STA303/1002 Portfolio

An exploration of linear mixed models and common misconceptions in statistics

Yuchen Zeng

2022-02-03

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 12

Task 4: Creating a reproducible example (reprex) 13

Task 5: Simulating p-values 15

Writing sample **19**

References 20

Reflection **21**

List of Figures

1 Scatter plot of Patch vs. Yield with treatments No netting, Netting, Scarecrow . 5

2 Histograms for the first three groups for each simulated dataset. 16

3 Histogram of p-values for the first three groups for each simulated dataset. 17

4 Q-Q plots with uniform as the quantile function for the first three groups for each simulated dataset. 18

Introduction

This portfolio is an assignment of STA303 at the University of Toronto. This course discusses the exploration of data sets, data visualizations, data interpretation, and ethics in data analysis. It also studies various statistical models using R. In this portfolio, I presented my statistical knowledge, data wrangling and visualization skills, writing skills, and self-reflection.

The first section is a presentation of statistical analysis and R skills. I presented how to set up libraries in R and explore sources of variance in a balanced experimental design. This involved writing the model formulation and calculating variances accordingly. Then, I applied and interpreted linear mixed models on the same set of data. After that, I showed my understanding of confidence intervals and p-values by creating interpreters for them respectively. Furthermore, I demonstrated the creation of reproducible examples. Lastly, I simulated p-values for different distributions under a null hypothesis and explored their distribution.

The second section is an article about increasing reproducibility of statistical research based on Motulsky's article in 2014. I discussed some of the major causes of low reproducibilities, such as p-hacking, overemphasizing p-value, and overusing statistical hypothesis tests. I also suggested solutions to these issues, such as mentioning details of our methods and experiments, including effect sizes and decisions that may affect reproducibility.

The last section is a reflection of my performance in this portfolio. I discussed the part I am proud of, what I have learned to apply in future work and study, and what I would do differently next time.

Statistical skills sample

Task 1: Setting up libraries and seed value

```
# Load tidyverse library
library(tidyverse)
# Set up last3digplus
# Original seed of 100+825 does not work
# Prof B gave permission to use an altered seed here.
# Ref: https://piazza.com/class/kx47tj4fmy65dg?cid=361
last3digplus = 100 + 826
```

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

```
# Don't edit this file
# Sourcing it makes a function available
source("grow_my_strawberries.R")
# run grow_my_strawberries
my_patch = grow_my_strawberries(seed = last3digplus)
# mutate my_patch so treatment is a factor with given order
my_patch <- my_patch %>%
  mutate(treatment = fct_relevel(treatment, "No netting", after = 0))
```

Plotting the strawberry patch

```
# Create the plot
ggplot(aes(x = patch, y = yield, fill = treatment, color = treatment),
       data = my_patch) +
  # Make points triangles
  geom_point(pch = 25) +
  theme_minimal() +
  # Colour the points
  scale_fill_manual(values = c("#78BC61", "#E03400", "#520048")) +
```

```
scale_color_manual(values = c("#78BC61", "#E03400", "#520048")) +
labs(caption = "Created by Yuchen Zeng in STA303/1002, Winter 2022")
```

