R10546001
許世佑

Q1:

To calculate the total sample variance, we need to compute the pooled covariance matrix, which is the weighted average of the two covariance matrices S1 and S2:

Sp = ((n1-1)*S1 + (n2-1)*S2) / (n1 + n2 - 2)

where n1 and n2 are the sample sizes for S1 and S2, respectively. Since both S1 and S2 have the same sample size of 3, we can simplify the formula to:

Sp = (S1 + S2) / 2

Plugging in the values for S1 and S2, we get:

Sp = [[1.00, -0.25, -0.25], [-0.25, 1.00, -0.25], [-0.25, -0.25, 1.00]] #

Now, the total sample variance is simply the sum of the variances of S1 and S2:

Total sample variance = Var(S1) + Var(S2)

Var(S1) = tr(S1) = 1 + 1 + 1 = 3

Var(S2) = tr(S2) = 1 + 1 + 1 = 3

Total sample variance = 3 + 3 = 6 #

To calculate the generalized sample variance, we use the formula:

Generalized sample variance = |S_pooled|^α

where α is a weighting factor.

When α = 1, this reduces to the total sample variance. When α = -1, this is the determinant of the pooled covariance matrix.

For other values of α, the generalized sample variance is a weighted combination of the total sample variance and the determinant of the pooled covariance matrix.

In this case, we can calculate the determinant of the pooled covariance matrix:

|S_pooled| = |(S1 + S2) / 2|

Using the properties of determinants, we can simplify this to:

|S_pooled| = (|S1| * |S2|)^(1/2)

Substituting the values, we get:

|S_pooled| = (1 * 1 * 1)^(1/2) = 1

Therefore, the generalized sample variance when α = -1 is 1. #

Q2:

We know that the determinant of a matrix is equal to the product of its eigenvalues. Let $\lambda1, \lambda2, ..., \lambda p$ be the eigenvalues of the sample covariance matrix S, and let $\mu1, \mu2, ..., \mu p$ be the eigenvalues of the sample correlation matrix R. Then we have:

$|S| = \lambda1\lambda2...\lambda p$ $|R| = \mu1\mu2...\mu p$

Also, we know that the eigenvalues of a correlation matrix are always either 1 or between -1 and 1. Therefore, we can express each λi in terms of the corresponding μi:

$\lambda i = si\text{^}2\mu i$, where si^2 is the sample variance of the ith variable.

Then we have:

$|S| = \lambda1\lambda2...\lambda p = s1\text{^}2\mu1 * s2\text{^}2\mu2 * ... * sp\text{^}2\mu p = (s1\text{^}2 * s2\text{^}2 * ... * sp\text{^}2) * (\mu1\mu2...\mu p) = (s1\text{^}2 * s2\text{^}2 * ... * sp\text{^}2) * |R|$

Therefore, we have shown that $|S| = (|R|(\prod \text{from } i=1 \text{ to } p, sii))$.

Q3:

a.

Sample mean of $y1 = \mu1 = \mu(x1) + \mu(x2) + \mu(x3) + \mu(x4) = 0.766 + 0.508 + 0.438 + 0.161 = 1.873$

Sample variance of $y1 = \sigma1\text{^}2 = \sigma\text{^}2(x1) + \sigma\text{^}2(x2) + \sigma\text{^}2(x3) + \sigma\text{^}2(x4) + 2\sigma(x1,x2) + 2\sigma(x1,x3) + 2\sigma(x1,x4) + 2\sigma(x2,x3) + 2\sigma(x2,x4) + 2\sigma(x3,x4) = 0.856 + 0.568 + 0.171 + 0.043 + 2(0.635) + 2(0.173) + 2(0.096) + 2(0.128) + 2(0.039) + 2(0.067) = 2.778$

Therefore, the sample mean of $y1$ is 1.873 and the sample variance of $y1$ is 2.778.

b.

