Assignment 3

Rory Sarten 301005654

29 September, 2020

Question 1

a)

A Runs Test tests a set of binary variables $X_1, ..., X_n$ to verify if the variables occur randomly.

H_0: variables occur randomly, i.e. knowing $X_1, ..., X_n$ does not help predict X_{n+1}.

H_A: variables are not random, i.e. knowing some part of the sequence can help predict subsequent variables.

As the variables are binary, they will take the value 0 or 1. The number of 0s is n_0 and the number of 1s is n_1, where:

$$n_0 = n - \sum_{i=1}^{n} X_i$$

$$n_1 = \sum_{i=1}^{n} X_i$$

To perform a Runs Test the observations are combined into one collection of $n = n_0 + n_1$ observations and arranged in increasing order of magnitude or observation. They are labeled according to which set they originally came from. A run is a group of two or more sequential values of 0 or 1.

Let R denote the number of runs in the combined ordered sample of $X \in \{0, 1\}$. Under H_0, R can be approximated as a normally distributed random variable, assuming both n_0 and n_1 are sufficiently large.

$R = 1 + \sum_{i=2}^{n} I_{(X_i, X_{i-1})}$, where $I_{(X_i, X_{i-1})} = 0$ if $X_i = X_{i-1}$ and $I_{(X_i, X_{i-1})} = 1$ if $X_i \neq X_{i-1}$

$$\bar{R} = \frac{2n_0 n_1}{n} + 1$$

$$Var(\bar{R}) = \frac{2n_0 n_1 (2n_0 n_1 - n)}{n^2 (n-1)}$$

With test statistic $Z = \dfrac{R - \bar{R}}{\sqrt{Var(\bar{R})}}$ where $Z \sim N(0, 1)$

b)

A small number of runs (a small value for R) would indicate that X_i is more likely to be the same as X_{i-1}. A large number means that X is fluctuating regularly between values and X is less likely to be the same as X_{i-1}.

c)

```
## 0 healthy, 1 diseased
X <- "HHHDDDHDDHHHHHDHHHHHHHHHHHHDDDHHHHDDDHHHHDHHHDHH" %>%
  stringr::str_split("") %>% unlist() %>% `==`("D") %>% as.integer()

n <- length(X)
n_0 <- n - sum(X)
n_1 <- sum(X)
```

```
R <- 1 + sum(X[2:n] != X[1:n-1])
R_est <- 1 + 2*n_0*n_1/n
R_var <- (2*n_0*n_1*(2*n_0*n_1 - n))/(n^2*(n - 1))
Z <- (R - R_est)/sqrt(R_var)
p <- pnorm(Z)
cbind(R, R_est, R_var, Z, p)
```

```
##       R    R_est    R_var         Z          p
## [1,] 15 20.65957 7.974767 -2.004125 0.02252834
```

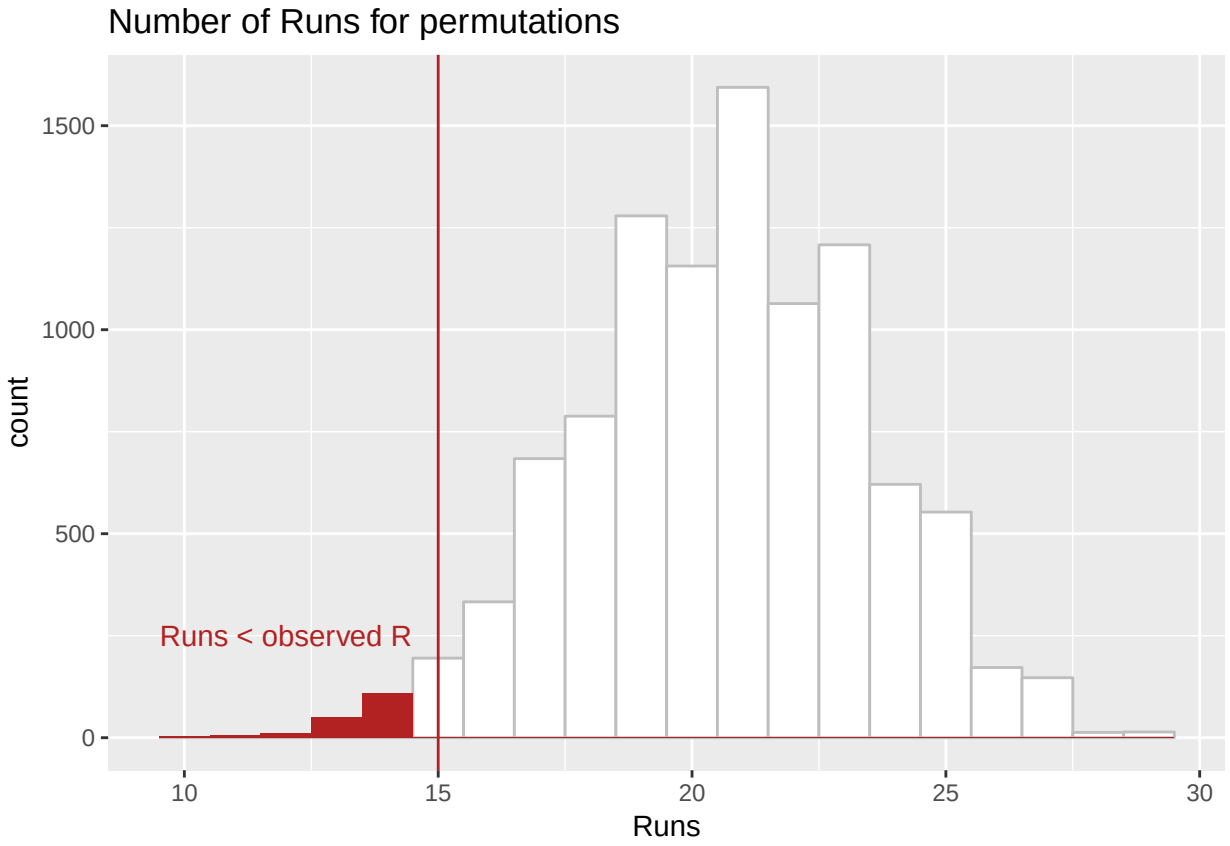
With p-value of 0.0225 we reject H_0 at the 5% level. We conclude that values are not randomly ordered.