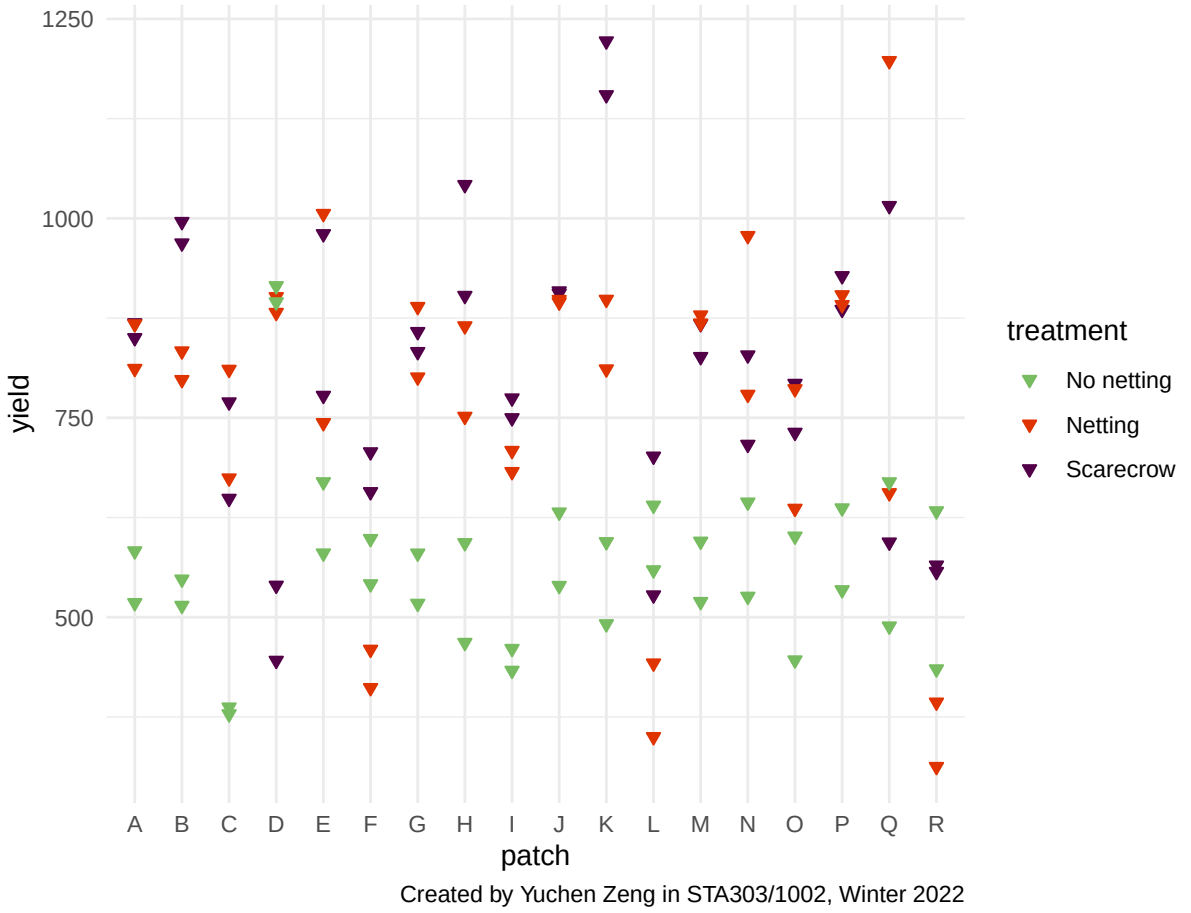

Figure 1: Scatter plot of Patch vs. Yield with treatments No netting, Netting, Scarecrow

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 strawberries yields (in kgs) in the k^{th} harvest time in the jth patch while applying treatment i.
- μ is the grand mean of strawberries yields.

- α_i are the fixed effects for treatment i, where i can be 1(No netting), 2(Netting), 3(Scarecrow).
- b_j are the random effects for patch j, where j can be patch $A, B, \ldots, R$(18 patches in total).
- $(\alpha b)_{ij}$ are the random effects for the interaction between the the treatment i and the patch j. There are a total of 54 interaction terms.
- $(\alpha b)_{ij} \sim N(0, \sigma^2_{\alpha b})$, $b_j \sim N(0, \sigma^2_b)$ and $\epsilon_{ijk} \sim N(0, \sigma^2)$
- All the random effects are mutually independent random variables.

```r
# Tibble group by patch and then summarize
agg_patch <- my_patch %>%
  # Group patch
  group_by(patch) %>%
  summarize(yield_avg_patch = mean(yield))

# Tibble with average strawberry yield for each patch and treatment combination
agg_int <- my_patch %>%
  # Group patch and treatment
  group_by(treatment, patch) %>%
  summarize(yield_avg_int = mean(yield), .groups = "drop")
```

```r
# Create an interaction model including main effects
int_mod <- lm(yield ~ patch * treatment, data = my_patch)

# Create an intercept only model
patch_mod <- lm(yield_avg_patch ~ 1, data = agg_patch)

# Create an main effects model
agg_mod <- lm(yield_avg_int ~ patch + treatment, data = agg_int)
```

```r
# Calculate variance in average yield patch-to-patch
var_patch <- (summary(patch_mod)$sigma)^2 - (summary(agg_mod)$sigma^2)/3

# Calculate residual variance
var_int <- (summary(int_mod)$sigma)^2

# Calculate variance in yield explained by interaction between patch and treatment
var_ab <- (summary(agg_mod)$sigma)^2 - var_int/2
```

```r
# Create the tibble of proportion of variances
tibble(`Source of variation` = c("patch:treatment",
                                 "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 = "Proportion of Variances Explained by Different Sources")
```

