```
import openpyxl
import pandas as pd
import numpy as np

# Load the Excel file
workbook = openpyxl.load_workbook("Temp - VBA.xlsm")

# Read the "Hi" worksheet
hi_sheet = workbook["Hi"]
company_col = pd.DataFrame([cell.value for cell in hi_sheet["A"]], columns=["Company"]).dro
time_col = pd.DataFrame([cell.value for cell in hi_sheet["N"]], columns=["TimeSet"]).dropna

print(company_col)
print(time_col)
```

```
  Company
0     AAA
1     BBB
2     CCC
     TimeSet
0  2024-06-20
1  2024-07-20
2  2024-08-20
3  2024-09-20
4  2024-10-20
5  2024-11-20
6  2024-12-20
7  2025-01-20
8  2025-02-20
9  2025-03-20
10 2025-04-20
11 2025-05-20
```

```
# Read the "Data" worksheet
data_sheet = workbook["Data"]
data2_sheet = workbook["Data2"]

# Initialize the benchmark dictionary
benchmark = {}

# Iterate over the rows in the benchmark data range
for row in data2_sheet.iter_rows(values_only=True):
    item_name = row[0]
    test_b_value = row[1]
    test_c_value = row[2]
    benchmark[item_name] = (test_b_value, test_c_value)

if "TestA" in benchmark:
    del benchmark["TestA"]

benchmark
```

```
{'apple': (10000, 10000),
 'banana': (10000, 20000),
 'gold': (10000, 30000),
 'king': (10000, 2000),
 'kite': (10000, 800),
 'yellow': (10000, 500),
 'green': (10000, 6000),
 'purple': (10000, 7000),
 'queen': (10000, 4000)}
```

```
# Create initial empty dataframes

def clear_charts():
    global more_item, less_item, more_item_value, less_item_value
    column_names = time_col["TimeSet"].tolist()
    more_item = pd.DataFrame(columns=["ItemName"] + column_names)
    less_item = pd.DataFrame(columns=["ItemName"] + column_names)
    more_item_value = pd.DataFrame(columns=["ItemName"] + column_names)
    less_item_value = pd.DataFrame(columns=["ItemName"] + column_names)

clear_charts()
```