d)

```
set.seed(101)

calc_R <- function(input)
  1 + sum(input[2:length(input)] != input[1:length(input)-1])

sample_R <- function(iter, input)
  input %>% sample() %>% calc_R()

obs_R <- calc_R(X)

N <- 1e4
perm_R <- 1:N %>% sapply(sample_R, input = X)

p <- sum(perm_R < obs_R)/length(perm_R)
p
```

```
## [1] 0.0179
```

```
ggplot() +
  geom_histogram(data = data.frame(x = perm_R[perm_R >= obs_R]),
                 aes(x = x), binwidth = 1, fill = "white", colour = "grey") +
  geom_histogram(data = data.frame(x = perm_R[perm_R < obs_R]),
                 aes(x = x), binwidth = 1, fill = "firebrick") +
  geom_vline(xintercept = obs_R, colour = "firebrick") +
  ggtitle("Number of Runs for permutations") +
  xlab("Runs") +
  geom_text(aes(x = 12, y = 250), label = "Runs < observed R", colour = "firebrick")
```



Because the number of runs is not a continuous variable, there is some ambiguity around whether the p-value should be calculated comparing values $<$ or $<=$ or even using some half point. I have decided to use $<$ as it gives the test the highest power.

Given this we find a p-value of 0.0179. We reject H_0 at the 5% level. We conclude that values are not randomly ordered.

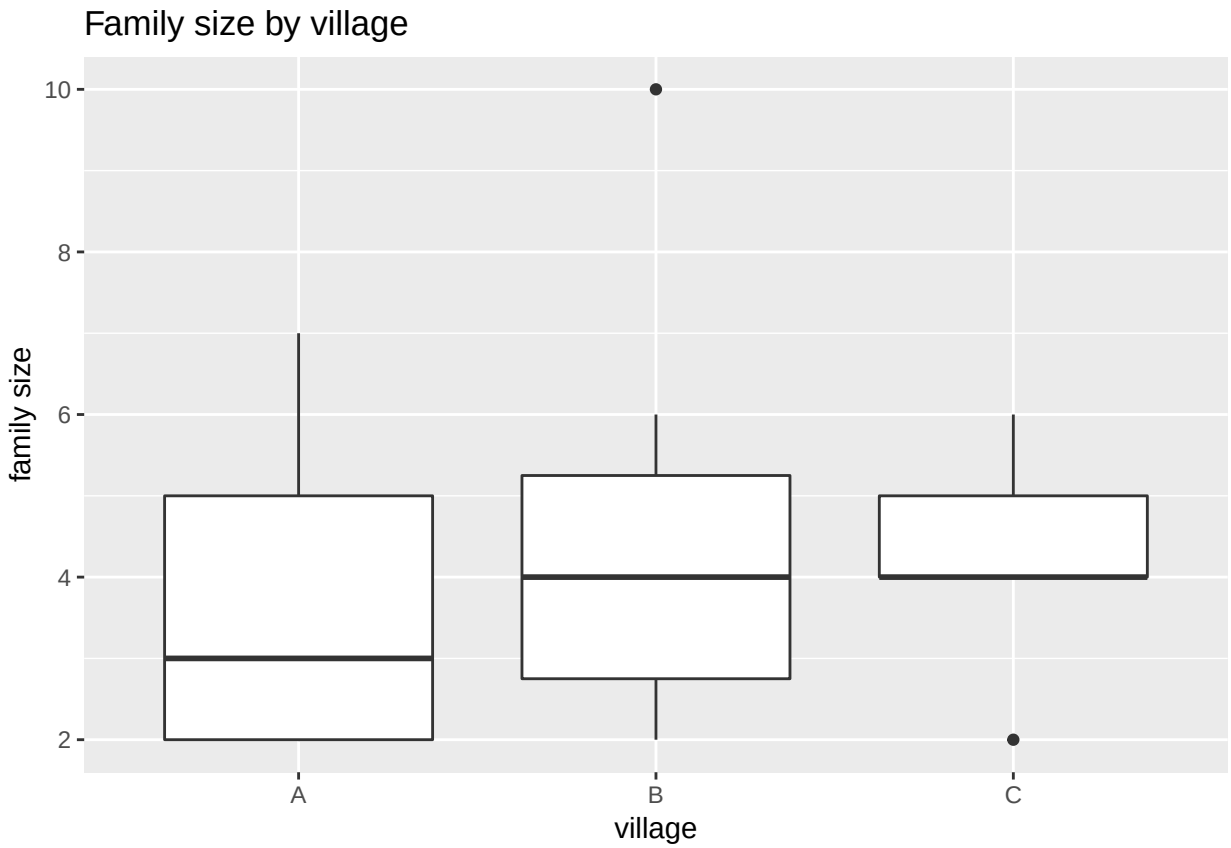
Question 2

a)

