STA303 Portfolio

An exploration of linear mixed models and common misconceptions in statistics

Smriti Kannan

2022-02-17

Contents

Introduction **3**

Statistical skills sample **4**

Task 1: Setting up libraries and seed value 4

Task 2a: Return to Statdew Valley: exploring sources of variance in a balanced experimental design (teaching and learning world) 4

Task 2b: Applying linear mixed models for the strawberry data (practical world) . . . 7

Task 3a: Building a confidence interval interpreter 9

Task 3b: Building a p value interpreter 10

Task 3c: User instructions and disclaimer 11

Task 4: Creating a reproducible example (reprex) 13

Task 5: Simulating p-values 14

Writing sample **18**

References 18

Reflection **20**

List of Figures

1 Strawberry patch yield based on different treatments 5

2 Distribution of simulated values from three different populations 15

3 Distribution of p values for different simulated data sets 16

4 Q-Q plots of p-values under each distribution 17

Introduction

This portfolio highlights specific skills across the three main components of the portfolio: the statistical skills samples, the writing sample and the reflection. The statistical skills samples display data visualization, analysis and modelling skills. For example, in Task 2 linear regression models (with a mix of fixed and/or random effects) are fitted and analyzed to find the best model for the data. The statistical skills samples also include function writing and simulation skills. For example, Task 3 involves building functions in R to interpret confidence intervals and p-values in hypothesis tests. Task 5 involves simulating data sets and understanding the distribution of p-values for these data sets using visualization tools such as Q-Q plots.

The writing skills samples portray critical reading and writing skills in the form of addressing an article written by Motulsky (2014), "Common misconceptions about data analysis and statistics." Some arguments made by Motulsky (2014) are reviewed along with why such arguments are important to consider. The main argument mentioned concerns how published research articles do not have reproducible results due to poor statistical practices such as p-hacking.

The reflection component highlights the strengths of this portfolio, how the skills displayed in this portfolio can prove to be useful in future work and suggestions for improvement of the portfolio.

Statistical skills sample

Task 1: Setting up libraries and seed value

```
# Setting up libraries and seed
library(tidyverse)
last3digitplus <- 100 + 272
```

Task 2a: Return to Statdew Valley: exploring sources of variance in a balanced experimental design (teaching and learning world)

Growinng your (grandmother's) strawberry patch

```
# Sourcing it makes a function available
source("grow_my_strawberries.R")

# Obtaining the data
my_patch <- grow_my_strawberries(seed = last3digitplus)

# Data wrangling to change the order of the levels of the treatment variable
my_patch <- my_patch %>%
  mutate(treatment = as_factor(treatment)) %>%
  mutate(treatment = fct_relevel(treatment, "No netting", after = 0)) %>%
  mutate(treatment = fct_relevel(treatment, "Netting", after = 1))
```

Plotting the strawberry patch

```
# Creating a scatter plot of the strawberry yield based on different treatments
my_patch %>%
  ggplot(aes(x=patch, y=yield, fill=treatment, color=treatment))+
  geom_point(pch=25)+
  theme_minimal() +
  scale_color_manual(values = c("#78BC61", "#E03400", "#520048"), name = "Treatment")
↪  +
  scale_fill_manual(values = c("#78BC61", "#E03400", "#520048"), name = "Treatment") +
  labs(x="Patch", y="Yield", caption = "Created by Smriti Kannan in STA303, Winter
↪  2022")
```