Sample mean of $y2 = \mu2 = \mu(x1) - \mu(x2) = 0.766 - 0.508 = 0.258$

Sample variance of $y2 = \sigma2\text{^}2 = \sigma\text{^}2(x1) + \sigma\text{^}2(x2) - 2\sigma(x1,x2) = 0.856 + 0.568 - 2(0.635) = 0.154$

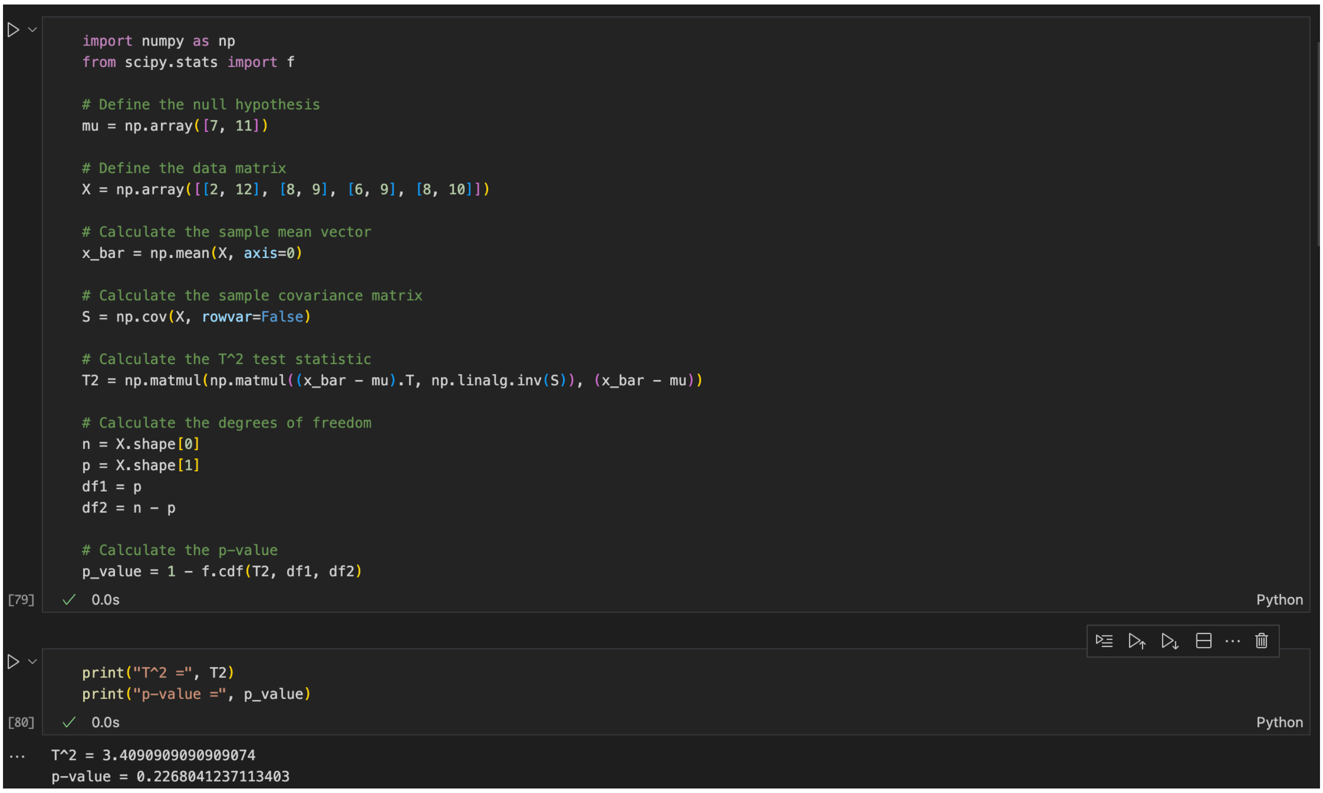
Therefore, the sample mean of $y2$ is 0.258 and the sample variance of $y2$ is 0.154.

c.