We could use a Poisson distribution

b)

```
dataset <- data.frame(
  size = c(2, 3, 2, 7, 5, 5, 3, 2, 6, 10, 3, 2, 2, 5, 6, 4, 4, 5, 4, 4, 6, 5, 4, 2),
  village = rep(c("A", "B", "C"), c(9, 8, 7))
)

ggplot(dataset, aes(x = village, y = size)) +
  geom_boxplot() +
  ggtitle("Family size by village") +
  ylab("family size")
```




```
## village means
tapply(dataset$size, dataset$village, mean) %>% round(digits = 4)
```

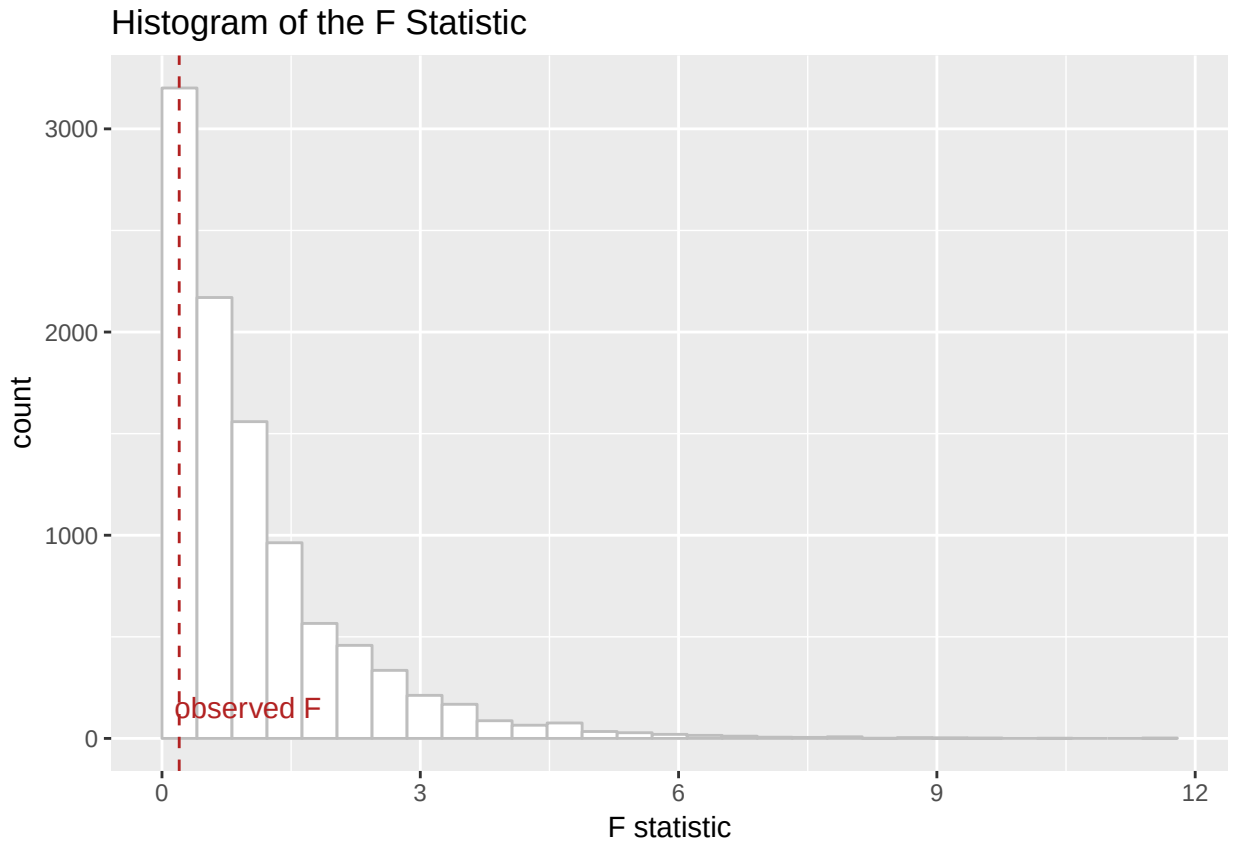
```
##      A      B      C
## 3.8889 4.5000 4.2857
```

c)

```
calc_F <- function(input)
  input[["fstatistic"]]["value"]

obs_F <- lm(size ~ village, data = dataset) %>%
  summary() %>% calc_F() %>% round(digits = 3)

N <- 1e4
set.seed(150)
perm_F <- 1:N %>%
  lapply(function(i) sample(dataset$village)) %>%
  lapply(function(groups) lm(dataset$size ~ groups)) %>%
  lapply(summary) %>%
  sapply(calc_F) %>%
  round(digits = 3)

p <- sum(perm_F >= obs_F)/N
p
```

```
## [1] 0.8319
```

```
ggplot() +
  geom_histogram(data = data.frame(x = perm_F), aes(x = x),
                 breaks = seq(from = 0, to = max(perm_F), length.out = 30),
                 fill = "white", colour = "grey") +
  geom_vline(xintercept = obs_F, colour = "firebrick", linetype = "dashed") +
  ggtitle("Histogram of the F Statistic") +
  xlab("F statistic") +
  geom_text(aes(x = 1, y = 150), label = "observed F", colour = "firebrick")
```



