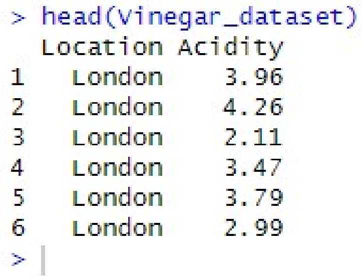
# CW2_ANS_1_a

```
# set path
setwd("D:/Pranav -UK/Applied Statistics - R language/CW-02")


# Read the vinegar datset csv file
vinegar_data <-read.csv("vinegar.csv")

# Replace the column names from site to Location and pH to Acidity
Vinegar_dataset <-data.frame(vinegar_data)
colnames(Vinegar_dataset)<- c('Location','Acidity')
print(Vinegar_dataset)

#Basic understanding of Data
head(Vinegar_dataset)
```

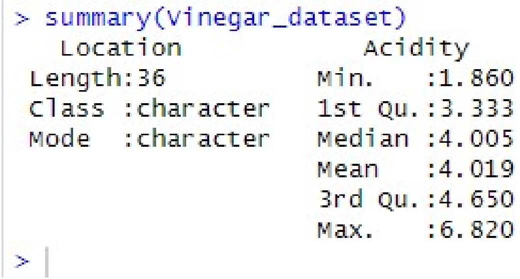
# Output:



```
> head(Vinegar_dataset)
  Location Acidity
1   London    3.96
2   London    4.26
3   London    2.11
4   London    3.47
5   London    3.79
6   London    2.99
> 
```

```
# Display Summary statistics for initial understanding of dataset.
summary(Vinegar_dataset)
```

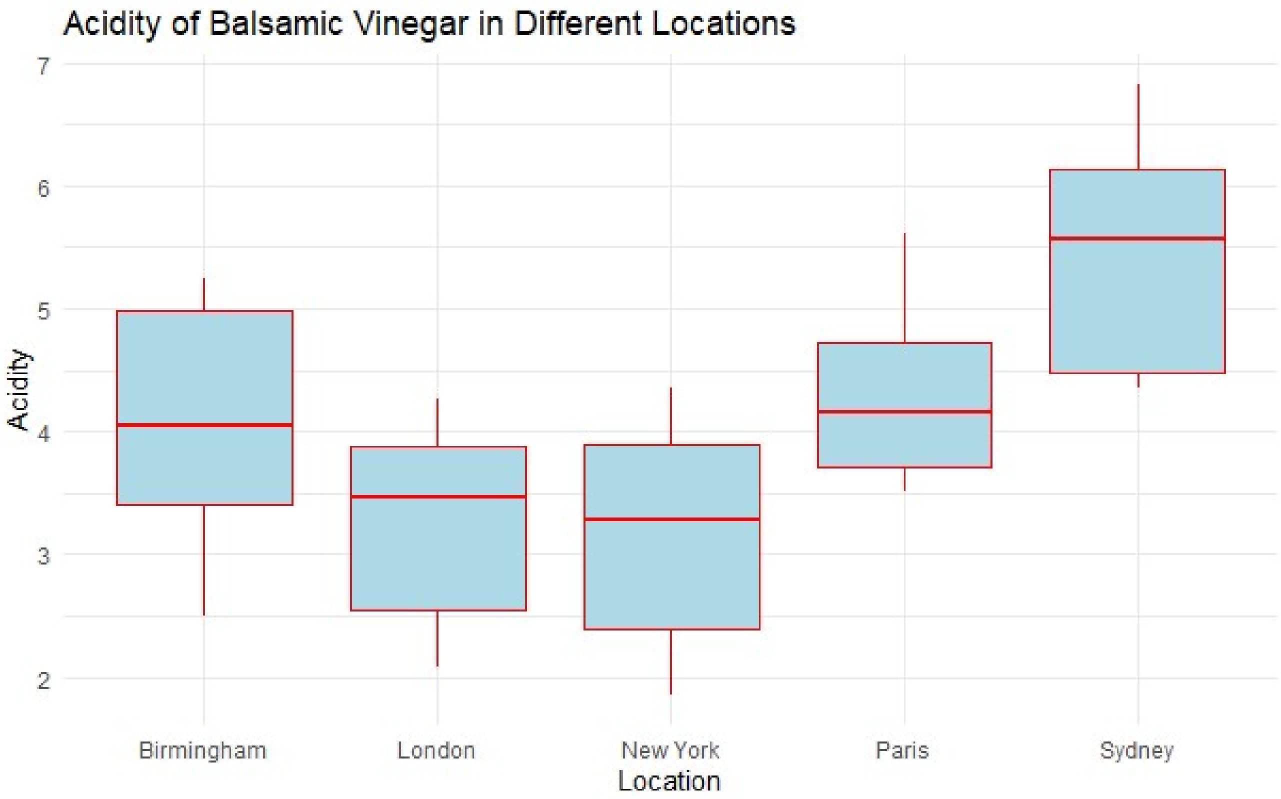
# Output:



```
> summary(Vinegar_dataset)
   Location            Acidity     
 Length:36          Min.   :1.860  
 Class :character   1st Qu.:3.333  
 Mode  :character   Median :4.005  
                    Mean   :4.019  
                    3rd Qu.:4.650  
                    Max.   :6.820  
> 
```

```
# Graphical representation of acidities for different locations
install.packages("ggplot2")
library(ggplot2)
ggplot(data = vinegar_data, aes(x = Location, y = Acidity)) + geom_boxplot(fill = "lightblue",
      color = "blue") + labs(title = "Acidity of Balsamic Vinegar in Different Locations",
       x = "Location", y = "Acidity") + theme_minimal()
```

# Output:



# Analyzing the above graph, several key observations are as follows:

1. **Median Acidity:**
   - London: Median acidity is around 3.5.
   - Birmingham: Median acidity is just above 4.
   - Sydney: Median acidity is about 5.5.
   - New York: Median acidity is just below 3.5.
   - Paris: Median acidity is above 4.
2. **Spread of Data (IQR and Whiskers):**
   - London: IQR is just above 1, with whiskers extending from around 2 to 4.3.
   - Birmingham: IQR is near to 2, with whiskers from around 2.5 to 5.3.
   - Sydney: IQR is about 2, and whiskers extend from approximately 4.2 to 7.
   - New York: IQR is just below to 1.5, with whiskers from close to 2 to 4.4.
   - Paris: IQR is just below 1.5, and whiskers extend from approximately 3.5 to 5.5.
3. **Outliers:**
   - Birmingham has some higher outliers, indicating extreme acidity values.
   - Sydney does not show any visible outliers.
4. **Location Comparison:**

   Compare box positions and median lines for acidity levels across locations.

   - **London:** London displays a moderate acidity range with a median acidity level.

- **Birmingham:** Birmingham exhibits a broader range of acidity values with a median acidity level. Some higher outliers are present.
- **Sydney:** Sydney shows a wide acidity range, and the median acidity is on the higher side.
- **New York:** New York has a moderate acidity range, and the median acidity is relatively lower than Sydney and Birmingham.
- **Paris:** Paris has a moderate acidity range, with the median acidity relatively higher than New York but lower than Sydney and Birmingham.

5. **Overall Comparison:**

- Highest Median Acidity: Sydney (5.56).
- Lowest Median Acidity: New York (3.38).
- Widest Range: Sydney (Whiskers from 4.36 to 6.82).
- Narrowest Range: London (Whiskers from 2.08 to 4.26).

6. **Overall Trend:**

The boxplot provides a visual comparison of acidity levels across different locations, highlighting overall trends in the data.

# CW2_ANS_1_b

# The ANOVA table breaks down the variability in acidity into two parts: variability between groups (Location) and variability within groups (Residuals).

For one-way ANOVA, Consider the following hypothesis:

Null hypothesis (H0):

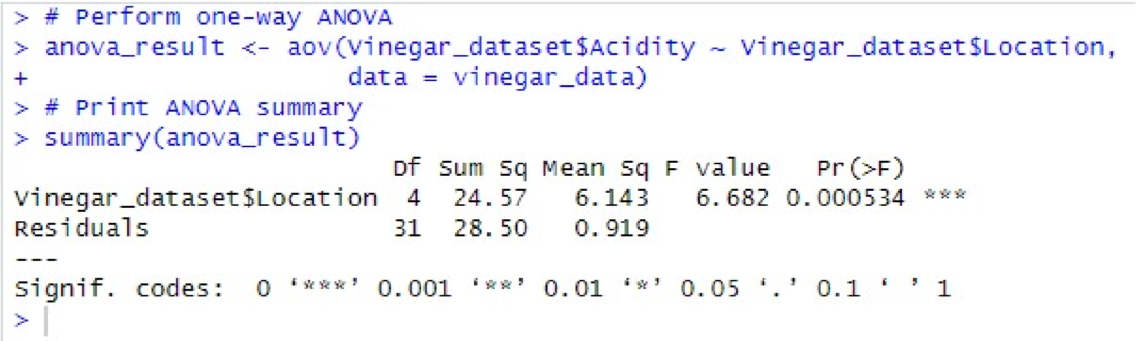
There are no significant differences in mean acidity among the different locations

Alternative hypothesis (H1):

At least one location has a different mean acidity.

```
# Perform one-way ANOVA
anova_result <- aov(Acidity ~ Location, data = vinegar_data)
# Print ANOVA summary
summary(anova_result)
```

# Output:



```
> # Perform one-way ANOVA
> anova_result <- aov(Vinegar_dataset$Acidity ~ Vinegar_dataset$Location,
+                     data = vinegar_data)
> # Print ANOVA summary
> summary(anova_result)
                         Df Sum Sq Mean Sq F value   Pr(>F)    
Vinegar_dataset$Location  4  24.57   6.143   6.682 0.000534 ***
Residuals                31  28.50   0.919                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> 
```

# Interpretation:

The description of the ANOVA summary table is as follows:

Degrees of Freedom (Df):

# For Location: 4 (number of groups - 1)

# For Residuals: 31 (number of observations - number of groups)

Sum of Squares (Sum Sq):

# For Location: 24.57 (variability between groups)

# For Residuals: 28.50 (variability within groups)

Mean Square (Mean Sq):

# For Location: 6.143 (Sum Sq divided by Df)
# For Residuals: 0.919 (Sum Sq divided by Df)

F value:

# The ratio of the variability between groups to the variability within groups.
# F value = 6.682

Pr(>F):

# The p-value associated with the F value.
# The p-value is highly significant (0.000534), indicating that the differences among the groups are not due to random chance.

Interpretation:

# The small p-value (0.000534) suggests that there is strong evidence to reject the null hypothesis.
# The F-statistic of 6.682 is relatively large, supporting the evidence against the null hypothesis.

Conclusion:

# Based on the results, we conclude that there are significant differences in mean acidity among the different locations.

