Lab 08

2024-09-19

Lecture Recap

```r
library(knitr)
```

ANOVA

When we want to compare two population means, we can perform a two-sample t-test.

- What if we have more than 2 independent groups?

ANOVA: a generalisation of the two-sided two-sample t-test.

ANOVA Decomposition

```r
knitr::include_graphics("images/lab1.png")
```

Decompose the total variation in the response variable Y (**Total Sum of Squares**) into two parts:

$$\sum_{i=1}^{g}\sum_{j=1}^{n_i}(y_{ij}-\bar{y}_{\bullet\bullet})^2 = \underbrace{\sum_{i=1}^{g}\underbrace{\sum_{j=1}^{n_i}(y_{ij}-\bar{y}_{i\bullet})^2}_{=(n_i-1)s_i^2}}_{\text{sample variances}} + \underbrace{\sum_{i=1}^{g} n_i(\bar{y}_{i\bullet}-\bar{y}_{\bullet\bullet})^2}_{\text{sample means}}$$

$$= \text{Residual SS} + \text{Treatment SS}$$

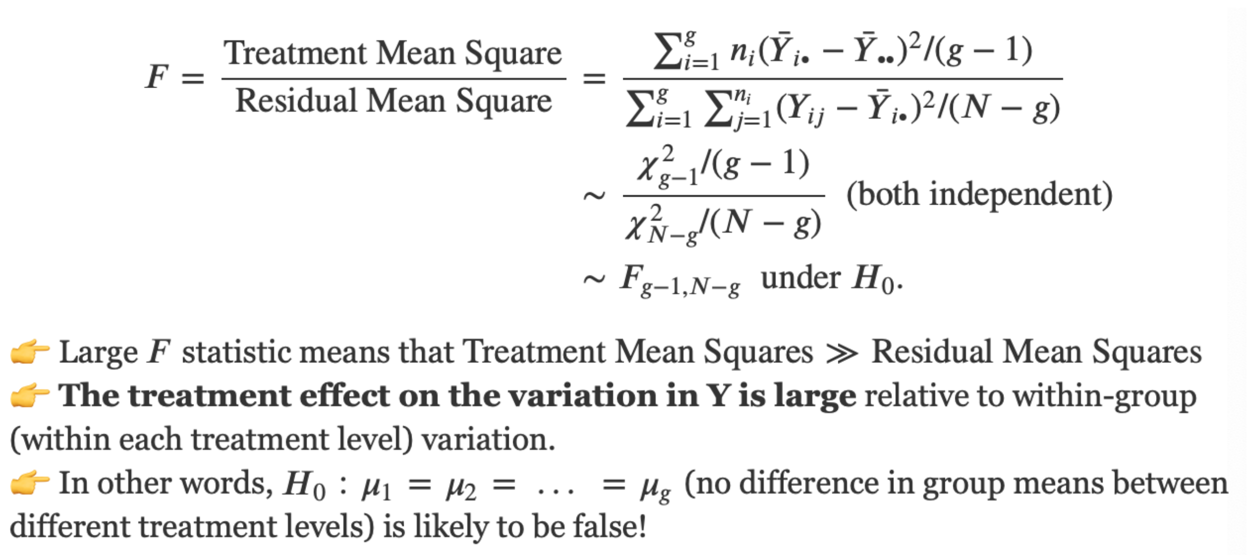
⚠️ **Intuition**

Don't be scared by the math!
Put simply, we're just decomposing the total variation in Y (measurement) into **two sources of variation: Treatment vs Residuals**
- **Treatment SS** measures between-group variation (between different treatment levels): **Treatment effect**
- **Residual SS** measures within-group variation (within each treatment level): **Non-treatment effect**

Then if we compute the ratio of these two sources of variation, we can see **if the treatment has an important effect on the response variable Y** (after normalising them by degrees of freedom, which account for the number of groups and sample size)
👉 The F statistic