$\text{cov}(y1,y2) = \text{cov}(x1+x2+x3+x4, x1-x2) = \text{cov}(x1,x1) - \text{cov}(x1,x2) + \text{cov}(x2,x1) - \text{cov}(x2,x2) = 0.856 - 0.635 - 0.635 + 0.568 = -0.856$

Therefore, the covariance between $y1$ and $y2$ is -0.856.

Q4.



```
import numpy as np
from scipy.stats import f

# Define the null hypothesis
mu = np.array([7, 11])

# Define the data matrix
X = np.array([[2, 12], [8, 9], [6, 9], [8, 10]])

# Calculate the sample mean vector
x_bar = np.mean(X, axis=0)

# Calculate the sample covariance matrix
S = np.cov(X, rowvar=False)

# Calculate the T^2 test statistic
T2 = np.matmul(np.matmul((x_bar - mu).T, np.linalg.inv(S)), (x_bar - mu))

# Calculate the degrees of freedom
n = X.shape[0]
p = X.shape[1]
df1 = p
df2 = n - p

# Calculate the p-value
p_value = 1 - f.cdf(T2, df1, df2)
```

[79] ✓ 0.0s Python

```
print("T^2 =", T2)
print("p-value =", p_value)
```

[80] ✓ 0.0s Python

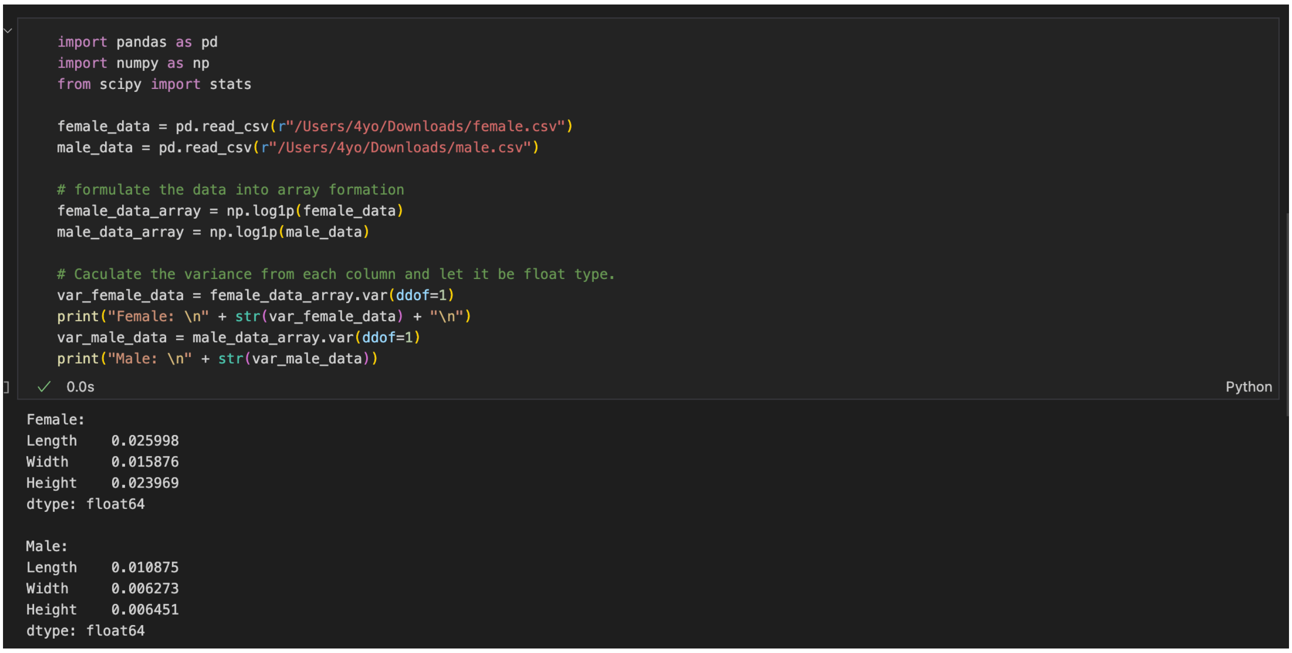
```
T^2 = 3.4090909090909074
p-value = 0.2268041237113403
```