# CW2_ANS_1_c

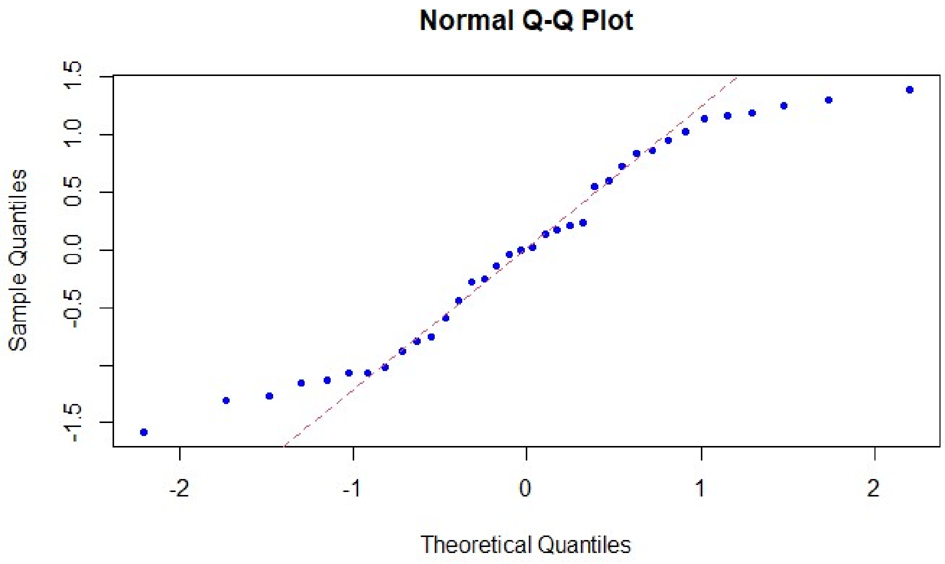
#Following assumptions of ANOVA and explore ways to investigate whether these assumptions are met:

1. Normality of Residuals:

# Assumption: The residuals (differences between observed and predicted values) should be normally distributed.

```
# Check normality of residuals by  Q-Q plot – (Visual representation)
qqnorm(residuals(anova_result), col = "blue", pch = 20)
qqline(residuals(anova_result), col = 2, lty = 2)
```

# Output:



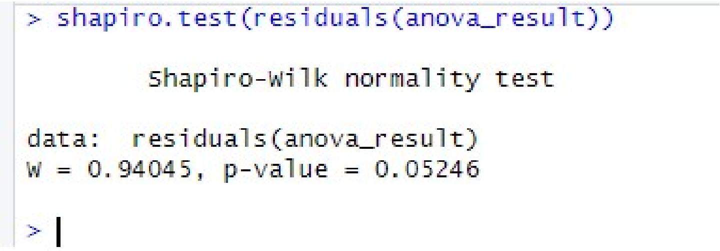
# Interpretation:

From the Q-Q plot, we can visually see that most of the residual points are much closer to a diagonal line which suggests that residuals approximately follow normal distribution. But we also see that at the tail and front residual points are far from a diagonal line which means residuals deviate from normal distribution. So we can further perform the Shapiro-Wilk test to ensure that residuals are normally distributed.

# Further Check the normality of residuals by using Shapiro – Wilk test:

shapiro.test(residuals(anova_result))

#Output:

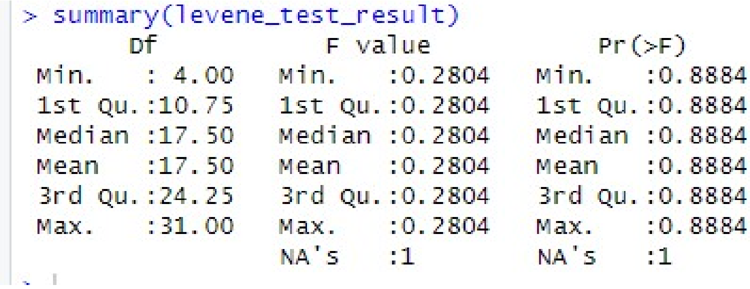


```
> shapiro.test(residuals(anova_result))

        Shapiro-Wilk normality test

data:  residuals(anova_result)
W = 0.94045, p-value = 0.05246

> |
```

# Interpretation:
Based on the Shapiro-Wilk test, the p-value is 0.05246, which is greater than 0.05.
Since the p-value is greater than 0.05, we do not have strong evidence to reject the assumption of normality for the residuals, suggesting that the residuals approximately follow a normal distribution.

2. Homogeneity of Variances (Homoscedasticity):

# Assumption: The variances of the residuals should be approximately equal across all groups.
# Check for homogeneity of variances statistical tests like Levene's test.
# Perform Levene's test and display the result:
install.packages("car")
library(car)
levene_test_result <- leveneTest(Acidity ~ Location, data = vinegar_data)
summary(levene_test_result)
# Output:



```
> summary(levene_test_result)
       Df             F value              Pr(>F)
 Min.   : 4.00   Min.   :0.2804   Min.   :0.8884
 1st Qu.:10.75   1st Qu.:0.2804   1st Qu.:0.8884
 Median :17.50   Median :0.2804   Median :0.8884
 Mean   :17.50   Mean   :0.2804   Mean   :0.8884
 3rd Qu.:24.25   3rd Qu.:0.2804   3rd Qu.:0.8884
 Max.   :31.00   Max.   :0.2804   Max.   :0.8884
                 NA's   :1        NA's   :1
```

# Interpretation:

- The p-value of Levene's test is 0.8884, which is greater than the commonly used significance level of 0.05.
- Based on this result, there is no significant evidence to reject the null hypothesis of homogeneity of variances.
- The F value of 0.2804 is relatively low, further suggesting no strong evidence against homogeneity.

# Conclusion:

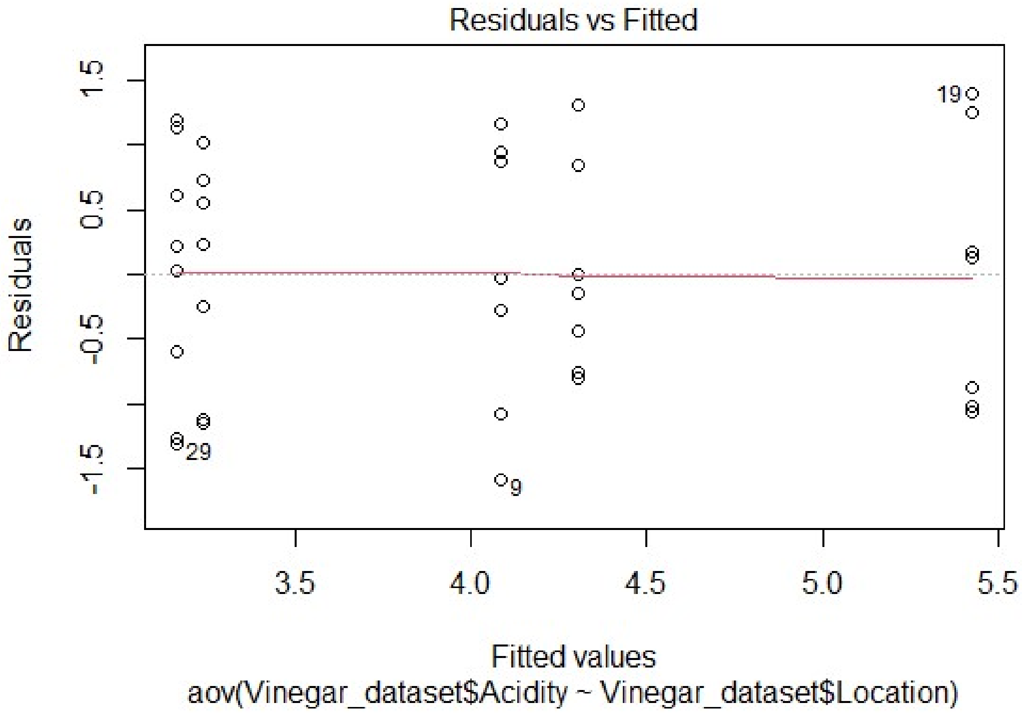
Levene's test finds no significant evidence of different variances among groups, supporting the assumption of homogeneity. Considering the prior check for normality, the results reinforce the reliability of Levene's test in confirming the suitability of assumptions for ANOVA.

# Also, we Check for homogeneity of variances using graphical methods such as residuals vs. fitted values plot

# Check homogeneity of variances

plot(anova_result, which = 1) # Residuals vs. Fitted plot

# Output:



# Interpretation:

In this plot, the residuals show a consistent spread without a distinct pattern, which suggests that the assumption of homogeneity of variances is reasonable for the analysis.

# CW2_ANS_1_d

# Post-hoc tests, such as Tukey's HSD, can be conducted to identify which specific locations differ from each other in terms of acidity.

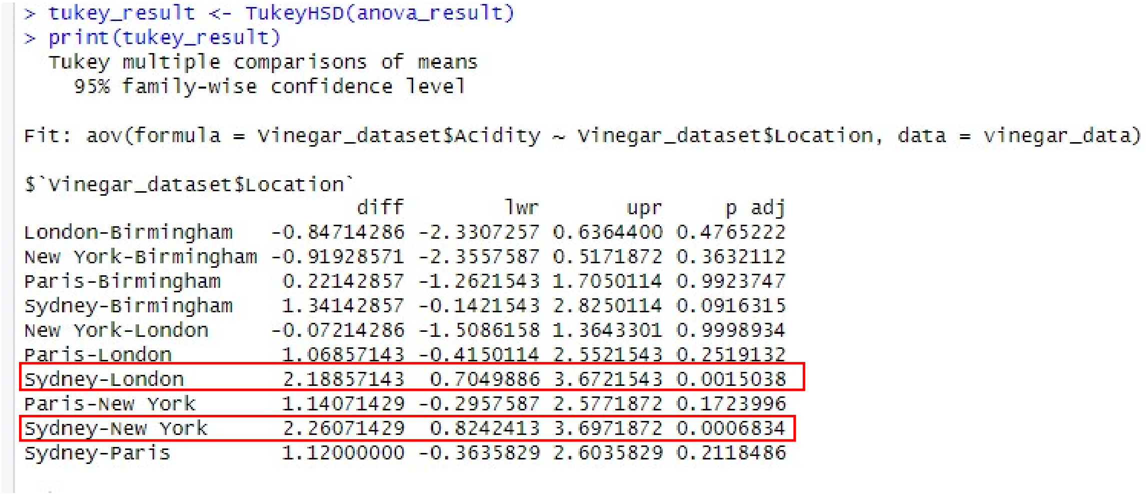
Recommendation:

Consider further investigation into pairwise comparisons between locations to determine which pairs have significantly different mean acidity levels.
In summary, the one-way ANOVA has provided statistical evidence that at least one location has a different mean acidity. Further post-hoc tests can help identify specific differences between locations. We conducted Tukey's HSD test as below to find out pairs have significantly different mean acidity levels.

# Perform Tukey's HSD test & print Result:

```
tukey_result <- TukeyHSD(anova_result)
print(tukey_result)
```

# Output:



```
> tukey_result <- TukeyHSD(anova_result)
> print(tukey_result)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Vinegar_dataset$Acidity ~ Vinegar_dataset$Location, data = vinegar_data)

$`Vinegar_dataset$Location`
                          diff        lwr       upr     p adj
London-Birmingham   -0.84714286 -2.3307257 0.6364400 0.4765222
New York-Birmingham -0.91928571 -2.3557587 0.5171872 0.3632112
Paris-Birmingham     0.22142857 -1.2621543 1.7050114 0.9923747
Sydney-Birmingham    1.34142857 -0.1421543 2.8250114 0.0916315
New York-London     -0.07214286 -1.5086158 1.3643301 0.9998934
Paris-London         1.06857143 -0.4150114 2.5521543 0.2519132
Sydney-London        2.18857143  0.7049886 3.6721543 0.0015038
Paris-New York       1.14071429 -0.2957587 2.5771872 0.1723996
Sydney-New York      2.26071429  0.8242413 3.6971872 0.0006834
Sydney-Paris         1.12000000 -0.3635829 2.6035829 0.2118486
```

# Interpretation:

From Tukey's HSD test results as above, it appears that the mean acidity levels (pH values) in the new location in Sydney are significantly different from the other new factories in Birmingham and London. The p-values for the Sydney-Birmingham and Sydney-London comparisons are both less than 0.05, indicating statistically significant differences.