The F statistic

```
knitr::include_graphics("images/lab2.png")
```

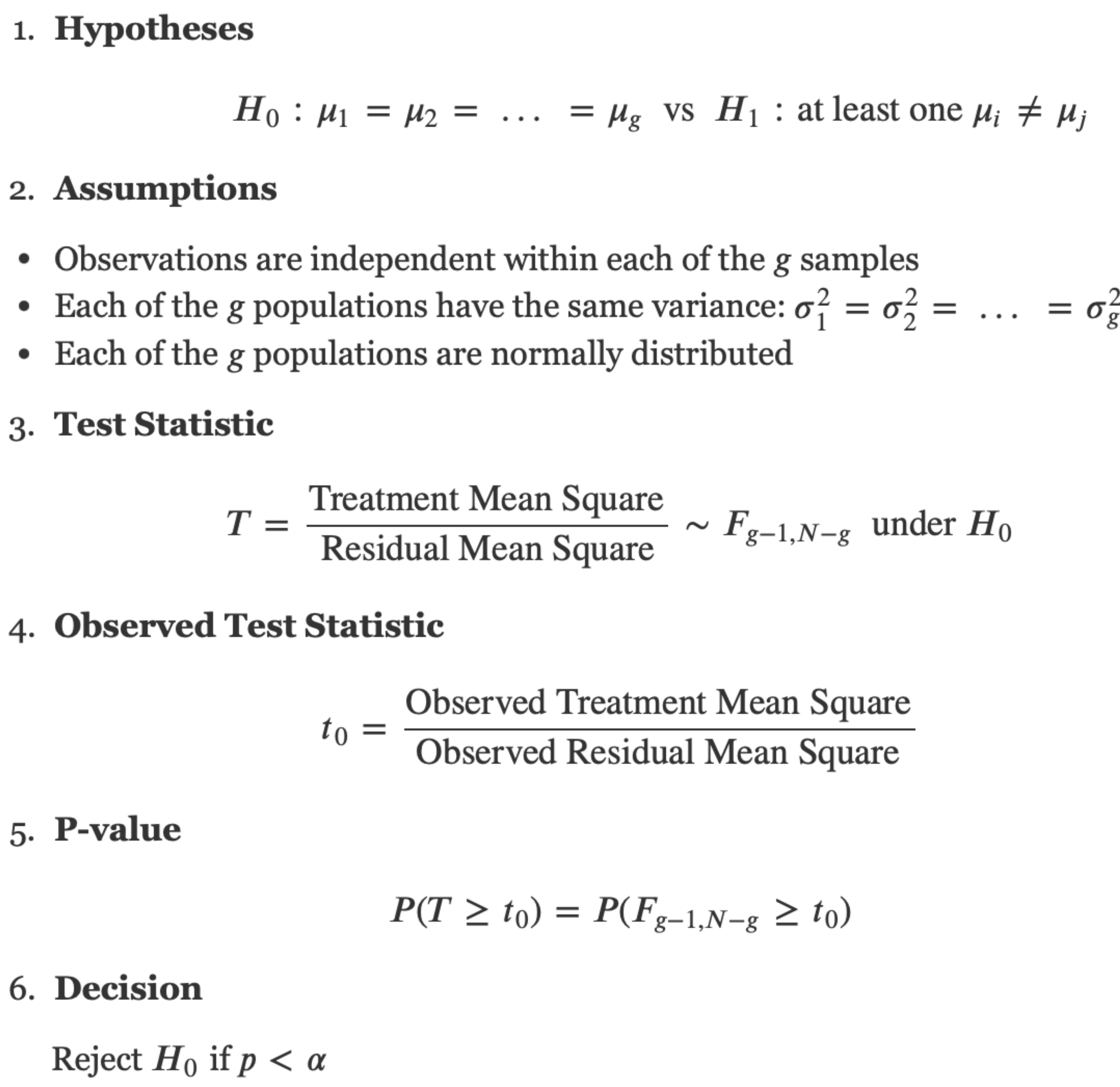


$$F = \frac{\text{Treatment Mean Square}}{\text{Residual Mean Square}} = \frac{\sum_{i=1}^{g} n_i(\bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet})^2/(g-1)}{\sum_{i=1}^{g}\sum_{j=1}^{n_i}(Y_{ij} - \bar{Y}_{i\bullet})^2/(N-g)}$$
$$\sim \frac{\chi^2_{g-1}/(g-1)}{\chi^2_{N-g}/(N-g)} \text{ (both independent)}$$
$$\sim F_{g-1,N-g} \text{ under } H_0.$$

👉 Large F statistic means that Treatment Mean Squares $\gg$ Residual Mean Squares
👉 **The treatment effect on the variation in Y is large** relative to within-group (within each treatment level) variation.
👉 In other words, $H_0 : \mu_1 = \mu_2 = \ \dots\ = \mu_g$ (no difference in group means between different treatment levels) is likely to be false!

One-way ANOVA

```r
knitr::include_graphics("images/lab3.png")
```



1. **Hypotheses**

$$H_0 : \mu_1 = \mu_2 = \ \dots \ = \mu_g \ \text{ vs } \ H_1 : \text{at least one } \mu_i \neq \mu_j$$

2. **Assumptions**

- Observations are independent within each of the g samples
- Each of the g populations have the same variance: $\sigma_1^2 = \sigma_2^2 = \ \dots \ = \sigma_g^2$
- Each of the g populations are normally distributed

3. **Test Statistic**

$$T = \frac{\text{Treatment Mean Square}}{\text{Residual Mean Square}} \sim F_{g-1,N-g} \ \text{ under } H_0$$

4. **Observed Test Statistic**

$$t_0 = \frac{\text{Observed Treatment Mean Square}}{\text{Observed Residual Mean Square}}$$

5. **P-value**

$$P(T \geq t_0) = P(F_{g-1,N-g} \geq t_0)$$

6. **Decision**

Reject H_0 if $p < \alpha$

Multiple Testing

```
knitr::include_graphics("images/lab4.png")
```

	True H_0 $(\theta = 0)$	False H_0 ()	Number of Tests
Conclusion $\theta = 0$	U	T	$m - R$
Conclusion $\theta \neq 0$	V	S	R
Number of Tests	m_0	$m - m_0$	m

⚠️ When performing m number of (independent) hypothesis tests, **the total number of false positives (falsely rejecting true H_0) will be** $m \times \alpha$, since α is a probability making false positives when performing a single hypothesis test.

The goal of the following two correction methods is **to reduce the total number of false positives when we perform multiple hypothesis tests**

Bonferroni Correction

```
knitr::include_graphics("images/lab5.png")
```

👉 Controls **the probability of making at least one or more false positives** (FWER) at level α.
In other words, $P(V \geq 1) < \alpha$.

🧐 How?

1. In each of the m tests, calculate the p-value as usual
2. Set a **new threshold** $\alpha* = \frac{\alpha}{m}$
3. For each test, reject H_0 **if the p-value is less than the new threshold** $\alpha*$. That is, reject H_0 if $p < \frac{\alpha}{m}$

⚠️ Alternative way

1. In each of the m tests, adjust the p-value: $p* = p \times m$
2. Reject H_0, if the adjusted p-value is less than *original* α. That is, reject H_0 if $p* < \alpha$

※ Note making α smaller ($\alpha* = \alpha/m$) and compare this adjusted $\alpha*$ with the unadjusted p-value is the same as making the individual p-value larger ($p* = p \times m$) and compare this adjusted p-value with unadjusted α, since $p < \frac{\alpha}{m}$ is the same as $p \times m < \alpha$

Pros and Cons

⭕ Pros: easy to calculate the adjusted significance level $\alpha*$

❌ Cons: **very conservative**!

👉 When m gets significantly large (e.g. $m = 1,000,000$), $\alpha* = \frac{\alpha}{m}$ **becomes extremely small**.
👉 With this extremely small $\alpha*$, **we'll almost always never reject any** H_0 regardless of whether it's true or not.
👉 Chances of not rejecting false H_0 ↑ (i.e., Type 2 error ↑)

Benjamini-Hochberg Correction

```
knitr::include_graphics("images/lab6.png")
```

👉 Controls the **false discovery rate (FDR) - expected proportion of false positives**: $E\left(\frac{V}{R}\right)$ by keeping FDR close to α.

🧐 How?

1. In each of the m tests, calculate the p-value as usual
2. Order p-values from smallest to largest: $p_{(1)} \leq p_{(2)} \leq \ \dots \ \leq p_{(m)}$
3. Find $j* = \max\ j$ such that $p_{(j)} \leq \frac{j}{m}\alpha$
   👉 In other words, we order m individual p-values and then find **a cutoff p-value $p_{(j)}$ that keeps the proportion of false positives (FDR) close to α**
   Note that **we're making α smaller again** by scaling α by $\frac{j}{m}$ because the maximum possible value for $\frac{j}{m}$ is 1, as we have m p-values.
4. For each test, reject H_{0i} if $p_{(i)} \leq \frac{j*}{m}\alpha$

Pros and Cons

⭕ Pros: less conservative than the Bonferroni correction

❌ Cons: because it's less conservative, it allows for more false positives

Lab Questions

If a light is flickering but at a very high frequency, it appears to not be flickering at all. Thus there exists a "critical flicker frequency" where the flickering changes from "detectable" to "not detectable" and this varies from person to person.

The critical flicker frequency and iris colour for 19 people were obtained as part of a study into the relationship between critical frequency flicker and eye colour.

