

# ASSIGNMENT 6

## Task 1: Demand-Supply Mismatch Analysis

**Objective:** Identify zones and regional zones with the highest mismatch between demand and supply.

mapper1.py:

```
#!/usr/bin/python3
```

```
"""mapper1.py"""
```

```
import sys
```

```
for line in sys.stdin:
```

```
    line = line.strip()
```

```
    if line:
```

```
        columns = line.split(',')
```

```
        if columns:
```

```
            zone = columns[4].strip()
```

```
            wh_regional_zone = columns[5].strip()
```

```
            product_wg_ton = columns[-1].strip()
```

```
            print('%s,%s,%s' % (zone, wh_regional_zone, product_wg_ton))
```

reducer1.py:

```
#!/usr/bin/python3
```

```
"""reducer1.py"""
```

```
import sys
```

```
data_dict = {}
```

```
#count_dict = {}
```

```
for line in sys.stdin:
```

```
    line = line.strip()
```

```
    zone, WH_regional_zone, product_wg_ton = line.split(',', 2)
```

```
    try:
```

```
        product_wg_ton = float(product_wg_ton)
```

```
    except ValueError:
```

```
        continue
```

```
    key = (zone, WH_regional_zone)
```

```
    if key in data_dict:
```

```
        data_dict[key] += product_wg_ton
```

```
#    count_dict[key] += 1
```

```
    else:
```

```
        data_dict[key] = product_wg_ton
```

```
#    count_dict[key] = 1
```

```
#for key in data_dict.keys():
#  data_dict[key] = data_dict[key]/count_dict[key]

for key, value in data_dict.items():
    print(f'Zone: {key[0]}, RegionalZone: {key[1]}, TotalSupply: {value}')
#  print("%s,%s" % (key, value))
```

```
hadoop@hadoop-VirtualBox:~/assignment6$ hadoop jar /usr/local/hadoop/share/hadoop/tools
lib/hadoop-streaming-2.7.6.jar -file /home/hadoop/assignment6/mapper1.py -mapper mapper
.py -file /home/hadoop/assignment6/reducer1.py -reducer reducer1.py -input /assignment6
FMCG_data.csv -output /assignment6/output1
packageJobJar: [/home/hadoop/assignment6/mapper1.py, /home/hadoop/assignment6/reducer1
y] [] /tmp/streamjob5762907958362666237.jar tmpDir=null
hadoop@hadoop-VirtualBox:~/assignment6$ hdfs dfs -cat /assignment6/output1/*
Zone: East, RegionalZone: Zone 1, TotalSupply: 872338.0
Zone: East, RegionalZone: Zone 3, TotalSupply: 2526684.0
Zone: East, RegionalZone: Zone 4, TotalSupply: 3306171.0
Zone: East, RegionalZone: Zone 5, TotalSupply: 1768074.0
Zone: East, RegionalZone: Zone 6, TotalSupply: 1274236.0
Zone: North, RegionalZone: Zone 1, TotalSupply: 18466131.0
Zone: North, RegionalZone: Zone 2, TotalSupply: 18966332.0
Zone: North, RegionalZone: Zone 3, TotalSupply: 21335735.0
Zone: North, RegionalZone: Zone 4, TotalSupply: 26254519.0
Zone: North, RegionalZone: Zone 5, TotalSupply: 42893115.0
Zone: North, RegionalZone: Zone 6, TotalSupply: 100249991.0
Zone: South, RegionalZone: Zone 1, TotalSupply: 14682866.0
Zone: South, RegionalZone: Zone 2, TotalSupply: 32467899.0
Zone: South, RegionalZone: Zone 3, TotalSupply: 18810119.0
Zone: South, RegionalZone: Zone 4, TotalSupply: 19230670.0
Zone: South, RegionalZone: Zone 5, TotalSupply: 24113697.0
Zone: South, RegionalZone: Zone 6, TotalSupply: 30235650.0
Zone: West, RegionalZone: Zone 1, TotalSupply: 10638197.0
Zone: West, RegionalZone: Zone 2, TotalSupply: 15146537.0
Zone: West, RegionalZone: Zone 3, TotalSupply: 20617692.0
Zone: West, RegionalZone: Zone 4, TotalSupply: 43804669.0
Zone: West, RegionalZone: Zone 5, TotalSupply: 32242727.0
Zone: West, RegionalZone: Zone 6, TotalSupply: 52661774.0
```

