.ticker import StrMethodFormatter

os.chdir('E:/WORK/DATA SCIENCE/02 - R Programming/ASSIGNMENTS/Assignment 03')
```

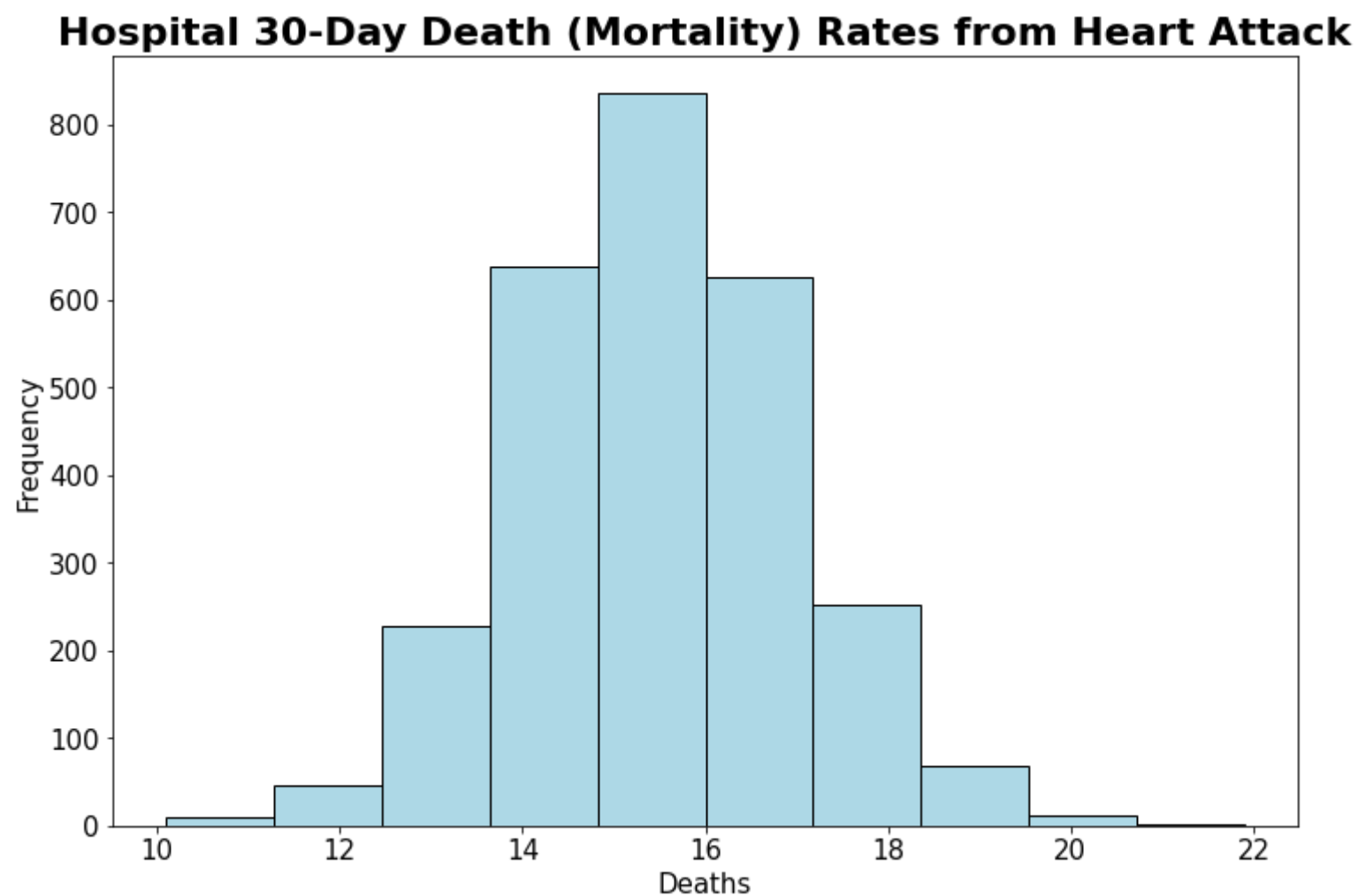
Part 1 - Plot the 30-day mortality rates for heart attack

```
In [2]: data = pd.read_csv("outcome-of-care-measures.csv")
df = data[data.columns[[10]]]
df.columns.values[0] = 'Heart attack'

zf = df[df['Heart attack'] != 'Not Available']
zf = zf.reset_index(drop=True)
zf = zf.astype(float)

zf.hist(figsize=(12,8), color='lightblue', zorder=2, grid=False, edgecolor = "black")

plt.xticks(size = 15)
plt.yticks(size = 15)
plt.xlabel('Deaths', size=15)
plt.ylabel('Frequency', size=15)
plt.title('Hospital 30-Day Death (Mortality) Rates from Heart Attack', size=22, fontweight="bold")
plt.show()
```



Part 2 - Finding the best hospital in a state