Critical Flicker Frequency

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
flicker = read_tsv("https://raw.githubusercontent.com/DATA2002/data/master/flicker.txt")
```

```
## Rows: 19 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (1): Colour
## dbl (1): Flicker
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

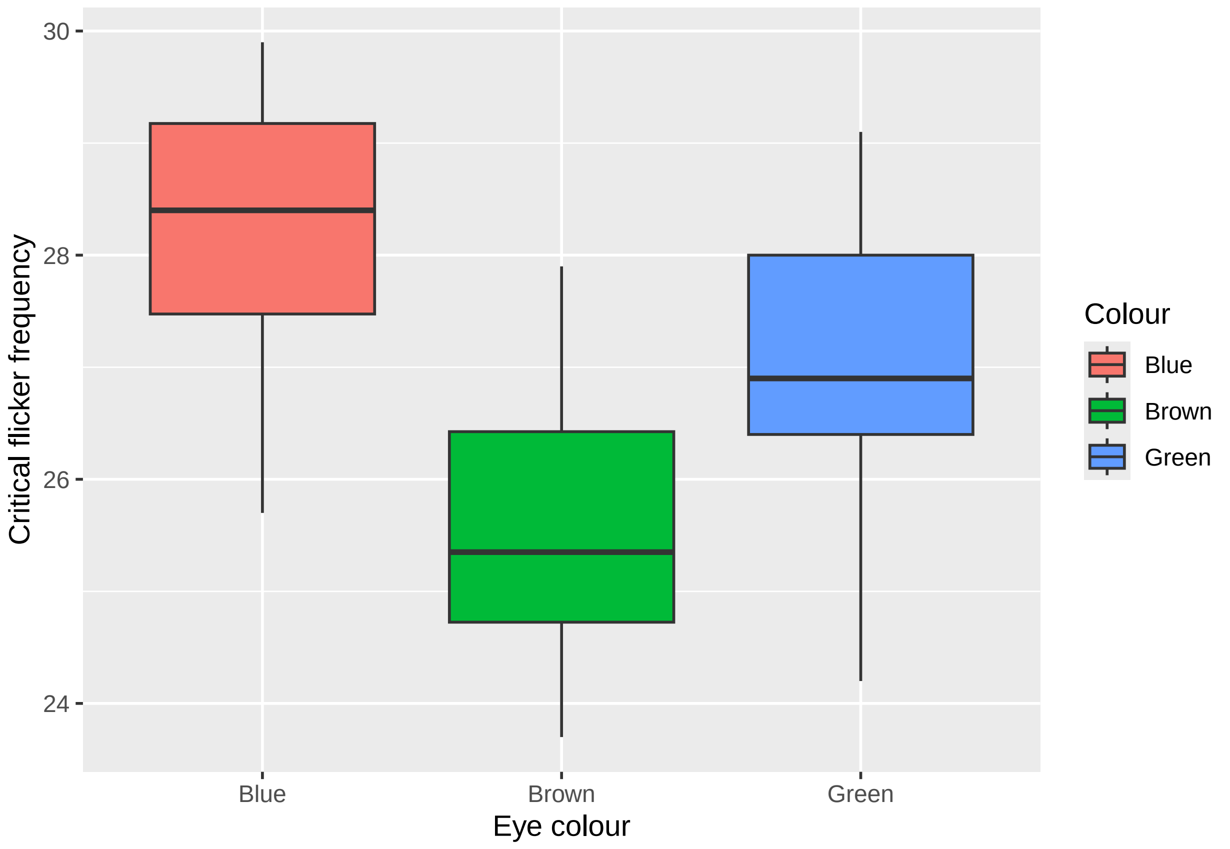
```r
glimpse(flicker)
```

```
## Rows: 19
## Columns: 2
## $ Colour  <chr> "Brown", "Brown", "Brown", "Brown", "Brown", "Brown", "Brown",~
## $ Flicker <dbl> 26.8, 27.9, 23.7, 25.0, 26.3, 24.8, 25.7, 24.5, 26.4, 24.2, 28~
```

already in long format

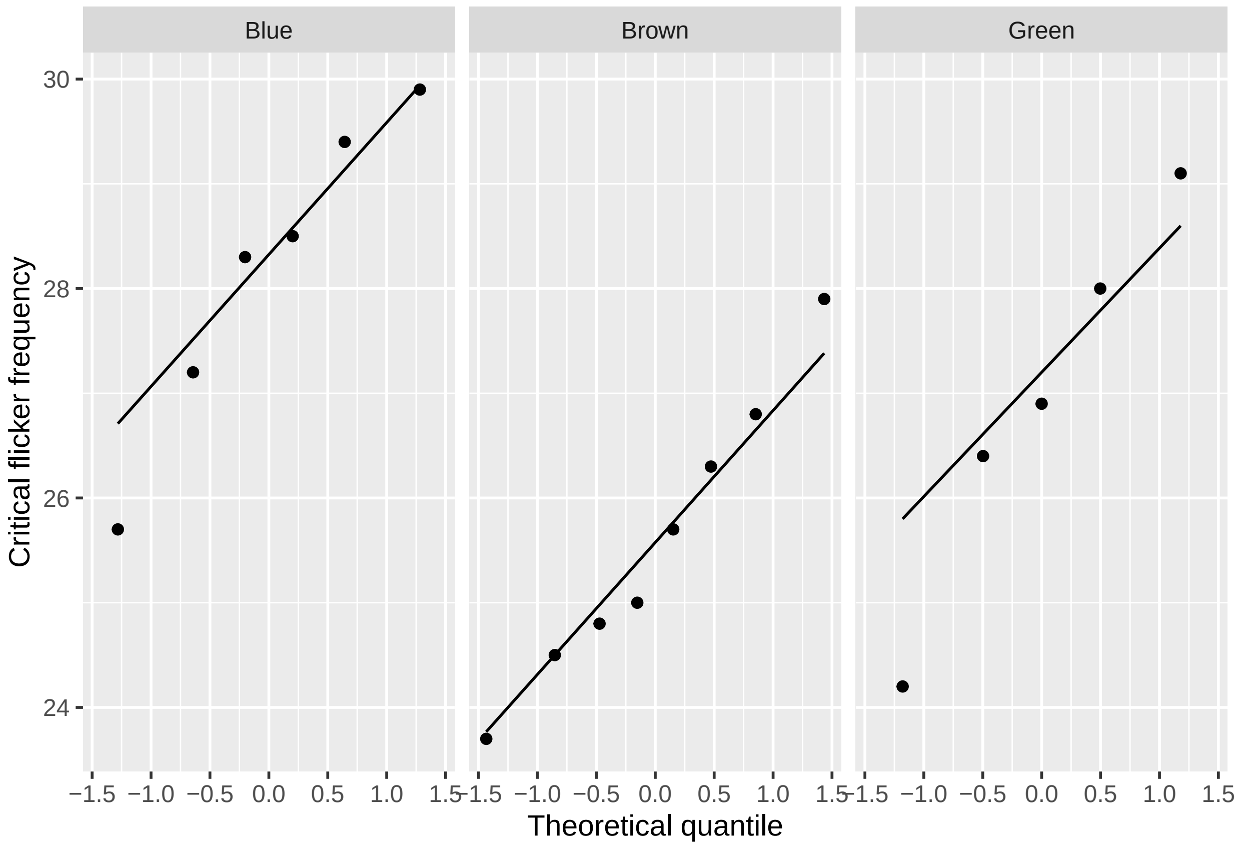
a. **Generate side by side box plots as well as normal Q-Q plots for the flicker data. Do your plots support the assumptions required for an ANOVA test to be valid? Explain.**

```r
ggplot(flicker, aes(x = Colour, y = Flicker, fill = Colour)) +
  geom_boxplot() +
  labs(y = "Critical flicker frequency", x = "Eye colour")
```