Table 1: Proportion of Variances Explained by Different Sources

Source of variation	Variance	Proportion
patch:treatment	14087.297	0.50
patch	4735.459	0.17
residual	9302.245	0.33

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

```r
# Fit linear model with only treatment
mod0 <- lm(yield ~ treatment, data=my_patch)
# Fit linear mixed model with treatment and patch
mod1 <- lme4::lmer(yield ~ treatment + (1 | patch), data=my_patch)
# Fit linear mixed model with treatment, patch and their interaction
mod2 <- lme4::lmer(yield ~ treatment + (1 | patch) + (1 | patch:treatment),
                   data=my_patch)

lmtest::lrtest(mod0, mod1, mod2)
```

```
## Likelihood ratio test
##
## Model 1: yield ~ treatment
## Model 2: yield ~ treatment + (1 | patch)
## Model 3: yield ~ treatment + (1 | patch) + (1 | patch:treatment)
##   #Df  LogLik Df  Chisq Pr(>Chisq)
```

```
## 1   4 -703.88
## 2   5 -684.84  1 38.075  6.808e-10 ***
## 3   6 -674.44  1 20.807  5.080e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

I am using REML. ML is used when we need to compare two nested models based on their fixed effects, but our main goal is estimating our model parameters with both random and fixed effects. mod0 and mod1 are nested model where mod1 has patch as an extra random effect. mod2 has an extra random effect, the interaction between patch and treatment, compared to mod1. In this case, REML is preferred.

```
summary(mod2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: yield ~ treatment + (1 | patch) + (1 | patch:treatment)
##    Data: my_patch
##
## REML criterion at convergence: 1348.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.4517 -0.4174  0.0300  0.4546  3.1648
##
## Random effects:
##  Groups          Name        Variance Std.Dev.
##  patch:treatment (Intercept) 14087    118.69
##  patch           (Intercept)  4735     68.81
##  Residual                      9302     96.45
## Number of obs: 108, groups:  patch:treatment, 54; patch, 18
##
## Fixed effects:
##                    Estimate Std. Error t value
## (Intercept)          565.73      36.11  15.666
## treatmentNetting     197.39      45.63   4.326
## treatmentScarecrow   242.59      45.63   5.317
##
## Correlation of Fixed Effects:
```

```
##              (Intr) trtmnN
## trtmntNttng -0.632
## trtmntScrcr -0.632  0.500
```

Justification and interpreation

mod2 is the most appropriate final model. When comparing mod0 and mod 1 using the Likelihood ratio test, we get a very small p-value <0.001, which means we have strong evidence against the null hypothesis that mod0 explains as much as mod1. Similarly, with a small p-value of about 0.012 when comparing mod1 and mod2, we have moderate evidence against the hypothesis that mod1 is as good as mod2. Therefore, mod2 is the best model.

For the fix effects, the value of the intercept means the average strawberry yields are 565.73 kg with no netting treatment. The value of treatmentNetting means there is an average increase of 197.39 kg in strawberry yields by applying netting treatment compared to no netting treatment. Similarly, the value of treatmentScarecrow means there is an average increase of 242.59 kg in strawberry yields compared to no netting. For the proportion of variance that is not explained by the fixed effect, 50% is explained by the interaction between patch and treatment, 17% is explained by patch, 33% is explained by residual.

Task 3a: Building a confidence interval interpreter

