STA303/1002 Portfolio

An exploration of linear mixed models and common misconceptions in statistics

Alexander Tran

2022-02-17

Contents

List of Figures

Introduction

This portfolio was written as an assignment for the course STA303/1002: Methods of Data Analysis II. This course aims to educate students about appropriately and ethically using data to apply statistical models in order accurately interpret results and communicate these findings to a range of audiences. In this portfolio, various statistical skills, as well as writing skills are demonstrated.

In the statistical skills sample, the R programming language is used to read or simulate data, which is then used in statistical methods to achieve desired results. Examples of concepts applied include simple and mixed linear models, which are explored to understand their applications and effectiveness, as well as confidence intervals and p-values, how they are defined, and how to interpret them. Other notable skills demonstrated in this section include formulating visually pleasing and easy-to-read figures such as plots or tables, as well as how to generate reproducible examples to be shared with peers.

In the writing sample, an example of a short commentary on an article is written. Knowing how to communicate ideas to other people is an invaluable skill, and so this section aims to demonstrate writing techniques and abilities in a professional environment. The reflection section, while also demonstrating writing ability, aims more to illustrate the capacity to reflect on oneself. This section shows how I am able to understand what I have done and what it achieves for me, as well as how I can improve myself in the future.

Statistical skills sample

Task 1: Setting up libraries and seed value

```
#load tidyverse
library(tidyverse)
#set seed
last3digplus <- 100 + 089
```

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

Growing your (grandmother's) strawberry patch

```
source("grow_my_strawberries.R")
#creating data
my_patch <- grow_my_strawberries(seed = last3digplus)
#turn treatment into a factor variable with the levels ordered as follows: "No
↪  netting," "Netting," "Scarecrow."
my_patch <- my_patch %>% mutate(treatment = fct_relevel(treatment, "No netting",
↪  "Netting", "Scarecrow"))
```

Plotting the strawberry patch

```
library(ggplot2)
my_patch %>% ggplot(aes(x = patch, y = yield, fill = treatment, colour = treatment)) +
  geom_point(pch = 25) +
  scale_fill_manual(values = c("#78BC61", "#E03400", "#520048")) +
  scale_color_manual(values = c("#78BC61", "#E03400", "#520048")) +
  theme_minimal() +
  labs(caption = "Created by Alexander Tran in STA303/1002, Winter 2022")
```