The box plots look symmetric, there are no outliers and they have similar spread. We can conclude that each population looks approximately normal and the equal variance assumption is reasonable.

```r
ggplot(flicker, aes(sample=Flicker)) +
  geom_qq() + geom_qq_line() +
  facet_wrap(vars(Colour)) +
  labs(y = "Critical flicker frequency", x = "Theoretical quantile")
```



The QQ-plots look OK, in that the points are reasonably close to the line.

b. **Use the `aov()` function to perform an ANOVA test for the equality of means flicker level across each of the three eye colours.**

Use the ANOVA workflow provided in the tut solution

```
flicker_anova = aov(Flicker ~ Colour, data = flicker)
flicker_anova
```

```
## Call:
##    aov(formula = Flicker ~ Colour, data = flicker)
##
## Terms:
##                    Colour Residuals
## Sum of Squares  22.99729  38.31008
## Deg. of Freedom        2        16
##
## Residual standard error: 1.547378
## Estimated effects may be unbalanced
```

```
summary(flicker_anova)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Colour       2  23.00  11.499   4.802 0.0232 *
## Residuals   16  38.31   2.394
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
pf(4.802, 2, 16, lower.tail = FALSE)
```

```
## [1] 0.02325398
```

c. **Using the output, write out the hypothesis test in full.** Be sure to state the null and alternative hypothesis, assumptions, test statistic (with distribution), observed test statistic, p-value and an appropriate conclusion.

1. **Hypotheses:** $H_0 : \mu_1 = \mu_2 = \mu_3$ vs H_1 : at least one $\mu_i \neq \mu_j$.
2. **Assumptions:** Observations are independent within each of the 3 samples. Each of the 3 populations have the same variance:
$$\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = \sigma$$
Each of the 3 populations are normally distributed (or the sample sizes are large enough such that you can rely on the central limit theorem).
3. **Test statistic:**
$$T = \frac{\text{Treatment Mean Sq}}{\text{Residual Mean Sq}} \quad \text{Under } H_0, \;\; T \sim F_{g-1,N-g}$$
where $g = 3$ is the number of groups and $N = 19$ is the total sample size.
4. **Observed test statistic:**
$$t_0 = \frac{11.499}{2.394} = 4.8$$
5. **p-value:**
$$P(T \geq t_0) = P(F_{2,16} \geq t_0) = 0.023$$
6. **Decision:** As the p-value is less than 0.05 we reject the null hypothesis and conclude that the population mean flicker sensitivity of at least one eye colour is significantly different to the others.

Blonds

In an investigation into the relationship between tolerance to pain and hair colour, men and women of various ages were divided into 4 groups based on hair colour and given a pain sensitivity test. Each person's "pain threshold score" (higher score means higher pain tolerance) is recorded in the file blonds.txt.

```r
pain = read_tsv("https://raw.githubusercontent.com/DATA2002/data/master/blonds.txt")
```

```
## Rows: 19 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (1): HairColour
## dbl (1): Pain
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
glimpse(pain)
```

```
## Rows: 19
## Columns: 2
## $ HairColour <chr> "LightBlond", "LightBlond", "LightBlond", "LightBlond", "Li~
## $ Pain       <dbl> 62, 60, 71, 55, 48, 63, 57, 52, 41, 43, 42, 50, 41, 37, 32,~
```

a. **Change `HairColour` so that the ordering is preserved from lightest to darkest.** Hint use: `factor()`

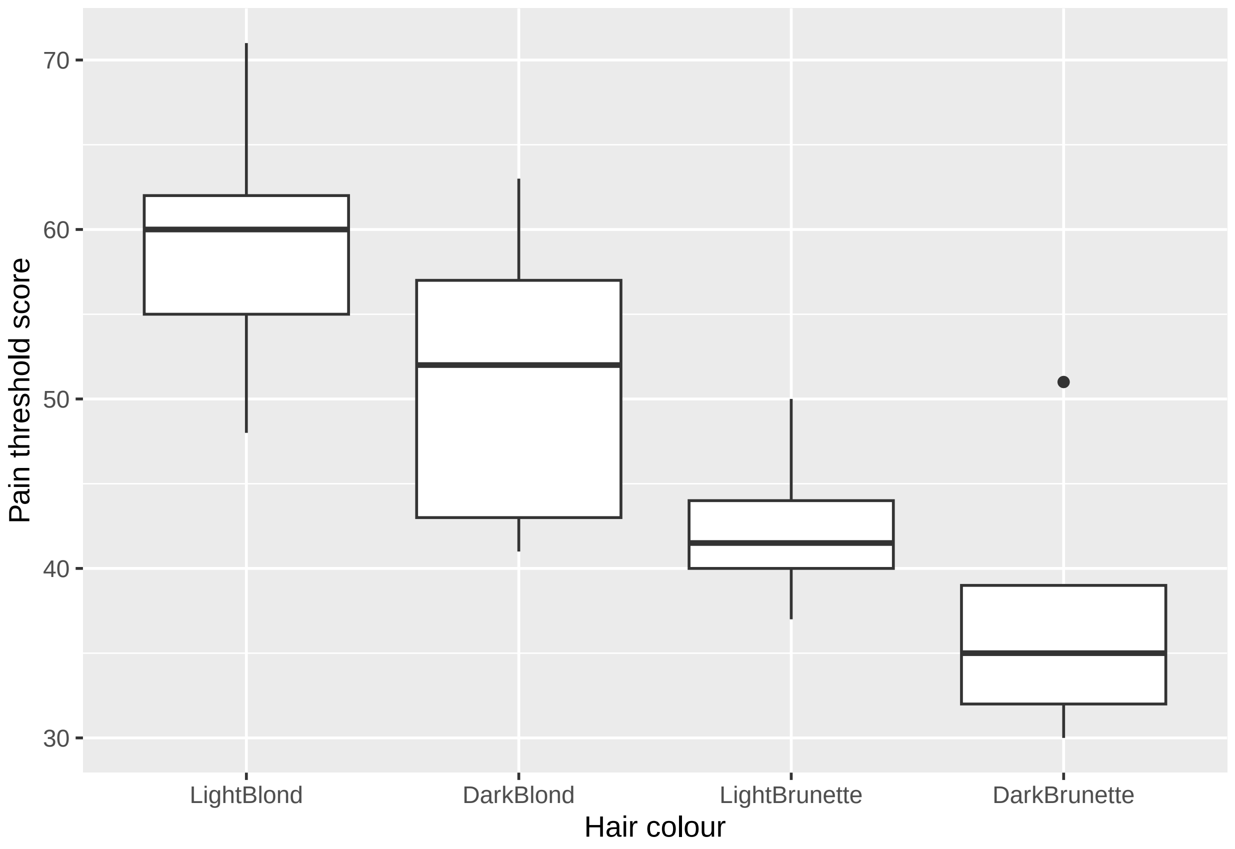
The order of these should be changed from alphabetical, there is a natural ordering, from lighter to darker. This is done as follows:

```
pain = pain |> mutate(
  HairColour = factor(HairColour, levels = c("LightBlond", "DarkBlond", "LightBrunette", "DarkBrunette"
)
levels(pain$HairColour)
```