```
interpret_ci <- function(lower, upper, ci_level, stat){
  if(!is.character(stat)) {
    # produce a warning if the statement of the parameter isn't a character string
    warning("
    Warning:
    stat should be a character string that describes the statistics of
    interest. ")
  } else if(!is.numeric(lower)) {
    # produce a warning if lower isn't numeric
    warning("Warning: lower should be a numeric value describing lower bound
            of the confidence interval.")
  } else if(!is.numeric(upper)) {
    # produce a warning if upper isn't numeric
    warning("Warning: upper should be a numeric value describing upper bound
            of the confidence interval.")
  } else if(!is.numeric(ci_level) | ci_level < 0 | ci_level > 100) {
    # produce a warning if ci_level isn't appropriate
```

```
    warning("Warning: ci_level should be a numeric value between 0 and 100,
            describing the confidence level this interval was calculated at.")
  } else{
    # print interpretation
  str_c("When taking a large amount of repeated samples with a fixed sample size,
        approximately ", ci_level,
        "% of the samples have the population ", stat,
        " lies within the interval between ", lower, " and ", upper,
        "." )
  }
}

# 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: When taking a large amount of repeated samples with a fixed sample size, approximately 99% of the samples have the population mean number of shoes owned by students lies within the interval between 10 and 20.

CI function test 2: Warning: ci_level should be a numeric value between 0 and 100, describing the confidence level this interval was calculated at.

CI function test 3: Warning: stat should be a character string that describes the statistics of interest.

Task 3b: Building a p value interpreter

```
# message=FALSE means we will not get the warnings
interpret_pval <- function(pval, nullhyp){
  if(!is.character(nullhyp)) {
    warning("
    Warning:
    nullhyp should be a character string describing the null hypothesis.")
  } else if(!is.numeric(pval)) {
```

```
    warning("Warning: pval should be a numeric value.")
  }  else if(pval > 1) {
    warning("
            Warning: pval should be a numeric value less than or equal to 1.")
  } else if(pval < 0){
    warning("
            Warning: pval should be a numeric value greater than or equal to 0.")
  } else if(pval > 0.1){
    str_c("The p value is ", round(pval, 3),
                ", there is no evidence against the null hypothesis that ", nullhyp,
                ↪  ".")
  } else if(pval > 0.05){
    str_c("The p value is ", round(pval, 3),
                ", there is weak evidence against the null hypothesis that ", nullhyp,
                ↪  ".")
  } else if(pval > 0.01){
    str_c("The p value is ", round(pval, 3),
                ", there is moderate evidence against the null hypothesis that ",
          nullhyp, ".")
  } else if(pval > 0.001){
    str_c("The p value is ", round(pval, 3),
                ", there is strong evidence against the null hypothesis that ",
          nullhyp, ".")
  } else if(pval <= 0.001){
    str_c("The p value is <.001, there is very strong evidence against the null
          hypothesis that ", nullhyp, ".")
  }
}

pval_test1 <- interpret_pval(0.0000000003,
                             "the mean grade for statistics students is the same as
                             ↪  for non-stats students")

pval_test2 <- interpret_pval(0.0499999,
                             "the mean grade for statistics students is the same as
                             ↪  for non-stats students")

pval_test3 <- interpret_pval(0.050001,
                             "the mean grade for statistics students is the same as
                             ↪  for non-stats students")

pval_test4 <- interpret_pval("0.05", 7)
```

p value function test 1: The p value is <.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: The p value is 0.05, there is 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: 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 describing the null hypothesis.

Task 3c: User instructions and disclaimer

Instructions

A population parameter is a fixed value calculated from all subjects from the field we are interested in. For example, a population parameter can be the average height of all University Students. Usually, we do not have all the data, so we calculate confidence intervals from our samples to learn information about the population parameter. The confidence interval interpreter interpret_ci requires numeric lower-bound and upper-bound of the confidence interval called lower and upper respectively. It also needs the significant level called ci_level, and the parameter you are interested in called stat. It is important to know that the confidence level does not describe the chance of the population parameter lying within the confidence interval because a population parameter is a fixed value that will not change. Also, you cannot tell the probability of the parameter lies in the confidence intervals since confidence intervals are different for each sample.