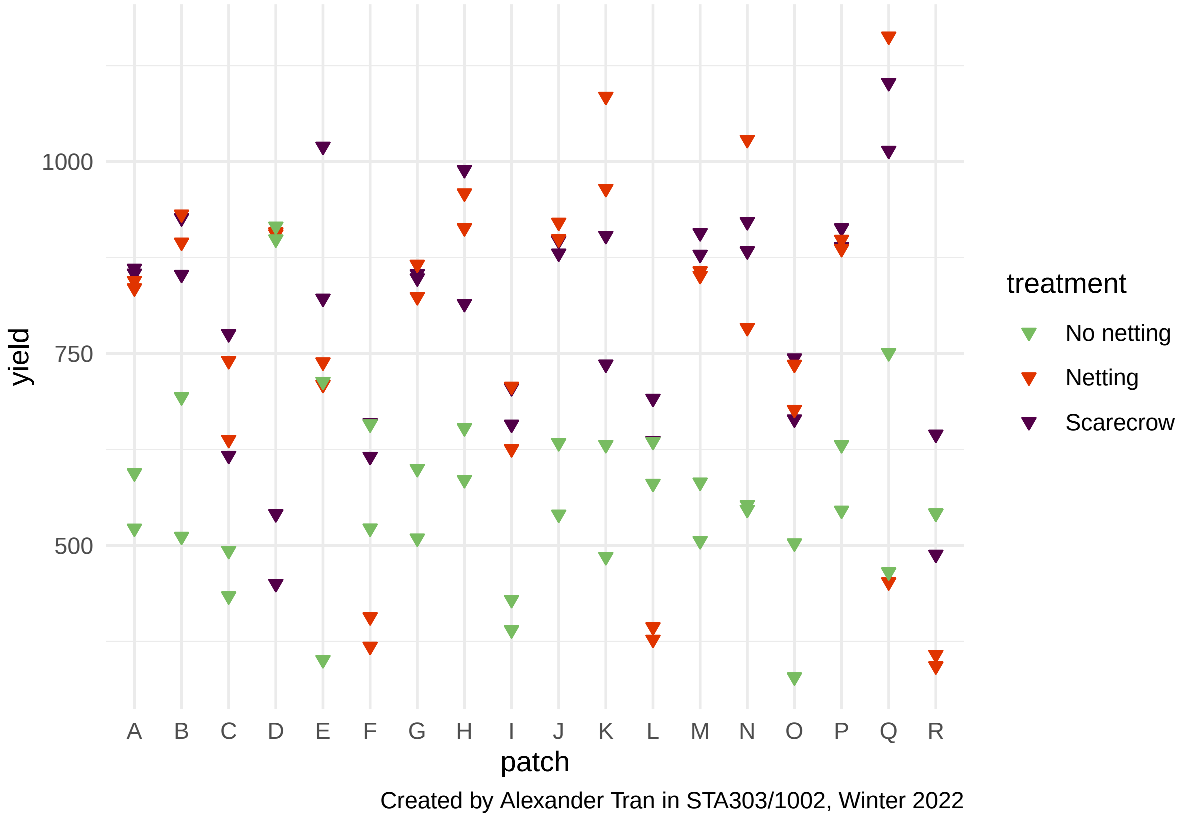

Figure 1: A Scatterplot of yield vs patch, under Various treatments

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

Model formula

$$y_{ij} = \mu + \tau_i + \rho_j + (\tau\rho)_{ij} + \epsilon_{ij}$$

where:

- y_{ij} is the amount of strawberries produced (in kilograms) by the j^{th} patch while under treatment i
- μ is the overall mean of strawberry production
- τ_i are the I fixed effects for *treatment*
- ρ_j are the random effects for patch j. $\rho_j \sim N(0, \sigma^2_\rho)$
- $(\tau\rho)_{ij}$ are the IJ interaction terms for the interaction between patch and the treatment. $(\tau\rho)_{ij}$ are random effects. $(\tau\rho)_{ij} \sim N(0, \sigma^2_{(\tau\rho)})$
- ϵ_{ij} is a random error term for the difference between the amount of strawberries produced (in kilograms) by the j^{th} patch while under treatment i from the expected amount under this model. $\epsilon_{ij} \sim N(0, \sigma^2)$

```
#Creating tibbles
agg_patch <- my_patch %>%
  group_by(patch) %>%
  summarise(yield_avg_patch = mean(yield))
agg_int <- my_patch %>%
  group_by(patch, treatment) %>%
  summarise(yield_avg_int = mean(yield), .groups = "drop")

#Creating models
int_mod <- lm(yield ~ patch * treatment, data = my_patch)
patch_mod <- lm(yield_avg_patch ~ 1, data = agg_patch)
agg_mod <- lm(yield_avg_int ~ patch + treatment, data = agg_int)

#Calculating numeric values
var_patch <- summary(patch_mod)$sigma^2 - (summary(agg_mod)$sigma^2)/3
var_int <- summary(int_mod)$sigma^2
var_ab <- summary(agg_mod)$sigma^2 - (summary(int_mod)$sigma^2)/2
```

```
#Creating a table to highlight variances and proportions
tibble(`Source of variation` = c("Patch-to-patch Variability",
                                 "Random Noise",
                                 "Variations in the Interaction Between the Patches
                                 ↪   and Treatments"),
       Variance = c(var_patch, var_int, var_ab),
       Proportion = c(round(var_patch/(var_patch + var_ab + var_int), 2),
                      round(var_int/(var_patch + var_ab + var_int),2),
                      round(var_ab/(var_patch + var_ab + var_int), 2) )) %>%
  knitr::kable(caption = "Variances and Proportions of Sources of Variation")
```

Table 1: Variances and Proportions of Sources of Variation

Source of variation	Variance	Proportion
Patch-to-patch Variability	6322.127	0.21
Random Noise	10728.181	0.36
Variations in the Interaction Between the Patches and Treatments	12857.680	0.43