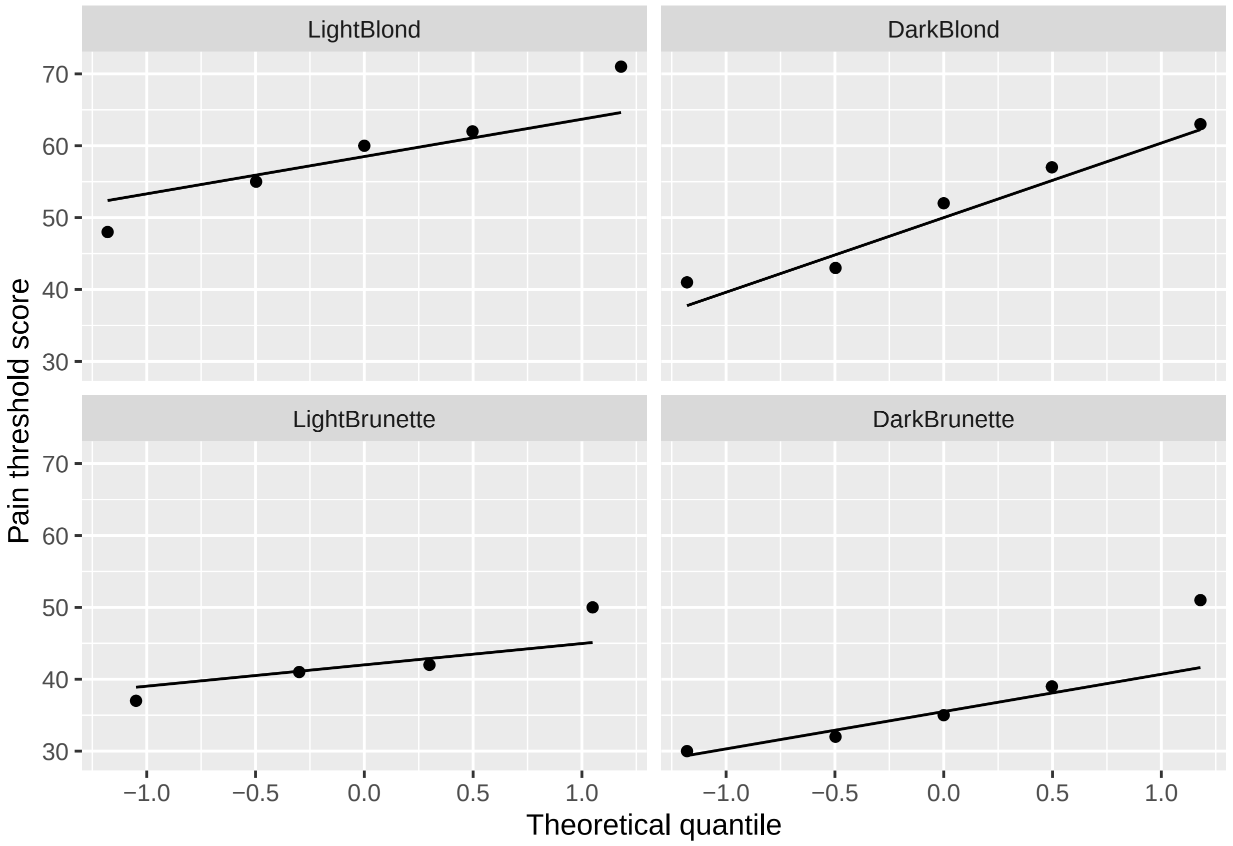
```
## [1] "LightBlond"    "DarkBlond"     "LightBrunette" "DarkBrunette"
```

b. **Generate box plots and Q-Q plots to check the ANOVA assumptions.**

```
ggplot(pain, aes(x = HairColour, y = Pain)) +
  geom_boxplot() +
  labs(y = "Pain threshold score", x = "Hair colour")
```



```
ggplot(pain, aes(sample = Pain)) +
  geom_qq() + geom_qq_line() +
  facet_wrap(~HairColour) +
  labs(y = "Pain threshold score", x = "Theoretical quantile")
```



It is hard to say anything conclusive about the ANOVA assumptions with so few observations in the different groups. Should be careful not to read too much into box plots with so few observations, but the spreads look roughly similar. Also with the QQ-plots, can't be too conclusive because of the low sample size, but the points are all reasonably close to the lines so the normality assumption doesn't appear to be violated.

c. **What do the box plots suggest about the null hypothesis that pain thresholds are the same regardless of hair colour?**

A shocking apparent effect! Looks like as hair colour darkens, pain thresholds decrease.

d. **Test this hypothesis formally using ANOVA. Does there seem to be a relationship between hair colour and pain threshold?!**