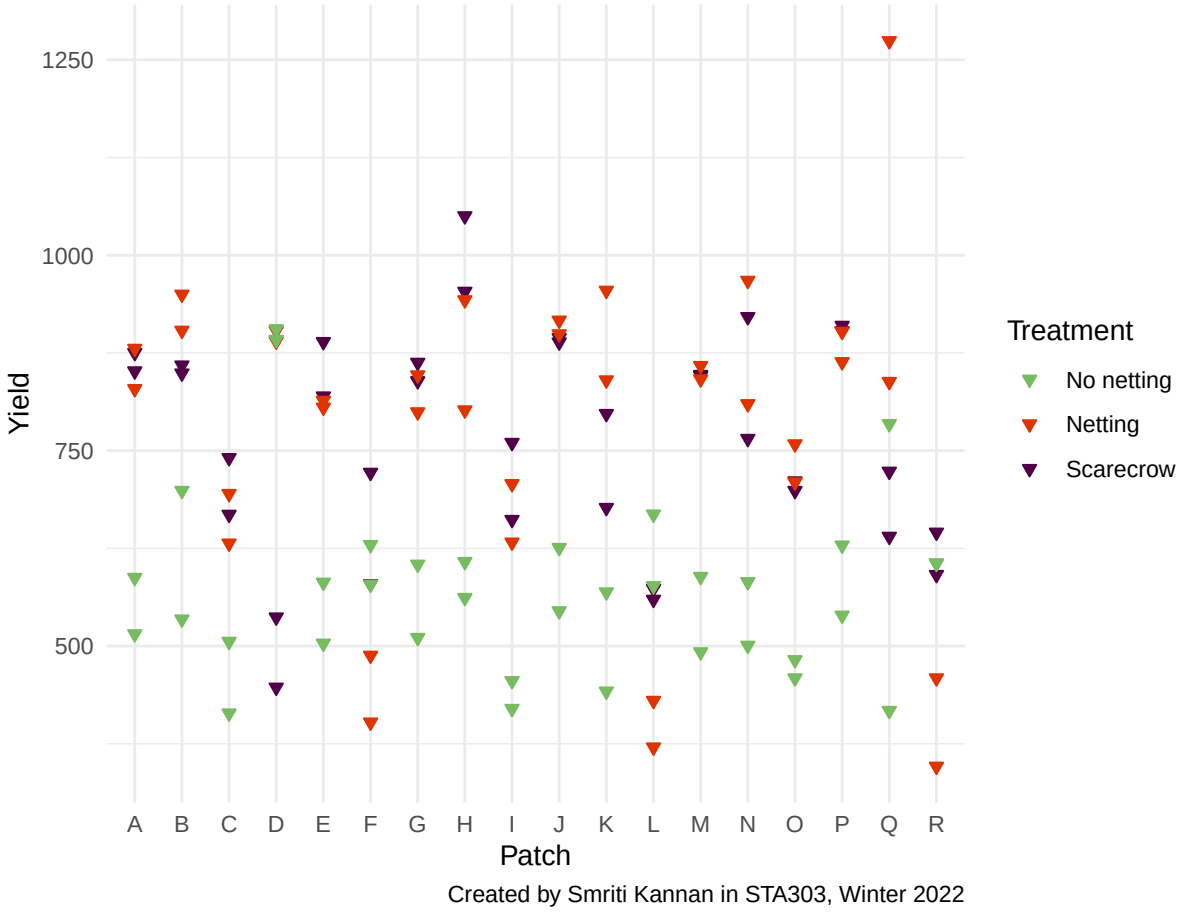

Figure 1: Strawberry patch yield based on different treatments

Demonstrating calculation of sources of variance in a least-squares modelling context

Model formula

$$y_{ijk} = \mu + \alpha_i + b_j + (\alpha b)_{ij} + \epsilon_{ijk}$$

where:

- y_{ijk} is the strawberry yield (in kg) produced with the ith treatment in patch j during the kth harvest
- μ is the mean strawberry yield
- α_i are the fixed effects for treatment
- b_j are the random effects for patch j where $b_j \sim N(0, \sigma_b^2)$
- $(\alpha b)_{ij}$ are the interaction terms for the interaction between the treatment and the patch where $(\alpha b)_{ij} \sim N(0, \sigma_{\alpha b}^2)$

- ϵ_{ijk} is the error term where $\epsilon_{ijk} \sim N(0, \sigma^2)$

```
# Creating a data set from my_patch that is aggregated across patch
agg_patch <- my_patch %>%
  group_by(patch) %>%
  summarize(yield_avg_patch = mean(yield))

# Creating a data set from my_patch that is aggregated across both patch and treatment
agg_int <- my_patch %>%
  group_by(patch, treatment) %>%
  summarize(yield_avg_int = mean(yield))

# Building an interaction model including the main effects
int_mod <- lm(yield ~ treatment*patch, data = my_patch)
# Building an intercept only model
patch_mod <- lm(yield_avg_patch ~ 1, data = agg_patch)
# Building a main effects model
agg_mod <- lm(yield_avg_int ~ treatment + patch, data = agg_int)

# Calculating variance in average yield patch-to-patch
var_patch <- summary(patch_mod)$sigma^2 -
↪  (summary(agg_mod)$sigma^2)/(nlevels(agg_int$treatment))
# Calculating residual variance
var_int <- summary(int_mod)$sigma^2
#Variance in yield explained by the interaction between patch and treatment
var_ab <- summary(agg_mod)$sigma^2 - var_int/(nrow(my_patch)/nrow(agg_int))
```

```
# Example tibble
tibble(`Source of variation` = c("Treatment:Patch",
                                 "Patch",
                                 "Residual"),
       Variance = c(var_ab, var_patch, var_int),
       Proportion = c(round(var_ab / (var_ab + var_patch + var_int), 2),
                      round(var_patch / (var_ab + var_patch + var_int), 2),
                      round(var_int / (var_ab + var_patch + var_int),2) )) %>%
  knitr::kable(caption = "Variance in strawberry yield based on different random
↪  effects")
```