Given a p-value of 0.8319 we fail to reject H_0 and find that there is not a difference in the mean family sizes.

d)

```
draw_hist <- function(perm_values, observed) {
  print(
    ggplot() +
      geom_histogram(data = data.frame(x = perm_values), aes(x = x),
                     breaks = seq(from = 0, to = max(perm_values), length.out = 10),
                     fill = "white", colour = "grey") +
      geom_vline(xintercept = observed, colour = "firebrick", linetype = "dashed") +
      ggtitle("Histogram of permutation values") +
      xlab("test statistic"))
}

permutation_test = function(test_func, dataset, niter, comparator) {
  test_stat <- lm(size ~ village, data = dataset) %>%
    summary() %>% test_func() %>% round(digits = 3)
```

```
  set.seed(150)
  perms <- 1:niter %>%
    lapply(function(i) sample(dataset$village)) %>%
    lapply(function(groups) lm(dataset$size ~ groups)) %>%
    lapply(summary) %>%
    sapply(test_func) %>%
    round(digits = 3)

  draw_hist(perms, test_stat)
  sum(comparator(perms, test_stat))/niter
}
```

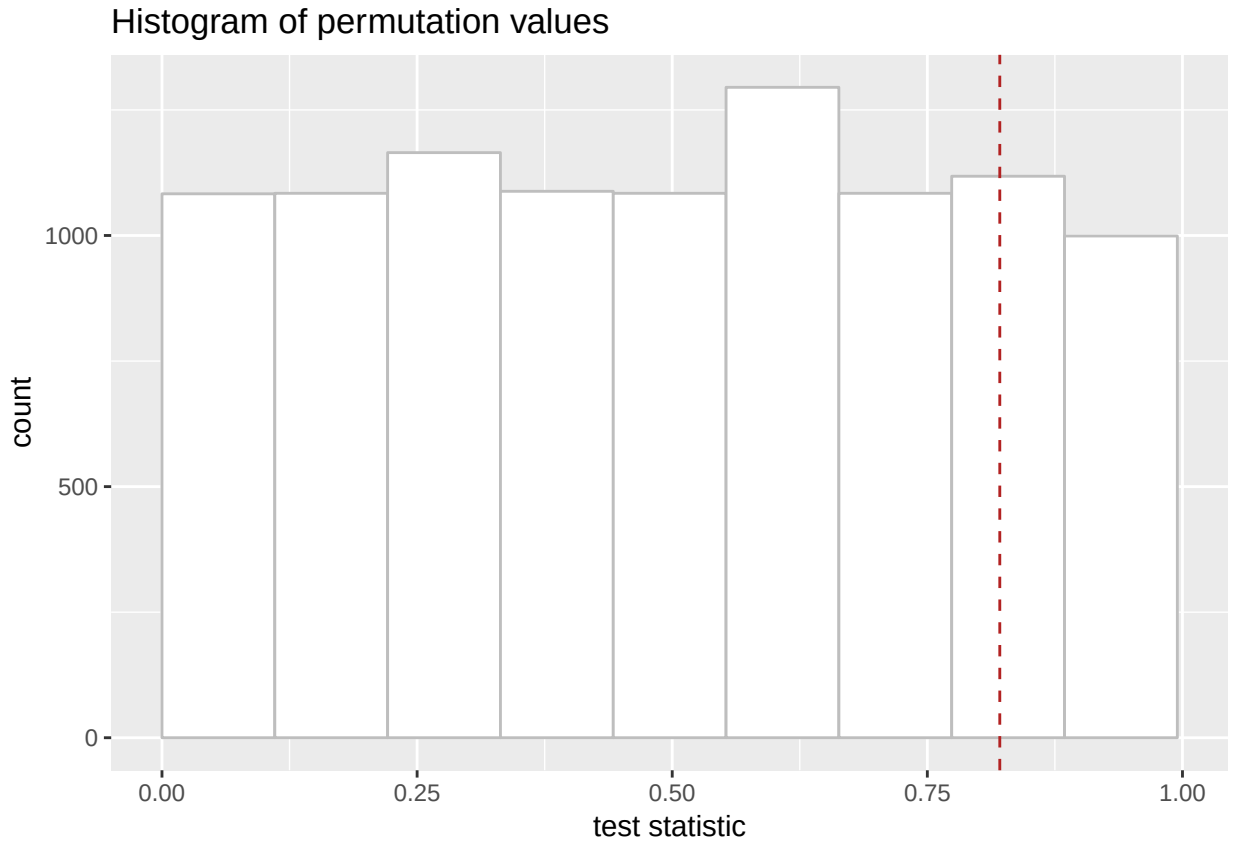
e)

A p-value is the probability of obtaining a test statistic as, or more, extreme than the test statistic observed under the null hypothesis. The p-value is a suitable test statistic to use in a permutation test as the relationship between the F statistic and the p-value means that the result of the ANOVA is preserved. Increasing the F statistic will always result in a decrease in the p-value so long as the degrees of freedom remain the same. Meaning that for all $F_x > F_y$, there will be $p_x < p_y$.

Using the same `set.seed` value will mean that the permutations of grouping values will be the same across tests even where the test statistic in question changes. Given the case above, the ratio of cases above/below the observed statistic will remain the same, giving the same final p-value.