Benjamini-H: - a lot less conservative, but still is

```r
pain_anova = aov(Pain ~ HairColour, data = pain)
summary(pain_anova)
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)   
## HairColour   3   1361   453.6   6.791 0.00411 **
## Residuals   15   1002    66.8                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

1. **Hypotheses:** $H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4$ vs $H_1 :$ at least one $\mu_i \neq \mu_j$.
2. **Assumptions:** Observations are independent within each of the 4 samples. Each of the 4 populations have the same variance:

$$\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = \sigma_4^2 = \sigma$$

   Each of the 4 populations are normally distributed.

3. **Test statistic:**
$$T = \frac{\text{Treatment Mean Sq}}{\text{Residual Mean Sq}} \quad \text{Under } H_0,\ T \sim F_{g-1,N-g}$$

where $g = 4$ is the number of groups and $N = 19$ is the total sample size.

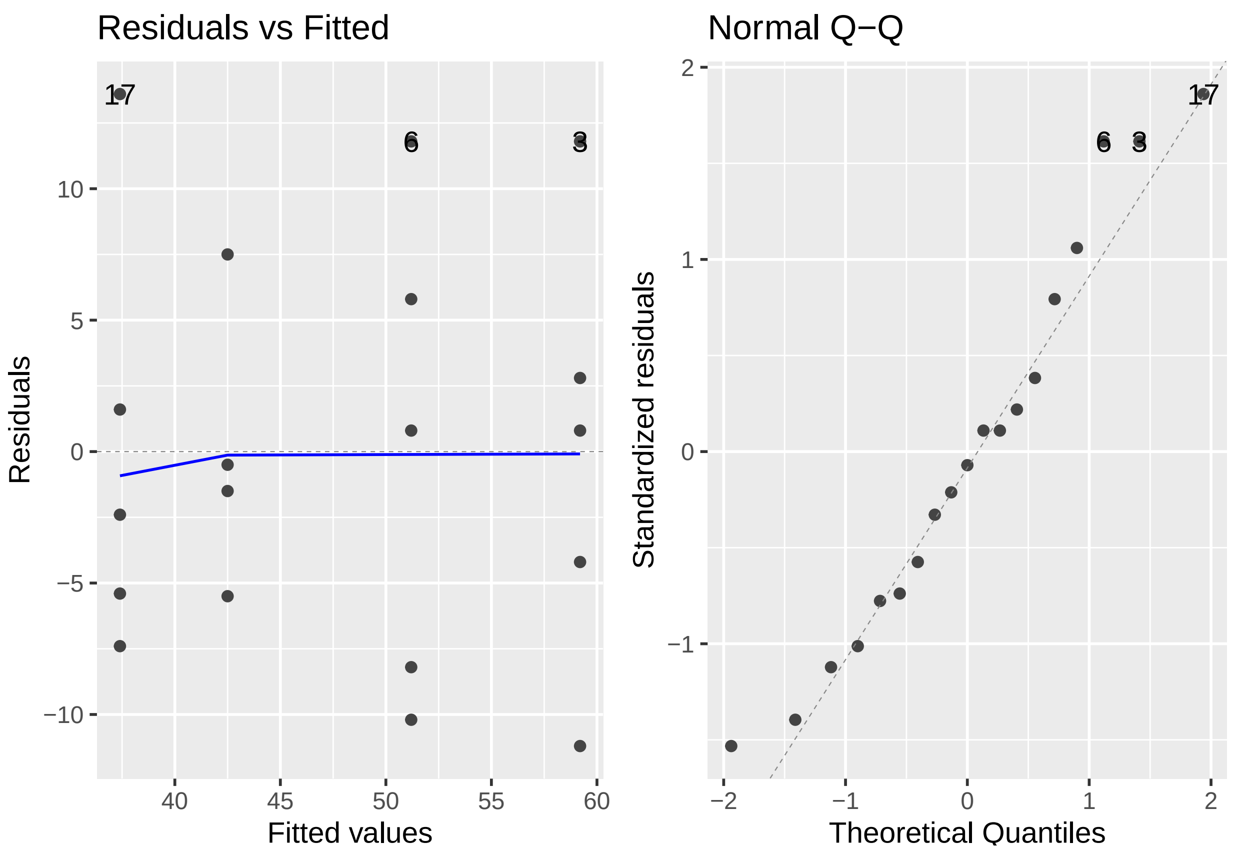
4. **Observed test statistic:**
$$t_0 = \frac{453.6}{66.9} = 6.791$$

5. **p-value:**
$$P(T \geq 6.791) = P(F_{3,16} \geq 6.791) = 0.004$$

6. **Decision:** As the p-value is less than 0.05 we reject the null hypothesis and conclude that the population mean pain threshold of at least one hair colour group is significantly different to the others.

The code and results below will be introduced in more detail in future weeks, but it provides a more overall way of assessing these assumptions. It's a similar idea, looking for roughly constant spread in the "residuals" across the range of "fitted values" and looking to check if the (standardised) residuals lie close to the dashed line in the normal Q-Q plot. In this case the spread of the residuals looks roughly similar across the range of fitted values (indicating the equal variance assumption is OK) and the points all lie reasonably close to the dashed line in the Q-Q plot indicating that the normality assumption is well satisfied.

```r
library(ggfortify)
autoplot(pain_anova, which = c(1,2))
```



Hedenfalk data

The package `sgof` has a data set Hedenfalk (Conde & Una Alvarez, 2020).

A shocking apparent effect! Looks like as hair colour darkens, pain thresholds decrease.

a. **How many p-values are in the data set?**