Table 1: Variance in strawberry yield based on different random effects

Source of variation	Variance	Proportion
Treatment:Patch	14989.775	0.62
Patch	3168.031	0.13
Residual	6042.466	0.25

Task 2b: Applying linear mixed models for the strawberry data (practical world)

```
# Loading the lme4 package in order to construct linear mixed model
library(lme4)

# Creating a linear model with only treatment (fixed effect)
mod0 <- lm(yield ~ treatment, data = my_patch)

# Creating a linear mixed model with treatment (fixed effect) and patch (random
↪  effect)
mod1 <- lmer(yield ~ treatment + (1|patch), data = my_patch)

# Creating a linear mixed model with treatment, patch and the intersection of
↪  treatment and patch
mod2 <- lmer(yield ~ treatment + (1|patch) + (1|patch:treatment), data = my_patch)

# Comparing mod0 and mod1 using a likelihood ratio test
lmtest::lrtest(mod0, mod1)
```

```
## Likelihood ratio test
##
## Model 1: yield ~ treatment
## Model 2: yield ~ treatment + (1 | patch)
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   4 -695.64
## 2   5 -677.34  1 36.594  1.455e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Comparing mod1 and mod3 using a likelihood ratio test
lmtest::lrtest(mod1, mod2)
```

```
## Likelihood ratio test
##
## Model 1: yield ~ treatment + (1 | patch)
## Model 2: yield ~ treatment + (1 | patch) + (1 | patch:treatment)
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   5 -677.34
## 2   6 -660.59  1 33.506  7.104e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The method used to compare the models is REML (restricted maximum likelihood) as it estimates our model parameters (both random and fixed). Additionally, REML can produce unbiased estimates of variance and covariance parameters. This is useful as we can compare the different model variances to the ones calculated above in part 2a.

Justification and interpreation

From the likelihood test when we compare mod0 and mod1, we can see that the p-value is very small. Therefore, we have strong evidence against the null hypothesis that the simple model (mod0), without the random effect of patch, is as good as the model with the fixed and random effect (mod1). Similarly, with the second likelihood test where we compare mod1 and mod2, we can see that the p-value is very small. Therefore, we once again have strong evidence against the null hypothesis that the model with fixed and random effects (mod2), is as good as the model with the treatment/patch interaction term (mod2).

Hence, the most appropriate model is mod2 which includes the fixed effects of treatment, random effects of patch and the treatment/patch interaction term. Among the fixed effects, the netting treatment leads to the largest increase in strawberry yield when compared to no netting and the scarecrow. The scarecrow also leads to an increase in the strawberry yield when compared to no netting but not as much as the netting treatment. Among the random effects, the greatest proportion of variability in strawberry yield is explained by the variations in the interaction between treatment and patch.

Task 3a: Building a confidence interval interpreter

```
# Building a confidence interval interpreter function
interpret_ci <- function(lower, upper, ci_level, stat){
  if(!is.character(stat)) {
    # Produces a warning if the statement of the parameter isn't a character string
    warning("Warning: stat should be a character string that describes the statistic
    ↪  of interest in your confidence interval.")
  } else if(!is.numeric(lower)) {
    # Produces a warning if lower isn't numeric
    warning("Warning: lower should be a numeric value that describes the lower bound
    ↪  of your confidence interval.")
  } else if(!is.numeric(upper)) {
    # Produces a warning if upper isn't numeric
    warning("Warning: upper should be numeric value that describes the upper bound of
    ↪  your confidence interval.")
  } else if(!is.numeric(ci_level) | ci_level < 0 | ci_level > 100) {
    # Produces a warning if ci_level isn't appropriate
    warning("Warning: ci_level should be a numeric value between 0 and 100 as it
    ↪  describes the certainty with which your confidence interval captures the true
    ↪  population parameter.")
  } else{
    # Prints an interpretation
  str_c("Hence, we can conclude that the true population value of the ", stat,
        " lies between ", lower, " and ", upper, " with a ", ci_level, "% level of
        ↪  certainty.")
  }
}

