STA303/1002 Portfolio

An exploration of linear mixed models and common misconceptions in statistics

Mark JP Sanchez

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 10

Task 3b: Building a p value interpreter 11

Task 3c: User instructions and disclaimer 12

Task 4: Creating a reproducible example (reprex) 13

Task 5: Simulating p-values 15

Writing sample **20**

References 20

Reflection **22**

List of Figures

1 Strawberry Plot by Yield 5

2 Histogram of first 3 groups by distribution 16

3 Distribution of p-values of T-Test (mu = 0) 17

4 QQ Plot of P-values under null hypothesis 18

Introduction

This portfolio was made for the University of Toronto STA302 (Methods of Data Analysis II) class. Within this portfolio shows examples of both my soft and technical statistical skills.

For soft skills, here you can see some data visualization, writing excerpts and explanations of statistical concepts. There are a multitude of graphs generated with ggplot as well as explanations of confidence intervals and p-values. There is also even an exploration of the concept of p-values and what they mean through the use of multiple simulated values taken from different distributions. There is also a writing sample which is a piece written with information taken from an article dealing with misconceptions of data analysis.

For technical skills, there are a variety of skills shown here. From understanding the more theoretical concepts around things such as confidence and p-values to model creation and interpretation. Within this portfolio, I attempt to fit a linear mixed model to data taken from a simulated strawberry patch. Here, I designate fixed and random effects as well as interaction terms. With the model and the data, I even derive the variance of the model as well as variance of each term.

Looking through this portfolio I hope you get a good understanding of my capabilities as well as my understanding of statistical concepts

Statistical skills sample

Task 1: Setting up libraries and seed value

```
library(tidyverse)

last3digplus <- 100 + 881
```

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

```
# Loading and reading in data
source("grow_my_strawberries.R")
my_patch <- grow_my_strawberries(seed = last3digplus)

# Ordering Treatment Levels
my_patch <- my_patch %>%
  mutate(treatment = fct_relevel(treatment, c("No netting", "Netting", "Scarecrow")))
```

Plotting the strawberry patch

```
my_patch %>% ggplot(aes(x = patch, y = yield, color = treatment, fill = 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 Mark JP Sanchez in STA303/1002, Winter 2022")
```