The p-value interpreter interpret_pval is used when you want to compare two or more groups. It requires a p-value called pval and a null hypothesis called nullhyp. The null hypothesis is a type of hypothesis that suggests there is no difference or relationship in the characteristic between the groups you are interested in. For example, when you want to compare the mean grade for statistics students and non-statistic students, the null hypothesis should be "the mean grade of statistics students and non-statistic students are the same." The p-value is a numeric value ranging from 0 to 1, explaining the likelihood to observe the value in our sample, assuming the null hypothesis is true. By inputting the p-value and null hypothesis, the interpreter will output the strength of evidence found in our sample against the null hypothesis. Furthermore, when getting an exact value on the boundary for each strength, the p-value interpreter will interpret the p-value in the lower range.

Disclaimer

Before using the p-value interpreter, it is the user's responsibility to make sure inputs are valid. According to the definition of p-value, pval should be a numeric value ranging from 0 to 1. Also, nullhyp should be a character string that describes the appropriate null hypothesis. Moreover, it is the user's responsibility to make sure assumptions for the test are satisfied before interpreting the p-value. Parametric tests make assumptions about the distribution of the population from the sample. For example, when using a one-sample t-test, the data should be continuous and normally distributed. Hypothesis tests also ask the sample to be a simple random sample from its population. When the assumptions are violated, the interpretation generated from interpret_pval is not reliable for your data.

Task 4: Creating a reproducible example (reprex)

A reprex is a reproducible example. When encountering a problem in the code, reprex packages codes in a way that others can easily reproduce them. The reprex package in the tidyverse library allows producing r code as a reprex. When I encounter a problem with my code, by producing a reprex, others can run my code easily and help me find the solution. When producing a reprex, I need to include all libraries used and the creation of objects that I used in the code. Otherwise, the code may not run properly on other people's computers. Also, I should present all warning messages, because sometimes they can help others quickly find the issue. Furthermore, reprex should be minimal. I should remove all codes and objects that are not directly related to the problem, which will make it easier for others to reproduce and figure out the solution faster.

```
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, 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

```
# Set seed to last3digplus
set.seed(last3digplus)

# Create simulated data sets
sim1 <- tibble(group = rep(1:1000, each = 100), val = rnorm(100000,0,1))
sim2 <- tibble(group = rep(1:1000, each = 100), val = rnorm(100000,0.2,1))
sim3 <- tibble(group = rep(1:1000, each = 100), val = rnorm(100000,1,1))
# Create a new dataset includes sim1 sim2 sim3
all_sim = bind_rows(sim1, sim2, sim3, .id="sim")
# Mutate all_sim so that sim values is numeric
all_sim <- all_sim %>%
  mutate(sim = as.numeric(sim))

# Create sim_description
# Dataset to merge with improved simulation names
sim_description <- tibble(sim = 1:4,
                          desc = c("N(0, 1)",
                                   "N(0.2, 1)",
                                   "N(1, 1)",
                                   "Pois(5)"))
# join all_sim with sim_description without including any irrelevant labels
all_sim <- all_sim %>%
  left_join(sim_description, by = "sim")
```