```r
# install.packages("sgof")
library(sgof)
```

```
## Loading required package: poibin
```

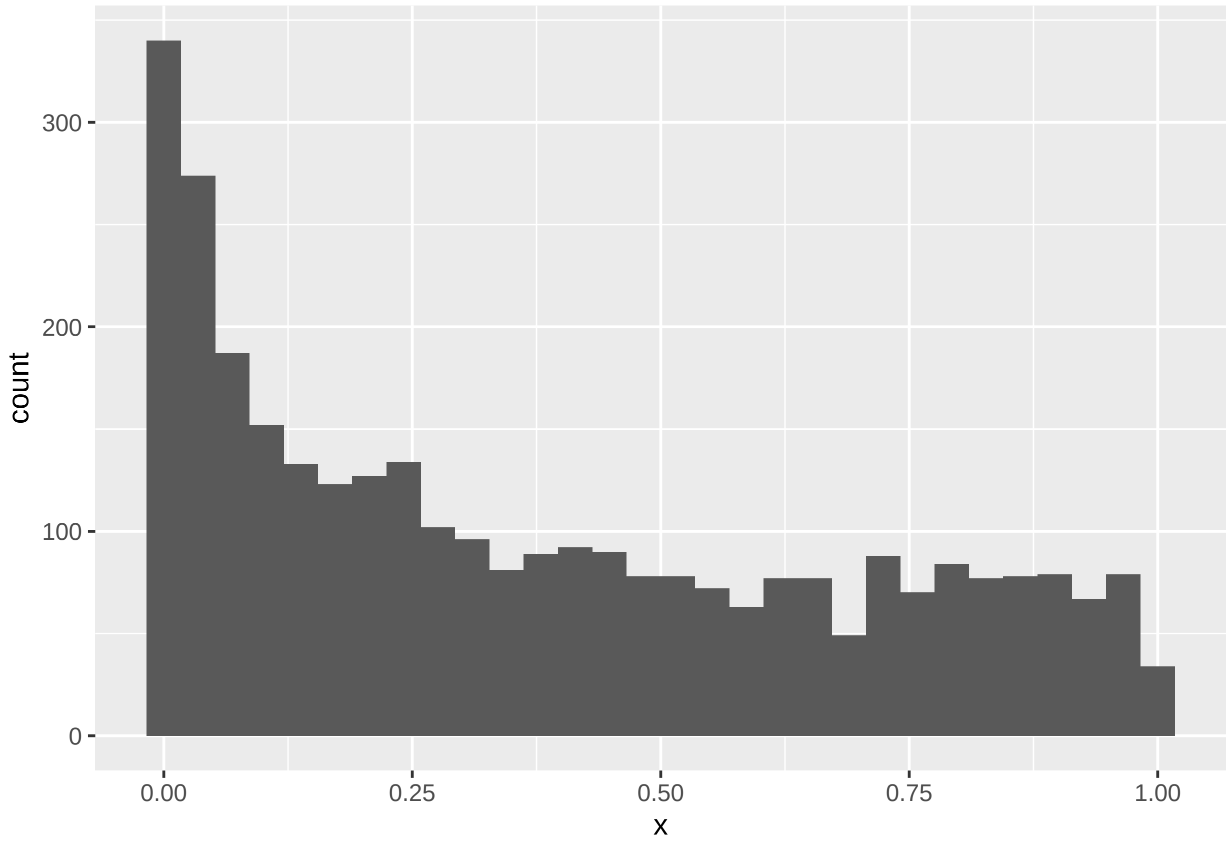
```r
# To find out more about this data set.
# ?Hedenfalk
glimpse(Hedenfalk)
```

```
## Rows: 3,170
## Columns: 1
## $ x <dbl> 0.0121261800, 0.0750252400, 0.9949211000, 0.0417854900, 0.8458139000~
```

b. **Generate a histogram of the (unadjusted) Hedenfalk p-values.**

```r
ggplot(Hedenfalk, aes(x = x)) +
  geom_histogram()
```

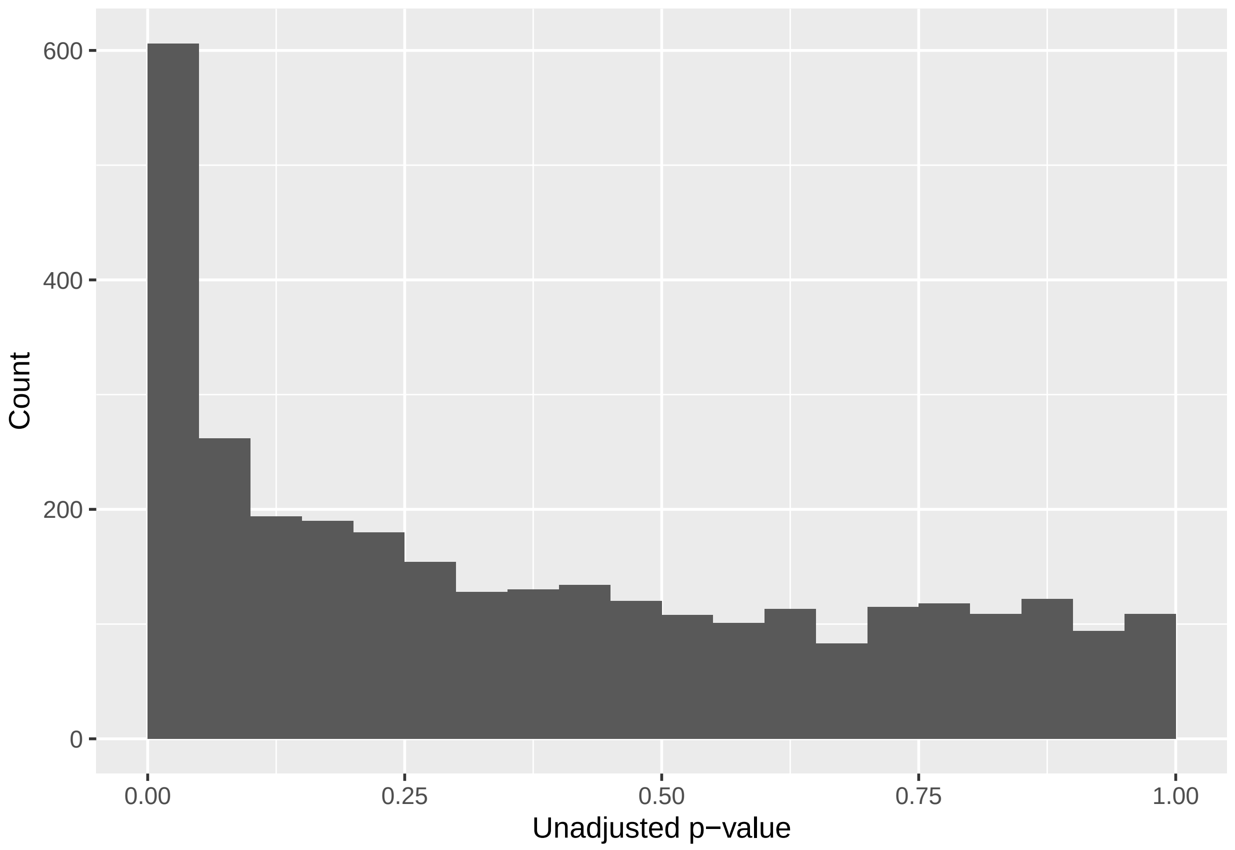
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



A

slightly improved histogram, where the bins don't go below 0 or above 1:

```r
ggplot(Hedenfalk, aes(x = x)) +
  geom_histogram(boundary=0, binwidth = 0.05) +
  labs(x = "Unadjusted p-value", y = "Count") +
  theme_grey()
```



c. **How many (unadjusted) p-values are significant at the 5% level of significance? What proportion of all p-values in the data set is this?**