The generated p-values appear to be drawn from a uniform distribution, and this is what we would expect under a null hypothesis.

```
calc_p <- function(input)
  pf(input$fstatistic, input$df[1]-1, input$df[2], lower.tail = FALSE)["value"]

p <- permutation_test(calc_p, dataset, 1e4, `<=`)
```



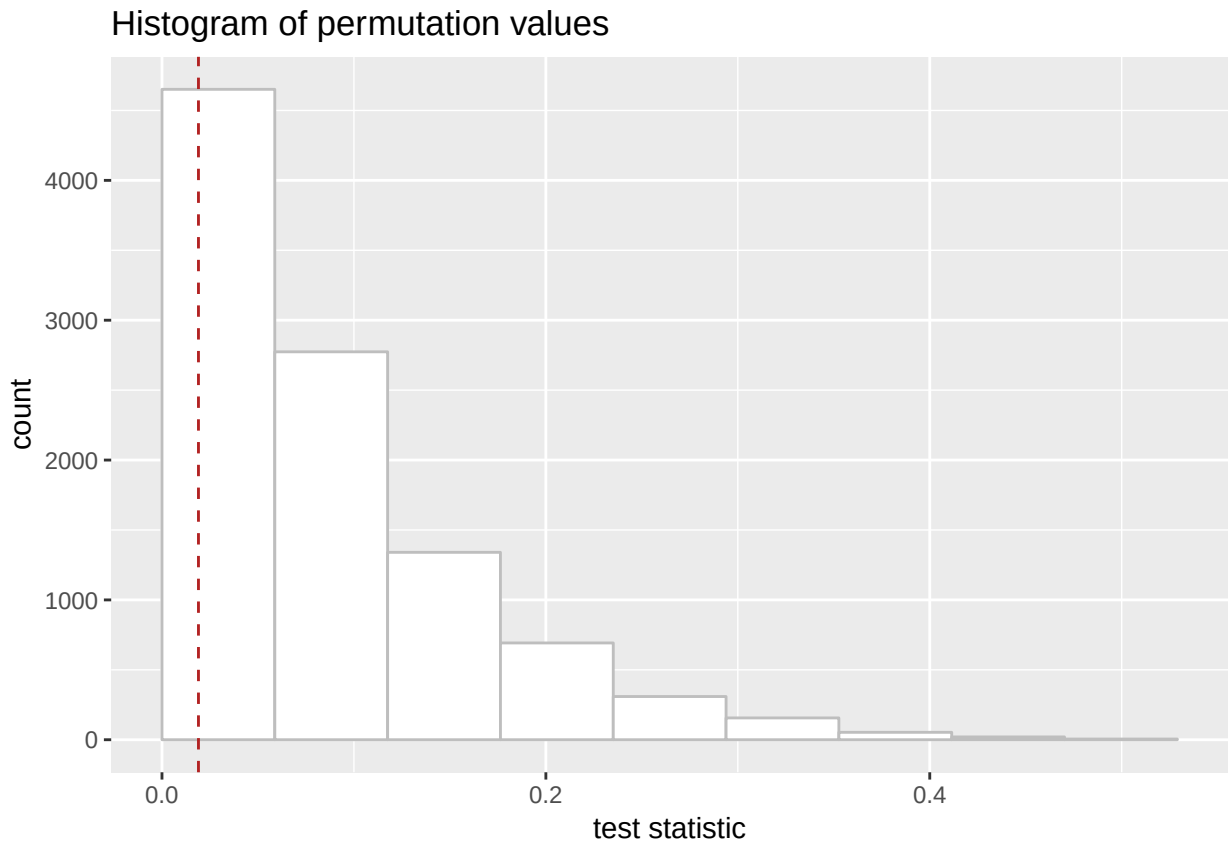

```
p
```

```
## [1] 0.8319
```

f)

A large R^2 value means the linear model fits the data well. This would also indicate that the F-statistic would be large as at least one of the coefficients in the model would be non-zero. This would mean that for all $F_x > F_y$ we have $R_x^2 > R_y^2$.

```
calc_r2 <- function(input)
  input[["r.squared"]]

r2 <- permutation_test(calc_r2, dataset, 1e4, `>=`)
```




```
r2
```

```
## [1] 0.8319
```

Question 3

a)

The chi-squared test statistic:

$\chi^2 = \sum \frac{(O_i - E_i)}{E_i}$ where O_i is the observed value in cell i and E_i is the expected value in cell i.

```
living_standards <- data.frame(
  significant_hardship = c(34, 27),
  fairly_comfortable = c(18, 22),
  comfortable = c(23, 52),
  good = c(9, 20),
  row.names = c("with_dep_children", "no_dep_children"))

chisq.test(living_standards)
```

```
##
##  Pearson's Chi-squared test
##
## data:  living_standards
## X-squared = 10.245, df = 3, p-value = 0.0166
```

With p-value of 0.0166 we reject H_0 at the 5% level. We conclude that there is a difference in living standards between families with dependent children and families without dependent children.

b)

The procedure for the permutation chi-square test is as follows:

1. Construct a data.frame with a column indicating the child dependency status and another indicating the living standard. There will be a single row giving the observed combination for each family.
2. Calculate the observed χ^2 value (χ^2_{obs}). Tabulate the observed table O then calculate the expected matrix E. The expected table is calculated as the outer product of the tabulated row and column sums of the observed counts.