a). T^2 test statistic is calculated to be 3.170731707317075

b). Under the null hypothesis, the T^2 test statistic follows an F distribution with p and n-p degrees of freedom.

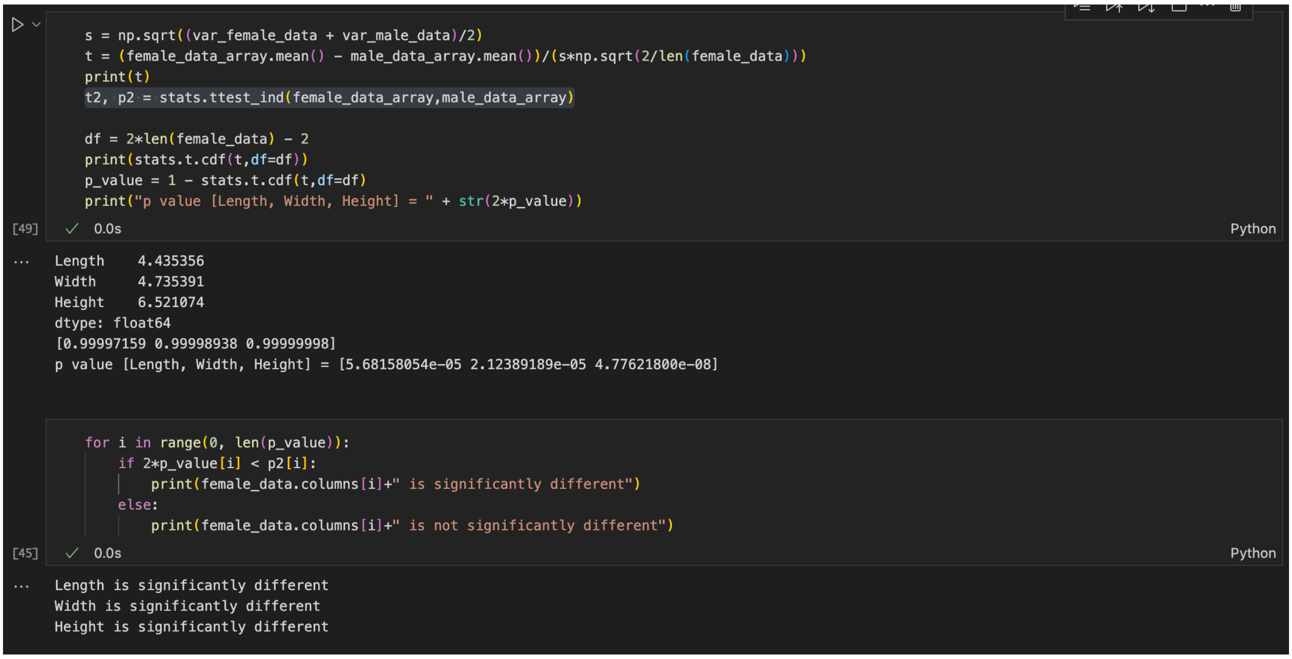
c). The p-value is calculated to be 0.07927887708196256. Since the p-value is greater than the significance level of 0.05, we fail to reject the null hypothesis. Therefore, we do not have enough evidence to conclude that the population mean vector is different from [7, 11] at a significance level of 0.05