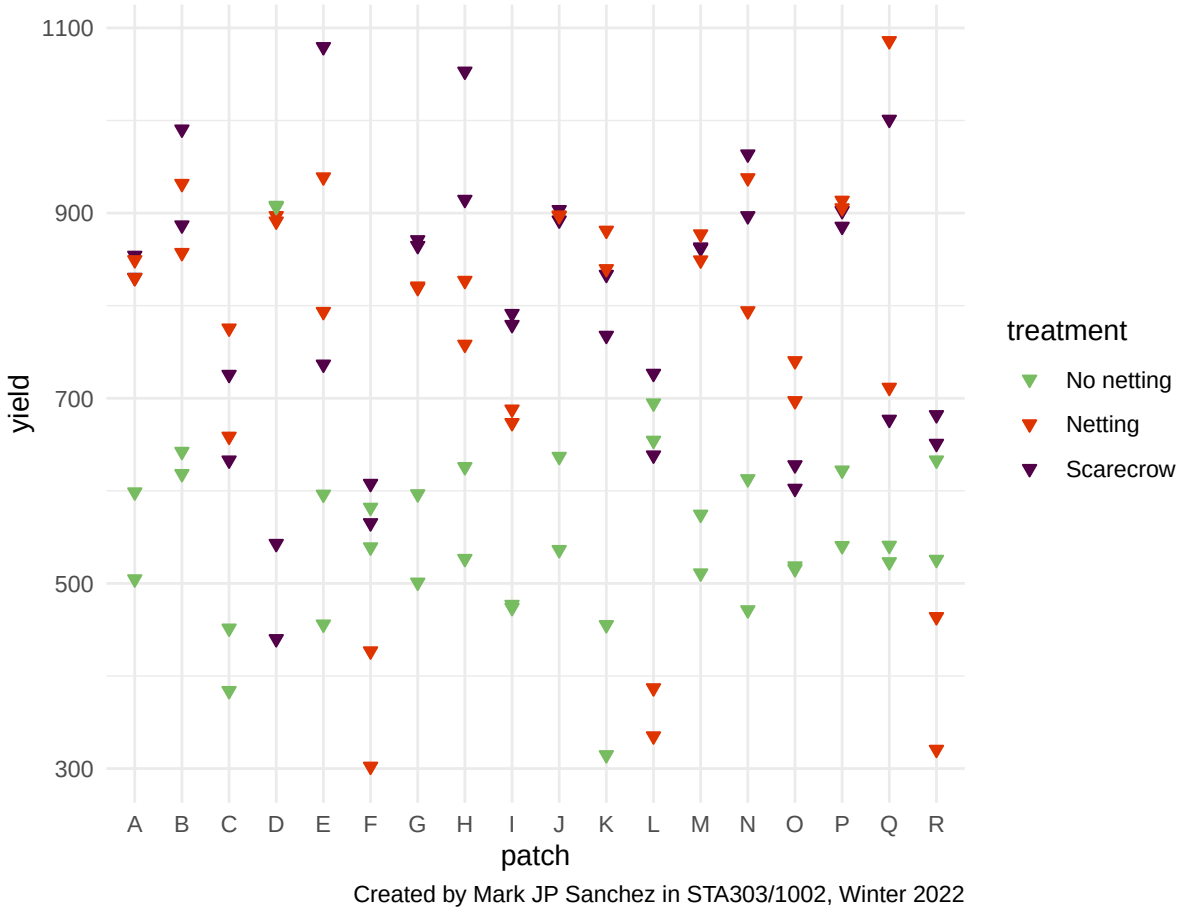

Figure 1: Strawberry Plot by Yield

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

Model formula

$$y_{ij} = \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ij}$$

where:

- y_{ij} is the yield using treatment i with patch j, and
- α_i are the fixed effects of treatment i, and
- $\beta_j \sim N(0, \sigma^2_\beta)$ are the random effects for patch j, and
- $(\alpha\beta)_{ij} \sim N(0, \sigma^2_{\alpha\beta})$ are the random interaction effects between the treatment and patch, and
- $\epsilon_{ij} \sim N(0, \sigma^2)$ are the error terms

```r
# Aggregated Model (by patch)
agg_patch <- my_patch %>%
  group_by(patch) %>%
  summarize(yield_avg_patch = mean(yield))

# Aggregated Model (by patch + treatment)
agg_int <- my_patch %>%
  group_by(patch, treatment) %>%
  summarize(yield_avg_int = mean(yield), .groups = "drop")

# Interaction Model
int_mod <- lm(yield ~ patch * treatment, data = my_patch)
# Intercept Only Model
patch_mod <- lm(yield_avg_patch ~ 1, data = agg_patch)
# Aggregate Model
agg_mod <- lm(yield_avg_int ~ patch + treatment, data = agg_int)

# Calculating error variance
var_int <- summary(int_mod)$sigma^2
# Calculating Interaction Variance
num_of_com <- nlevels(my_patch$treatment) * length(unique(my_patch$patch))
K <- nrow(my_patch) / num_of_com
var_ab <- summary(agg_mod)$sigma^2 - (var_int / K)
# Calculating patch to patch variance
num_of_patch <- length(unique(my_patch$patch))
var_patch <- summary(patch_mod)$sigma^2 - (summary(agg_mod)$sigma^2)/num_of_patch
```

```r
tibble(`Source of variation` = c("residuals",
                                 "patch:treatment",
                                 "patch"),
       Variance = c(var_int, var_ab, var_patch),
       Proportion = c(round(var_int / (var_int + var_ab + var_patch), 2),
                      round(var_ab / (var_int + var_ab + var_patch), 2),
                      round(var_patch / (var_int + var_ab + var_patch), 2) )) %>%
  knitr::kable(caption = "Variation of Random Effects and Proportions")
```

Table 1: Variation of Random Effects and Proportions

Source of variation	Variance	Proportion
residuals	6212.957	0.20
patch:treatment	16323.947	0.53
patch	8451.728	0.27

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

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
# Defining Models
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)
```

Here we are using restricted maximum likelihood (REML) because all of our models have the same fixed effects and we wanted to perform likelihood ratio tests on them. If the fixed effects were different in the models then we would have opted to use maximum likelihood. We also use REML as we are trying to estimate our model parameters.

Justification and interpretation

```r
# Performing tests
lmtest::lrtest(mod0, mod1)
```

```
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): original model was of
## class "lm", updated model is of class "lmerMod"

## Likelihood ratio test
##
## Model 1: yield ~ treatment
## Model 2: yield ~ treatment + (1 | patch)
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   4 -698.64
## 2   5 -680.57  1 36.152  1.825e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
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 -680.57
## 2   6 -662.95  1 35.227  2.934e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Checking final model
summary(mod2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: yield ~ treatment + (1 | patch) + (1 | patch:treatment)
##    Data: my_patch
##
## REML criterion at convergence: 1325.9
```