$$E = \frac{rowsum(O) * colsum^T(O)}{n}$$

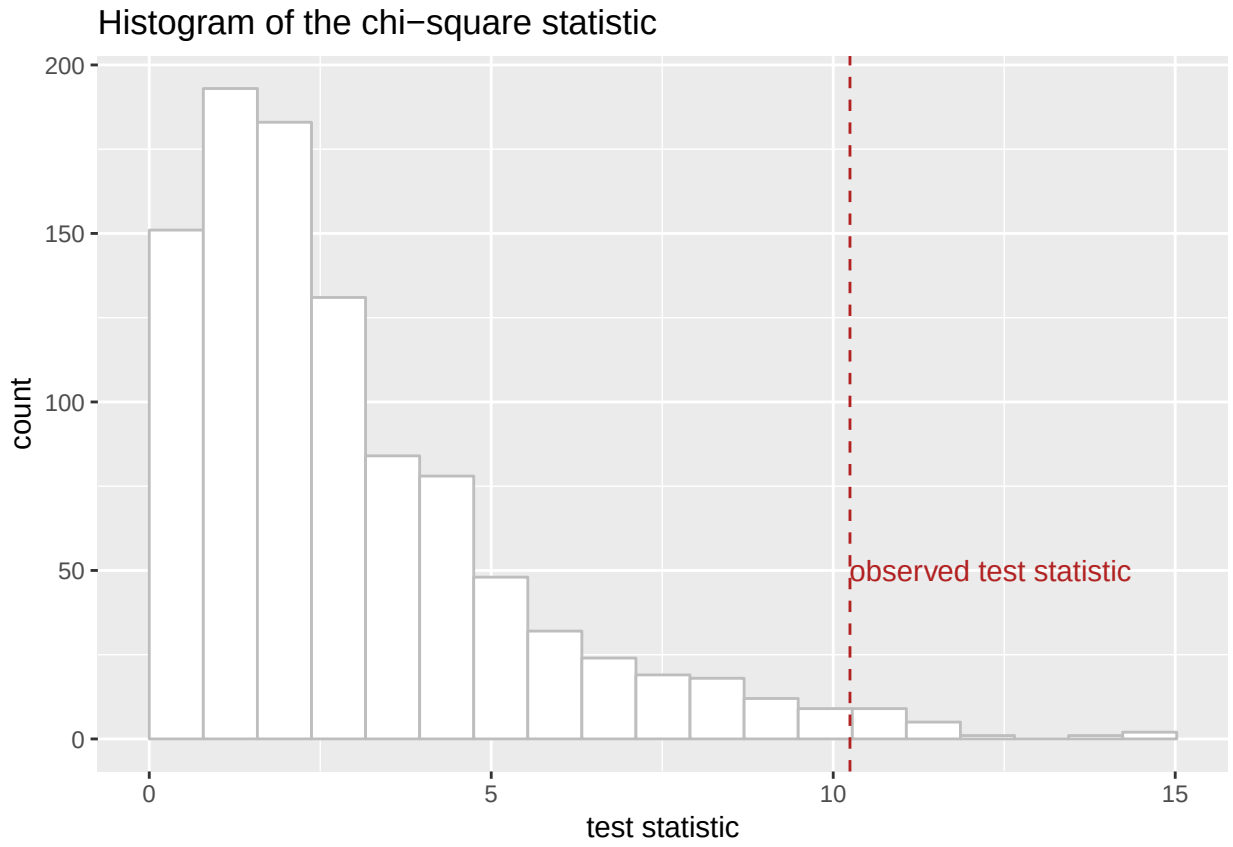
The chi-square test statistic can then be calculated using the formula in a).

3. For each permutation randomly reallocate the values for dependent children, while holding the living standard fixed. Under the null hypothesis each value for the dependent children is equally likely for the levels of living standards. The chi-square statistic for each permutation χ^2_{perm} is calculated as above.
4. p-value is calculated as:

$$\text{p-value} = \frac{\text{No of } \chi^2_{perm} >= \chi^2_{obs}}{\text{No of permutations}}$$

```
families <- data.frame(
  dependent_children = c(rep("Yes", 84L), rep("No", 121L)),
  living_standard = c(rep("significant_hardship", 34L),
                      rep("fairly_comfortable", 18L),
                      rep("comfortable", 23L),
                      rep("good", 9L),
                      rep("significant_hardship", 27L),
                      rep("fairly_comfortable", 22L),
                      rep("comfortable", 52L),
                      rep("good", 20L)))

## calculates E, then formula for chi-squared
calc_X2 <- function(observed) {
  expected <- outer(rowSums(observed), colSums(observed))/sum(observed)
  sum((observed - expected)^2/expected)
}

## calculate observed chi-square test statistic
obs_X2 <- calc_X2(table(families)) %>% round(digits = 4)

N <- 1000
set.seed(101)
perm_X2 <- 1:N %>%
  ## shuffle dependent child values
  lapply(function(i) sample(families$dependent_children)) %>%
  ## generate table against fixed living standard values
  lapply(table, families$living_standard) %>%
  ## calculate chi-square test statistic
  sapply(calc_X2) %>%
```

```
  round(digits = 4)

p <- sum(perm_X2 >= obs_X2)/N

ggplot() +
  geom_histogram(data = data.frame(x = perm_X2), aes(x = x),
                 breaks = seq(from = 0, to = max(perm_X2), length.out = 20),
                 fill = "white", colour = "grey") +
  geom_vline(xintercept = obs_X2, colour = "firebrick", linetype = "dashed") +
  ggtitle("Histogram of the chi-square statistic") +
  xlab("test statistic") +
  geom_text(aes(x = 12.3, y = 50), label = "observed test statistic", colour = "firebrick")
```



