[CC511] Homework 10 20204817 Federico Berto

November 23, 2020

1 Homework 10 - Federico Berto

[88]:
```
# Importing useful libraries
import numpy as np
import pandas as pd
import math
import matplotlib.pyplot as plt
from scipy import stats
import statsmodels.stats.weightstats as sms
from scipy.stats import t
from scipy.stats import f
from scipy.stats import norm
from statsmodels.stats.anova import anova_lm
from statsmodels.formula.api import ols
from statsmodels.stats.multicomp import pairwise_tukeyhsd
from IPython.display import display, Math
```

1.1 Exercise 11.1.4

The p-value will not change. We can prove this by showing the new F' statistics is the same as the original F.

Given x_{ij} is replaced by the value $ax_{ij} + b$, then we have to calculate first the $MSTr$ and MSE of the function. In this case we have:

$$SSTr' = \sum_{i=1}^{k} n_i \overline{x}'^2_{i\cdot} - n_T \overline{x}'^2_{\cdot\cdot}$$

Then,

$$SSE = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (x'_{ij} - \overline{x}'_{i\cdot})^2$$

Also, we can notice that by the properties of mean and variance:

$$\mu'_{i\cdot} = a \times \mu_{i\cdot} + b$$

$$\sigma'^2_{i\cdot} = a^2 \times \sigma^2_{i\cdot}$$

$$\sigma'^2_{\cdot\cdot} = a^2 \times \sigma^2_{\cdot\cdot}$$

Therefore:

$$SSTr' = a^2 \times SSTr$$

$$SSE' = a^2 \times SSE$$

By which we can get the F' value as

$$F' = \frac{\frac{SSTr'}{k-1}}{\frac{SSE'}{n_T-k}} = \frac{a^2 \times \frac{SSTr}{k-1}}{a^2 \times \frac{SSE}{n_T-k}} = F$$

Hence, the p-value will not change.

1.2 Exercise 11.1.8

a) First, we get the $q_{\alpha,k,\nu} = 3.49$. Therefore, we can calculate the ranges with the following:

$$\left(\overline{x}_{i_1} - \overline{x}_{i_2} - \hat{\sigma}\frac{q_{\alpha,k,\nu}}{\sqrt{2}}\sqrt{\frac{1}{n_{i_1}} + \frac{1}{n_{i_2}}}, \overline{x}_{i_1} - \overline{x}_{i_2} + \hat{\sigma}\frac{q_{\alpha,k,\nu}}{\sqrt{2}}\sqrt{\frac{1}{n_{i_1}} + \frac{1}{n_{i_2}}}\right)$$

So:

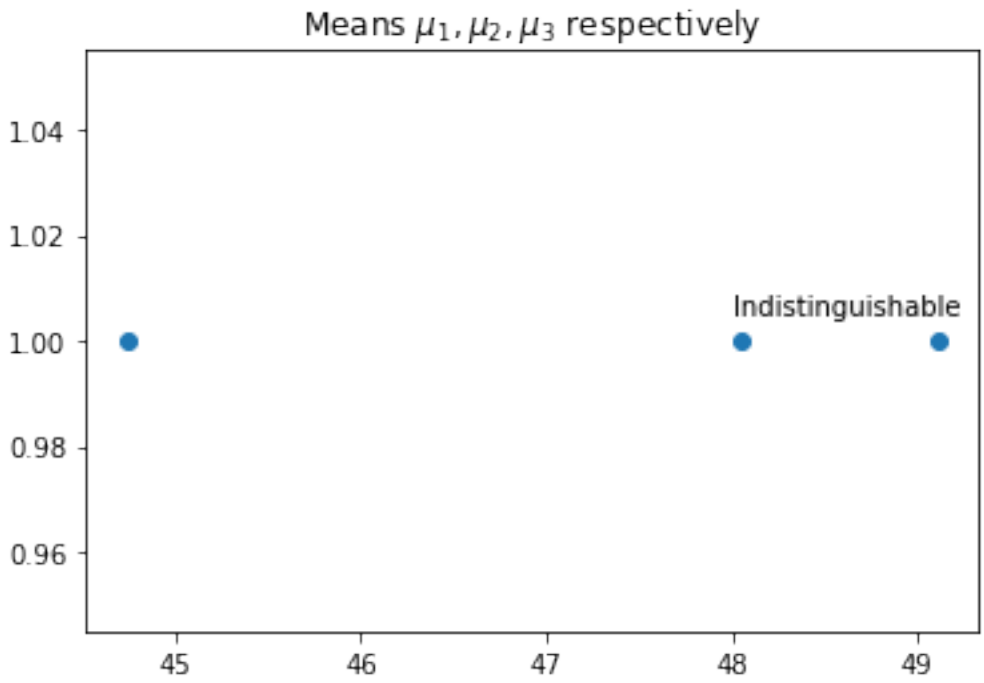
$$\mu_1 - \mu_2 \in \left(48.05 - 44.74 - \frac{\sqrt{4.96} \times 3.49}{\sqrt{11}}, 48.05 - 44.74 + \frac{\sqrt{4.96} \times 3.49}{\sqrt{11}}\right) = (0.97, 5.65)$$