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

```
#Creating models
library(lme4)
mod0 <- lm(yield ~ treatment, data = my_patch)
mod1 <- lmer(yield ~ treatment + (1|patch), data = my_patch)
mod2 <- lmer(yield ~ treatment + (1|patch) + (1|patch:treatment), data = my_patch)
#Perform a likelihood test
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 -707.24
## 2   5 -686.90  1 40.692  1.782e-10 ***
## 3   6 -678.84  1 16.106  5.990e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: yield ~ treatment + (1 | patch) + (1 | patch:treatment)
##    Data: my_patch
##
## REML criterion at convergence: 1357.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.4734 -0.4677  0.1105  0.3398  3.3897
##
## Random effects:
##  Groups          Name        Variance Std.Dev.
##  patch:treatment (Intercept) 12858    113.39
##  patch           (Intercept)  6322     79.51
##  Residual                    10728    103.58
## Number of obs: 108, groups:  patch:treatment, 54; patch, 18
##
## Fixed effects:
##                     Estimate Std. Error t value
## (Intercept)           568.79      36.93  15.403
## treatmentNetting      194.86      45.00   4.331
## treatmentScarecrow    227.33      45.00   5.052
##
## Correlation of Fixed Effects:
##             (Intr) trtmnN
## trtmntNttng -0.609
## trtmntScrcr -0.609  0.500
```

In mod0, we are using ML because are fitting a simple linear model that has only one fixed effect. In mod1, we are using REML because we are fitting a linear mixed model with *patch* as a fixed effect and *treatment* as a random effect. In mod2, we are using REML because we are fitting a linear mixed model with *patch* as a fixed effect, *treatment* as a random effect, and their interaction as a random effect.

In the likelihood ratio test, we are using REML because we have the same fixed effects and are only comparing nested random effects.

Justification and interpretation

mod2 is the most appropriate as the final model. This is the model that is fitted with *patch* as a fixed effect, *treatment* as a random effect, and their interaction as a random effect. This model was chosen because after conducting likelihood ratio tests, it can be seen that we have significant evidence against the hypothesis that the simpler model performs as good as the more complex model. Thus, we reject this null hypothesis at most significance levels when we compare mod1 to mod0, and mod2 to mod1. Choosing the more complex model both times, we end up with mod2.

Interpreting the fixed effect coefficients tell us that when there is no treatment, the mean strawberry yield is 568.79 kilograms. Similarly, when *netting* and *scarecrow* are the given treatments, the means yields are 194.86 kg and 227.33 kg, respectively. Looking at the variance that is explained by each source of variance, we see that 21% of the total variance is explained by patch-to-patch variability and 43% is explained by the variations in the interactions between the patches and treatments.

Task 3a: Building a confidence interval interpreter

```r
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
    # the spacing is a little weird looking so that it prints nicely in your pdf
    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 representing the lower bound of
    ↪  the CI.")
  } else if(!is.numeric(upper)) {
    # produce a warning if upper isn't numeric
    warning("Warning: upper should be a numeric value representing the upper bound of
    ↪  the CI.")
  } 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 between 0 and 100, inclusive.")
  } else{
    # print interpretation
  str_c("We are ", ci_level,
        "% confident that the true value of ", stat,
        " is 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, 95, 99)
```

CI function test 1: We are 99% confident that the true value of mean number of shoes owned by students is between 10 and 20.

CI function test 2: Warning: ci_level should be between 0 and 100, inclusive.