# Test 1
ci_test1 <- interpret_ci(10, 20, 99, "mean number of shoes owned by students")

# Test 2
ci_test2 <- interpret_ci(10, 20, -1, "mean number of shoes owned by students")

# Test 3
ci_test3 <- interpret_ci(10, 20, -1, tibble(stat = 3))
```

CI function test 1: Hence, we can conclude that the true population value of the mean number of shoes owned by students lies between 10 and 20 with a 99% level of certainty.

CI function test 2: Warning: ci_level should be a numeric value between 0 and 100 as

it describes the certainty with which your confidence interval captures the true population parameter.

CI function test 3: Warning: stat should be a character string that describes the statistic of interest in your confidence interval.

Task 3b: Building a p value interpreter

```
# Building a p value interpreter function
interpret_pval <- function(pval, nullhyp){
  if(!is.character(nullhyp)) {
    # Produces a warning if the null hypothesis is not a character string
    warning("Warning: nullhyp should be a character string that describes the null
    ↪  hypothesis of your experiment.")
  } else if(!is.numeric(pval)) {
    # Produces a warning if the p value is not numeric
    warning("Warning: pval should be a number as it represents the p value.")
  }  else if(pval > 1) {
    # Produces a warning if the p value is greater than 1
    warning("Warning: pval cannot be greater than 1 as it denotes the p-value of your
    ↪  data which represents how likely your data would be observed under the null
    ↪  hypothesis being true.")
  } else if(pval < 0){
    # Produces a warning if the p value is less than 0
    warning("Warning: pval cannot be negative as it denotes the p-value of your data
    ↪  which represents how likely you data would be observed under the null
    ↪  hypothesis being true.")
    # Prints an interpretation based on the value of the p value
  } else if(pval > 0.1){
    str_c("Since the p-value is ", round(pval, 3), ", there is no evidence against the
    ↪  null hypothesis that ", nullhyp, ".")
  } else if(pval > 0.05 && pval <= 0.1){
    str_c("Since the p-value is ", round(pval, 3), ", there is weak evidence against
    ↪  the null hypothesis that ", nullhyp, ".")
  } else if(pval > 0.01 && pval <= 0.05){
    str_c("Since the p-value is ", round(pval,3), ", there is some or moderate
    ↪  evidence against the null hypothesis that ", nullhyp, ".")
  } else if(pval >= 0.001 && pval <= 0.01){
    str_c("Since the p-value is ", round(pval, 3), ", there is strong evidence against
    ↪  the null hypothesis that ", nullhyp, ".")
  } else if(pval < 0.001){
```

```
    str_c("Since the p-value is < 0.001 there is very strong evidence against the null
    ↪  hypothesis that ", nullhyp, ".")
  }

}

# Test 1
pval_test1 <- interpret_pval(0.0000000003,
                             "the mean grade for statistics students is the same as
                             ↪  for non-stats students")
# Test 2
pval_test2 <- interpret_pval(0.0499999,
                             "the mean grade for statistics students is the same as
                             ↪  for non-stats students")
# Test 3
pval_test3 <- interpret_pval(0.050001,
                             "the mean grade for statistics students is the same as
                             ↪  for non-stats students")
# Test 4
pval_test4 <- interpret_pval("0.05", 7)
```

p value function test 1: Since the p-value is < 0.001 there is very strong evidence against the null hypothesis that the mean grade for statistics students is the same as for non-stats students.

p value function test 2: Since the p-value is 0.05, there is some or moderate evidence against the null hypothesis that the mean grade for statistics students is the same as for non-stats students.

p value function test 3: Since the p-value is 0.05, there is weak evidence against the null hypothesis that the mean grade for statistics students is the same as for non-stats students.

p value function test 4: Warning: nullhyp should be a character string that describes the null hypothesis of your experiment.

Task 3c: User instructions and disclaimer

Instructions

The confidence interval interpreter takes in the lower and upper bounds of your confidence interval, the confidence level and the parameter of interest as inputs. A confidence interval predicts the range of values the true population parameter might take, with a specified degree of certainty. A

population parameter is a statistical measure for a given population. For example, the mean of the population is a population parameter. Most of the time, we work with sample data and do not have access to the entire population data set. Therefore, the true population parameter is unknown and we can use confidence intervals to estimate a range of values the population parameter might take (with a certain degree of confidence). The confidence interval interpreter function produces a useful output which can be used to understand what the confidence interval values mean.

The p-value interpreter takes in the p-value and the null hypothesis of your test as inputs. The null hypothesis of a test is deemed to be "true" until proved otherwise with data. A good start for writing out a null hypothesis is to first think about what your research question is. Once a research question has been clearly set, it can be rephrased into a null hypothesis appropriately. For example, suppose you are interested in finding out if doing extra homework has an effect on students' grades. The null hypothesis in this situation would be that the mean grades of students who do extra homework and those who do not are the same.

The p-value obtained from null hypothesis testing refers to the probability of obtaining results at least as extreme as the ones observed in your data. The p-value gives you a good idea of whether or not you have evidence against your null hypothesis. For example, if your p-value was less than 0.001, then you would have very strong evidence against the null hypothesis being true. The p-value interpreter helps one understand if they have evidence against their null hypothesis or not.

Disclaimer

The confidence interval interpreter assumes that it is appropriate to calculate a confidence interval and was done so correctly since the function only takes in the confidence interval bounds as inputs and does not calculate the confidence interval.

The p-value interpreter assumes that an appropriate test was performed to produce the p-value and that no assumptions were violated during the null hypothesis testing. Additionally, the p-value interpreter rounds the p-value given to three decimal places in the output but the original p-value is used to decide the strength of evidence.. Furthermore, the interpreter will give an output commenting on the strength of evidence (if any) against the null hypothesis based on the p-value. It is important to note that p-value interpretations and thresholds vary and this is just one method of interpreting p-values.

Task 4: Creating a reproducible example (reprex)

A reprex is an example that is easily reproducible by another person. For example, a reprex can be used to easily convey the error in one's code to their peers and allow their peers to reproduce the same error and help fix the code. When creating a reprex, one needs to ensure that the necessary libraries and data are included in the reprex in order to allow someone else to reproduce the code with ease. Additionally one should try to avoid paths to files in their reprex as a file path will not work on someone else's computer.

An example of a simple reprex where we are trying to create a new data set which contains the mean of each group:

```
library(tidyverse)
my_data <- tibble(group = rep(1:10, each=10),
                  value = c(16, 18, 19, 15, 15, 23, 16, 8, 18, 18, 16, 17, 17,
                            16, 37, 23, 22, 13, 8, 35, 20, 19, 21, 18, 18, 18,
                            17, 14, 18, 22, 15, 27, 20, 15, 12, 18, 15, 24, 18,
                            21, 28, 22, 15, 18, 21, 18, 24, 21, 12, 20, 15, 21,
                            33, 15, 15, 22, 23, 27, 20, 23, 14, 20, 21, 19, 20,
                            18, 16, 8, 7, 23, 24, 30, 19, 21, 25, 15, 22, 12,
                            18, 18, 24, 23, 32, 22, 11, 24, 11, 23, 22, 26, 5,
                            16, 23, 26, 20, 25, 34, 27, 22, 28))

my_summary <- my_data %>%
  summarize(group_by = group, mean_val = mean(value))

glimpse(my_summary)
#> Rows: 100
#> Columns: 2
#> $ group_by <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3...
#> $ mean_val <dbl> 19.67, 19.67, 19.67, 19.67, 19.67, 19.67, 19.67, 19.67, 19.67...
```

Task 5: Simulating p-values

Setting up simulated data

```
# Setting seed
set.seed(last3digitplus)

# Creating three data frames with simulated values
sim1 <- tibble(
  group = rep(1:1000, each = 100),
  val = rnorm(100000, mean = 0, sd = 1))
sim2 <- tibble(
  group = rep(1:1000, each = 100),
  val = rnorm(100000, mean = 0.2, sd = 1))
sim3 <- tibble(
  group = rep(1:1000, each = 100),
  val = rnorm(100000, mean = 1, sd = 1))

# Creating one data set which combines sim1, sim2 and sim3
all_sim <- bind_rows(sim1, sim2, sim3, .id = "sim")

# Creating a data set to merge with all_sim for improved simulation names
sim_description <- tibble(sim = 1:4,
                          desc = c("N(0, 1)",
                                   "N(0.2, 1)",
                                   "N(1, 1)",
                                   "Pois(5)"))
# Altering the all_sim data set to ensure it can be joined with sim_description
all_sim <- all_sim %>%
  mutate(sim = as.integer(sim))

# Joining the all_sim and sim_description data sets
all_sim <- left_join(all_sim, sim_description, by = "sim")
```

```
# Creating histograms to visualize the distributions of data in all_sim
all_sim %>%
  filter(group <= 3) %>%
  ggplot(aes(x = val)) +
  geom_histogram(bins = 40) +
  facet_wrap(desc~group, nrow = 3) +
  theme_minimal() +
```