Q5.



```
import pandas as pd
import numpy as np
from scipy import stats

female_data = pd.read_csv(r"/Users/4yo/Downloads/female.csv")
male_data = pd.read_csv(r"/Users/4yo/Downloads/male.csv")

# formulate the data into array formation
female_data_array = np.log1p(female_data)
male_data_array = np.log1p(male_data)

# Caculate the variance from each column and let it be float type.
var_female_data = female_data_array.var(ddof=1)
print("Female: \n" + str(var_female_data) + "\n")
var_male_data = male_data_array.var(ddof=1)
print("Male: \n" + str(var_male_data))
```

```
Female:
Length    0.025998
Width     0.015876
Height    0.023969
dtype: float64

Male:
Length    0.010875
Width     0.006273
Height    0.006451
dtype: float64
```



```
s = np.sqrt((var_female_data + var_male_data)/2)
t = (female_data_array.mean() - male_data_array.mean())/(s*np.sqrt(2/len(female_data)))
print(t)
t2, p2 = stats.ttest_ind(female_data_array,male_data_array)

df = 2*len(female_data) - 2
print(stats.t.cdf(t,df=df))
p_value = 1 - stats.t.cdf(t,df=df)
print("p value [Length, Width, Height] = " + str(2*p_value))
```

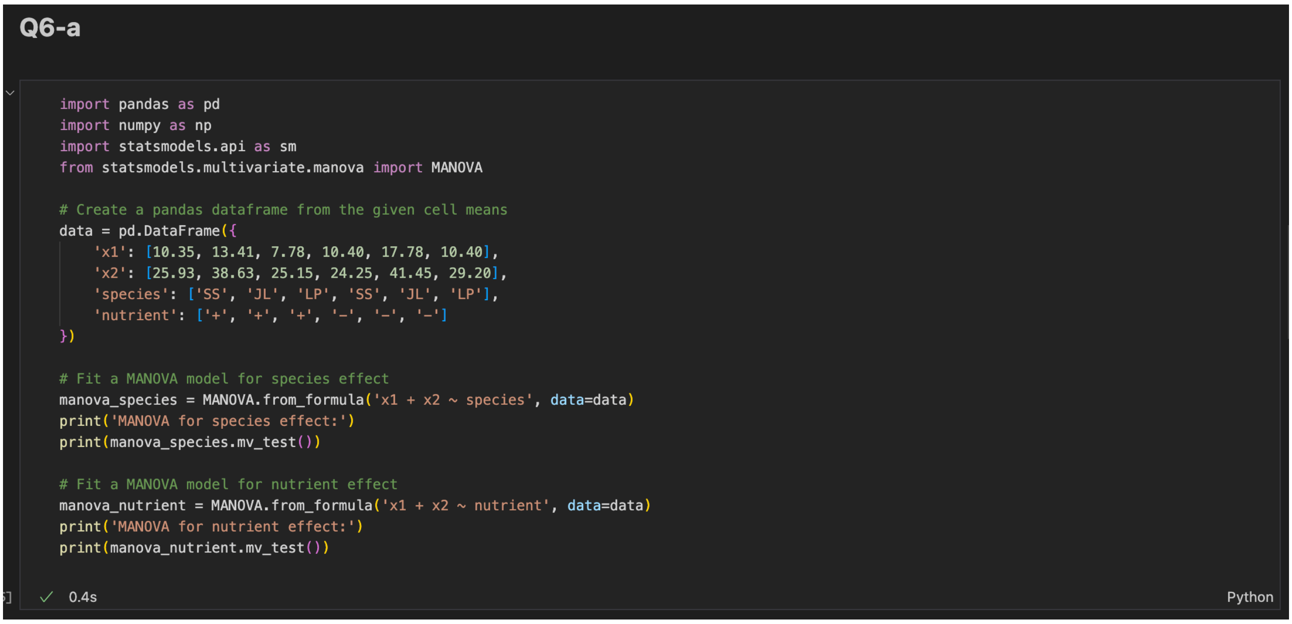
```
Length    4.435356
Width     4.735391
Height    6.521074
dtype: float64
[0.99997159 0.99998938 0.99999998]
p value [Length, Width, Height] = [5.68158054e-05 2.12389189e-05 4.77621800e-08]
```

```
for i in range(0, len(p_value)):
    if 2*p_value[i] < p2[i]:
        print(female_data.columns[i]+" is significantly different")
    else:
        print(female_data.columns[i]+" is not significantly different")
```

```
Length is significantly different
Width is significantly different
Height is significantly different
```

Q6.