$$\mu_1 - \mu_3 \in \left(48.05 - 49.11 - \frac{\sqrt{4.96} \times 3.49}{\sqrt{11}}, 48.05 - 49.11 + \frac{\sqrt{4.96} \times 3.49}{\sqrt{11}}\right) = (-3.40, 1.28)$$

$$\mu_2 - \mu_3 \in \left(44.74 - 49.11 - \frac{\sqrt{4.96} \times 3.49}{\sqrt{11}}, 44.74 - 49.11 + \frac{\sqrt{4.96} \times 3.49}{\sqrt{11}}\right) = (-6.71, -2.03)$$

b) Diagram:

[43]:
```
# Use matplotlib to plot the means and their groupings

plt.scatter([44.74, 48.05, 49.11], [1, 1, 1])
plt.title('Means $\mu_1, \mu_2, \mu_3$ respectively')
plt.annotate('Indistinguishable',(48, 1.005))
plt.show()
```



c) We know that $L = 2q_{\alpha,k,\nu}\sqrt{\frac{s^2}{n}}$ and $L \leq 2.0$, so:

$$n \geq \frac{4s^2 q^2_{\alpha,k,\nu}}{L^2} = \frac{4 \cdot 4.96 \cdot 3.49^2}{2^2} = 60.4$$

Hence, n is at least 61; then 61-11 = 50 new observations from each factor level should be collected.

1.3 Exercise 11.1.16

We transfer the data to Python to perform the hard-coded calculations without using packages.

[108]:
```
# Load the data
data = pd.read_excel('DS 11.1.4.xls') # Tranform file to csv

def eliminate_nan(x):
    return x[np.logical_not(np.isnan(x))]

# Transfor layouts to arrays
l1 = eliminate_nan(data['Layout 1'].to_numpy())
l2 = eliminate_nan(data['Layout 2'].to_numpy())
l3 = eliminate_nan(data['Layout 3'].to_numpy())
layout_list = [l1, l2, l3]

# Hard coded calculations
k = 3
```

```
df_treatments = k - 1
nt = np.count_nonzero(l1) + np.count_nonzero(l2) + np.count_nonzero(l3)
df_error = nt -k
df_total = nt -1

# Sum of Squares
stack = np.hstack((l1, l2, l3))
SST = 0
for i in range(len(stack)):
    SST += (stack[i] - stack.mean())**2
SSTr = 0
for i in range(3):
    SSTr += len(layout_list[i]) * (layout_list[i].mean() - stack.mean())**2
SSE = SST - SSTr

# Mean of Squares
MSTr = SSTr / (k-1)
MSE = SSE / (nt-k)

# F-value
F = MSTr/MSE

# p-value
p = f.sf(F, k-1, nt-k)

# sigma
sigma = math.sqrt(MSE)

# Return the left and right confidence interval
def print_confidence_interval(data1, data2, q):
    diff = data1.mean() - data2.mean()
    left = diff - sigma*q/(math.sqrt(2)) * math.sqrt( 1/len(data1) + 1/
 ↪len(data2))
    right = diff + sigma*q/(math.sqrt(2)) * math.sqrt( 1/len(data1) + 1/
 ↪len(data2))
    return left, right

# Critical value
q = 3.49

# Print functions
display(Math(r'F: {:.4f} \\ p-value: {:.4f}'.format(F, p)))
left, right = confidence_interval(l1, l2, q)
display(Math(r'\mu_1 - \mu_2 \in ({:.4f}, {:.4f})'.format(left, right)))
left, right = confidence_interval(l1, l3, q)
display(Math(r'\mu_1 - \mu_3 \in ({:.4f}, {:.4f})'.format(left, right)))
left, right = confidence_interval(l2, l3, q)
```

```
display(Math(r'\mu_2 - \mu_3 \in ({:.4f}, {:.4f})'.format(left, right)))
```