```
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.12253 -0.42331  0.04962  0.40291  2.62752
##
## Random effects:
##  Groups          Name        Variance Std.Dev.
##  patch:treatment (Intercept) 16324    127.77
##  patch           (Intercept)  3054     55.27
##  Residual                     6213     78.82
## Number of obs: 108, groups:  patch:treatment, 54; patch, 18
##
## Fixed effects:
##                     Estimate Std. Error t value
## (Intercept)           563.05      35.34  15.931
## treatmentNetting      194.54      46.46   4.187
## treatmentScarecrow    229.83      46.46   4.946
##
## Correlation of Fixed Effects:
##             (Intr) trtmnN
## trtmntNttng -0.657
## trtmntScrcr -0.657  0.500
```

With our 3 models we perform likelihood ratio tests to see which model we will use for our analysis. The log likelihood values for our model with only the treatment variable (mod0), our model with treatment and patch (mod1) and our model with treatment, patch and their interaction (mod2) are -699, -681 and -662.95 respectively. Our lrtest show that these results and difference are statistically significant as the lrtest between mod0-mod1 and mod1-mod2 had p-values that were both very small as they were less than 0.05. From this we can see that we should choose model with the largest log likelihood value which is mod2.

Looking at mod2 we can take a look at our treatment variable. Here we can see that our intercept is 563.05 which correlates to our no netting treatment. Here we can see that taking into account our other variables, on average we produce 563.05kgs of strawberries. Using netting produces on average 195 more kgs of strawberries while using a scarecrow produces on average 230 kgs of strawberries.

Task 3a: Building a confidence interval interpreter

```
interpret_ci <- function(lower, upper, ci_level, stat){
  if(!is.character(stat)) {
    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 number.")
  } else if(!is.numeric(upper)) {
    # produce a warning if upper isn't numeric
    warning("Warning: upper should be a number")
  } 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 number between 0 and a 100 exclusive")
  } else{
    # print interpretation
    # this is the main skill I want to see, writing a good CI interpretation.
  str_c("We are ", ci_level,
        "% confident that the ", stat,
        " is in 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 mean number of shoes owned by students is in between 10 and 20

CI function test 2: Warning: ci_level should be a number between 0 and a 100 exclusive

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 string denoting the null hypothesis")
  } else if(!is.numeric(pval)) {
    warning("Warning: pval should be a number.")
  }  else if(pval > 1) {
    warning("
            Warning: pval should not be greater than 1.")
  } else if(pval < 0){
    warning("
            Warning: pval should not be less than 0.")
  } else if(pval > 0.1){
    str_c("The p value is ", round(pval, 5),
                ", we fail to reject the hyothesis that ", nullhyp)
  } else if(pval > 0.05){
    str_c("The p value is ", round(pval, 5),
                ", we have weak evidence against the idea that ", nullhyp, ".")
  } else if(pval > 0.01){
    str_c("The p value is ", round(pval, 5),
                ", we have some evidence against the idea that ", nullhyp, ".")
  } else if(pval > 0.001){
    str_c("The p value is ", round(pval, 5),
                ", we have strong evidence against the idea that ", nullhyp, ".")
  } else {
    str_c("The p value is ", round(pval, 5),
                ", we have very strong evidence against the idea 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 0, we have very strong evidence against the idea 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, we have some evidence against the idea 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, we have weak evidence against the idea 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 string denoting the null hypothesis

Task 3c: User instructions and disclaimer

Instructions

To utilize the interpret_ci function, you need to fill in its four parameters. The four parameters are lower, upper, ci_level and stat which corresponds to the lower bound, upper bound, confidence level and statistic measuring, respectively. Once you fill in the parameters, the function will spit out an interpretation of the confidence level.

There are some caveats and common pitfalls for confidence interval interpretations you should be aware of. For example, the confidence interval does not specify the likelihood of capturing the population parameter, the population parameter being the true value your statistic is trying to estimate. The confidence level is a measurement of our confidence in the method. For example, a 95% confidence level does not denote a 95% chance of capturing our population parameter but instead that if we create confidence interval for x different samples, 95% of them will capture our population parameter.

To utilize the interpret_pval function, you need to fill in its two parameters. The two parameters are pval and nullhyp which correspond to the p-value and null hypothesis respectively. The function will then spit out an interpretation of the p-value based on the strength of the results. Here the null hypothesis is an assumption we are performing our test against. Our p-value is the likelihood that our result is true assuming our null hypothesis is true.

Disclaimer

Be wary of trusting the p-value interpreters too heavily. First of all you should make sure you are actually performing the right test correctly. In other words, make sure that the assumptions of the test you are performing have been satisfied. For example, in a one sample t-test, you need to ensure that your data is independent and normally distributed. Failure to meet these assumptions causes the p-value you get from the test to be meaningless. Also be aware that even if assumptions are met there is still the possibility of making a Type 1. A type 1 error is where you incorrectly reject the null hypothesis. Another common misconceptions is that failing to reject the null hypothesis is not equivalent to accepting it.