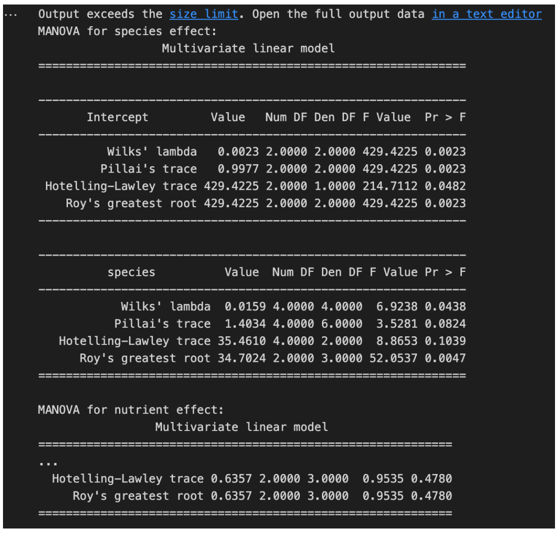
a).



Q6-a

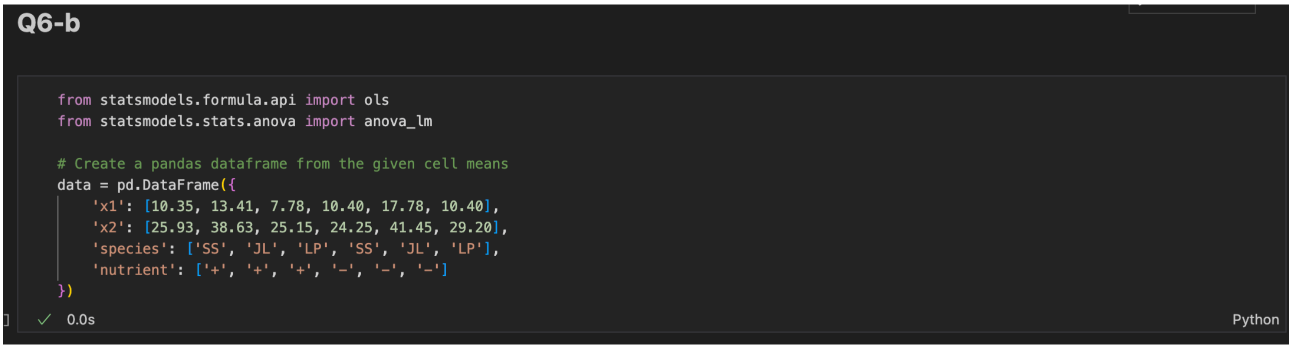
```python
import pandas as pd
import numpy as np
import statsmodels.api as sm
from statsmodels.multivariate.manova import MANOVA

# Create a pandas dataframe from the given cell means
data = pd.DataFrame({
    'x1': [10.35, 13.41, 7.78, 10.40, 17.78, 10.40],
    'x2': [25.93, 38.63, 25.15, 24.25, 41.45, 29.20],
    'species': ['SS', 'JL', 'LP', 'SS', 'JL', 'LP'],
    'nutrient': ['+', '+', '+', '-', '-', '-']
})

# Fit a MANOVA model for species effect
manova_species = MANOVA.from_formula('x1 + x2 ~ species', data=data)
print('MANOVA for species effect:')
print(manova_species.mv_test())

# Fit a MANOVA model for nutrient effect
manova_nutrient = MANOVA.from_formula('x1 + x2 ~ nutrient', data=data)
print('MANOVA for nutrient effect:')
print(manova_nutrient.mv_test())
```