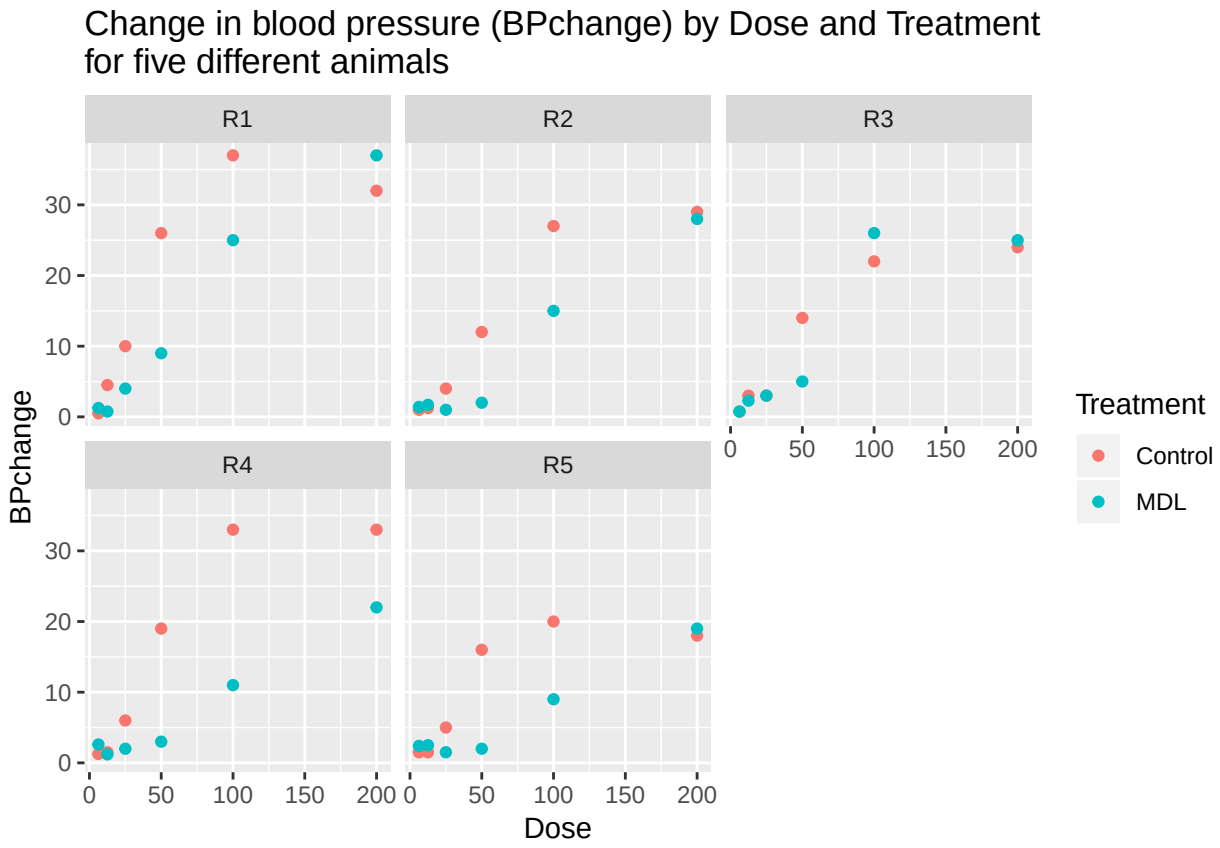
With p-value of 0.019 we reject H_0 at the 5% level. We conclude that there is a difference in living standards between families with dependent children and families without dependent children.

The histogram above clearly demonstrates that χ^2_{obs} lies well into the tail of the χ^2_{perm} distribution.

Question 4

a)

```
rabbits <- readr::read_csv("bp_rabbits.csv", col_types = readr::cols())
ggplot(data = rabbits, aes(x = Dose, y = BPchange, colour = Treatment)) +
  geom_point() +
  facet_wrap(~ Animal) +
  ggtitle("Change in blood pressure (BPchange) by Dose and Treatment\nfor five different animals")
```