F : 52.8394
$p - value$: 0.0000

$\mu_1 - \mu_2 \in (-5.1168, -2.8532)$

$\mu_1 - \mu_3 \in (-0.7420, 1.4647)$

$\mu_2 - \mu_3 \in (3.1915, 5.5013)$

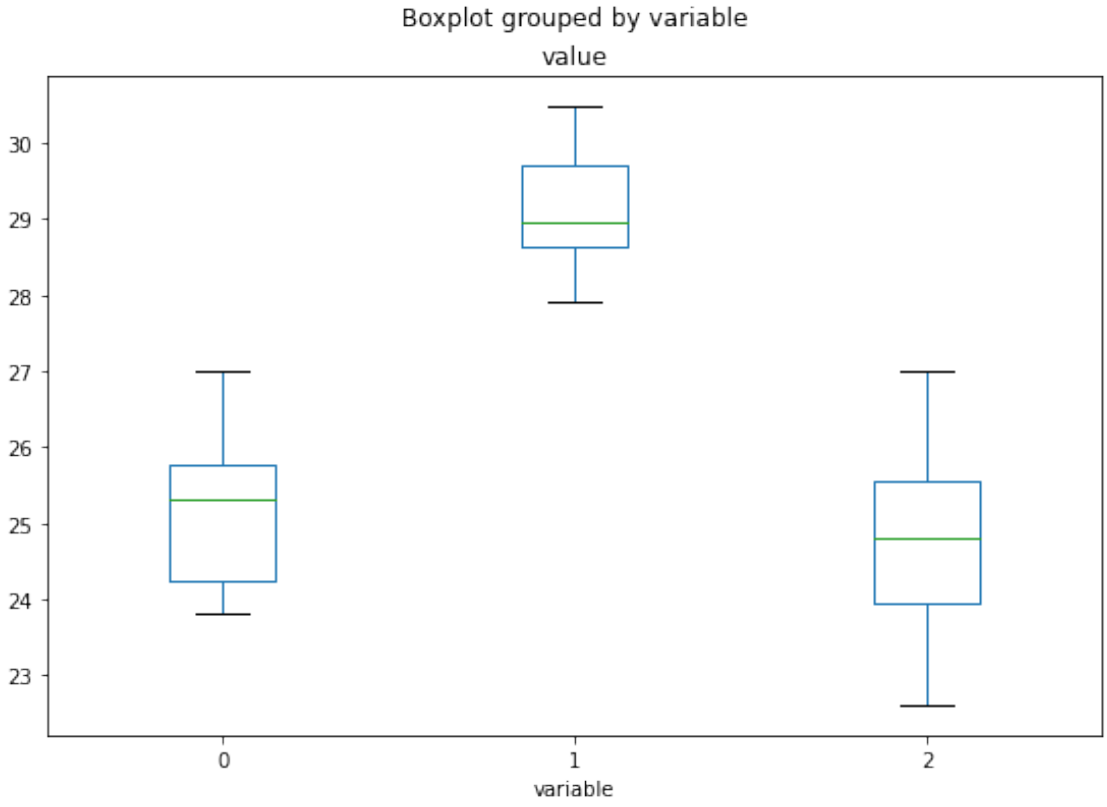
Therefore, we have enough evidence to conclude that layout 2 is slower than the other layouts.