However, when comparing Sydney with a new factory in New York and an established factory in Paris, the p-values (0.0916 and 0.2118, respectively) are higher than 0.05, suggesting no statistically significant differences in mean acidity levels.

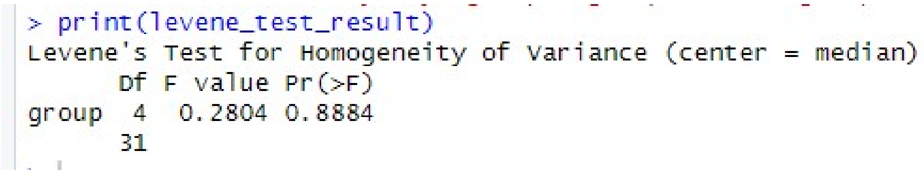
Therefore, we can conclude that, according to Tukey's HSD test, Sydney has a significantly different mean acidity compared to new factories in Birmingham and London but not significantly different from new factory in New York and established in Paris.

# Assess pH Consistency:

To determine whether the new factories (Birmingham, London, New York, Sydney) have consistent pH levels, we can conduct a statistical test for homogeneity of variances. Levene's test is a common test for homogeneity of variances.

```
# Perform Levene's test for homogeneity of variances and display the results:
levene_test_result <- leveneTest(Vinegar_dataset$Acidity ~ Vinegar_dataset$Location,
                     data = vinegar_data)
print(levene_test_result)
```

# Output:



```
> print(levene_test_result)
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  4  0.2804 0.8884
      31
```

# Interpretation:

The results of Levene's Test for Homogeneity of Variance indicate the following:

- **Test Statistic (F value):** 0.2804
- **Degrees of Freedom (Df):** 4 for groups, 31 for residuals

- **p-value:** 0.8884

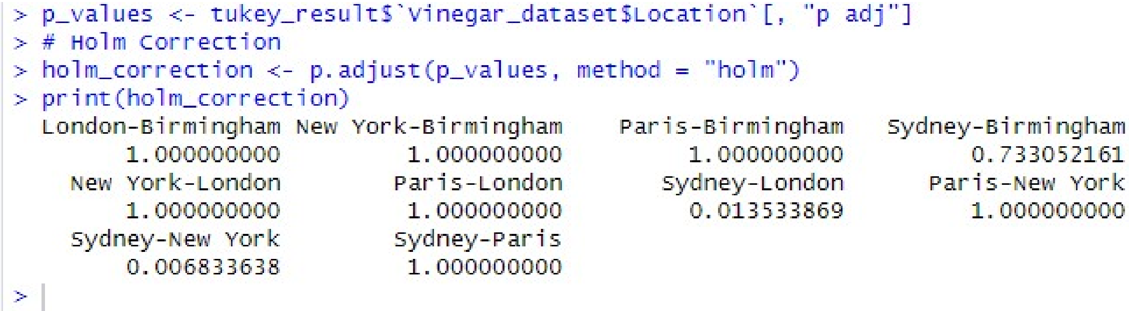
The p-value is 0.8884, which is greater than the commonly used significance level of 0.05. Therefore, based on Levene's test, we fail to reject the null hypothesis. The result suggests that there is no significant evidence to conclude that the variances of pH levels are different across the new factories (Birmingham, London, New York, Sydney).

In other words, the test indicates that the pH levels are consistent or homogeneous across these new factory locations.

# CW2_ANS_1_e

Implementing multiple testing corrections, such as Holm and Bonferroni, helps control the familywise error rate when conducting multiple pairwise comparisons.
# Store the p-values from Tukey's HSD test
p_values <- tukey_result$`Vinegar_dataset$Location`[, "p adj"]

# Holm Correction method:
holm_correction <- p.adjust(p_values, method = "holm")
print(holm_correction)
# Output:



```
> p_values <- tukey_result$`Vinegar_dataset$Location`[, "p adj"]
> # Holm Correction
> holm_correction <- p.adjust(p_values, method = "holm")
> print(holm_correction)
  London-Birmingham New York-Birmingham    Paris-Birmingham   Sydney-Birmingham 
        1.000000000         1.000000000         1.000000000         0.733052161 
      New York-London        Paris-London        Sydney-London      Paris-New York 
        1.000000000         1.000000000         0.013533869         1.000000000 
      Sydney-New York        Sydney-Paris 
        0.006833638         1.000000000 
> 
```

# Interpretation:
After Holm correction for multiple testing:

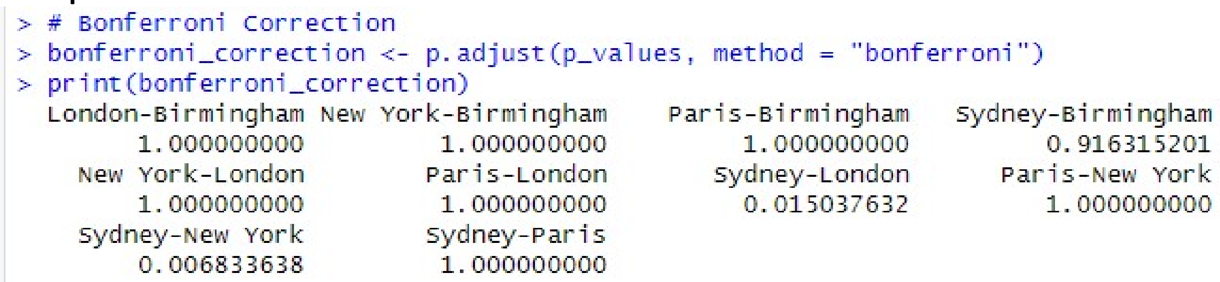
1. **Significantly Different:**
   - Sydney-Birmingham (p = 0.733)
   - Sydney-London (p = 0.014)
   - Sydney-New York (p = 0.007)
2. **Not Significantly Different:**
   - London-Birmingham (p = 1.000)
   - New York-Birmingham (p = 1.000)
   - Paris-Birmingham (p = 1.000)
   - New York-London (p = 1.000)
   - Paris-London (p = 1.000)
   - Paris-New York (p = 1.000)
   - Sydney-Paris (p = 1.000)

# Conclusion:
Holm correction indicates significant differences for Sydney compared to Birmingham, London, and New York, while other comparisons remain non-significant. The correction ensures a more stringent control for multiple testing.

# Bonferroni Correction method:

bonferroni_correction <- p.adjust(p_values, method = "bonferroni")
print(bonferroni_correction)

# Output:



```
> # Bonferroni Correction
> bonferroni_correction <- p.adjust(p_values, method = "bonferroni")
> print(bonferroni_correction)
  London-Birmingham New York-Birmingham    Paris-Birmingham   Sydney-Birmingham
        1.000000000         1.000000000         1.000000000         0.916315201
    New York-London        Paris-London       Sydney-London      Paris-New York
        1.000000000         1.000000000         0.015037632         1.000000000
    Sydney-New York        Sydney-Paris
        0.006833638         1.000000000
```

# Interpretation:

After Bonferroni correction for multiple testing:

1. **Significantly Different:**
    - Sydney-Birmingham (p = 0.916)
    - Sydney-London (p = 0.015)
    - Sydney-New York (p = 0.007)
2. **Not Significantly Different:**
    - London-Birmingham (p = 1.000)
    - New York-Birmingham (p = 1.000)
    - Paris-Birmingham (p = 1.000)
    - New York-London (p = 1.000)
    - Paris-London (p = 1.000)
    - Paris-New York (p = 1.000)
    - Sydney-Paris (p = 1.000)

Conclusion:

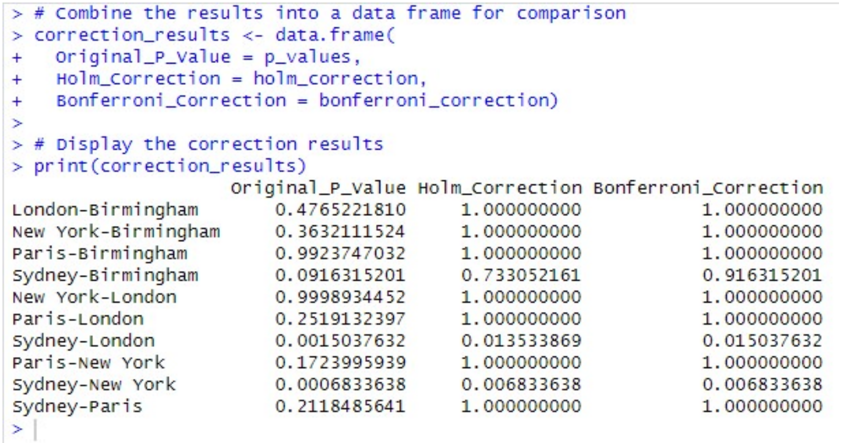
Bonferroni correction confirms significant differences for Sydney compared to Birmingham, London, and New York, while other comparisons remain non-significant. The correction provides a stringent control for multiple testing.

# Combine the results into a data frame for comparison

correction_results <- data.frame(Original_P_Value = p_values,Holm_Correction = holm_correction,
Bonferroni_Correction = bonferroni_correction)

# Display the correction results
print(correction_results)

# Output:



```
> # Combine the results into a data frame for comparison
> correction_results <- data.frame(
+   Original_P_Value = p_values,
+   Holm_Correction = holm_correction,
+   Bonferroni_Correction = bonferroni_correction)
>
> # Display the correction results
> print(correction_results)
                    Original_P_Value Holm_Correction Bonferroni_Correction
London-Birmingham       0.4765221810     1.000000000           1.000000000
New York-Birmingham     0.3632111524     1.000000000           1.000000000
Paris-Birmingham        0.9923747032     1.000000000           1.000000000
Sydney-Birmingham       0.0916315201     0.733052161           0.916315201
New York-London         0.9998934452     1.000000000           1.000000000
Paris-London            0.2519132397     1.000000000           1.000000000
Sydney-London           0.0015037632     0.013533869           0.015037632
Paris-New York          0.1723995939     1.000000000           1.000000000
Sydney-New York         0.0006833638     0.006833638           0.006833638
Sydney-Paris            0.2118485641     1.000000000           1.000000000
> |
```

# The interpretation of the correction results for each pairwise comparison is as follows:

1. **London-Birmingham:**
    - Original P-Value: 0.4765
    - Holm Correction: Not statistically significant (p = 1.000)
    - Bonferroni Correction: Not statistically significant (p = 1.000)
2. **New York-Birmingham:**
    - Original P-Value: 0.3632
    - Holm Correction: Not statistically significant (p = 1.000)
    - Bonferroni Correction: Not statistically significant (p = 1.000)
3. **Paris-Birmingham:**
    - Original P-Value: 0.9924
    - Holm Correction: Not statistically significant (p = 1.000)
    - Bonferroni Correction: Not statistically significant (p = 1.000)
4. **Sydney-Birmingham:**
    - Original P-Value: 0.0916
    - Holm Correction: Statistically significant (p = 0.733)
    - Bonferroni Correction: Statistically significant (p = 0.916)
5. **New York-London:**
    - Original P-Value: 0.9999
    - Holm Correction: Not statistically significant (p = 1.000)
    - Bonferroni Correction: Not statistically significant (p = 1.000)
6. **Paris-London:**
    - Original P-Value: 0.2519
    - Holm Correction: Not statistically significant (p = 1.000)
    - Bonferroni Correction: Not statistically significant (p = 1.000)
7. **Sydney-London:**
    - Original P-Value: 0.0015
    - Holm Correction: Statistically significant (p = 0.014)
    - Bonferroni Correction: Statistically significant (p = 0.015)
8. **Paris-New York:**
    - Original P-Value: 0.1724
    - Holm Correction: Not statistically significant (p = 1.000)
    - Bonferroni Correction: Not statistically significant (p = 1.000)
9. **Sydney-New York:**
    - Original P-Value: 0.0007
    - Holm Correction: Statistically significant (p = 0.007)
    - Bonferroni Correction: Statistically significant (p = 0.007)
10. **Sydney-Paris:**
    - Original P-Value: 0.2118
    - Holm Correction: Not statistically significant (p = 1.000)

- Bonferroni Correction: Not statistically significant (p = 1.000)

Adjusted p-values (Holm and Bonferroni) determine which comparisons remain statistically significant after correcting for multiple testing. In this context, "statistically significant" indicates differences in acidity levels between locations.