```
In [3]: def best(state, outcome):
data = pd.read_csv("outcome-of-care-measures.csv")
df = data[data.columns[[1,6,10,16,22]]]

df.columns.values[0] = 'Name'
df.columns.values[1] = 'State'
df.columns.values[2] = 'Heart attack'
df.columns.values[3] = 'Heart failure'
df.columns.values[4] = 'Pneumonia'

# Getting a List of all the states
states = df["State"].tolist()
unique_states = []
for i in states:
    if i not in unique_states:
        unique_states.append(i)

# List of all possible outcomes
outcomes = ["Heart attack", "Heart failure", "Pneumonia"]

if (outcome not in outcomes):
    return 'Invalid Outcome'
if (state not in unique_states):
    return "Invalid State"

# Dropping rows with value "Not Available"
zf = df[df[outcome] != 'Not Available']
zf = zf.reset_index(drop=True)

sf = zf[zf["State"] == state]      # Getting only the rows with our state value
sf = sf.reset_index(drop=True)
vf = sf[outcome].values.tolist()  # Convert the data frame to list
vf = [float(i) for i in vf]       # Convert the values in the list to float

# Getting the name of hospital in that state with lowest 30-day death rate
min_value = min(vf)
row_num = sf[sf[outcome]==str(min_value)].index.values
f = (sf["Name"].iloc[row_num]).tolist()

return f[0]
```

Testing of best() Function

```
In [4]: print(best("TX", "Heart attack"))      # CYPRESS FAIRBANKS MEDICAL CENTER
print(best("TX", "Heart failure"))             # FORT DUNCAN MEDICAL CENTER
print(best("MD", "Heart attack"))              # JOHNS HOPKINS HOSPITAL, THE
print(best("MD", "Pneumonia"))                 # GREATER BALTIMORE MEDICAL CENTER
print(best("BB", "Heart attack"))              # Error in best("BB", "heart attack") : invalid state
print(best("NY", "Hert attack"))               # Error in best("NY", "hert attack") : invalid outcome

CYPRESS FAIRBANKS MEDICAL CENTER
FORT DUNCAN MEDICAL CENTER
JOHNS HOPKINS HOSPITAL, THE
GREATER BALTIMORE MEDICAL CENTER
Invalid State
Invalid Outcome
```

Part 3 - Ranking hospitals by outcome in a state

```
In [5]: def rankhospital(state, outcome, num):
data = pd.read_csv("outcome-of-care-measures.csv")
df = data[data.columns[[1,6,10,16,22]]]

df.columns.values[0] = 'Name'
df.columns.values[1] = 'State'
df.columns.values[2] = 'Heart attack'
df.columns.values[3] = 'Heart failure'
df.columns.values[4] = 'Pneumonia'

# Getting a List of all the states
states = df["State"].tolist()
unique_states = []
for i in states:
    if i not in unique_states:
        unique_states.append(i)

# List of all possible outcomes
outcomes = ["Heart attack", "Heart failure", "Pneumonia"]

if (outcome not in outcomes) or (state not in unique_states) or (num!="best" and num!="worst" and float(num)%1!=
0):
    return 'Invalid Input'

# Dropping rows with value "Not Available"
zf = df[df[outcome] != 'Not Available']
zf = zf.reset_index(drop=True)

# Getting only the rows with our state value
sf = zf[zf["State"] == state]
sf = sf.reset_index(drop=True)

if num!="best" and num!="worst" and num>len(sf):
    return "NA"
if isinstance(num, int):
    sf[outcome] = sf[outcome].astype(float)    # Converting the column values to float when num is a numeric value

# Sort the data frame by first names and then outcome values
of = sf.sort_values(by=["Name"], ascending=[True])
of = of.sort_values(by=[outcome], ascending=[True])
of = of.reset_index(drop=True)

vf = of[outcome].values.tolist()    # Convert the data frame to list
vf = [float(i) for i in vf]         # Convert the values in the list to float

# Getting the index of the name associated with the outcome
if num == "best":
    min_value = min(vf)
    row_num = of[of[outcome]==str(min_value)].index.values
elif num == "worst":
    max_value = max(vf)
    row_num = of[of[outcome]==str(max_value)].index.values
else:
    row_num = [(num-1)]