```
# Plot histograms for the first three groups for each simulated dataset in one plot
all_sim %>%
  filter(group <= 3) %>%
  ggplot(aes(x = val)) +
  geom_histogram(bins = 40) +
  facet_wrap(desc~group, nrow = 3) +
  theme_minimal() +
  labs(caption = "Created by Yuchen Zeng in STA303/1002, Winter 2022")
```

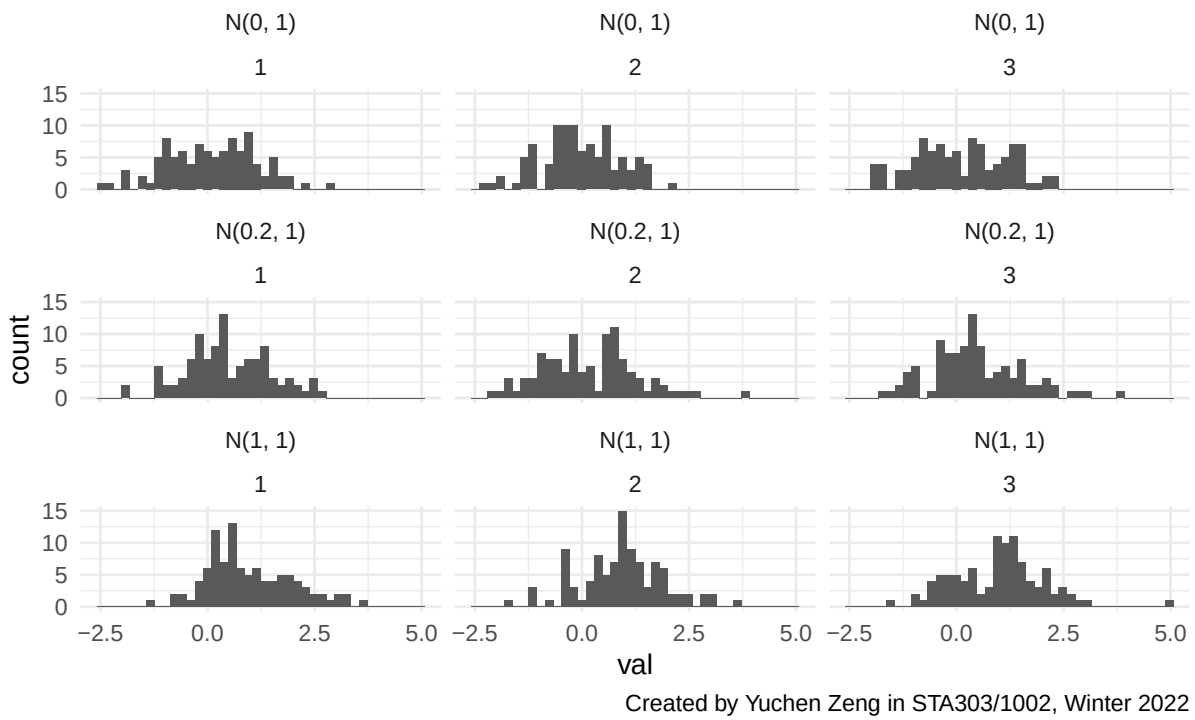

Figure 2: Histograms for the first three groups for each simulated dataset.

Calculating *p* values

```
# Create a new dataset of all_sim groups by both desc and group
pvals <- all_sim %>%
  group_by(desc, group) %>%
  # Summarizes to find the p value for each group by one sample, 2-sided t.test
  summarise(pval = t.test(val, mu = 0)$p.value, .groups = "drop")
```

```
pvals %>%
  ggplot(aes(x = pval)) +
  geom_histogram(boundary = 0, binwidth = 0.05, fill = "grey", color = "black") +
  xlim(0,1) +
  facet_wrap(~ desc, scales = "free_y") +
  theme_minimal() +
  labs(caption = "Created by Yuchen Zeng in STA303/1002, Winter 2022")
```

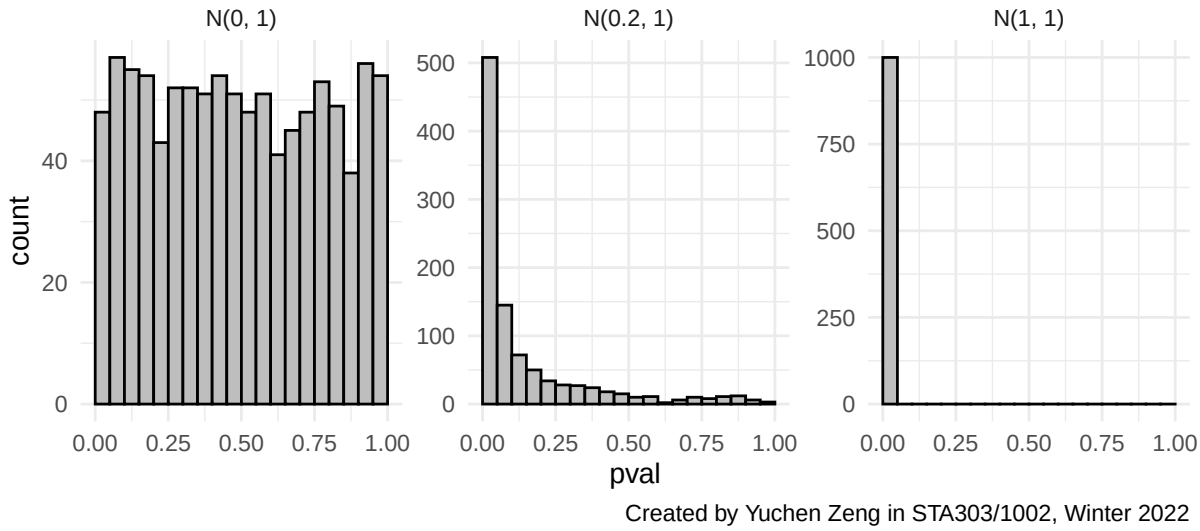

Figure 3: Histogram of p-values for the first three groups for each simulated dataset.

Drawing Q-Q plots