```
  labs(x = "Value", y = "Count", caption = "Created by Smriti Kannan in STA303, Winter
↪  2022")
```

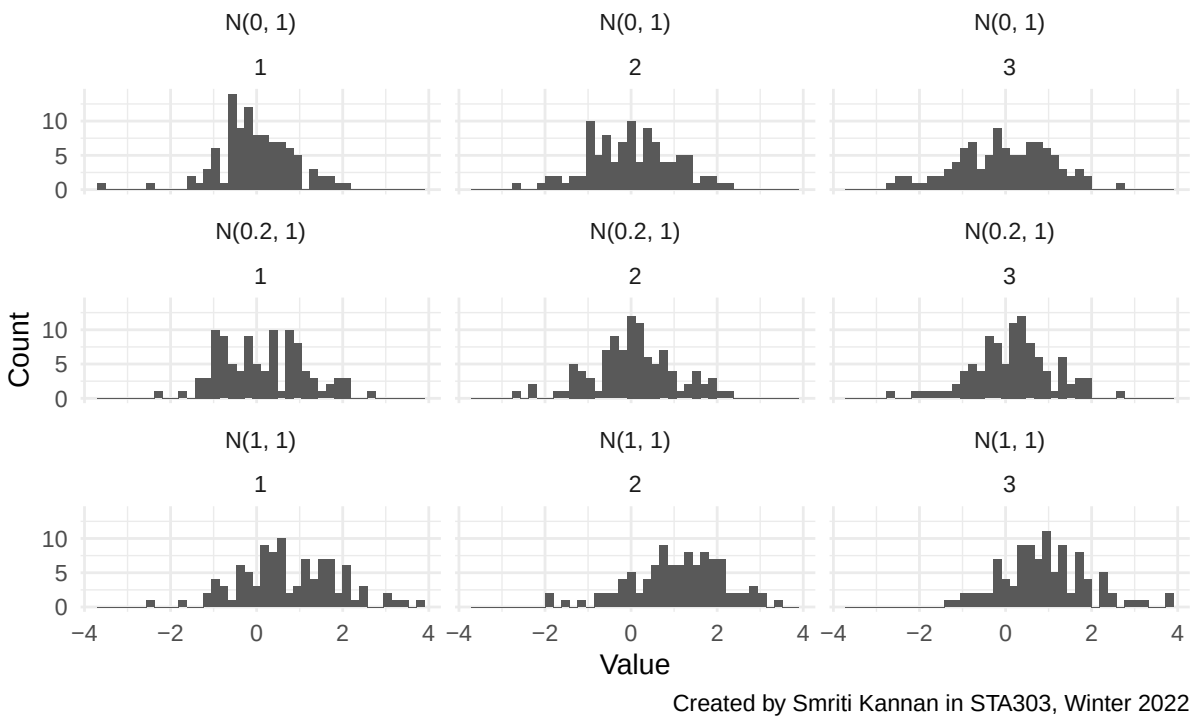

Figure 2: Distribution of simulated values from three different populations

Calculating *p* values

```
# Creating a new data set which includes p-values for the data in all_sim
pvals <- all_sim %>%
  group_by(desc, group) %>%
  mutate(pval = t.test(val, mu = 0)$p.value)
```

```
# Creating histograms for the p-values of all the simulated data
pvals %>%
  ggplot(aes(x=pval)) +
  geom_histogram(boundary = 0, binwidth = 0.05, fill = "grey", color = "black") +
  facet_wrap(~desc, nrow = 1, scales = "free_y") +
  theme_minimal() +
```

```
  xlim(0,1) +
  labs(x = "P-value", y = "Count", caption = "Created by Smriti Kannan in STA303,
↪  Winter 2022")
```

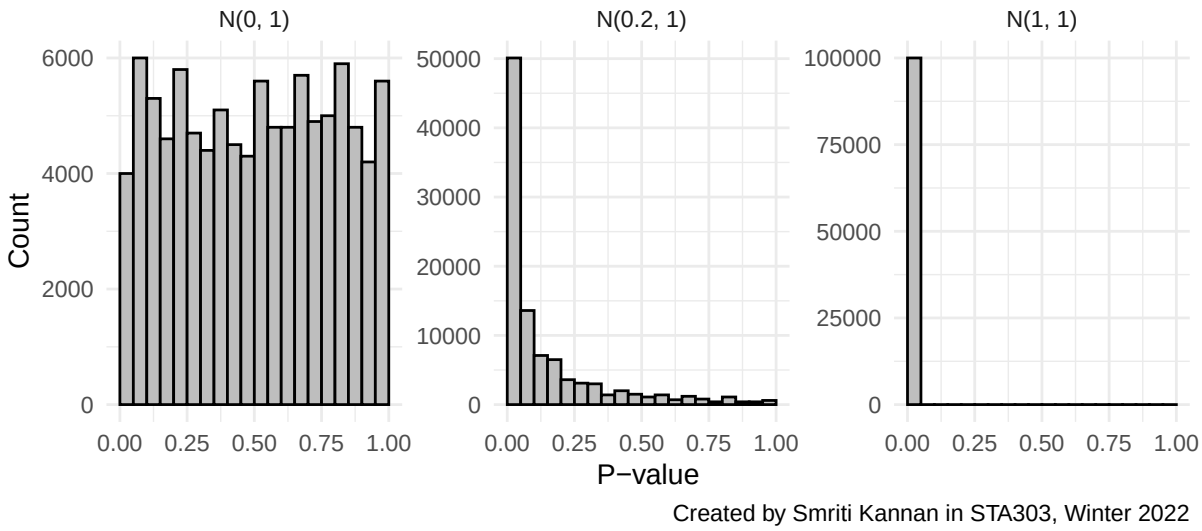

Figure 3: Distribution of p values for different simulated data sets

Drawing Q-Q plots