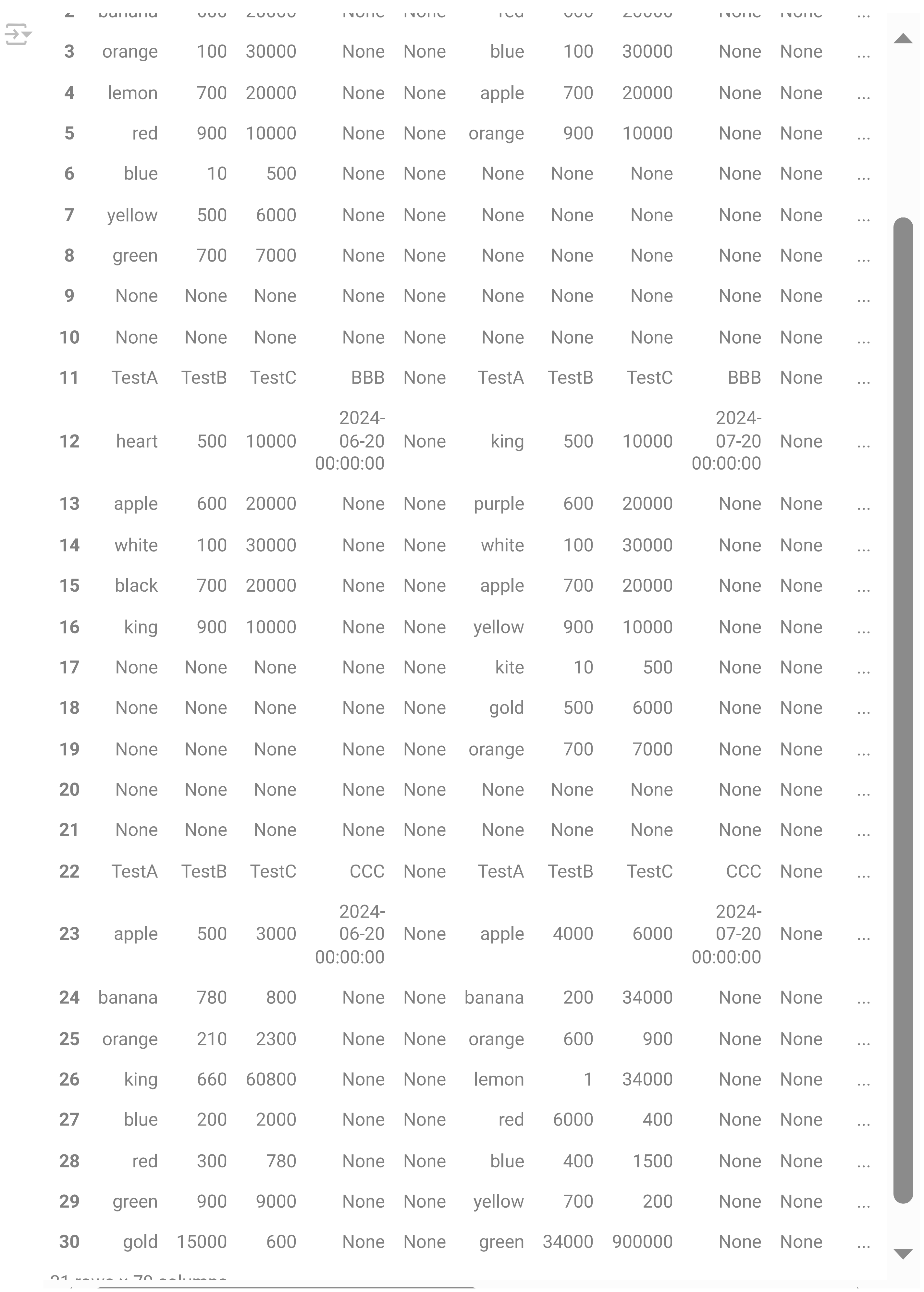
```
# Create a list to store the data
data = []

# Iterate through the rows in the "Data" worksheet
for row in data_sheet.iter_rows(values_only=True):
    data.append(row)

# Create the DataFrame
data_df = pd.DataFrame(data)

# Set the column names (if available)
if data_sheet.max_row > 0:
    data_df.columns = data_sheet[1]

# Display the DataFrame
data_df.columns = range(data_df.shape[1])
data_df
```



2	banana	600	20000	None	None	red	600	20000	None	None	...
3	orange	100	30000	None	None	blue	100	30000	None	None	...
4	lemon	700	20000	None	None	apple	700	20000	None	None	...
5	red	900	10000	None	None	orange	900	10000	None	None	...
6	blue	10	500	None	None	None	None	None	None	None	...
7	yellow	500	6000	None	None	None	None	None	None	None	...
8	green	700	7000	None	None	None	None	None	None	None	...
9	None	None	None	None	None	None	None	None	None	None	...
10	None	None	None	None	None	None	None	None	None	None	...
11	TestA	TestB	TestC	BBB	None	TestA	TestB	TestC	BBB	None	...
12	heart	500	10000	2024-06-20 00:00:00	None	king	500	10000	2024-07-20 00:00:00	None	...
13	apple	600	20000	None	None	purple	600	20000	None	None	...
14	white	100	30000	None	None	white	100	30000	None	None	...
15	black	700	20000	None	None	apple	700	20000	None	None	...
16	king	900	10000	None	None	yellow	900	10000	None	None	...
17	None	None	None	None	None	kite	10	500	None	None	...
18	None	None	None	None	None	gold	500	6000	None	None	...
19	None	None	None	None	None	orange	700	7000	None	None	...
20	None	None	None	None	None	None	None	None	None	None	...
21	None	None	None	None	None	None	None	None	None	None	...
22	TestA	TestB	TestC	CCC	None	TestA	TestB	TestC	CCC	None	...
23	apple	500	3000	2024-06-20 00:00:00	None	apple	4000	6000	2024-07-20 00:00:00	None	...
24	banana	780	800	None	None	banana	200	34000	None	None	...
25	orange	210	2300	None	None	orange	600	900	None	None	...
26	king	660	60800	None	None	lemon	1	34000	None	None	...
27	blue	200	2000	None	None	red	6000	400	None	None	...
28	red	300	780	None	None	blue	400	1500	None	None	...
29	green	900	9000	None	None	yellow	700	200	None	None	...
30	gold	15000	600	None	None	green	34000	900000	None	None	...