## **Task 2: Warehouse Refill Frequency Correlation**

**Objective:** Determine the correlation between warehouse capacity and refill frequency.

mapper2.py:

```
#!/usr/bin/python3
"""mapper2.py"""
```

```
import sys
```

```
for line in sys.stdin:
    line = line.strip()
    if line:
        columns = line.split(',')
        if columns:
            wh_capacity_size = columns[3].strip()
```

```

num_refill_req_l3m = columns[6].strip()
print('%s,%s' % (wh_capacity_size, num_refill_req_l3m))

reducer2.py:
#!/usr/bin/python3
"""reducer2.py"""

import sys
import numpy as np

data_dict = {}
count_dict = {}
for line in sys.stdin:
    line = line.strip()
    wh_capacity_size, num_refill_req = line.split(',', 1)

    try:
        num_refill_req = int(num_refill_req)
    except ValueError:
        continue

    wh_capacity_value = 1 if wh_capacity_size=='Small' else 2 if wh_capacity_size=='Mid' else
3 if wh_capacity_size=='Large' else 4
    # wh_capacity_value_list.append(wh_capacity_value)
    # num_refill_req_list.append(num_refill_req)
    if wh_capacity_value in data_dict:
        data_dict[wh_capacity_value] += num_refill_req
        count_dict[wh_capacity_value] += 1
    else:
        data_dict[wh_capacity_value] = num_refill_req
        count_dict[wh_capacity_value] = 1

for key in data_dict.keys():
    data_dict[key] = data_dict[key]/count_dict[key]

wh_capacity_value_list = []
num_refill_req_list = []

correlation_value = np.corrcoef(list(data_dict.keys()), list(data_dict.values()))[0,1]
print(f"Correlation between warehouse capacity and number of refills is:
{round(correlation_value, 5)}")

```

```
hadoop@hadoop-VirtualBox:~/assignment6$ hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.6.jar -file /home/hadoop/assignment6/mapper2.py -mapper mapper2.py -file /home/hadoop/assignment6/reducer2.py -reducer reducer2.py -input /assignment6/output2 -output /assignment6/output2
packageJobJar: [/home/hadoop/assignment6/mapper2.py, /home/hadoop/assignment6/reducer2.py] [] /tmp/streamjob751240838340007030.jar tmpDir=null
hadoop@hadoop-VirtualBox:~/assignment6$ hdfs dfs -cat /assignment6/output2/*
Correlation between warehouse capacity and number of refills is: -0.73499
hadoop@hadoop-VirtualBox:~/assignment6$
```

**Insight:** Correlation coefficient between warehouse capacity and average number of refill requests is 0.73499, thus there is a positive correlation

### **Task 3. Transport Issue Impact Analysis**

**Objective:** Analyse the impact of transport issues on warehouse supply efficiency.

mapper3.py:

```
#!/usr/bin/python3
"""mapper3.py"""
```

```
import sys
```

```
for line in sys.stdin:
    line = line.strip()
    if line:
        columns = line.split(',')
        if columns:
            transport_issue_l1y = columns[7].strip()
            product_wg_ton = columns[23].strip()
            print('%s,%s' % (transport_issue_l1y, product_wg_ton))
```

recucer3.py:

```
#!/usr/bin/python3
"""reducer3.py"""
```

```
import sys
```

```
import numpy as np
```

```
data_dict = {}
##count_dict = {}
for line in sys.stdin:
    line = line.strip()
    transport_issues, product_wg_ton = line.split(',', 1)

    try:
        transport_issues = int(transport_issues)
        product_wg_ton = float(product_wg_ton)
    except ValueError:
        continue

    if transport_issues in data_dict:
        data_dict[transport_issues] += [product_wg_ton]
```