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

Task 3b: Building a p value interpreter

```
interpret_pval <- function(pval, nullhyp){
  if(!is.character(nullhyp)) {
    warning("
            Warning: nullhyp should be a character string that describes the
            null hypothesis being tested.")
  } else if(!is.numeric(pval)) {
    warning("Warning: pval should be a numeric value representing your p value.")
  }  else if(pval > 1) {
    warning("
            Warning: pval should not be greater than 1.")
  } else if(pval < 0){
    warning("
            pval should not be less than 0.")
  } else if(pval >= 0.1){
    str_c("The p value is ", round(pval, 3),
                ". There is no evidence against the hypothesis that ", nullhyp)
  } else if((pval >= 0.05) & (pval < 0.1)){
    str_c("The p value is ", round(pval, 3),
                ". There is weak evidence against the hypothesis that ", nullhyp)
  } else if((pval >= 0.01) & (pval < 0.05)){
    str_c("The p value is ", round(pval, 3),
                ". There is moderate evidence against the hypothesis that ", nullhyp)
  } else if((pval >= 0.001) & (pval < 0.01)){
    str_c("The p value is ", round(pval, 3),
                ". There is strong evidence against the hypothesis that ", nullhyp)
  } else if(pval < 0.001){
    str_c("The p value is <.001. There is very strong evidence against the 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 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 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 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 being tested.

Task 3c: User instructions and disclaimer

Instructions

interpret_ci outputs an interpretation of a given confidence interval, and takes *lower*, *upper*, $ci_levelevel$, and *stat* as arguments. *lower* and *upper* represent the lower bound and upper bounds of your confidence interval. $ci_levelevel$ is the confidence level at which your interval was calculated at. *stat* is the population parameter in question. A population parameter is a statistic that describes an entire group or population of things, as opposed to describing just a sample. For example, the mean lifespan of all males in Toronto is a population parameter, where the population is all males in Toronto.

interpret_pval outputs an interpretation of a given p value, and takes *pval* and *nullhyp* as arguments. *pval* is the calculated p value from your hypothesis testing. *nullhyp* is the null hypothesis that hypothesis testing was done under. Often, a null hypothesis follows the lines of "there is no effect on this variable", or "there is no difference under different conditions". For example, if you are investigating whether hummingbirds prefer apple juice or orange juice, the null hypothesis is “Hummingbirds have no preference. They prefer them the same.”.

Note that there should be some caution in interpreting frequentist confidence intervals. The correct interpretation is the one that is outputted. It is not equivalent to other interpretations such as “there is a 95% chance that the confidence interval contains the true population parameter.”, which is incorrect because the population parameter is a fixed unknown value that is either inside or outside the interval with 100% certainty. What the outputted interpretation actually says is that "the method used constructs an interval that captures the population parameter 95% of the time.

Disclaimer

When using the p value interpreter, there are several things that should be noted to avoid potential misinterpretations or incorrect results. The interpreter does not state about whether or not the null hypothesis should be rejected.

The interpreter talks about how strong the evidence is against the null hypothesis, but it is ultimately up to the user to decide what significance cutoff to use and how this information is incorporated when deciding whether or not to reject the null hypothesis after testing. A p value also does not provide information about how large an effect is, and should not be interpreted as such.

Another thing to note is that when values of pval that are close to significance level cutoffs are passed to interpret_pval, some interesting results may occur that the user should be wary about. For example, if a value of 0.0499999 is inputted, the output will read: “The p value is 0.05. There is moderate evidence against the hypothesis that. . . ”, and if 0.050001 was inputted the output reads: “The p value is 0.05. There is weak evidence against the hypothesis that. . . ”. Despite appearing to be the same p value, they are interpreted to indicate different levels of evidence against the null. This is simply due to rounding when printing the output string, and does not affect the actual decision for the strength of the evidence.

Task 4: Creating a reproducible example (reprex)

A reprex is a reproducible example, often generated and used to receive assistance with some issue by allowing others to easily reproduce the same errors or behaviors. In order to produce a reprex, it is important to provide the necessary packages if you are using functions from their libraries. The given code in this task made use of functions such as summarize() and glimpse(), but did not include a call to load tidyverse, so that had to be included in order to create a reprex.

Furthermore, if functions that generate randomly are involved, it may be useful to set a seed so that the same data can be replicated.

```
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

```r
set.seed(last3digplus)

#Generating normally distributed data sets
sim1 <- tibble(group = rep(1:1000, each = 100), val = rnorm(n = 100000, mean = 0, sd =
↪  1))
sim2 <- tibble(group = rep(1:1000, each = 100), val = rnorm(n = 100000, mean = 0.2, sd
↪  = 1))
sim3 <- tibble(group = rep(1:1000, each = 100), val = rnorm(n = 100000, mean = 1, sd =
↪  1))

#Stacking
all_sim <- bind_rows(sim1, sim2, sim3, .id = "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)"))

#Changing sum from char to numeric in all_sim so that it can be joined
all_sim <- all_sim %>% mutate(sim = as.numeric(sim))
all_sim <- left_join(all_sim, sim_description, by = "sim")
```

```r
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 Alexander Tran in STA303/1002, Winter 2022")
```