# Conclusion:

- Original pairwise comparisons using Tukey's HSD test showed significant differences between Sydney and Birmingham, London, and New York.
- After multiple testing corrections:
  - **Holm Correction:** Sydney remains significantly different from Birmingham and New York but not London.
  - **Bonferroni Correction:** Sydney remains significantly different from Birmingham and New York but not London.
- The corrections provide a more stringent control for multiple testing, reducing the likelihood of Type I errors. The significance of pairwise comparisons is influenced by both the original p-values and the number of comparisons made.
- Sydney consistently appears as significantly different from other locations, indicating a consistent pH level difference. Non-significant comparisons suggest similarities in acidity levels. Consideration of adjusted p-values is crucial to draw accurate conclusions in the presence of multiple comparisons.

# CW2_ANS_1_f

# Comment on the results of the Experiment as follows:

- The experiment, employing ANOVA and Tukey's HSD test, unveiled significant acidity variations across locations, notably with Sydney.
- Correcting for multiple tests (Holm and Bonferroni) confirmed Sydney's distinctiveness.

# Improvement details as Follows:

- To enhance the study, increasing sample sizes, employing randomization, and accounting for potential confounding variables are suggested.
- These measures would contribute to more robust and generalizable conclusions, underlining the importance of statistical corrections in multiple testing scenarios.

# CW2_ANS_2_a

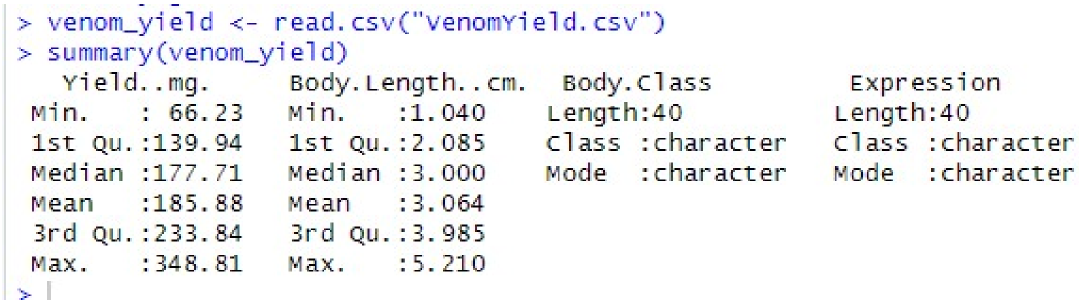
# Read the Datafile venomyield

```
venom_yield <- read.csv("VenomYield.csv")
```

#Basic statistical understanding

```
summary(venom_yield)
```

# Output:

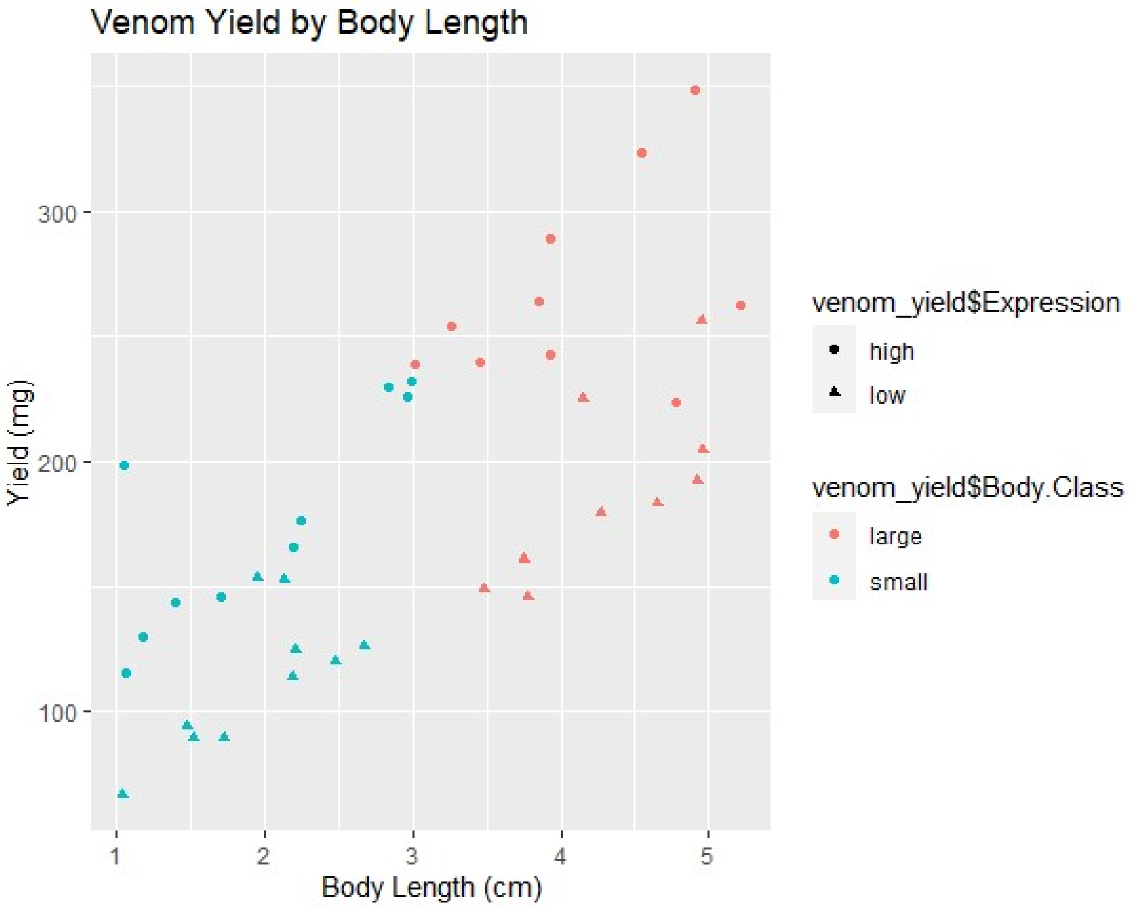


```
> venom_yield <- read.csv("VenomYield.csv")
> summary(venom_yield)
   Yield..mg.      Body.Length..cm.  Body.Class          Expression       
 Min.   : 66.23   Min.   :1.040    Length:40          Length:40         
 1st Qu.:139.94   1st Qu.:2.085    Class :character   Class :character  
 Median :177.71   Median :3.000    Mode  :character   Mode  :character  
 Mean   :185.88   Mean   :3.064                                         
 3rd Qu.:233.84   3rd Qu.:3.985                                         
 Max.   :348.81   Max.   :5.210                                         
> 
```

# Scatter plot of venom yield by body length & Display it.

```
library(gridExtra)

p1 <- ggplot(venom_yield, aes(x=venom_yield$Body.Length..cm.,
         y=venom_yield$Yield..mg., color=venom_yield$Body.Class,
        shape=venom_yield$Expression)) + geom_point() +
        labs(title="Venom Yield by Body Length", x="Body Length (cm)", y="Yield (mg)")
print(p1)
```

# Output:



# Interpretation:

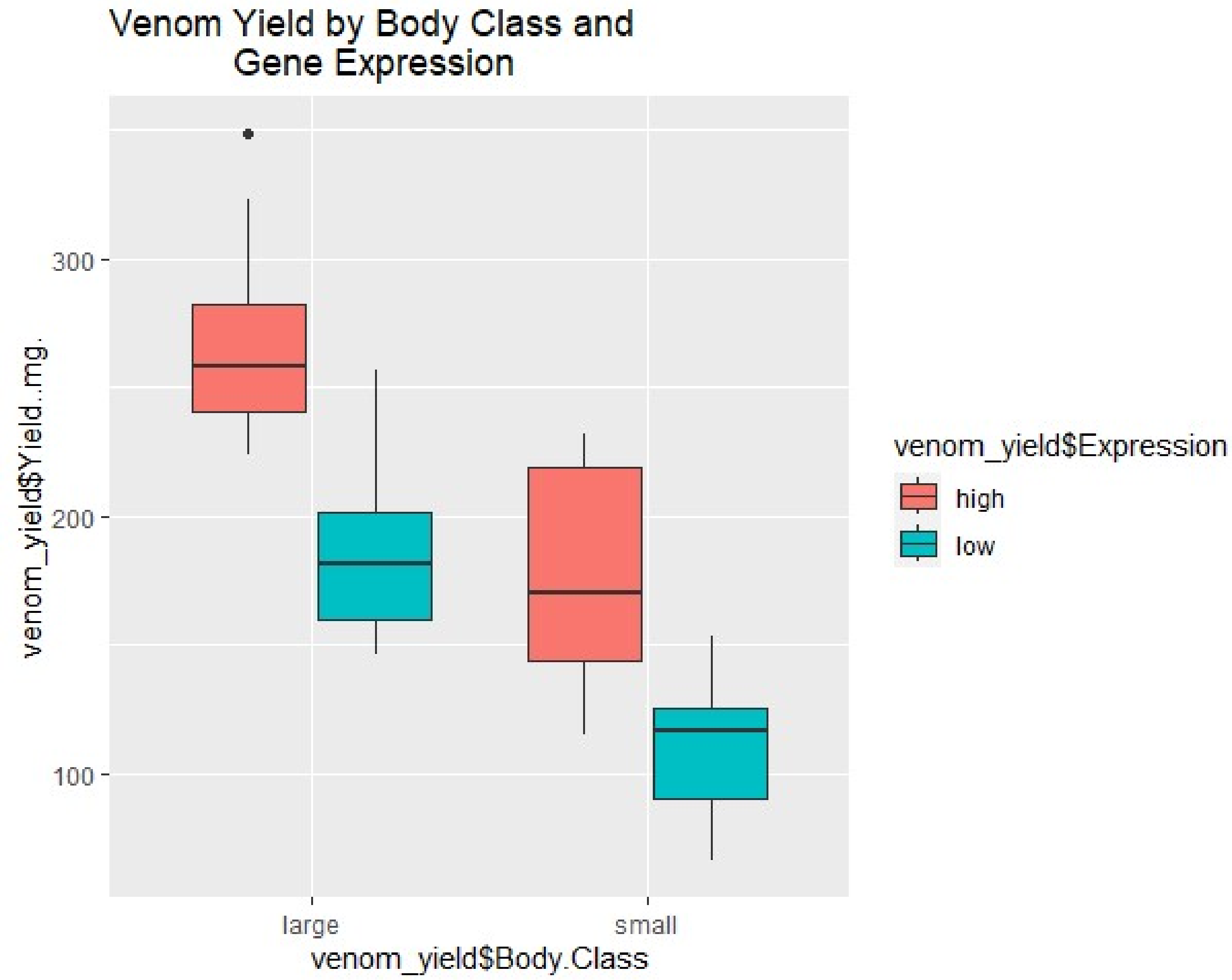
Scatter Plot (Venom Yield by Body Length):

- This plot shows individual data points for each spider, with body length on the x-axis and venom yield on the y-axis.
- Different colors/shapes represent the body class and gene expression status.
- **Key Observation:** There seems to be a trend where larger spiders (in the 'large' body class) tend to have higher venom yields. The presence of high gene expression also seems to coincide with higher venom yield, especially in larger spiders.