```
pvals %>%
  ggplot(aes(sample = pval)) +
  geom_qq(distribution = qunif) +
  geom_abline(intercept = 0, slope = 1) +
  facet_wrap(~desc) +
  theme_minimal() +
  labs(caption = "Created by Yuchen Zeng in STA303/1002, Winter 2022")
```

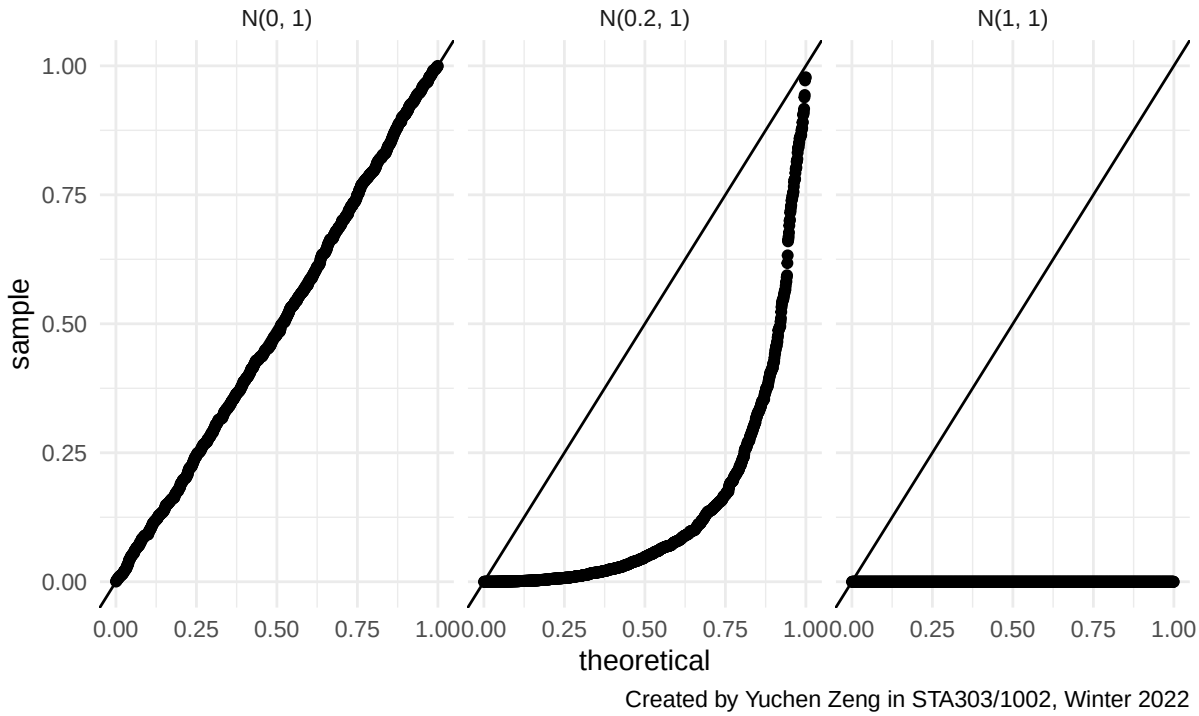

Figure 4: Q-Q plots with uniform as the quantile function for the first three groups for each simulated dataset.

Conclusion and summary

By definition, the p-value describes the likelihood of observing the sample assuming the null hypothesis is true. Assuming the population mean is 0, we found that the p-values for samples generated from a distribution with mean truly being 0 distributed approximately equally between 0 and 1. For samples generated from a distribution with a mean not equal to 0, most p-value we got from the hypothesis test is close to 0, meaning that it is very unlikely to observe such a sample when the null hypothesis is true.

Regarding the question from the pre-knowledge check, assuming the null hypothesis is true, as mentioned above, p-values for the samples generated from N(0,1) distributed approximately equally between 0 and 1, which follows U(0,1). Thus, for each 0.1 length interval from 0 to 1, there are approximately the same amount of p-values. Therefore, the right answer for this question should be approximately 10% of the p-values will be between 0.9 and 1.

Writing sample

Every researcher wishes their experimental result to be reproducible by others. However, a large amount of published journals has concerning reproducibility nowadays. One potential reason is the misunderstanding of statistical concepts. Hence, statistic students should learn to realize misconceptions when working with statistical analysis. In this article, I discussed some major misunderstandings of statistical concepts, including p-hacking, overemphasis on p-values, overuse of statistical hypotheses. I also described how to identify or avoid these issues in an experiment.

First of all, p-hacking can largely affect the reproducibility of statistical results. It is the general term for modifying or introducing more data to the experiment to get a statistically significant p-value. As stated by Motulsky, p-hacking introduces bias into the analysis, because researchers will not check the modified result if the first analysis is already statistically significant(2014). Common p-hacking issues include varying the sample size and hypothesizing after getting the result. To notify other experimenters of the reproducibility issue, Motulsky suggests explicitly states whether the sample size is fixed in advance, and labels "preliminary" in the conclusion if p-hacking is used(2014).

Also, we need to acknowledge that the p-value depends on sample size. P-value is the likelihood of observing the current experiment under the null hypothesis. It does not provide information on the level of differences between the two groups. Then, when two experiments yield the same p-value, the strength of evidence is similar but the effect size can be very different. Thus, a large p-value may not mean there is no difference, and a small p-value may not provide as much information as expected. Therefore, reporting the p-value can be misleading, and we should emphasize the effect size and its confidence interval instead(Motulsky, 2014).