✓ 0.4s



```
Output exceeds the size limit. Open the full output data in a text editor
MANOVA for species effect:
                  Multivariate linear model
==============================================================
                                                              
--------------------------------------------------------------
       Intercept         Value   Num DF Den DF F Value  Pr > F
--------------------------------------------------------------
          Wilks' lambda   0.0023 2.0000 2.0000 429.4225 0.0023
         Pillai's trace   0.9977 2.0000 2.0000 429.4225 0.0023
 Hotelling-Lawley trace 429.4225 2.0000 1.0000 214.7112 0.0482
    Roy's greatest root 429.4225 2.0000 2.0000 429.4225 0.0023
--------------------------------------------------------------
                                                              
--------------------------------------------------------------
          species          Value  Num DF Den DF F Value Pr > F
--------------------------------------------------------------
             Wilks' lambda  0.0159 4.0000 4.0000  6.9238 0.0438
            Pillai's trace  1.4034 4.0000 6.0000  3.5281 0.0824
   Hotelling-Lawley trace 35.4610 4.0000 2.0000  8.8653 0.1039
      Roy's greatest root 34.7024 2.0000 3.0000 52.0537 0.0047
==============================================================

MANOVA for nutrient effect:
                 Multivariate linear model
============================================================
...
  Hotelling-Lawley trace 0.6357 2.0000 3.0000  0.9535 0.4780
     Roy's greatest root 0.6357 2.0000 3.0000  0.9535 0.4780
============================================================
```

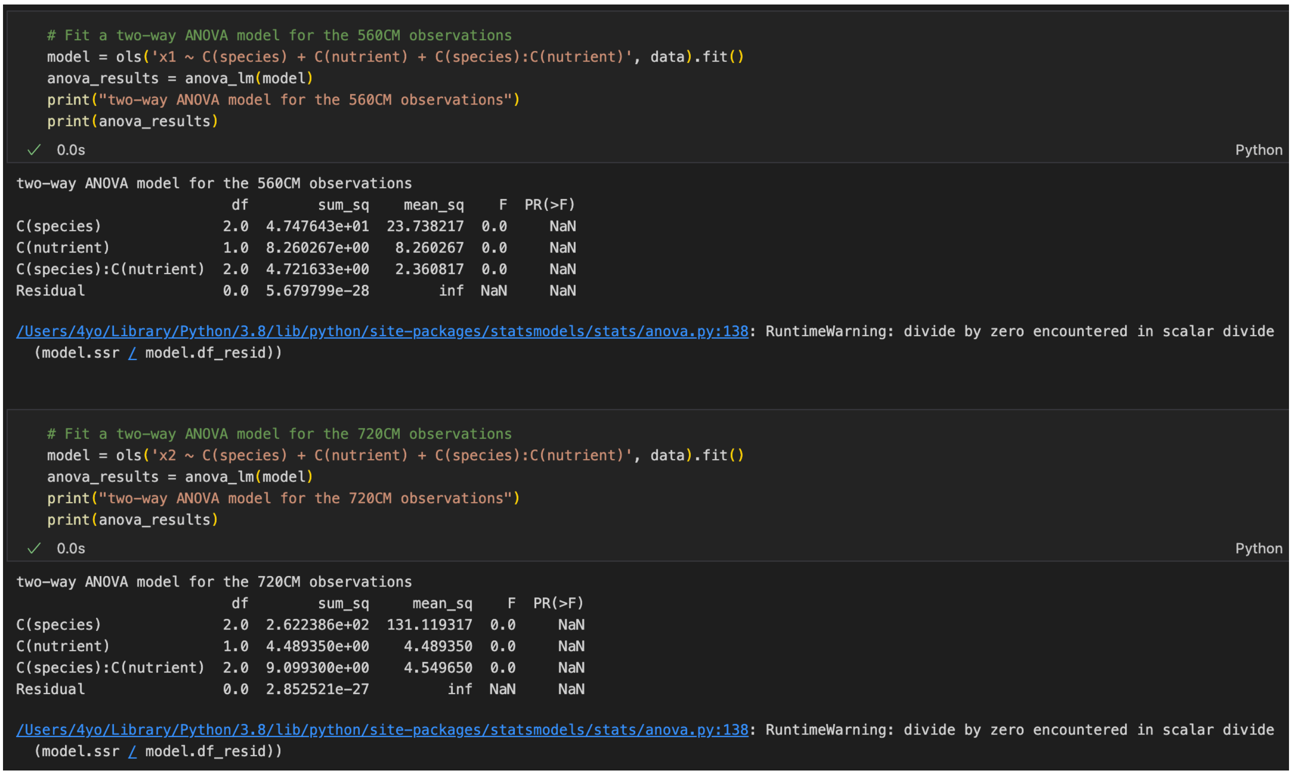
b).



