MANOVA Analysis in Python – Complete Step-by-Step Guide

SECTION 1: General Python Commands and Setup

1. Check Python & PIP Versions

```
python --version # Check Python version
```

pip --version # Check pip version

2. See Installed Python Packages

pip list # List all installed packages

3. Install Required Libraries

Install the core libraries needed for MANOVA and data analysis:

pip install pandas numpy matplotlib seaborn statsmodels openpyxl

4. Run Python Scripts in VSCode

- Create and Run a Script
 - 1. Open VSCode.
 - 2. Create a new file: manova_analysis.py.
 - 3. Write your code inside it.
 - 4. Save the file.
 - 5. Run it in terminal with:

python manova_analysis.py

✓ 5. Read Excel Dataset in Python

import pandas as pd

```
# Read Excel file (make sure 'openpyxl' is installed)

df = pd.read_excel("Manova data.xlsx", sheet_name="Upload-sums")
```

■ SECTION 2: Wide vs Long Format – Pivoting

What Do "Wide" and "Long" Format Mean?

Feature Wide Format Long Format

Shape	One row per subject; many columns	Multiple rows per subject
Used for	Classic MANOVA, Machine Learning	ANOVA, Repeated Measures, Post-Hoc
Example	Columns = [Reno, Cubic, Vegas,]	Columns = [Subject, Algorithm, Value]

• Example

Wide Format:

Observatio n	Cubic	Reno	DCTCP	Vega s	
1	30.2	28.1	29.4	22.5	:
2	31.7	27.8	30.1	23.9	

Long Format:

Observatio n	Algorithm	UploadThroughput
1	Cubic	30.2
1	Reno	28.1

1	DCTCP	29.4
---	-------	------

SECTION 3: MANOVA Analysis in Python

▼ Step 1: Load & Clean Dataset, Create Wide and Long Format

```
import pandas as pd
# Load data
df = pd.read_excel("Manova data.xlsx", sheet_name="Upload-sums")
# Clean column names
df.columns = (
    df.columns
    .str.strip()
    .str.replace(' ', '_', regex=False)
    .str.replace('-', '_', regex=False)
    .str.replace('+', 'plus', regex=False)
)
# Add Observation ID
```

```
df = df.reset_index().rename(columns={"index": "Observation"})

# Wide Format

df_wide = df.copy()

# Long Format (for post-hoc, repeated measures)

df_long = df.melt(
    id_vars=["Observation"],
    var_name="Algorithm",
    value_name="UploadThroughput"
)
```

Step 2: MANOVA on Wide Format (Classical)

```
from statsmodels.multivariate.manova import MANOVA

y_columns = df_wide.columns.drop("Observation")

formula = ' + '.join(y_columns) + ' ~ 1' # No independent variable

manova_wide = MANOVA.from_formula(formula, data=df_wide)

print(manova_wide.mv_test())
```

Use this when:

You're analyzing multiple outcome variables at once (each algorithm's performance is a separate variable).

why Manova on long Format does not work

```
maov = MANOVA.from_formula("UploadThroughput ~ Algorithm",
data=df_long)
print(maov.mv_test())
```

Use this when:

You're analyzing whether one repeated-measure variable (UploadThroughput) changes across levels of Algorithm.

Error Explained:

ValueError: There must be more than one dependent variable to fit MANOVA!

This happens because in your long-format data, you're trying to apply MANOVA with only one dependent variable (UploadThroughput).

But MANOVA (Multivariate Analysis of Variance) is only valid when you have multiple dependent variables — that's why it works with wide-format data (where each algorithm is a separate column = separate dependent variable).

Step 3: Tukey's HSD (Post-hoc Comparison)

from statsmodels.stats.multicomp import pairwise_tukeyhsd

tukey = pairwise_tukeyhsd(

```
endog=df_long['UploadThroughput'],
   groups=df_long['Algorithm'],
   alpha=0.05
)
print(tukey.summary())
```

Use this to:

Compare which pairs of algorithms are statistically significantly different in performance.

Step 4: Visualize Results with Boxplot

python

CopyEdit

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))
sns.boxplot(data=df_long, x='Algorithm', y='UploadThroughput')
plt.xticks(rotation=45)
plt.title("Upload Throughput by TCP Algorithm")
plt.tight_layout()
plt.show()
```