b)

```
model <- lm(BPchange ~ Dose + Treatment + Animal, data = rabbits)
summary(model)
```

```
##
## Call:
## lm(formula = BPchange ~ Dose + Treatment + Animal, data = rabbits)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.342  -3.702  -1.263   2.027  14.554
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.74087    1.93350   4.521  3.5e-05 ***
## Dose          0.13992    0.01084  12.902  < 2e-16 ***
## TreatmentMDL -4.68000    1.46786  -3.188  0.00240 **
## AnimalR2     -5.30417    2.32089  -2.285  0.02632 *
## AnimalR3     -4.85000    2.32089  -2.090  0.04146 *
## AnimalR4     -4.28750    2.32089  -1.847  0.07028 .
## AnimalR5     -7.38333    2.32089  -3.181  0.00245 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.685 on 53 degrees of freedom
## Multiple R-squared:  0.7796, Adjusted R-squared:  0.7547
## F-statistic: 31.25 on 6 and 53 DF,  p-value: 9.529e-16
```

```r
anova(model)
```

```
## Analysis of Variance Table
##
## Response: BPchange
##           Df Sum Sq Mean Sq  F value    Pr(>F)
## Dose       1 5380.3  5380.3 166.4726 < 2.2e-16 ***
## Treatment  1  328.5   328.5  10.1653  0.002402 **
## Animal     4  351.4    87.9   2.7185  0.039217 *
## Residuals 53 1712.9    32.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Animal is included as a predictor to account for any of the effects or differences between individual rabbits. There are multiple observations of a single rabbit, each of which may have different reactions and baseline blood pressures when subjected to the drug, this needs to be accounted for.

c)

The procedure for the permutation test is as follows:

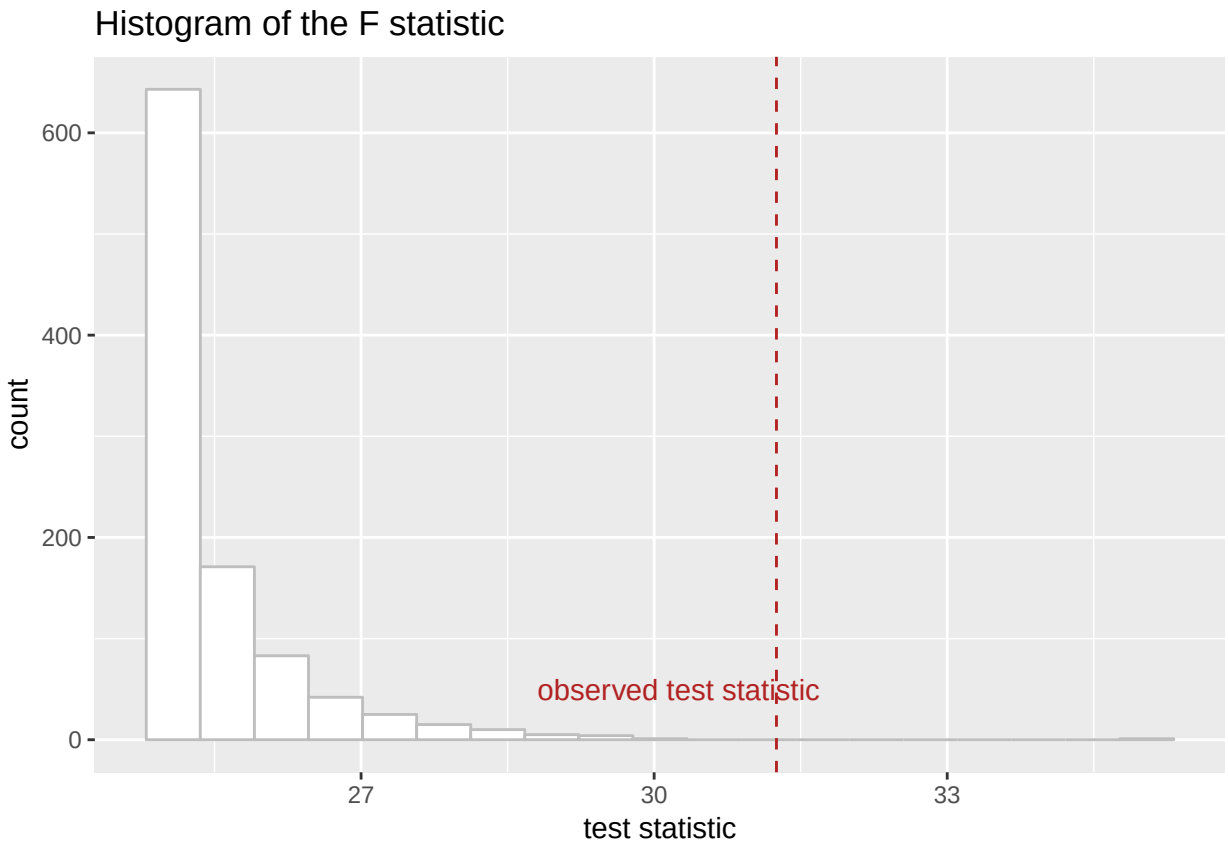
1. Calcualte the observed F statistic F_{obs}
2. For each permutation randomly allocate the 60 values for `Treatment` while hold `Dose` and `Animal` in place.
3. Calculate the permuted F statistic F_{perm}
4. The p-value is calculate as:

$$\text{p-value} = \frac{\text{No of } F_{perm} > F_{obs}}{\text{No of permutations}}$$

```r
calc_F <- function(treatment, dataset) {
  lm(BPchange ~ Dose + treatment + Animal, data = dataset) %>%
    summary() %>% .[["fstatistic"]] %>% .["value"] %>% unname()
}

obs_F <- calc_F(rabbits$Treatment, rabbits)
obs_F
```

```
## [1] 31.252
```

```r
N <- 1000
set.seed(101)
perm_F <- 1:N %>%
  lapply(function(i) {sample(rabbits$Treatment)}) %>%
  sapply(calc_F, rabbits)

p <- sum(perm_F >= obs_F)/N

ggplot() +
  geom_histogram(data = data.frame(x = perm_F), aes(x = x),
                 breaks = seq(from = min(perm_F), to = max(perm_F), length.out = 20),
                 fill = "white", colour = "grey") +
  geom_vline(xintercept = obs_F, colour = "firebrick", linetype = "dashed") +
  ggtitle("Histogram of the F statistic") +
  xlab("test statistic") +
  geom_text(aes(x = obs_F - 1, y = 50), label = "observed test statistic", colour = "firebrick")
```



With p-value of 0.001 we reject H_0 at the 5% level. We conclude that Treatment has an effect on the change in blood pressure after allowing for Dose and Animal.