Based on our discussion on the problems with p-value, it is reasonable that we should not overuse statistical hypothesis testing and the concept of "statistical significance." In Motulsky's article, he mentioned that we usually do not need to make a yes or no decision in basic research, and statistical hypothesis testing is usually not useful when a crisp decision is not needed(2014). Furthermore, when making decisions based on p-value, it is possible to make type-I errors or type-II errors. Therefore, we should avoid using hypothesis testing when no decision-making is needed. When making a decision, we should also state the p-value and threshold instead of discussing using significance(Motulsky, 2014).

In conclusion, to increase the reproducibility of our statistical results, we should be careful when making decisions in our analysis. We should avoid p-hacking and unnecessary statistical hypothesis testing. We also should mention details of our methods and experiments, including effect sizes and decisions that may affect reproducibility. By acknowledging and preventing misleading statistics concepts, we can increase the reproducibility of our statistical research

effectively in the future.

Word count: 460 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 proud of my problem-solving process in task2. I encountered a special issue with my seed which produces problematic observations. At first, I was very confused with the negative variance I produced. I checked my calculation and I did not find any error. Then I produced the mixed model by lmer which was a bizarre result too. So I changed the seed and found that the values were correct for other seeds. Then, I sought help from Professor. Bolton and we came up with the solution. This experience tells me the importance of troubleshooting and seeking help from others when needed.

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

The linear mixed model is very useful for future statistical analysis. It is a new concept to me and I think it is very fascinating. Before I learned about the mixed model, I will usually consider these effects as the limitation of my models such as noises and confounding variables. The introduction of random effects provides a more descriptive model and it will be useful for future statistical research. Also, I think the writing assignment is valuable. It helps me learn more about statistical ethics. Moreover, I can post my article on LinkedIn to present my writing skill and statistical knowledge.

What is something I'd do differently next time?

One thing I would do differently is to work on the assignment earlier. This week I have five assignments to work on, and I realized how crucial time management is when studying in university. When I started this assignment, I quickly realized the workload was heavier than I expected. I ended up spending two whole days working on it. Although I did not apply for an extension and finished the portfolio on time, I believe I can do better if I start early. Next time, I will plan my time on the portfolio more rigidly so that I have more time to polish my writing and present my work more nicely.