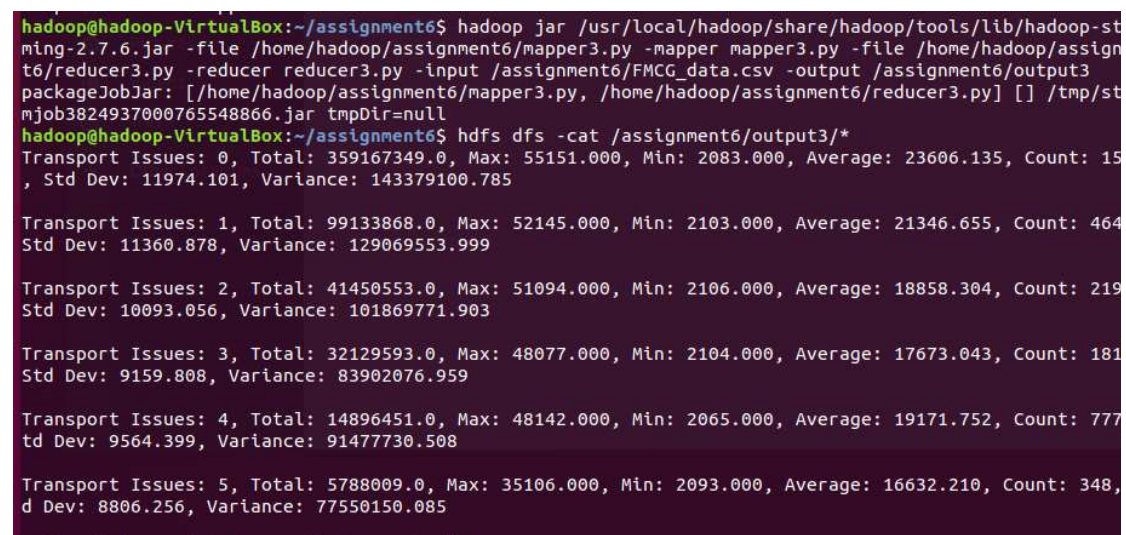
```

##     count_dict[transport_issues] += 1
    else:
        data_dict[transport_issues] = [product_wg_ton]
##     count_dict[transport_issues] = 1

#for key in data_dict.keys():
#  data_dict[key] = round(data_dict[key]/count_dict[key], 3)

data_dict = sorted(data_dict.items(), key=lambda x:x[0])
for (key, value) in data_dict:
    value = np.array(value)
    total = np.sum(value)
    max_value = np.max(value)
    min_value = np.min(value)
    average = np.mean(value)
    count = len(value)
    std_dev = np.std(value)
    variance = np.var(value)
    print(f"Transport Issues: {key}, Total: {total}, Max: {max_value:.3f}, Min: {min_value:.3f},
Average: {average:.3f}, Count: {count}, Std Dev: {std_dev:.3f}, Variance: {variance:.3f}\n")

```



```

hadoop@hadoop-VirtualBox:~/assignment6$ hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.6.jar -file /home/hadoop/assignment6/mapper3.py -mapper mapper3.py -file /home/hadoop/assignment6/reducer3.py -reducer reducer3.py -input /assignment6/FMCG_data.csv -output /assignment6/output3
packageJobJar: [/home/hadoop/assignment6/mapper3.py, /home/hadoop/assignment6/reducer3.py] [] /tmp/stmjob3824937000765548866.jar tmpDir=null
hadoop@hadoop-VirtualBox:~/assignment6$ hdfs dfs -cat /assignment6/output3/*
Transport Issues: 0, Total: 359167349.0, Max: 55151.000, Min: 2083.000, Average: 23606.135, Count: 15
, Std Dev: 11974.101, Variance: 143379100.785

Transport Issues: 1, Total: 99133868.0, Max: 52145.000, Min: 2103.000, Average: 21346.655, Count: 464
Std Dev: 11360.878, Variance: 129069553.999

Transport Issues: 2, Total: 41450553.0, Max: 51094.000, Min: 2106.000, Average: 18858.304, Count: 219
Std Dev: 10093.056, Variance: 101869771.903

Transport Issues: 3, Total: 32129593.0, Max: 48077.000, Min: 2104.000, Average: 17673.043, Count: 181
Std Dev: 9159.808, Variance: 83902076.959

Transport Issues: 4, Total: 14896451.0, Max: 48142.000, Min: 2065.000, Average: 19171.752, Count: 777
Std Dev: 9564.399, Variance: 91477730.508

Transport Issues: 5, Total: 5788009.0, Max: 35106.000, Min: 2093.000, Average: 16632.210, Count: 348,
Std Dev: 8806.256, Variance: 77550150.085

```

**Insight:** Comparing Transport Issues with average product weight supplied, we see there is an *inverse relation* - ie, as the *number of transport issues increase*, *average weight of products supplied decreases*.