# Return name of the hospital
f = (of["Name"].iloc[row_num]).tolist()
return f[0]
```

Testing rankhospital() Function

```
In [6]: print(rankhospital("NC", "Heart attack", "worst"))    # WAYNE MEMORIAL HOSPITAL
print(rankhospital("TX", "Heart failure", 4))                # DETAR HOSPITAL NAVARRO
print(rankhospital("MD", "Heart attack", "worst"))           # HARFORD MEMORIAL HOSPITAL
print(rankhospital("MN", "Heart attack", 5000))              # NA
```

```
WAYNE MEMORIAL HOSPITAL
DEAR HOSPITAL NAVARRO
HARFORD MEMORIAL HOSPITAL
NA
```

Part 4 - Ranking hospitals in all states

```

In [7]: def rankall(outcome, num = "best"):
    data = pd.read_csv("outcome-of-care-measures.csv")
    df = data[data.columns[[1,6,10,16,22]]]

    df.columns.values[0] = 'Name'
    df.columns.values[1] = 'State'
    df.columns.values[2] = 'Heart attack'
    df.columns.values[3] = 'Heart failure'
    df.columns.values[4] = 'Pneumonia'

    # Getting a List of all the states
    states = df["State"].tolist()
    unique_states = []
    for i in states:
        if i not in unique_states:
            unique_states.append(i)
    unique_states = sorted(unique_states)

    # List of all possible outcomes
    outcomes = ["Heart attack", "Heart failure", "Pneumonia"]
    if (outcome not in outcomes) or (num!="best" and num!="worst" and float(num)%1!=0):
        return 'Invalid Input'

    # Dropping rows with value "Not Available"
    zf = df[df[outcome] != 'Not Available']
    zf = zf.reset_index(drop=True)

    dataframe = pd.DataFrame(None, columns = ["Hospital", "State"]) # New data frame

    for j in unique_states:
        sf = zf[zf["State"] == j] # Getting only the rows with our state value
        sf = sf.reset_index(drop=True)

        if isinstance(num, int):
            sf[outcome] = sf[outcome].astype(float) # Converting the column values to float when num is a numeric va
Lue

        if num!="best" and num!="worst" and num>len(sf):
            new_row = {'Hospital':'NA', 'State':j}
            dataframe = dataframe.append(new_row, ignore_index=True)
            continue

        # Sort the data frame by first names and then outcome values
        of = sf.sort_values(by=["Name"], ascending=[True])
        of = of.sort_values(by=[outcome], ascending=[True])
        of = of.reset_index(drop=True)

        vf = of[outcome].values.tolist() # Convert the data frame to List
        vf = [float(i) for i in vf]

        if num == "best":
            min_value = min(vf)
            row_num = of[of[outcome]==str(min_value)].index.values
        elif num == "worst":
            max_value = max(vf)
            row_num = of[of[outcome]==str(max_value)].index.values
        else:
            row_num = [(num-1)]

        # Return name of the hospital
        f = (of["Name"].iloc[row_num]).tolist()

        new_row = {'Hospital':f[0], 'State':j}
        dataframe = dataframe.append(new_row, ignore_index=True)

        f = None
        sf = None

    return dataframe

```

Testing of rankall() function

```
In [8]: t1 = rankall("Heart attack", 20)
t2 = rankall("Pneumonia", "worst")
t3 = rankall("Heart failure")

print(t1.head(10))
print('')
print(t2.tail(3))
print('')
print(t3.tail(10))
```

	Hospital	State
0		NA AK
1	D W MCMILLAN MEMORIAL HOSPITAL	AL
2	ARKANSAS METHODIST MEDICAL CENTER	AR
3	JOHN C LINCOLN DEER VALLEY HOSPITAL	AZ
4	SHERMAN OAKS HOSPITAL	CA
5	SKY RIDGE MEDICAL CENTER	CO
6	MIDSTATE MEDICAL CENTER	CT
7		NA DC
8		NA DE
9	DOCTORS HOSPITAL INC	FL

	Hospital	State
51	MAYO CLINIC HEALTH SYSTEM - NORTHLAND, INC	WI
52	PLATEAU MEDICAL CENTER	WV
53	NORTH BIG HORN HOSPITAL DISTRICT	WY

	Hospital	State
44	WELLMONT HAWKINS COUNTY MEMORIAL HOSPITAL	TN
45	FORT DUNCAN MEDICAL CENTER	TX
46	VA SALT LAKE CITY HEALTHCARE - GEORGE E. WAHLE...	UT
47	SENTARA POTOMAC HOSPITAL	VA
48	GOV JUAN F LUIS HOSPITAL & MEDICAL CTR	VI
49	SPRINGFIELD HOSPITAL	VT
50	HARBORVIEW MEDICAL CENTER	WA
51	AURORA ST LUKES MEDICAL CENTER	WI
52	FAIRMONT GENERAL HOSPITAL	WV
53	CHEYENNE VA MEDICAL CENTER	WY