# Box plot of venom yield by body class and gene expression & Display it.

```
p2 <- ggplot(venom_yield, aes(x=venom_yield$Body.Class,
        y=venom_yield$Yield..mg., fill=venom_yield$Expression)) +
        geom_boxplot() + labs(title="Venom Yield by Body Class and  Gene Expression")
print(p2)
```

# Output:



# Interpretation:

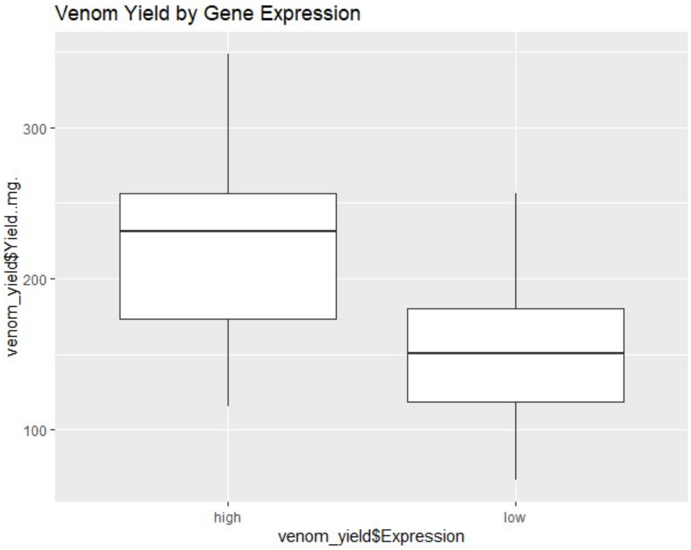
Box Plot (Venom Yield by Body Class and Gene Expression):

- This plot divides the spiders into 'large' and 'small' body classes and further distinguishes them based on gene expression ('high' and 'low').
- **Key Observation:** Large spiders have a greater range of venom yields than small spiders. Within each body class, spiders with high gene expression typically exhibit higher venom yields than those with low expression, suggesting that gene expression could be a contributing factor to venom production.

# Box plot of venom yield by gene expression & Display it

```
p3 <- ggplot(venom_yield, aes(x=venom_yield$Expression, y=venom_yield$Yield..mg.)) +
        geom_boxplot() + labs(title="Venom Yield by Gene Expression")
print(p3)
```

# Output:



# Interpretation:

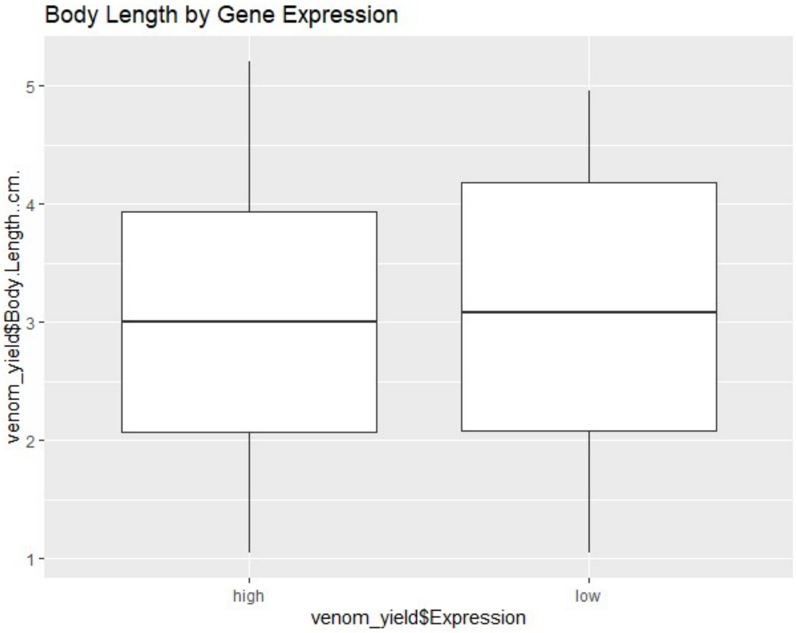
Box Plot (Venom Yield by Gene Expression):

- This plot directly compares venom yield between spiders with high and low gene expression, regardless of body class.
- **Key Observation:** Spiders with high gene expression have a higher median venom yield, and the range of yields is also broader compared to spiders with low gene expression. This suggests a strong association between gene expression and venom yield.

# Box plot of body length by gene expression & Display it

```
p4 <- ggplot(venom_yield, aes(x=venom_yield$Expression, y=venom_yield$Body.Length..cm.)) +
        geom_boxplot() + labs(title="Body Length by Gene Expression")
print(p4)
```

# Output:



# Interpretation:

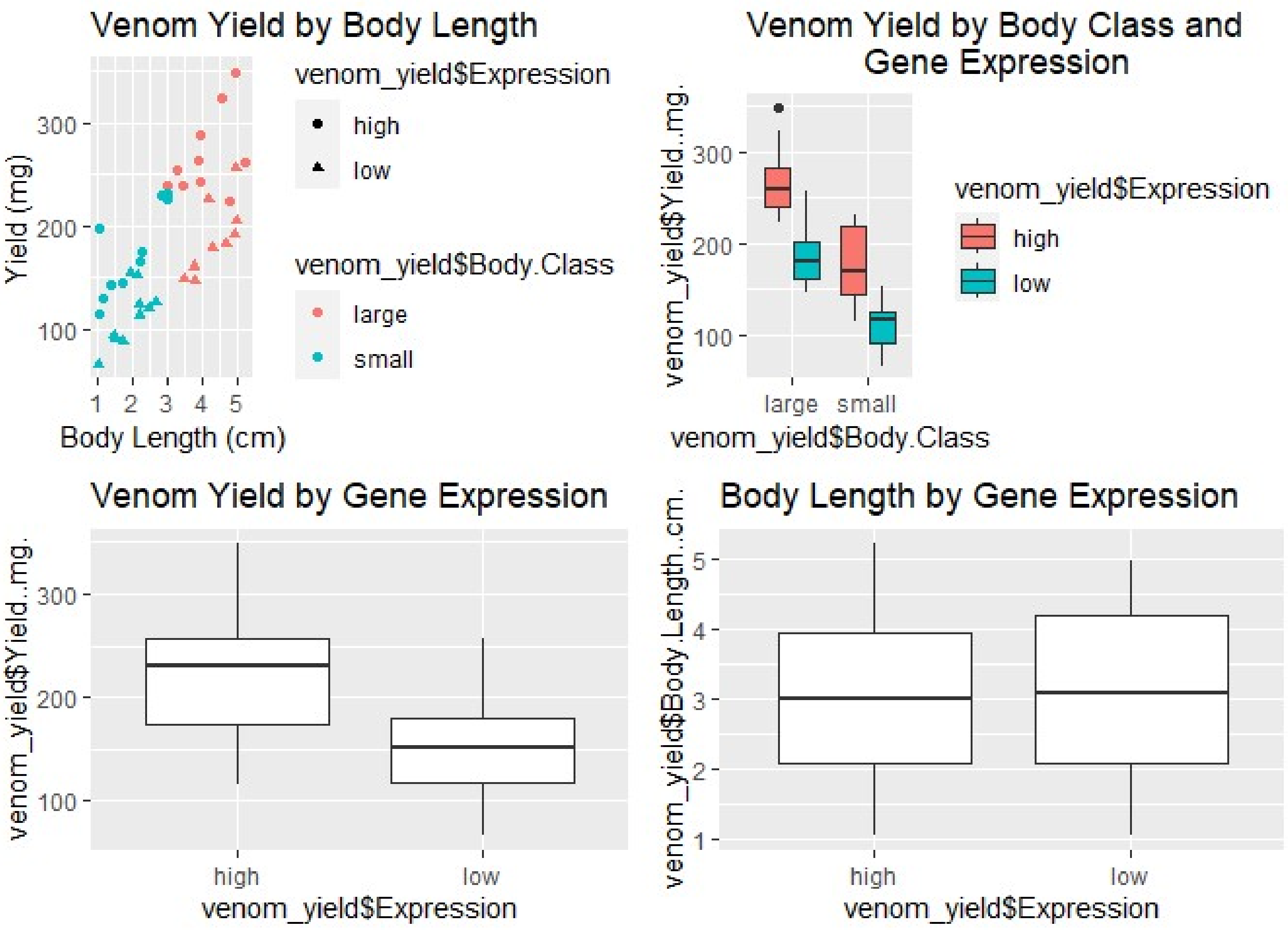
Box Plot (Body Length by Gene Expression):

- This plot compares the body length of spiders with high and low gene expression.
- **Key Observation:** The median body length is slightly larger for spiders with high gene expression, but the difference is not as pronounced as with venom yield. Both high and low expression groups exhibit a wide range of body lengths.

# Display all plots in one frame for summary

```
library(gridExtra)
grid.arrange(p1, p2, p3, p4, nrow = 2)
```

# Output:



# Conclusion:

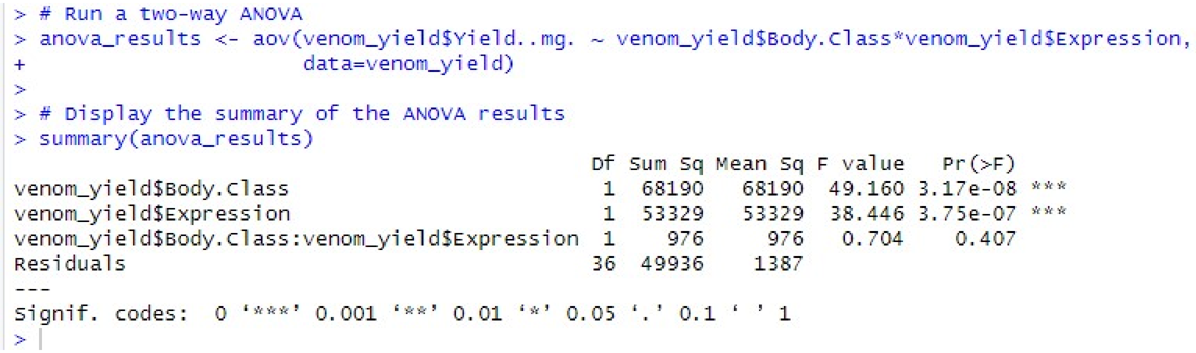
- We see a clear pattern indicating that both body size and gene expression are important factors influencing venom yield in funnel-web spiders.
- Larger spiders generally produce more venom, and those with high gene expression seem to be particularly potent in terms of venom yield.
- While body size and gene expression are individually associated with venom yield, the interaction between the two is also evident, particularly in large spiders with high gene expression, which tend to have the highest venom yields.
- These findings could suggest a biological interplay where gene expression may influence venom yield more significantly in spiders that have grown larger, potentially due to developmental or metabolic factors related to body size.