#### Task 4. Storage Issue Analysis

**Objective:** Evaluate the impact of storage issues on warehouse performance.

```

mapper4.py:
#!/usr/bin/python3
"""mapper4.py"""

```

```

import sys

```

```

for line in sys.stdin:
    line = line.strip()
    if line:
        columns = line.split(',')
        if columns:
            storage_issues_reported = columns[18].strip()
            product_wg_ton = columns[23].strip()
            print('%s,%s' % (storage_issues_reported, product_wg_ton))

```

reducer4.py:  
#!/usr/bin/python3  
"""reducer4.py"""

```

import sys
import numpy as np

```

```

data_dict = {}
count_dict = {}
for line in sys.stdin:
    line = line.strip()
    storage_issues_reported, product_wg_ton = line.split(',', 1)

```

```

    try:
        storage_issues_reported = int(storage_issues_reported)
        product_wg_ton = float(product_wg_ton)
    except ValueError:
        continue

```

```

    if storage_issues_reported in data_dict:
        data_dict[storage_issues_reported] += [product_wg_ton]
        count_dict[storage_issues_reported] += 1
    else:
        data_dict[storage_issues_reported] = [product_wg_ton]
        count_dict[storage_issues_reported] = 1

```

```

#for key in data_dict.keys():
#    data_dict[key] = round(data_dict[key]/count_dict[key], 3)
#
#for key, value in data_dict.items():
#    print("%s,%s" % (key, value))
data_dict = sorted(data_dict.items(), key=lambda x:x[0])
for (key, value) in data_dict:
    value = np.array(value)
    total = np.sum(value)
    max_value = np.max(value)
    min_value = np.min(value)
    average = np.mean(value)
    count = len(value)
    std_dev = np.std(value)

```



```

variance = np.var(value)
print(f"Storage Issues: {key}, Total: {total}, Max: {max_value:.3f}, Min: {min_value:.3f},
Average: {average:.3f}, Count: {count}, Std Dev: {std_dev:.3f}, Variance: {variance:.3f}")