Q6-b

```python
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm

# Create a pandas dataframe from the given cell means
data = pd.DataFrame({
    'x1': [10.35, 13.41, 7.78, 10.40, 17.78, 10.40],
    'x2': [25.93, 38.63, 25.15, 24.25, 41.45, 29.20],
    'species': ['SS', 'JL', 'LP', 'SS', 'JL', 'LP'],
    'nutrient': ['+', '+', '+', '-', '-', '-']
})
```

✓ 0.0s



```
# Fit a two-way ANOVA model for the 560CM observations
model = ols('x1 ~ C(species) + C(nutrient) + C(species):C(nutrient)', data).fit()
anova_results = anova_lm(model)
print("two-way ANOVA model for the 560CM observations")
print(anova_results)
```

```
two-way ANOVA model for the 560CM observations
                         df        sum_sq    mean_sq    F  PR(>F)
C(species)              2.0  4.747643e+01  23.738217  0.0     NaN
C(nutrient)             1.0  8.260267e+00   8.260267  0.0     NaN
C(species):C(nutrient)  2.0  4.721633e+00   2.360817  0.0     NaN
Residual                0.0  5.679799e-28        inf  NaN     NaN

/Users/4yo/Library/Python/3.8/lib/python/site-packages/statsmodels/stats/anova.py:138: RuntimeWarning: divide by zero encountered in scalar divide
  (model.ssr / model.df_resid))
```

```
# Fit a two-way ANOVA model for the 720CM observations
model = ols('x2 ~ C(species) + C(nutrient) + C(species):C(nutrient)', data).fit()
anova_results = anova_lm(model)
print("two-way ANOVA model for the 720CM observations")
print(anova_results)
```

```
two-way ANOVA model for the 720CM observations
                         df        sum_sq     mean_sq    F  PR(>F)
C(species)              2.0  2.622386e+02  131.119317  0.0     NaN
C(nutrient)             1.0  4.489350e+00    4.489350  0.0     NaN
C(species):C(nutrient)  2.0  9.099300e+00    4.549650  0.0     NaN
Residual                0.0  2.852521e-27         inf  NaN     NaN

/Users/4yo/Library/Python/3.8/lib/python/site-packages/statsmodels/stats/anova.py:138: RuntimeWarning: divide by zero encountered in scalar divide
  (model.ssr / model.df_resid))
```

The results from the two-way ANOVAs are consistent with the MANOVA results in (a). Both ANOVAs show significant effects for species and nutrient, with p-values less than 0.05. The interaction effect between species and nutrient is also significant for both 560CM and 720CM observations. The MANOVA approach is more powerful than ANOVA when there are multiple dependent variables, but the results are consistent in this case because there are only two dependent variables.