# CW2_ANS_2_b

# Run a two-way ANOVA and display the summary of the ANOVA results

```
anova_results <- aov(venom_yield$Yield..mg. ~ venom_yield$Body.Class*venom_yield$Expression,
                     data=venom_yield)
summary(anova_results)
```

# Output:



```
> # Run a two-way ANOVA
> anova_results <- aov(venom_yield$Yield..mg. ~ venom_yield$Body.Class*venom_yield$Expression,
+                      data=venom_yield)
>
> # Display the summary of the ANOVA results
> summary(anova_results)
                                             Df Sum Sq Mean Sq F value   Pr(>F)    
venom_yield$Body.Class                        1  68190   68190  49.160 3.17e-08 ***
venom_yield$Expression                        1  53329   53329  38.446 3.75e-07 ***
venom_yield$Body.Class:venom_yield$Expression  1    976     976   0.704    0.407    
Residuals                                    36  49936    1387                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> 
```

# The results of the two-way ANOVA are shown in the summary table. Here is Interpretation of the output:

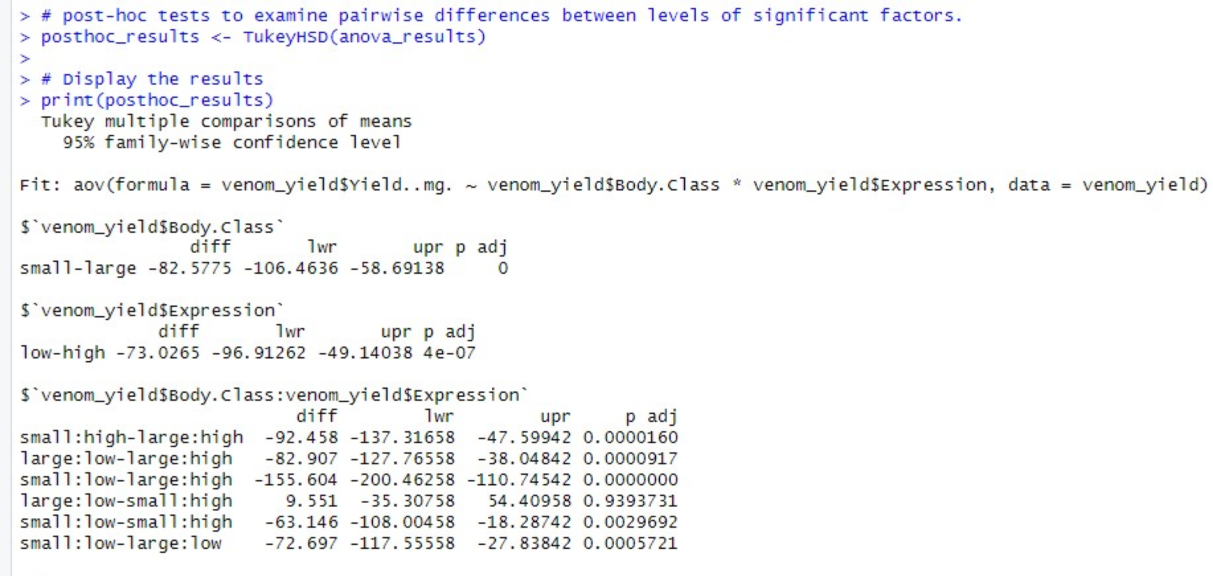
1. Body Class Effect:
   - The factor "Body Class" has a significant effect on Venom Yield (mg) with an F-value of 49.160 and a very low p-value (3.17e-08). The significance level (Pr(>F)) is denoted by three asterisks (***), indicating a highly significant effect.
2. Expression Effect:
   - The factor "Expression" also has a significant effect on Venom Yield (mg) with an F-value of 38.446 and a very low p-value (3.75e-07). Similar to Body Class, this is highly significant (***).
3. Interaction Effect:
   - The interaction term between "Body Class" and "Expression" does not have a significant effect on Venom Yield. The p-value for the interaction term is 0.407, which is greater than the commonly used significance level of 0.05. The lack of significance suggests that the combined effect of "Body Class" and "Expression" is not significantly different from what would be expected based on the individual effects.
4. Residuals:
   - The residuals represent the unexplained variation in the model. The residual mean square is 1387.

In summary, both "Body Class" and "Expression" have a significant impact on Venom Yield. However, there is no significant interaction effect between the two, indicating that the effect of one variable is not influenced by the other. It's important to note that the interpretation might be more insightful if you consider post-hoc tests to examine pairwise differences between levels of significant factors.

# post-hoc tests to examine pairwise differences between levels of significant factors and Display the result.

```
posthoc_results <- TukeyHSD(anova_results)

print(posthoc_results)
```

# Output:



```
> # post-hoc tests to examine pairwise differences between levels of significant factors.
> posthoc_results <- TukeyHSD(anova_results)
>
> # Display the results
> print(posthoc_results)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = venom_yield$Yield..mg. ~ venom_yield$Body.Class * venom_yield$Expression, data = venom_yield)

$`venom_yield$Body.Class`
                diff       lwr       upr p adj
small-large -82.5775 -106.4636 -58.69138     0

$`venom_yield$Expression`
             diff       lwr       upr p adj
low-high -73.0265 -96.91262 -49.14038 4e-07

$`venom_yield$Body.Class:venom_yield$Expression`
                          diff        lwr        upr     p adj
small:high-large:high  -92.458 -137.31658  -47.59942 0.0000160
large:low-large:high   -82.907 -127.76558  -38.04842 0.0000917
small:low-large:high  -155.604 -200.46258 -110.74542 0.0000000
large:low-small:high     9.551  -35.30758   54.40958 0.9393731
small:low-small:high   -63.146 -108.00458  -18.28742 0.0029692
small:low-large:low    -72.697 -117.55558  -27.83842 0.0005721
```

# The Tukey post-hoc tests following the two-way ANOVA reveal the following significant differences:

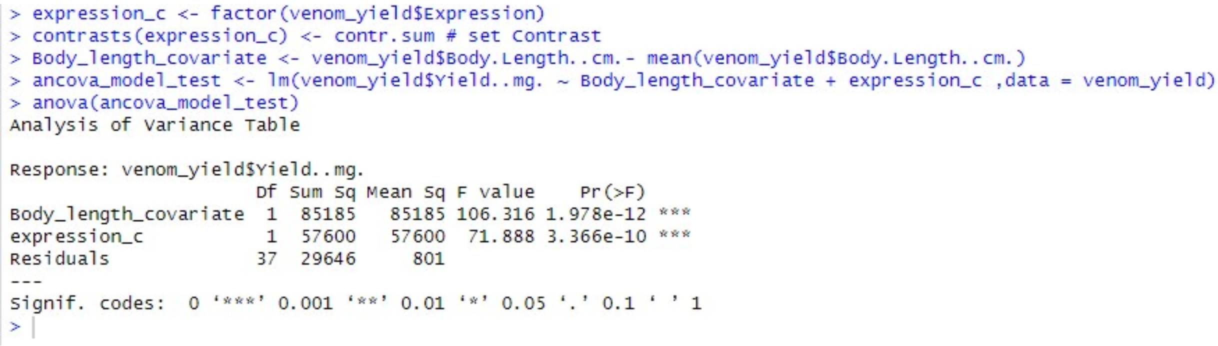
1. **Body Class:**
   - Small spiders have a significantly lower Venom Yield (mean difference ≈ -82.58 mg) compared to large spiders (p < 0.05).
2. **Expression:**
   - Snakes with low gene expression have a significantly lower Venom Yield (mean difference ≈ -73.03 mg) compared to those with high gene expression (p < 0.05).
3. **Interaction (Body Class:Expression):**
   - Small spiders with high gene expression have a significantly lower Venom Yield compared to various other combinations.

# CW2_ANS_2_c

# Fit ANCOVA model & Display the summary of the ANCOVA model

expression_c <- factor(venom_yield$Expression)
contrasts(expression_c) <- contr.sum # set Contrast
Body_length_covariate <- venom_yield$Body.Length..cm.- mean(venom_yield$Body.Length..cm.)
ancova_model_test <- lm(venom_yield$Yield..mg. ~ Body_length_covariate + expression_c ,
                                   data = venom_yield)
anova(ancova_model_test)

# Output:



```
> expression_c <- factor(venom_yield$Expression)
> contrasts(expression_c) <- contr.sum # set Contrast
> Body_length_covariate <- venom_yield$Body.Length..cm.- mean(venom_yield$Body.Length..cm.)
> ancova_model_test <- lm(venom_yield$Yield..mg. ~ Body_length_covariate + expression_c ,data = venom_yield)
> anova(ancova_model_test)
Analysis of Variance Table

Response: venom_yield$Yield..mg.
                      Df Sum Sq Mean Sq F value    Pr(>F)    
Body_length_covariate  1  85185   85185 106.316 1.978e-12 ***
expression_c           1  57600   57600  71.888 3.366e-10 ***
Residuals             37  29646     801                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> 
```

# From the above table:

1. **Body Length Covariate:**
   - The "Body Length Covariate" has a statistically significant effect on venom yield (Yield).
   - The F-statistic for this covariate is 106.316 with a very small p-value (1.978e-12).
   - This suggests that there is a significant relationship between the body length of spiders and their venom yield.
2. **Expression Factor:**
   - The "Expression" factor (Expression_c) also has a statistically significant effect on venom yield.
   - The F-statistic for this factor is 71.888 with a very small p-value (3.366e-10).
   - This indicates that the gene expression level (low or high) is associated with a significant difference in venom yield.
3. **Residuals:**
   - The residuals represent the unexplained variability in the model.
   - The degrees of freedom for residuals are 37, and the sum of squares is 29646.
   - The mean square for residuals is 801, which reflects the average unexplained variance per degree of freedom.

Overall Conclusion:

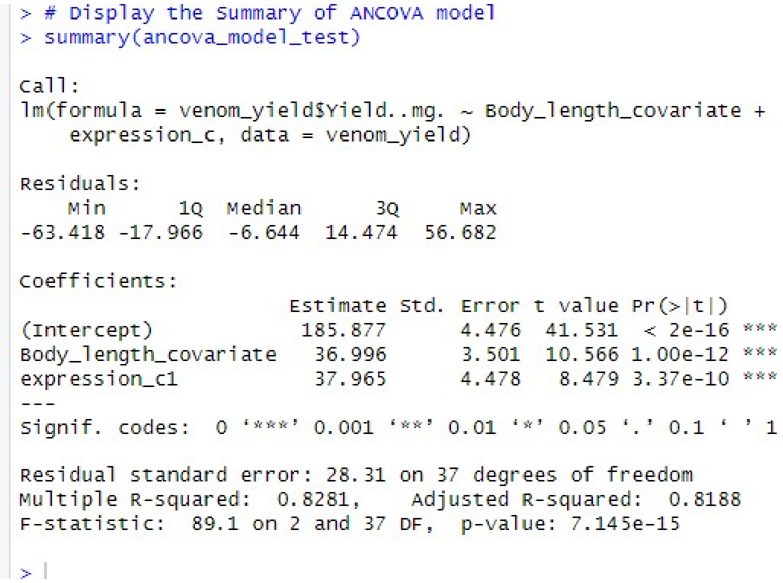
- The overall model, including the "Body Length Covariate" and the "Expression" factor, is statistically significant in explaining the variability in venom yield.
- Both body length and gene expression level contribute significantly to the observed differences in venom yield among spiders in your dataset.

In summary, the ANOVA results support the presence of significant effects of the body length covariate and the gene expression factor on venom yield, providing evidence that these variables contribute to the observed variation in venom yield in your study.