We can check the manually calculated results by the Python packages to verify our results:

[36]:
```
# Data loader
data = pd.read_excel('DS 11.1.4.xls') # Tranform file to csv
print(data)

# Data processing (groups will start from 0)
value_vars = []
for i in range(len(data.columns)):
    value_vars.append(i) # count number of groups
data = data.T.reset_index(drop=True).T
proc_data = pd.melt(data, value_vars=value_vars)
proc_data.dropna(subset = ['value'], inplace=True)

# Anova table
model = ols('value ~ C(variable)', proc_data).fit()
print("\nANOVA TABLE\n",anova_lm(model))

# Tukey comparison
comp = pairwise_tukeyhsd(proc_data['value'], proc_data['variable'], alpha=0.05)
print('\n', comp)

# Boxplots
proc_data.boxplot('value', by='variable', grid=False, figsize=(9,6))
plt.show()
```

```
     Layout 1  Layout 2  Layout 3
0   23.799999      30.2      27.0
1   25.600000      29.9      25.4
2   24.000000      29.1      25.6
3   25.100000      28.8      24.2
4   25.500000      29.1      24.8
5   26.100000      28.6      24.0
6   23.799999      28.3      25.5
7   25.700001      28.7      23.9
8   24.299999      27.9      22.6
9   26.000000      30.5      26.0
```

```
10  24.600000       NaN      23.4
11  27.000000       NaN       NaN

ANOVA TABLE
                df      sum_sq    mean_sq         F        PR(>F)
C(variable)    2.0  121.238197  60.619098  52.83944  1.476166e-10
Residual      30.0   34.416961   1.147232       NaN           NaN

 Multiple Comparison of Means - Tukey HSD, FWER=0.05
====================================================
group1 group2 meandiff p-adj   lower   upper  reject
----------------------------------------------------
     0      1    3.985  0.001  2.8546  5.1154   True
     0      2  -0.3614 0.6879 -1.4633  0.7406  False
     1      2  -4.3464  0.001 -5.4998 -3.1929   True
----------------------------------------------------
```



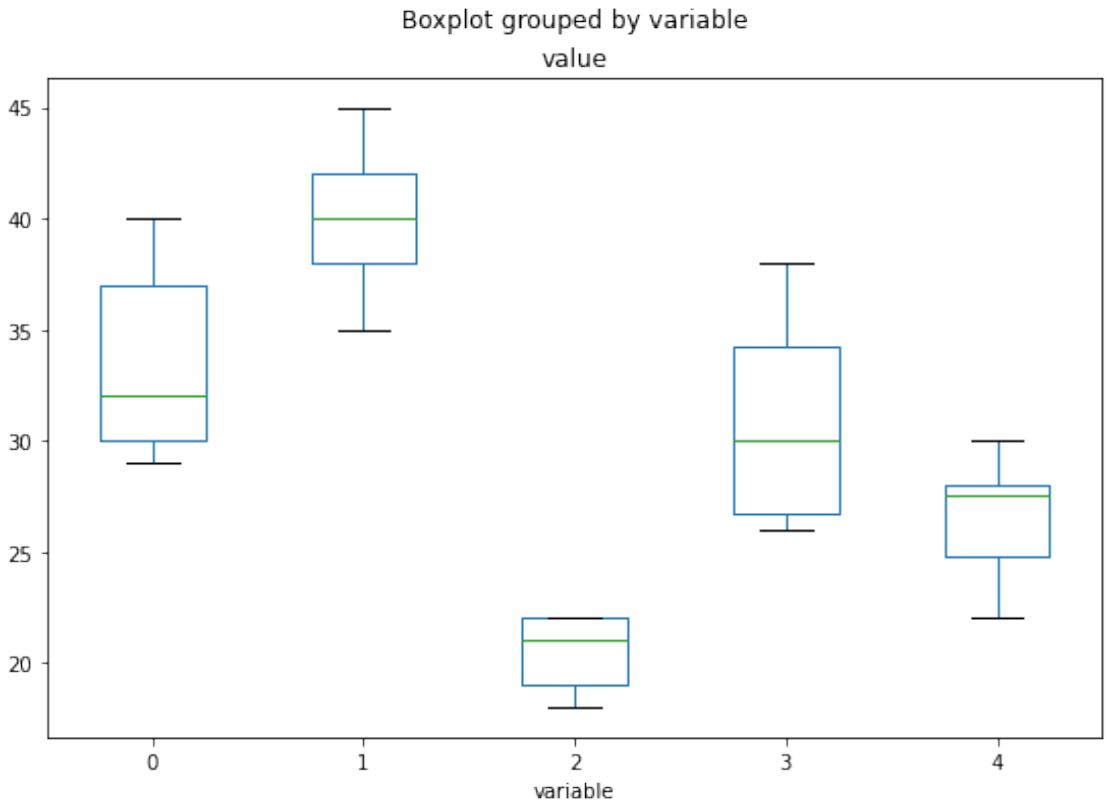
The results are the same as the manual calculations. As we can see, the time taken to perform a numerical task by the Layout 3 is considerably greater than the other 2. Layout 1 and 3 are similar there is some evidence they have the same mean, however layout 3 has a smaller mean.