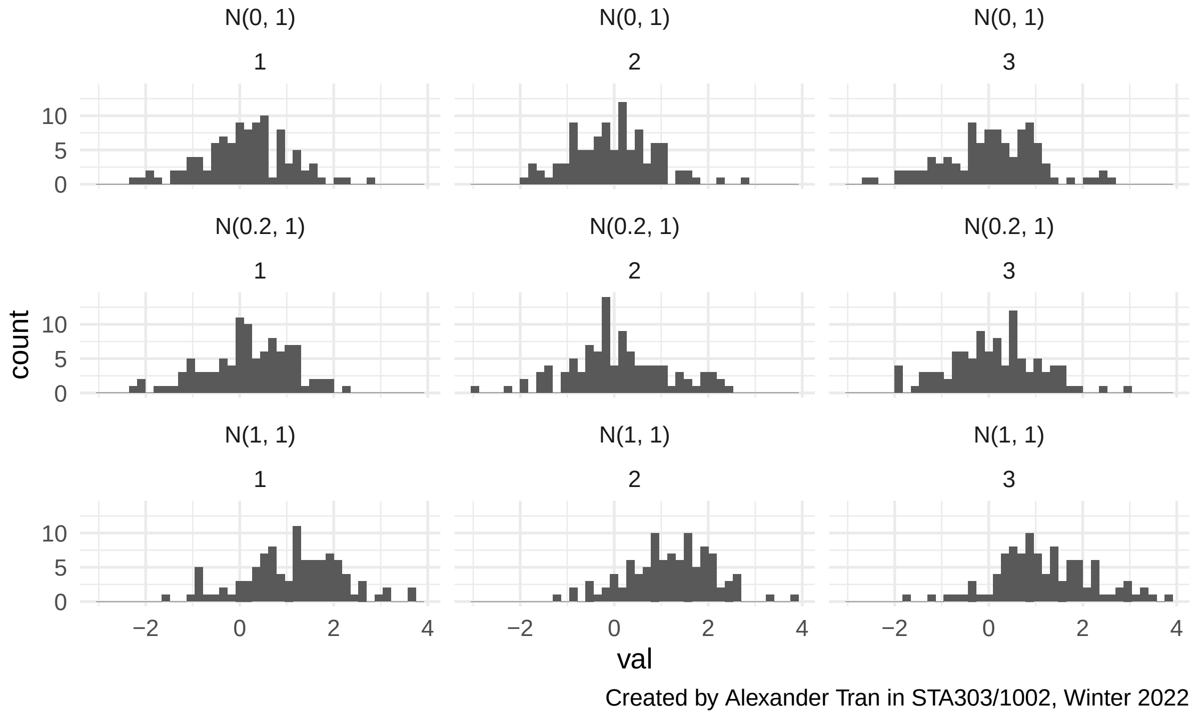

Figure 2: Histograms of Normal Distributions of Varying Means

Calculating *p* values

```
#As instructed in instructions
pvals <- all_sim %>%
  group_by(desc, group) %>%
  summarize(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 Alexander Tran in STA303/1002, Winter 2022")
```