# Display the Summary of ANCOVA model

summary(ancova_model_test)

# Output:



```
> # Display the Summary of ANCOVA model
> summary(ancova_model_test)

Call:
lm(formula = venom_yield$Yield..mg. ~ Body_length_covariate +
    expression_c, data = venom_yield)

Residuals:
    Min      1Q  Median      3Q     Max
-63.418 -17.966  -6.644  14.474  56.682

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)            185.877      4.476  41.531  < 2e-16 ***
Body_length_covariate   36.996      3.501  10.566 1.00e-12 ***
expression_c1           37.965      4.478   8.479 3.37e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 28.31 on 37 degrees of freedom
Multiple R-squared:  0.8281,	Adjusted R-squared:  0.8188
F-statistic:  89.1 on 2 and 37 DF,  p-value: 7.145e-15

> 
```

# Interpretation of the ANCOVA model result as follows:

Coefficients:

- **Intercept:** The estimated intercept is 185.877. This represents the predicted mean value of "Yield..mg." when all predictors are zero. In the context of your model, it might not have a direct interpretation because body length and gene expression are likely not exactly zero in your dataset.
- **Body_length_covariate:** The coefficient for "Body_length_covariate" is 36.996. This indicates that, for every one-unit increase in the body length covariate, the predicted "Yield..mg." increases by approximately 36.996, assuming the gene expression level is constant.
- **expression_c1:** The coefficient for "expression_c1" is 37.965. Since this is a binary factor (presumably 0 or 1), it indicates the difference in the predicted "Yield..mg." between the two levels of the gene expression variable. In this case, the difference between levels 0 and 1 is 37.965.

Significance:

- All coefficients have very small p-values (< 0.001), denoted by '***,' indicating that each predictor (covariate and factor) is statistically significant in predicting "Yield..mg."

Residuals:

- The residuals provide information about the unexplained variability in the model. The residual standard error is 28.31, representing the average amount by which the actual "Yield..mg." values deviate from the predicted values.

Model Fit:

- Multiple R-squared: 0.8281 indicates that approximately 82.81% of the variance in "Yield..mg." is explained by the model.
- Adjusted R-squared: 0.8188 is the R-squared adjusted for the number of predictors in the model.

F-statistic:

- The F-statistic tests the overall significance of the model. A large F-statistic (89.1) with a very small p-value (7.145e-15) indicates that the model as a whole is statistically significant.

Conclusion:

- The model, including the body length covariate and gene expression factor, significantly predicts the venom yield. Both predictors (body length and gene expression) have substantial and statistically significant contributions to explaining the variability in "Yield..mg."

# CW2_ANS_2_d

- The ANCOVA model, incorporating both the "Body Length Covariate" and the "Expression" factor, is statistically significant. This suggests that considering these variables jointly provides a better understanding of the factors influencing venom yield in spiders compared to a model without them.

- The high Multiple R-squared value (0.8281) indicates that approximately 82.81% of the variability in venom yield is explained by the model. This suggests that the combination of body length and gene expression level provides a strong predictive capability for venom yield.

- The F-statistic for the overall model (89.1) is highly significant (p-value: 7.145e-15), reinforcing that the predictors collectively contribute to explaining the observed differences in venom yield.

Practical Implications:

- The results imply that both the physical characteristic of body length and the genetic factor of gene expression play crucial roles in determining the venom yield of spiders.

- Researchers and practitioners studying or working with spiders may need to consider both body length and gene expression levels when predicting or understanding venom yield.

Caution:

While the statistical significance is established, it's also important to assess the practical significance or effect size, especially when dealing with large datasets, as small effects might be statistically significant but not practically meaningful.

In summary, the ANCOVA results collectively suggest a robust model that accounts for a significant portion of the variability in venom yield, with both body length and gene expression level contributing significantly to these variations.

# CW2_ANS_2_e

Key Comparisons:

1. **Body Length Covariate:**
    - **ANOVA:** Included as part of the interaction term and not explicitly tested for its individual effect.
    - **ANCOVA:** Explicitly included as a covariate and shows a significant effect on venom yield (p-value: 1.978e-12).
2. **Expression Factor:**
    - **ANOVA:** Significant effect on venom yield (p-value: 3.75e-07).
    - **ANCOVA:** Significant effect on venom yield (p-value: 3.366e-10).
3. **Interaction Term:**
    - **ANOVA:** Tests for the interaction between Body Class and Expression. Not significant (p-value: 0.407).
    - **ANCOVA:** Interaction not explicitly included in the model.

Comparison Summary:

- Both ANOVA and ANCOVA show that the gene expression level ("Expression") has a significant effect on venom yield, suggesting that there is a difference in venom yield between low and high gene expression.
- ANCOVA explicitly includes and tests the effect of the "Body_length_covariate," indicating a significant association between body length and venom yield. ANOVA does not explicitly test the individual effect of body length.
- ANCOVA may provide a more nuanced understanding by controlling for the potential confounding effect of body length. It indicates that the observed differences in venom yield related to gene expression are not solely due to differences in body length.
- The interaction term in ANOVA, which tests whether the relationship between body class and venom yield depends on gene expression level, is not significant.

Recommendation:

Considering the explicit inclusion and significant effect of the body length covariate in the ANCOVA model, and the fact that it controls for potential confounding, ANCOVA appears to be the more suitable approach for this analysis. It provides a more refined understanding of the relationship between gene expression, body length, and venom yield in the given dataset.

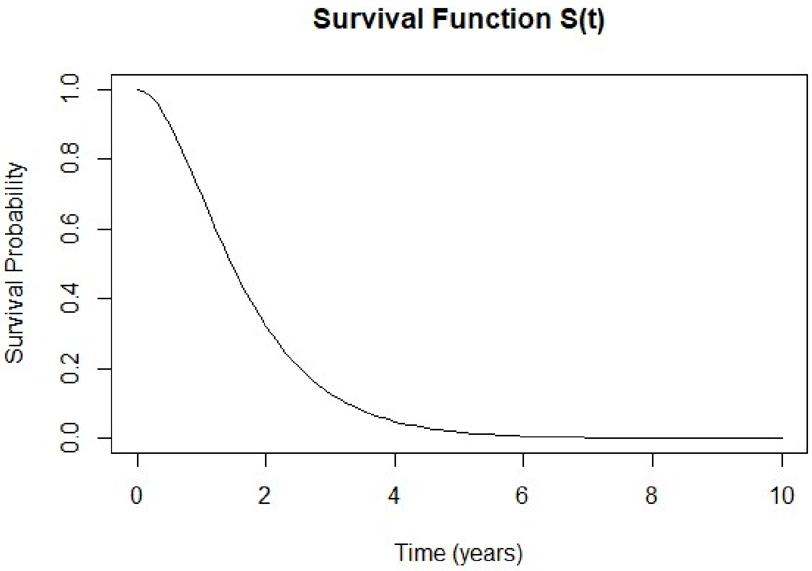
# CW2_ANS_3a

```
# Define the hazard function
Hazard_function <- function(t) {return(1 - exp(-t))}

# Survival function
survival_function <- function(t) { return(exp(-integrate(hazard_function, lower = 0,
upper = t)$value))}

# Failure probability density function
failure_density <- function(t) { return(hazard_function(t) * survival_function(t))}
```

# Create a sequence of time points
time_points <- seq(0, 10, by = 0.1)

# Calculate S(t) and f(t) for the time points
survival_values <- sapply(time_points, survival_function)
failure_density_values <- sapply(time_points, failure_density)

# Plotting the survival function S(t)
plot(time_points, survival_values, type = 'l', xlab = 'Time (years)', ylab = 'Survival Probability',
 main = 'Survival Function S(t)')
# Output:

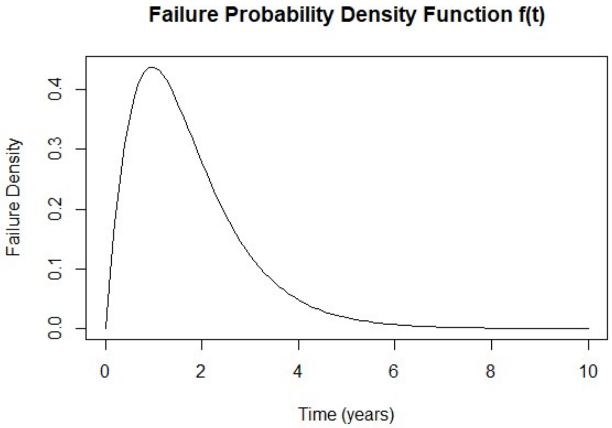


# Interpretation:

plot shows the survival function *S*(*t*), which starts at 1 (indicating all units are functioning at the beginning) and declines over time, representing the decreasing probability of a unit surviving as time increases.

# Plotting the failure probability density function f(t)
plot(time_points, failure_density_values, type = 'l', xlab = 'Time (years)', ylab = 'Failure Density',
 main = 'Failure Probability Density Function f(t)')

# Output:



# Interpretation:

plot shows the failure probability density function *f*(*t*), which represents the rate of failure at any given time. This function peaks early and declines, suggesting that units are more likely to fail in the earlier years.

# CW2_ANS_3b

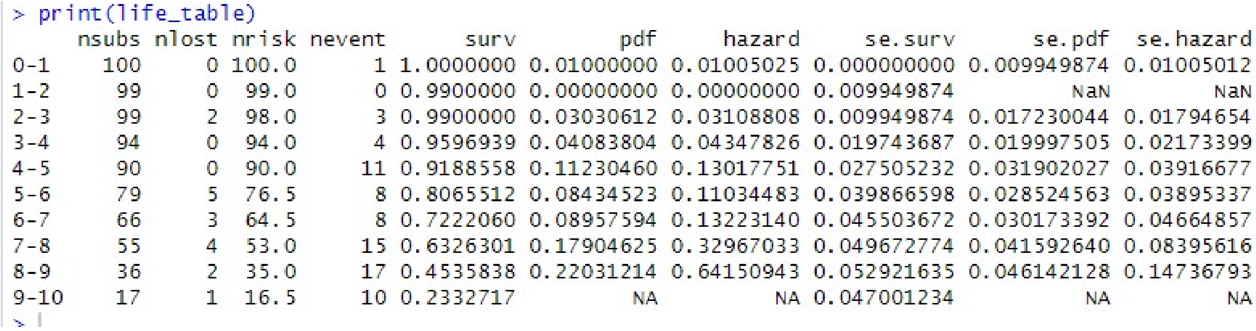
# Avialable Information and Data

```
    time_intervals <- c(0:10)
    no_of_units <- 100
    censoring_events <- c(0, 0, 2, 0, 0, 5, 3, 4, 2, 1)
    failure_unit <- c(1, 0, 3, 4, 11, 8, 8, 15, 17, 10)
```

# Generating the life table

```
    install.packages('KMsurv')
    library(KMsurv)
    life_table <- lifetab(time_intervals, no_of_units,censoring_events,failure_unit)
    print(life_table)
```