31 rows x 70 columns

```
# Get processing company and time, delete extra columns, rename
def get_block(data_df, row_extract_start, row_extract_end, col_extract_start, col_extract_e
    block_df = data_df.iloc[row_extract_start:row_extract_end, col_extract_start:col_extrac
    processing_company = block_df.iloc[0, 3]
    processing_time = block_df.iloc[1, 3]
    block_df = block_df.drop(block_df.columns[3], axis=1)
    block_df = block_df.dropna(how='all', axis=0)
    block_df.columns = block_df.iloc[0]
    block_df = block_df.iloc[1:].reset_index(drop=True)
    return block_df, processing_company, processing_time

# Check if the item is in the more_item list already. Add if not.
def check_or_add_item(item_name):
    global more_item, less_item, more_item_value, less_item_value
    if not more_item.empty:
        if item_name in more_item["ItemName"].values:
            return
    new_row = pd.DataFrame({"ItemName": [item_name]})
    more_item = pd.concat([more_item, new_row]).fillna(0)
    less_item = pd.concat([less_item, new_row]).fillna(0)
    more_item_value = pd.concat([more_item_value, new_row]).fillna(0)
    less_item_value = pd.concat([less_item_value, new_row]).fillna(0)
```

```
# Clear previous records
clear_charts()

# Slice the data_df slowly
row_extract_start = 0
row_extract_end = 10
col_extract_start = 0
col_extract_end = 4

while row_extract_end <= len(data_df)+5:

    while col_extract_end <= len(data_df.columns):
        block_df, processing_company, processing_time = get_block(data_df, row_extract_star
        diff = 0

        for _, row in block_df.iterrows():

            # Check item added in four charts or not; also for dictionary
            check_or_add_item(row["TestA"])
            if row["TestA"] not in benchmark:
                benchmark.setdefault(row["TestA"], (0, 0))

           # Compare with benchmark
            if row["TestA"] not in benchmark:
                diff = row["TestB"]
            else:
                diff = row["TestB"] - benchmark[row["TestA"]][0]

            # Update the dataframes
            if diff > 0:
                more_item.loc[more_item["ItemName"] == row["TestA"], processing_time] += 1
                more_item_value.loc[more_item_value["ItemName"] == row["TestA"], processing
            elif diff < 0:
                less_item.loc[less_item["ItemName"] == row["TestA"], processing_time] += 1
                less_item_value.loc[less_item_value["ItemName"] == row["TestA"], processing
            else:
                pass

        col_extract_start += 5
        col_extract_end += 5

        print(processing_company, processing_time)

    row_extract_start += 11
    row_extract_end += 11
    col_extract_start = 0
    col_extract_end = 4
```

```
AAA 2024-06-20 00:00:00
AAA 2024-07-20 00:00:00
AAA 2024-08-20 00:00:00
AAA 2024-09-20 00:00:00
```

```
     AAA 2024-06-20 00:00:00
     AAA 2024-07-20 00:00:00
     AAA 2024-08-20 00:00:00
     AAA 2024-09-20 00:00:00
     AAA 2024-06-20 00:00:00
     AAA 2024-07-20 00:00:00
     AAA 2024-08-20 00:00:00
     AAA 2024-09-20 00:00:00
     AAA 2024-06-20 00:00:00
     AAA 2024-07-20 00:00:00
     AAA 2024-08-20 00:00:00
     AAA 2024-09-20 00:00:00
     BBB 2024-06-20 00:00:00
     BBB 2024-07-20 00:00:00
     BBB 2024-08-20 00:00:00
     BBB 2024-09-20 00:00:00
     BBB 2024-06-20 00:00:00
     BBB 2024-07-20 00:00:00
     BBB 2024-08-20 00:00:00
     BBB 2024-09-20 00:00:00
     BBB 2024-06-20 00:00:00
     BBB 2024-07-20 00:00:00
     BBB 2024-08-20 00:00:00
     BBB 2024-09-20 00:00:00
     BBB 2024-06-20 00:00:00
     BBB 2024-07-20 00:00:00
     BBB 2024-08-20 00:00:00
     BBB 2024-09-20 00:00:00
     CCC 2024-06-20 00:00:00
     CCC 2024-07-20 00:00:00
     CCC 2024-08-20 00:00:00
     CCC 2024-09-20 00:00:00
     CCC 2024-06-20 00:00:00
     CCC 2024-07-20 00:00:00
     CCC 2024-08-20 00:00:00
     CCC 2024-09-20 00:00:00
     CCC 2024-06-20 00:00:00
     CCC 2024-07-20 00:00:00
     CCC 2024-08-20 00:00:00
     CCC 2024-09-20 00:00:00
     CCC 2024-06-20 00:00:00
     CCC 2024-07-20 00:00:00
     CCC 2024-08-20 00:00:00
     CCC 2024-09-20 00:00:00
```

```
more_item
```