```
# Graphing QQ plots for p-values under each distribution in the simulation
pvals %>%
  ggplot(aes(sample = pval)) +
  geom_qq(distribution = stats::qunif) +
  geom_abline(intercept = 0, slope = 1) +
  facet_wrap(~desc) +
  theme_minimal() +
  labs(x = "Theoretical", y = "Sample", caption = "Created by Smriti Kannan in STA303,
↪  Winter 2022")
```

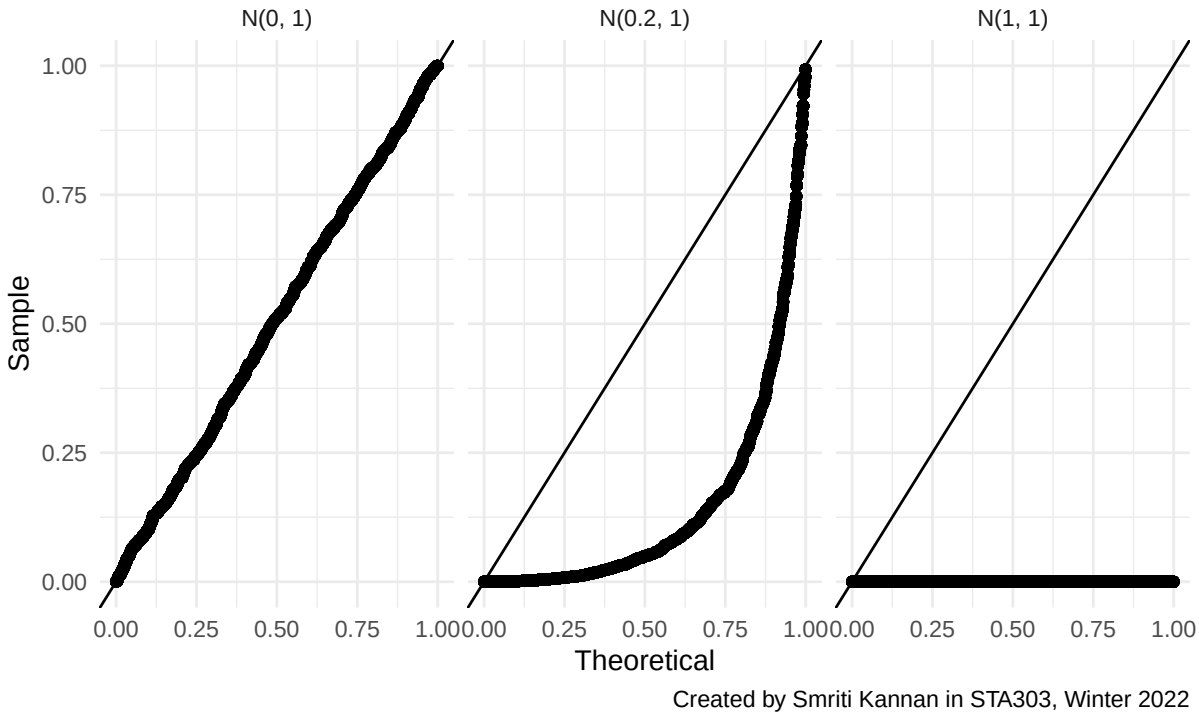

Figure 4: Q-Q plots of p-values under each distribution

Conclusion and summary

In this task, three data sets were simulated from normal distribution with different means (0, 0.2 and 1). Then, the p-values were calculated based on a one-sample, two sided t-test, with the null hypothesis that the population mean is zero. The p-values for each distribution were visualized using histograms and Q-Q plots where the quantile function was the uniform distribution. Looking at these figures, we can see that the data with a normal distribution mean of zero has p-values which follow a uniform distribution. Furthermore, when we look at the Q-Q plot for this distribution, all the points are on the line and form a straight line. This is quite different from the figures for the other two distributions (with mean 0.2 and 1). The p-values for the other two distributions are heavily skewed and a very small number of points line on the straight line in the Q-Q plot.

Therefore, when we simulate multiple data sets of normally distributed data points (with mean 0 and variance 1) and perform a one-sample t-test on each dataset (where the null hypothesis is that the population mean is zero), the resulting histogram of p-values will show that the p-values have a uniform distribution. Additionally, approximately 10% of the p-values will lie between 0.9 and 1, since the p-values are uniformly distributed.

Writing sample

In the article, "Common misconceptions about data analysis and statistics" Motulsky (2014) raises the concern about how many published research articles do not have reproducible findings. For example, only 6 in 53 studies in basic cancer biology were reproducible by investigators at Agmen (Motulsky, 2014). Motulsky (2014) attributes this issue in reproducibility to many reasons, one being poor understanding of statistical concepts which leads to mistakes such as p-hacking being made. Motulsky (2014) highlights the many ways (some subtle and some more obvious) in which p-hacking can take place in research.

Some of the findings presented by Motulsky (2014) can be useful when conducting your own research and data analysis in order to avoid introducing biases into your data. Motulsky (2014) goes over some common methods of p-hacking which include, ad hoc sample size selection and hypothesizing after the result is known. Ad hoc sample size selection involves not choosing a sample size before conducting data analysis and changing the sample size until desirable results are produced (Motulsky, 2014). Hypothesizing after the result is known (HARKing) involves forming a hypothesis after data visualization and analysis have been conducted but are presented in the research prior to the analysis (Motulsky, 2014). It is important to keep in mind such examples of how data analysis can become biased when we design our own research and experiments.