```

```

hadoop@hadoop-VirtualBox:~/assignment6$ hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.6.jar -file /home/hadoop/assignment6/mapper4.py -mapper mapper4.py -file /home/hadoop/assignment6/reducer4.py -reducer reducer4.py -input /assignment6/FMCG_data.csv -output /assignment6/output4
packageJobJar: [/home/hadoop/assignment6/mapper4.py, /home/hadoop/assignment6/reducer4.py] [] /tmp/streamjob8962293787686474835.jar tmpDir=null
hadoop@hadoop-VirtualBox:~/assignment6$ hdfs dfs -cat /assignment6/output4/*
Storage Issues: 0, Total: 4930869.0, Max: 14149.000, Min: 2065.000, Average: 5430.472, Count: 908, Std Dev: 2687.874, Variance: 7224666.058
Storage Issues: 4, Total: 5602095.0, Max: 6151.000, Min: 4055.000, Average: 5182.327, Count: 1081, Std Dev: 537.922, Variance: 289360.185
Storage Issues: 5, Total: 8645439.0, Max: 8148.000, Min: 5055.000, Average: 6399.289, Count: 1351, Std Dev: 591.158, Variance: 349467.584
Storage Issues: 6, Total: 8158616.0, Max: 9149.000, Min: 6058.000, Average: 7725.962, Count: 1056, Std Dev: 646.817, Variance: 418371.756
Storage Issues: 7, Total: 4393171.0, Max: 11102.000, Min: 8055.000, Average: 8947.395, Count: 491, Std Dev: 683.976, Variance: 467823.392
Storage Issues: 8, Total: 4206084.0, Max: 12136.000, Min: 9055.000, Average: 10149.468, Count: 406, Std Dev: 737.380, Variance: 543729.405
Storage Issues: 9, Total: 9165459.0, Max: 14142.000, Min: 10055.000, Average: 11646.072, Count: 787, Std Dev: 871.812, Variance: 760055.892
Storage Issues: 10, Total: 8259859.0, Max: 15150.000, Min: 11056.000, Average: 12966.812, Count: 637, Std Dev: 975.407, Variance: 951417.974
Storage Issues: 11, Total: 12270859.0, Max: 17151.000, Min: 12055.000, Average: 14153.240, Count: 867, Std Dev: 1015.574, Variance: 1031389.599
Storage Issues: 12, Total: 11436927.0, Max: 18150.000, Min: 13060.000, Average: 15476.221, Count: 739, Std Dev: 1120.308, Variance: 1255090.028
Storage Issues: 13, Total: 12163798.0, Max: 20150.000, Min: 14113.000, Average: 16754.543, Count: 726, Std Dev: 1136.632, Variance: 1291931.474
Storage Issues: 14, Total: 14535116.0, Max: 21146.000, Min: 16055.000, Average: 17704.161, Count: 821, Std Dev: 1239.221, Variance: 1535667.614
Storage Issues: 15, Total: 17281171.0, Max: 23149.000, Min: 17055.000, Average: 19032.127, Count: 908, Std Dev: 1307.804, Variance: 1710351.307
Storage Issues: 16, Total: 19200310.0, Max: 24151.000, Min: 18055.000, Average: 20469.414, Count: 938, Std Dev: 1332.039, Variance: 1774326.643
Storage Issues: 17, Total: 16416984.0, Max: 26125.000, Min: 19055.000, Average: 21918.537, Count: 749, Std Dev: 1408.354, Variance: 1983461.247
Storage Issues: 18, Total: 24289887.0, Max: 27133.000, Min: 20055.000, Average: 22700.829, Count: 1070, Std Dev: 1490.871, Variance: 2222697.531
Storage Issues: 19, Total: 24569176.0, Max: 29091.000, Min: 21057.000, Average: 24040.290, Count: 1022, Std Dev: 1525.795, Variance: 2328049.351
Storage Issues: 20, Total: 27006058.0, Max: 30108.000, Min: 23055.000, Average: 25357.801, Count: 1065, Std Dev: 1554.632, Variance: 2416880.477
Storage Issues: 21, Total: 18581712.0, Max: 31149.000, Min: 24055.000, Average: 27047.616, Count: 687, Std Dev: 1767.348, Variance: 3123517.285
Storage Issues: 22, Total: 25472459.0, Max: 32138.000, Min: 25056.000, Average: 27930.328, Count: 912, Std Dev: 1697.084, Variance: 2882130.988
Storage Issues: 23, Total: 26797528.0, Max: 33145.000, Min: 26058.000, Average: 29223.040, Count: 917, Std Dev: 1755.331, Variance: 3081186.782
Storage Issues: 24, Total: 42904667.0, Max: 34151.000, Min: 27055.000, Average: 30129.682, Count: 1424, Std Dev: 1715.050, Variance: 2941396.862
Storage Issues: 25, Total: 39461458.0, Max: 36149.000, Min: 28055.000, Average: 31268.984, Count: 1262, Std Dev: 1674.469, Variance: 2803845.321
Storage Issues: 26, Total: 19958755.0, Max: 37148.000, Min: 29075.000, Average: 32772.997, Count: 609, Std Dev: 1924.221, Variance: 3702625.974
Storage Issues: 27, Total: 19849883.0, Max: 39132.000, Min: 31055.000, Average: 33931.424, Count: 585, Std Dev: 1998.050, Variance: 3992205.277
Storage Issues: 28, Total: 12281089.0, Max: 40150.000, Min: 33081.000, Average: 36550.860, Count: 336, Std Dev: 1842.360, Variance: 3394289.483
Storage Issues: 29, Total: 12068423.0, Max: 41150.000, Min: 34055.000, Average: 37596.333, Count: 321, Std Dev: 1897.569, Variance: 3600768.378
Storage Issues: 30, Total: 13109614.0, Max: 43142.000, Min: 35065.000, Average: 38900.932, Count: 337, Std Dev: 2076.622, Variance: 4312358.978
Storage Issues: 31, Total: 11698085.0, Max: 44143.000, Min: 37055.000, Average: 40477.803, Count: 289, Std Dev: 2038.498, Variance: 4155475.902
Storage Issues: 32, Total: 12244881.0, Max: 46148.000, Min: 38055.000, Average: 41367.841, Count: 296, Std Dev: 2187.246, Variance: 4784043.532
Storage Issues: 33, Total: 12650336.0, Max: 47151.000, Min: 39055.000, Average: 42882.495, Count: 295, Std Dev: 2241.955, Variance: 5026361.775
Storage Issues: 34, Total: 12750651.0, Max: 48148.000, Min: 40071.000, Average: 44273.094, Count: 288, Std Dev: 2324.740, Variance: 5404413.981
Storage Issues: 35, Total: 8440349.0, Max: 50150.000, Min: 43057.000, Average: 46631.762, Count: 181, Std Dev: 2086.902, Variance: 4355158.026
Storage Issues: 36, Total: 7722257.0, Max: 51148.000, Min: 44059.000, Average: 47964.329, Count: 161, Std Dev: 2129.223, Variance: 4533588.569
Storage Issues: 37, Total: 6921399.0, Max: 53150.000, Min: 45126.000, Average: 49087.936, Count: 141, Std Dev: 2088.861, Variance: 4363339.294

```

**Insight:** Comparing Storage Issues with product weight supplied, we see that even when there is an increase in storage issues, the warehouses have higher average and total product weight supplied. Thus, there is *a positive coreelation between Storage Issue and Total Product Weight* [also with Average Product Weight].

Also, maximum and minimum product weight supplied is directly proportional to storage issues.

This suggests that warehouses that ships more products will have more storage issues compared to warehouses that ships less products.