	ItemName	2024-06-20 00:00:00	2024-07-20 00:00:00	2024-08-20 00:00:00	2024-09-20 00:00:00	2024-10-20 00:00:00	2024-11-20 00:00:00	2024-12-20 00:00:00	2025- 00:0
0	apple	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0	banana	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0	orange	8.0	12.0	8.0	4.0	0.0	0.0	0.0	
0	lemon	4.0	4.0	4.0	4.0	0.0	0.0	0.0	
0	red	8.0	8.0	8.0	8.0	0.0	0.0	0.0	
0	blue	8.0	8.0	8.0	8.0	0.0	0.0	0.0	
0	yellow	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0	green	0.0	4.0	0.0	0.0	0.0	0.0	0.0	
0	kite	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0	queen	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0	gold	4.0	0.0	0.0	0.0	0.0	0.0	0.0	
0	heart	4.0	0.0	4.0	4.0	0.0	0.0	0.0	
0	white	4.0	4.0	0.0	0.0	0.0	0.0	0.0	
0	black	4.0	0.0	4.0	0.0	0.0	0.0	0.0	
0	king	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0	purple	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

後續步驟： 使用 more_item生成程式碼 查看建議的圖表

```
less_item
```

	ItemName	2024-06-20 00:00:00	2024-07-20 00:00:00	2024-08-20 00:00:00	2024-09-20 00:00:00	2024-10-20 00:00:00	2024-11-20 00:00:00	2024-12-20 00:00:00	2025- 00:0
0	apple	12.0	12.0	12.0	12.0	0.0	0.0	0.0	
0	banana	8.0	4.0	4.0	8.0	0.0	0.0	0.0	
0	orange	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0	lemon	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0	red	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0	blue	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0	yellow	4.0	12.0	8.0	12.0	0.0	0.0	0.0	
0	green	8.0	0.0	4.0	8.0	0.0	0.0	0.0	
0	kite	0.0	4.0	8.0	4.0	0.0	0.0	0.0	
0	queen	0.0	0.0	8.0	4.0	0.0	0.0	0.0	
0	gold	0.0	4.0	8.0	8.0	0.0	0.0	0.0	
0	heart	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0	white	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0	black	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0	king	8.0	4.0	4.0	4.0	0.0	0.0	0.0	
0	purple	0.0	4.0	0.0	0.0	0.0	0.0	0.0	

後續步驟： 使用 less_item生成程式碼 查看建議的圖表

```
more_item_value
```