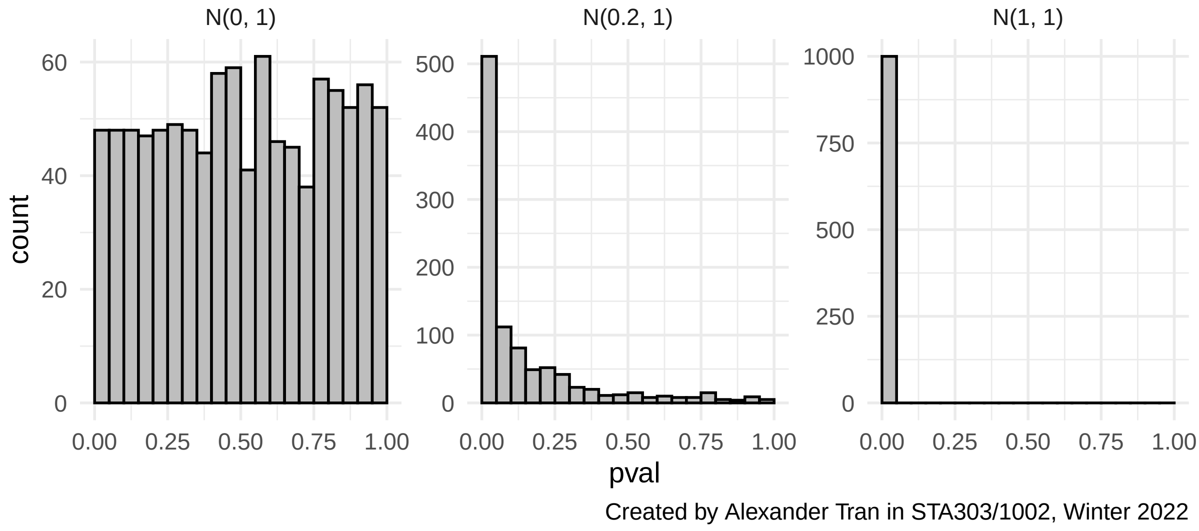

Figure 3: Histograms of p values for Normal Distributions of Varying Means

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 Alexander Tran in STA303/1002, Winter 2022")
```

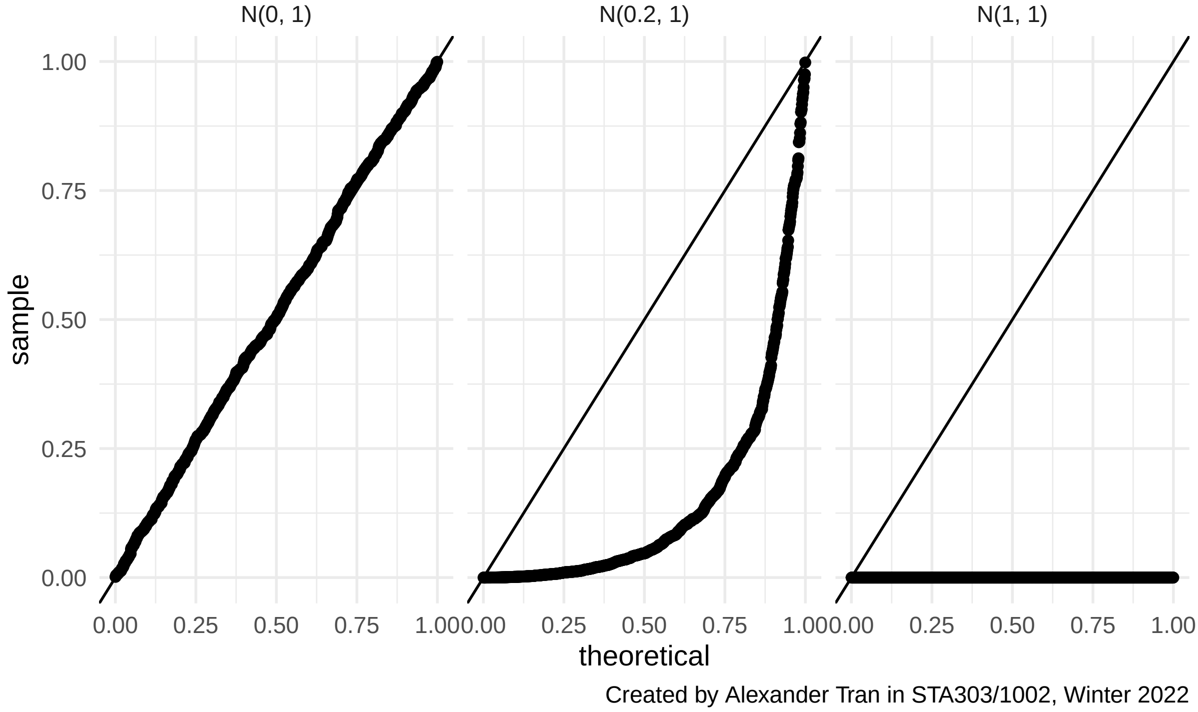

Figure 4: Simulated Quantiles vs Expected Quantiles for p-values

Conclusion and summary

In this task, we simulated 1000 sets of 100 normally distributed data points three times; one for each of three different means: 0, 0.2, and 1. Then, for all 3000 sets, we performed one sample, 2-sided t-tests, where the null hypothesis is the the population mean is 0. Finally, for each of our three normal distributions, we compared the distributions of each set's p-values to that of a $U(0,1)$ by plotting their quantiles against each other.

The results show that the p values of the sets drawn from $N(0,1)$ closely follow $U(0,1)$. This makes sense, as we ran t-tests under the assumption that the null hypothesis was true, that is to say the test statistic t followed a $N(0,1)$. The p-value is the probability of observing a t this "extreme" under the null hypothesis, so it makes sense that the p-value is uniformly distributed between 0 and 1.

This is precisely what was being asked in question 16 of the pre-knowledge check, where a simulation was proposed. Since approximately 10% of values lie between any two values with difference of 0.1 in $U(0,1)$, the correct answer is "approximately 10% of the p-values will be between 0.9 and 1". Furthermore, we can see that the alternative answers are incorrect by referring to the histograms and q-q plots above.

Writing sample

Reproducibility is irrefutably an integral part of science. It is evidence of well-designed study, and gives credibility to its findings. Recently, I read "Common misconceptions about data analysis and statistics" Motulsky (2014). This article shed light on many common mistakes that occur during data analysis, and highlighted how these mistakes result in a lack of reproducibility by a large proportion of published findings. Motulsky (2014) states that oftentimes, these mistakes arose from having a poor understanding of statistical concepts. Having some background knowledge in statistics, I found it quite surprising that some of these mistakes were being made in professional writing. On the other hand, the article mentions misconceptions that I myself held, that I did not know were mistakes that I have made.

The first of these mistakes is that "P-hacking is OK". P-hacking is something that I have been taught time and time again not to do, so it was a little shocking to see that this was being done by professionals. Though to be fair, it may be possible that they simply were not aware that what they were doing was p-hacking, or perhaps due to external forces, they felt pressured to produce results and out of desperation used p-hacking. As suggested in the article, it can be very easy to fall into this trap, especially when not getting the desired results.

Another mistake that I would like to talk about is how details are not fully reported in the methods sections of papers. I find this to be rather inexcusable. If the statistical methods being used are to analyze data gathered from experimental methods used in the field, one may naturally think of the statistical methods as just a supplement, and that they are not as important. Motulsky argues that one is just as important as the other, especially when it comes to reproducibility. I agree with this statement, and believe that all details regarding the statistical methods being used, including the resulting P-value and whether the data was transformed should be reported in full.