1.4 Exercise 11.1.20

[33]:
```
# Data loader
data = pd.read_excel('DS 11.1.7.xls') # Tranform file to csv
print(data)

# Data processing
value_vars = []
for i in range(len(data.columns)):
    value_vars.append(i) # count number of colums
data = data.T.reset_index(drop=True).T
proc_data = pd.melt(data, value_vars=value_vars)
proc_data.dropna(subset = ['value'], inplace=True)

# Anova table
model = ols('value ~ C(variable)', proc_data).fit()
print("\nANOVA TABLE\n",anova_lm(model))

# Tukey comparison
comp = pairwise_tukeyhsd(proc_data['value'], proc_data['variable'], alpha=0.05)
print('\n', comp)

# Boxplots
proc_data.boxplot('value', by='variable', grid=False, figsize=(9,6))
plt.show()
```

```
   Treatment1  Treatment2  Treatment3  Treatment4  Treatment5
0        37.0        45.0        22.0        33.0          30
1        40.0        38.0        21.0        38.0          27
2        29.0        40.0        19.0        26.0          22
3        30.0        42.0        18.0        27.0          28
4        32.0        35.0        22.0         NaN          28
5         NaN         NaN         NaN         NaN          24

ANOVA TABLE
                df    sum_sq  mean_sq         F    PR(>F)
C(variable)   4.0  1102.74  275.685  18.50856  0.000002
Residual     20.0   297.90   14.895       NaN       NaN

  Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====================================================
group1 group2 meandiff p-adj   lower    upper   reject
-----------------------------------------------------
     0      1      6.4 0.1039  -0.9046  13.7046  False
     0      2    -13.2  0.001 -20.5046  -5.8954   True
     0      3     -2.6  0.834 -10.3477   5.1477  False
     0      4     -7.1 0.0455 -14.0937  -0.1063   True
     1      2    -19.6  0.001 -26.9046 -12.2954   True
```

```
1      3     -9.0  0.018 -16.7477  -1.2523   True
1      4    -13.5  0.001 -20.4937  -6.5063   True
2      3     10.6 0.0046   2.8523  18.3477   True
2      4      6.1 0.1063  -0.8937  13.0937  False
3      4     -4.5 0.3988 -11.9553   2.9553  False
-----------------------------------------------------
```



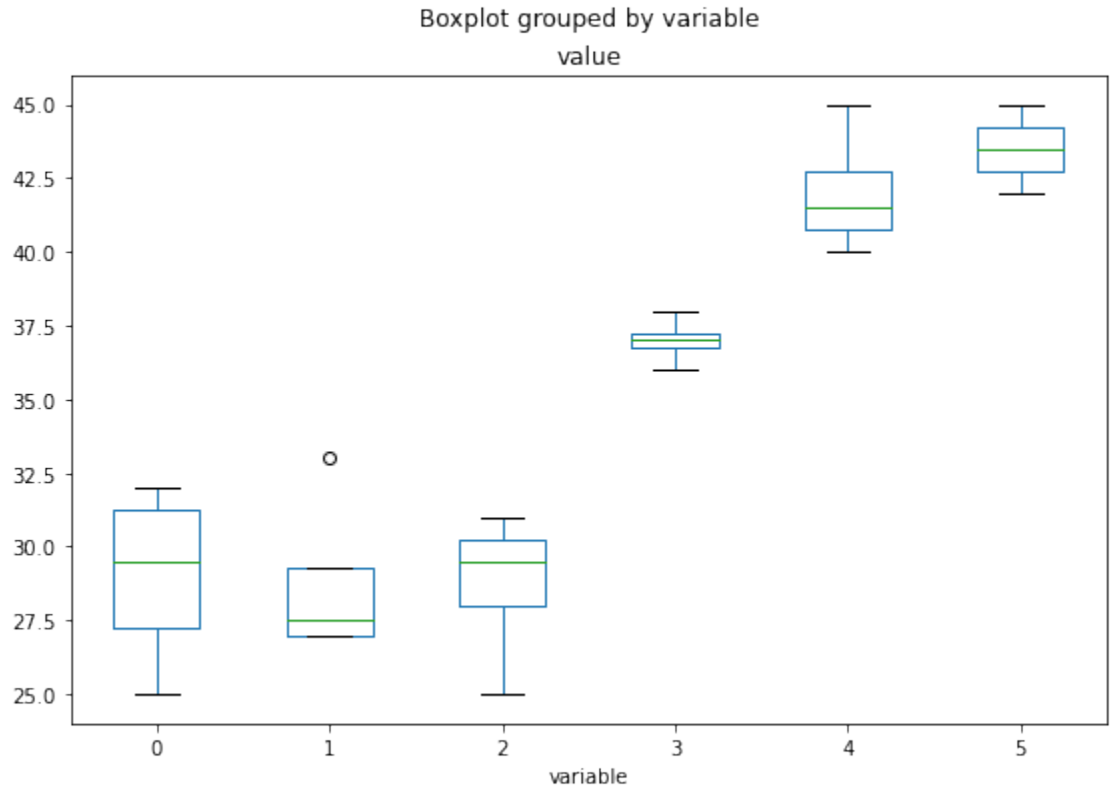
As we can see from the graph, the highest mean is from Treatment 2, while the lowest is from Treatment 3.