```
length(Hedenfalk$x) # we will have 3170 p-values
```

```
## [1] 3170
```

```
sum(Hedenfalk$x < 0.05) # 605
```

```
## [1] 605
```

```
mean(Hedenfalk$x < 0.05) # 0.1909
```

```
## [1] 0.1908517
```

d. **Why is it a good idea to consider adjusted p-values?**

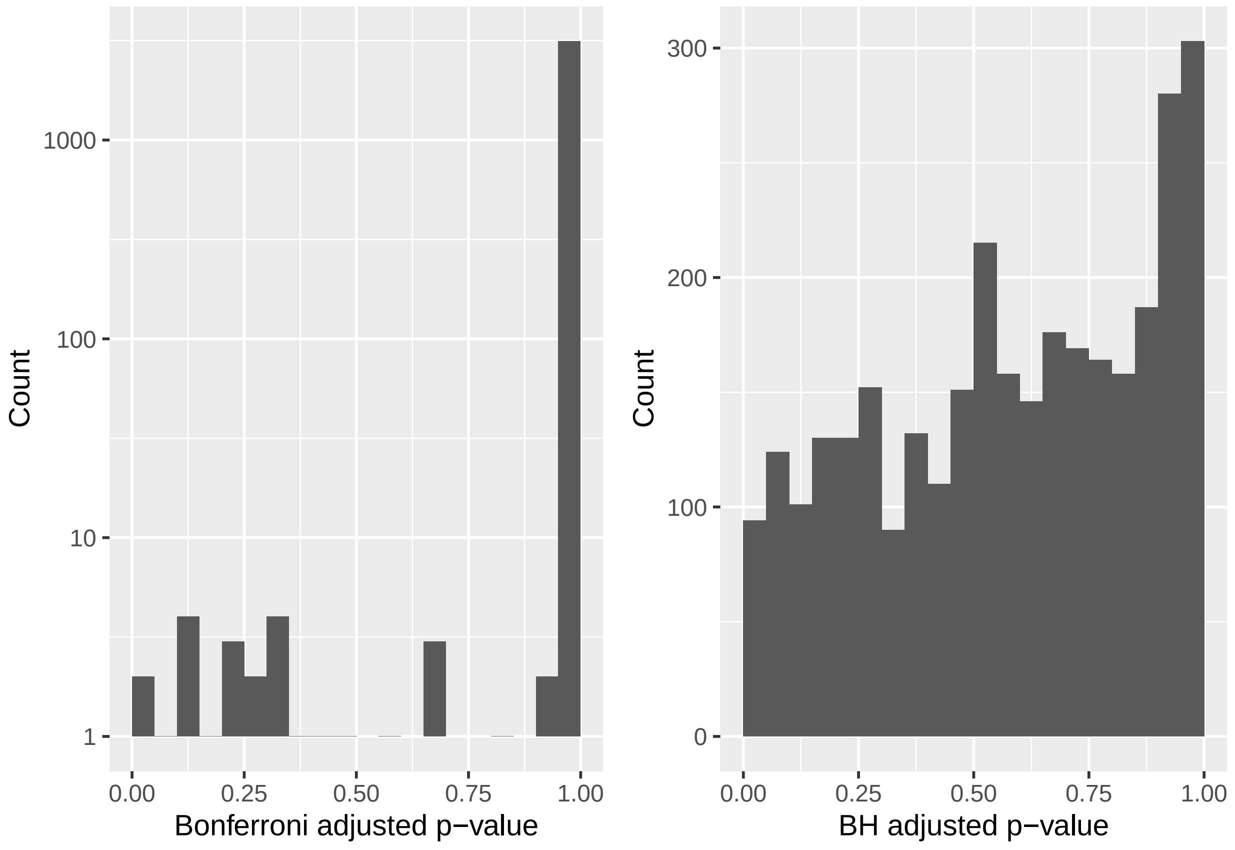
With so many tests, it's likely we're seeing a substantial number of false positives. With 3170 p-values, even if none of them should be rejected (i.e. even if in reality the null hypothesis is true for all tests), we'd expect to see $3170 \cdot 0.05 = 158.5$ significant p-values by chance alone.

e. **Using `p.adjust()` find the Bonferroni and BH p-values. Plot histograms of each and find the number of significant results after adjustment for both.**

```
Hedenfalk = Hedenfalk |>
  mutate(
    bonf_p = p.adjust(x, method = "bonferroni"),
    BH_p = p.adjust(x, method = "BH")
  )
p1 = Hedenfalk |>
  ggplot() + aes(x = bonf_p) +
  geom_histogram(boundary = 0, binwidth = 0.05) +
  theme_grey() +
  labs(x = "Bonferroni adjusted p-value",
       y = "Count") +
  scale_y_log10()
p2 = Hedenfalk |>
  ggplot() + aes(x = BH_p) +
  geom_histogram(boundary = 0, binwidth = 0.05) +
  labs(x = "BH adjusted p-value",
       y = "Count") +
  theme_grey()
gridExtra::grid.arrange(p1,p2,ncol=2)
```

```
## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

## Warning: Removed 5 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```



```
# sum(Hedenfalk$bonf_p < 0.05) mean(Hedenfalk$bonf_p < 0.05)
# sum(Hedenfalk$BH_p < 0.05) mean(Hedenfalk$BH_p < 0.05)
Hedenfalk %>%
    summarise_at(.vars = vars(bonf_p, BH_p), # for each of the two columns: bonf_p and BH_p
                 .funs = list(n_sig = function(x) sum(x < 0.05), # find the number of Bonferroni/BH adj
                              prop_sig = function(x) mean(x < 0.05))) %>% # find the proportion of Bonf
    knitr::kable()
```

bonf_p_n_sig	BH_p_n_sig	bonf_p_prop_sig	BH_p_prop_sig
2	94	0.0006309	0.029653

f. **Comment on the difference between the Bonferroni method and the BH method.**

- The Bonferroni method seeks to control the family wise error rate, and can be very conservative.
  - Bon: p-value * num of test
    - very conservative
    - Cap at 1 mark, since dens cannot > 1
- The Benjamini–Hochberg (BH) method looks to control the false discovery rate and tends to allow more false positives.
  - less conservative

We can see this in the results, where the Bonferroni method finds only two significantly differentially expressed genes whereas the BH procedure identified 94.

Extra The following code will prove that we can adjust α and compare adjusted α^* with unadjusted p-values for the Bonferroni correction method.

```
Hedenfalk %>%
    summarise_at(.vars = vars(x), # for unadjusted p-values x
                 .funs = list(n_sig = function(x) sum(x < 0.05/3170), # find the number of unadjusted p
                              prop_sig = function(x) mean(x < 0.05/3170))) %>% # find the proportion of
    knitr::kable()
```

n_sig	prop_sig
2	0.0006309

As you can see, the total number of false positives after the Bonferroni correction is the same as above (see bonf_p_n_sig and bonf_p_prop_sig in question e above).