A misconception that the article pointed out to me was that "statistical hypothesis testing and reports of 'statistical significance' are necessary in experimental research". I was under the impression that this was true, that statistical tests were there to make decisions in studies. However, Motulsky makes several arguments for avoiding the use of statistical hypothesis testing, and supports his arguments well enough that they have changed my mind, and is something that I will keep in mind in the future.

In reading Motulsky's commentary on statistical misconceptions and reproducibility, I have learned quite a bit about the role that statistics and its methods play in other fields of study. Seeing the oversights that others make keeps me mindful about my own habits in analysis, so that I do not repeat the very same mistakes. This article is a lesson to me, and I will be sure to remember it as I move forward in my career in statistics. **Word count:** 497 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?

First of all, I am proud that I was able to complete it to my standard of work. I believe that the results that I have produced in this portfolio reflect my efforts. The bulk of the time spent on this portfolio was in the statistical skills and writing samples, so naturally I am more proud of the work that was created in those sections.

In the statistical skills sample, I sometimes had trouble figuring out what was being requested or how to execute the instructions. Looking back, I don't regret struggling at all. In fact, I believe that it was a good thing that I had trouble, so that I could train myself to be capable of doing things on my own, and learn things along the way, improving myself. That aside, the tasks in this section taught me plenty of useful things.

The writing sample gave me a chance to create a piece of writing that I don't often get the chance to do. Writing is a very useful skill in statistics, so I found this to be a valuable opportunity to exercise my writing ability.

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

I am planning to continue studying statistics, and go into its related fields. This means that what I've learned and demonstrated in this portfolio will without a doubt be useful in the future.

In the statistical skills sample, I learned many useful methods and how to present their results effectively. I learned practical applications of simple and mixed linear models, how to produce reproducible examples, and how to simulate random data. The ability to create aesthetically pleasing and easy to read figures is invaluable, so learning how to do this was great. The other tasks also refreshed previously learned concepts such as how we define and interpret confidence intervals and p values.

Knowing how to write is also a skill that is arguably as valuable as knowing how to use statistical methods. Ideas are nothing if they cannot be communicated. In a field like statistics, it is very relevant to know how to present your results so that your peers can understand, so I can definitely apply what I've learned here to future work, where I will more often than not be working alongside other people.

What is something I'd do differently next time?

First of all, I would begin this assignment earlier to reduce the amount of stress it has given me. While this would be beneficial to me, it also gives me the freedom to help other people (on Piazza). I found that a lot of people had similar questions, and I may have been able to provide some guidance to them.

That aside, I would maybe begin with the writing sample next time. I found this particular prompt to be fairly useful, as well as relevant to me. While the order in which sections are done are mostly inconsequential, this article taught me some useful ideas about statistical methods, especially regarding p-values. This information would have been useful when I worked on the statistical skills sample later, mostly when working on the task involving interpreting p-values.