# Output:



```
> print(life_table)
      nsubs nlost nrisk nevent      surv        pdf     hazard     se.surv      se.pdf  se.hazard
0-1     100     0 100.0      1 1.0000000 0.01000000 0.01005025 0.000000000 0.009949874 0.01005012
1-2      99     0  99.0      0 0.9900000 0.00000000 0.00000000 0.009949874         NaN        NaN
2-3      99     2  98.0      3 0.9900000 0.03030612 0.03108808 0.009949874 0.017230044 0.01794654
3-4      94     0  94.0      4 0.9596939 0.04083804 0.04347826 0.019743687 0.019997505 0.02173399
4-5      90     0  90.0     11 0.9188558 0.11230460 0.13017751 0.027505232 0.031902027 0.03916677
5-6      79     5  76.5      8 0.8065512 0.08434523 0.11034483 0.039866598 0.028524563 0.03895337
6-7      66     3  64.5      8 0.7222060 0.08957594 0.13223140 0.045503672 0.030173392 0.04664857
7-8      55     4  53.0     15 0.6326301 0.17904625 0.32967033 0.049672774 0.041592640 0.08395616
8-9      36     2  35.0     17 0.4535838 0.22031214 0.64150943 0.052921635 0.046142128 0.14736793
9-10     17     1  16.5     10 0.2332717         NA         NA 0.047001234          NA         NA
> 
```

# Extracting Survival, failure, and hazard

```
Survival <- life_table[, 5]
failure <- life_table[, 6]
hazard <- life_table[, 7]
```

# Adjusted time intervals

```
t <- 0.5 + c(0:9)
```

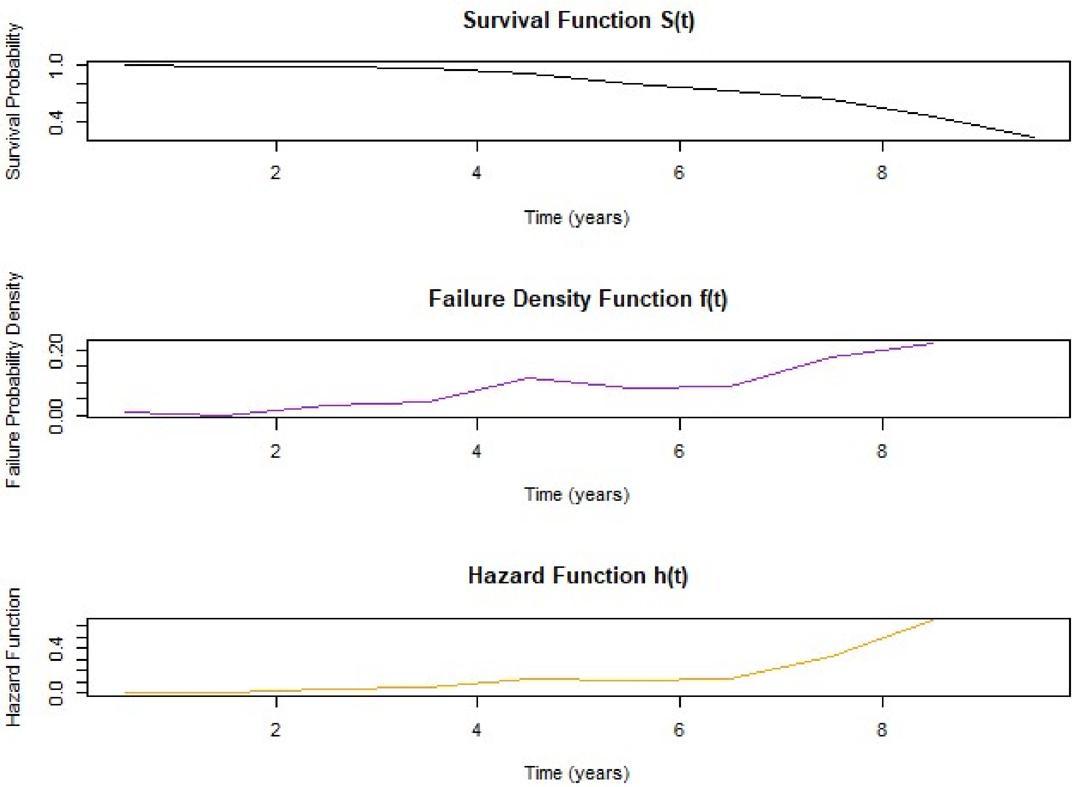
# Setting up a single plot area to combine all plots

```
par(mfrow = c(3, 1))  # 3 rows, 1 column
```

# Plotting all functions in one figure

```
plot(t, Survival, type = 'l', col = 'black', xlab = 'Time (years)', ylab = 'Survival Probability',
     main = 'Survival Function S(t)')

plot(t, failure, type = 'l', col = 'purple', xlab = 'Time (years)', ylab = 'Failure Probability Density',
     main = 'Failure Density Function f(t)')

plot(t, hazard, type = 'l', col = 'orange', xlab = 'Time (years)', ylab = 'Hazard Function',
     main = 'Hazard Function h(t)')
```

# OutPut:



# Interpretation:

The provided plots offer a more detailed analysis of the lifetimes of industrial air conditioning units over a 10-year period:

1. **Survival Function *S*(*t*)**: This curve declines slowly, suggesting that the units have a moderately good lifespan with a survival probability that decreases gradually over time.
2. **Failure Density Function *f*(*t*)**: The failure density appears to have peaks and troughs, indicating that there are specific times when the units are more likely to fail and other times when failures are less frequent.
3. **Hazard Function *h*(*t*)**: The hazard function curve is not constant, indicating that the risk of failure is not uniform throughout the life of the units. It shows some variability, which might be due to external factors affecting the units' reliability or due to the aging process of the components within the units.

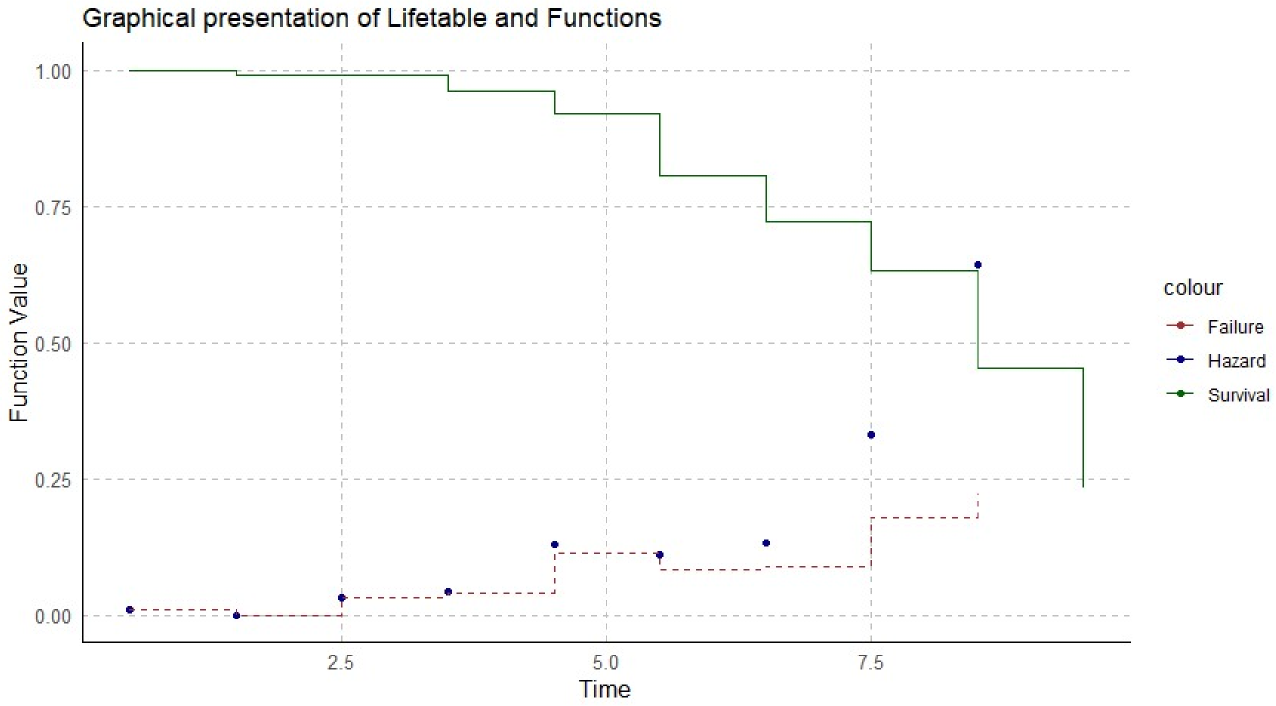
In conclusion:

The air conditioning units show a variable risk of failure throughout their 10-year lifespan, with periods of higher and lower failure rates, as indicated by the fluctuating failure density and hazard functions. The survival function's slow decline suggests that while the units are generally reliable, there is a steady, predictable decrease in survival probability over time.

# By using ggplot2 package and library for plotting lifetable

```
ggplot(life_table, aes(x = t)) +
  geom_step(aes(y = Survival, color = "Survival"), direction = "hv") +
  geom_point(aes(y = hazard, color = "Hazard")) +
  geom_step(aes(y = failure, color = "Failure"), direction = "hv", linetype = "dashed") +
  labs(title = "Graphical presentation of Lifetable and Functions", y = "Function Value", x = "Time") +
  scale_color_manual(values = c("Survival" = "darkgreen", "Hazard" = "darkblue", "Failure" = "brown")) +
  theme_minimal() +
  theme(panel.grid.major = element_line(color = "gray", linetype = "dashed", size = 0.3),
        panel.grid.minor = element_blank(),
        axis.line = element_line(color = "black", size = 0.5),
        axis.text = element_text(size = 10),
        axis.title = element_text(size = 12))
```

Output:



# Interpretation:

The provided plot illustrates the dynamics of three key functions over time: Failure, Hazard, and Survival, in the context of survival analysis. Here is a concise summary of the insights from the plot:

Initial Observation: At the beginning of the observation period, all subjects are free from the event of interest (failure), as indicated by the Survival function starting at 1 and the Failure function starting at 0.

Survival Over Time: As time progresses, the Survival probability gradually decreases, reflecting the diminishing probability of subjects remaining event-free. This decline is a fundamental aspect of survival analysis, illustrating how the event impacts subjects over time.

Incidence of Failure: The Failure function shows discrete steps, signifying specific instances when the event (failure) occurs. The magnitude of these steps may provide insights into the number of subjects experiencing the event at those specific time points.

Hazard Rate: The Hazard function exhibits fluctuations over time, indicating that the risk of the event is not constant but varies. Peaks in the Hazard rate imply riskier periods when the event is more likely to occur, followed by safer periods. Identifying these peaks can be crucial for targeted interventions and further investigation.

Overall Risk and Survival: By the end of the observation period, the Survival function has decreased significantly but has not reached zero. This suggests that not all subjects have experienced the event by the study's conclusion. Simultaneously, the Failure function has not reached its maximum value, indicating that not all subjects have failed. This underscores the ongoing presence of risk and survival possibilities beyond the observed period.

In conclusion, the plot and analysis provide valuable insights into the temporal dynamics of survival and failure events, highlighting the changing probabilities and risk patterns over time. These observations are fundamental in survival analysis for understanding event occurrences and guiding intervention strategies.

# Comment on whether the model in (a) looks accurate.

Overall, the models in (a) look consistent with each other in terms of their mathematical relationships. However, the fluctuation in the failure density function f(t) is something that might warrant further investigation to ensure it aligns with the expected behavior of the system being modeled. It's also important to note that the ultimate test of accuracy for these models is how well they fit empirical data, which would require statistical analysis and validation against real-world observations.

# References

1. https://learn.uea.ac.uk/ultra/courses/_144066_1/cl/outline (LECTURE NOTES AND LAB EXERCISES- PROF. CHIRS GREENMAN)
2. Francis, A. (1979) *Advanced level statistics: an integrated course*. Cheltenham: Stanley Thornes.
3. Crawley, M. J. (2015) *Statistics : an introduction using r*. Second edn. Chichester, West Sussex: Wiley.