1.5 Exercise 11.1.24

[32]:
```python
# Data loader
data = pd.read_excel('DS 11.1.9.xls') # Tranform file to csv
print(data)

# Data processing
value_vars = []
for i in range(len(data.columns)):
    value_vars.append(i) # count number of colums
data = data.T.reset_index(drop=True).T
proc_data = pd.melt(data, value_vars=value_vars)
```

```
proc_data.dropna(subset = ['value'], inplace=True)

# Anova table
model = ols('value ~ C(variable)', proc_data).fit()
print("\nANOVA TABLE\n",anova_lm(model))

# Tukey comparison
comp = pairwise_tukeyhsd(proc_data['value'], proc_data['variable'], alpha=0.05)
print('\n', comp)

# Boxplots
proc_data.boxplot('value', by='variable', grid=False, figsize=(9,6))
plt.show()
```

```
   Location 1  Location 2  Location 3  Location 4  Location 5   Location 6
0          32          27          25          36          42           45
1          28          28          31          37          40           42
2          25          33          30          38          45           44
3          31          27          29          37          41           43

ANOVA TABLE
                df      sum_sq     mean_sq          F        PR(>F)
C(variable)    5.0  956.833333  191.366667  35.695337  9.856804e-09
Residual      18.0   96.500000    5.361111        NaN           NaN

 Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====================================================
group1 group2 meandiff p-adj   lower   upper  reject
-----------------------------------------------------
     0      1    -0.25    0.9 -5.4534  4.9534  False
     0      2    -0.25    0.9 -5.4534  4.9534  False
     0      3      8.0 0.0014  2.7966 13.2034   True
     0      4     13.0  0.001  7.7966 18.2034   True
     0      5     14.5  0.001  9.2966 19.7034   True
     1      2      0.0    0.9 -5.2034  5.2034  False
     1      3     8.25  0.001  3.0466 13.4534   True
     1      4    13.25  0.001  8.0466 18.4534   True
     1      5    14.75  0.001  9.5466 19.9534   True
     2      3     8.25  0.001  3.0466 13.4534   True
     2      4    13.25  0.001  8.0466 18.4534   True
     2      5    14.75  0.001  9.5466 19.9534   True
     3      4      5.0 0.0638 -0.2034 10.2034  False
     3      5      6.5 0.0098  1.2966 11.7034   True
     4      5      1.5    0.9 -3.7034  6.7034  False
-----------------------------------------------------
```



The p-value is small and therefore we can conclude that there is sufficient evidence that E. Coli pollution levels are not the same in all the five locations.

The location with the highest mean E. Coli concentration is the location 6, while location 2 has the lowest mean. The lowest total pollution level is then either in location 1, 2 or 3 while the highest is either at 5 or 6. These data make physical sense, because we notice the concentration of the bacterium level decreases the further downstream we go; which means, we are moving away from the upstream source.