Additionally, Motulsky (2014) advises authors of research papers on what to do concerning p-hacking. Motulsky (2014) states that authors should clearly mention if the sample size was chosen in advance for tables and figures in a report. Motulsky (2014) also advises that if any form of p-hacking was used in the analysis to publish the conclusions as "preliminary". These are great ways to ensure that the readers of the research understand the methodologies used and also comprehend the significance of the findings appropriately. While this is very important, it is also critical to understand how to avoid such biases in your analysis. Some ways to avoid this would be to set sample sizes before the data analysis and also form a hypothesis before data analysis and testing.

Therefore, by understanding the different ways in which p-hacking can take place in research, we have a better understanding of why many studies do not have reproducible findings. Furthermore, increasing awareness of the different types of p-hacking can help researchers avoid such mistakes and produce better quality data analysis.

Word count: 400 words

References

Motulsky, H. J. (2014). Common misconceptions about data analysis and statistics. *Naunyn-Schmiedeberg's Archives of Pharmacology*, *387*(11), 1017–1023. https://doi.org/10.1007/s00210-014-1037-6

Reflection

What is something specific that I am proud of in this portfolio?

I am quite proud of the data visualization and model explanation in Task 2. Figure 1 clearly displays the strawberry yield based on patch and treatment type (with different colours for each treatment) making it easy to interpret the data. Additionally, writing out the model formula using LaTeX makes it very easy to understand what the different variables in the model are. Both the data visualization and model formula with the explanation of variables makes it easy for a reader to understand what type of analysis is being conducted. It is also highly useful when presenting the data analysis to others.

How might I apply what I've learned and demonstrated in this portfolio in future work and study, after STA303/1002?

I have increased my statistical model knowledge by learning about linear mixed models and applying them, specifically in Task 1. This is a useful addition to my data analysis and modelling knowledge which can be used in future research and data analysis projects. Additionally, I have also improved and demonstrated my function writing skills in R in Task 3. Writing functions in R to interpret p-values and confidence intervals has improved my interpretation skills of such statistical values which can be useful in reports for explaining a statistical model and its results. Furthermore, this task has improved my function building skills in R which can be useful for future projects where building a function could prove to be more useful than just hardcoding.

What is something I'd do differently next time?

One thing I would do differently next time would be to slightly modify the data visualization in Task 5, specifically make some modifications for Figure 3 and Figure 4. In order to improve the visualization, I think it would be better to add a fill to the histograms and change the colour of the Q-Q plots according to the distribution of the data. Having different colours for the different distributions(N(0, 1), N(0.2, 1) and N(1, 1)) would make it easier to understand and interpret what the graphs are trying to convey about p-values and the distributions of the data set.