Task 4: Creating a reproducible example (reprex)

A reproducible example (reprex) is an example of code you can show to someone that they can easily reproduce. Within a reprex, you need to show the result as well as the code you used to get that result. When creating a reprex you need to ensure that a user can easily copy and paste your code and with very minimal alterations, achieve the same results as you. This includes things as ensuring you have given them the data as well as all the steps you took to reach your final output. For example, a person can not recreate your results if you don't give them the data set or, if you are simulating the data, the random seed you used to get your values. You know your reprex is viable if you can run the code on a fresh new set up. You should also list or show any unknown or unique packages that you might be using.

```
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(last3digplus)

# Generating sims
sim1 <- tibble(group = rep(1:1000, each = 100),
               val = rnorm(100000))
sim2 <- tibble(group = rep(1:1000, each = 100),
               val = rnorm(100000, mean = 0.2))
sim3 <- tibble(group = rep(1:1000, each = 100),
               val = rnorm(100000, mean = 1))

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

# Adding desc
sim_description <- tibble(sim = 1:4,
                          desc = c("N(0, 1)",
                                   "N(0.2, 1)",
                                   "N(1, 1)",
                                   "Pois(5)"))
all_sim <- merge(all_sim, sim_description, by="sim", all.x = T, all.y = F)
```

```
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 Mark JP Sanchez in STA303/1002, Winter 2022")
```

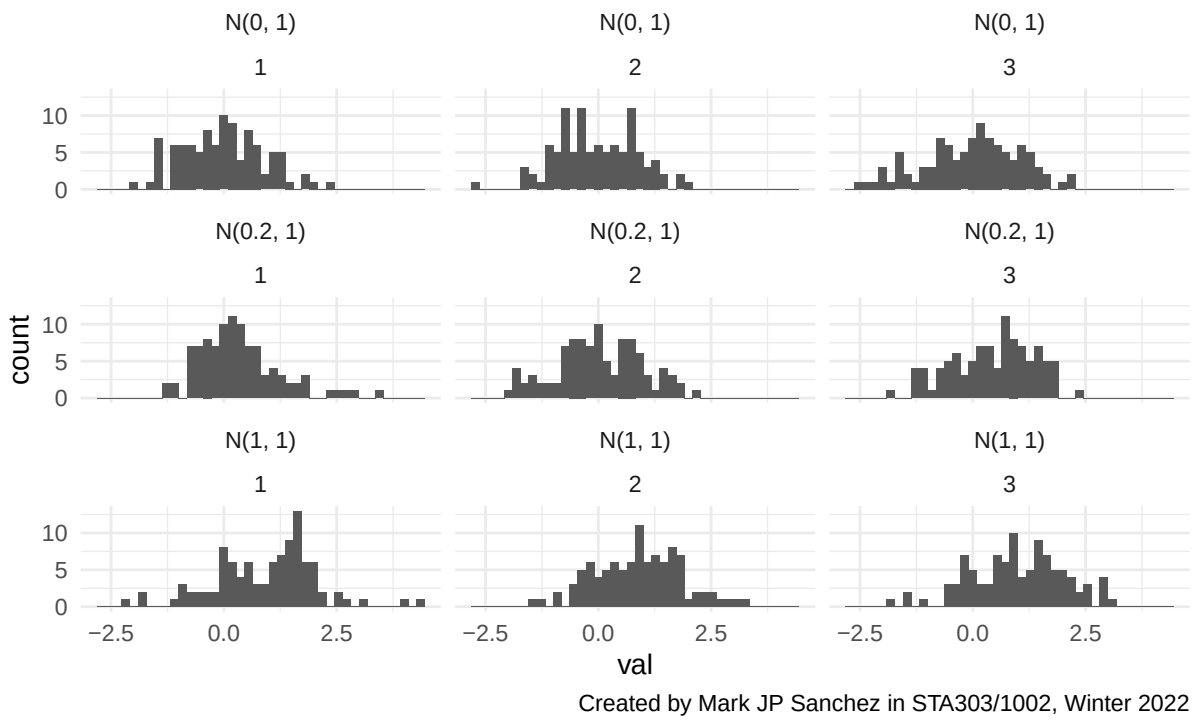

Figure 2: Histogram of first 3 groups by distribution

Calculating *p* values

```
pvals <- all_sim %>%
  group_by(desc, group) %>%
  summarise(pval = t.test(val, mu = 0)$p.value, .groups = "drop")
```

```
pvals %>%
  ggplot(aes(x = pval)) +
  geom_histogram() +
  theme_minimal() +
  facet_wrap(~desc, scales = "free_y") +
  labs(caption = "Created by Mark JP Sanchez in STA303/1002, Winter 2022")
```

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

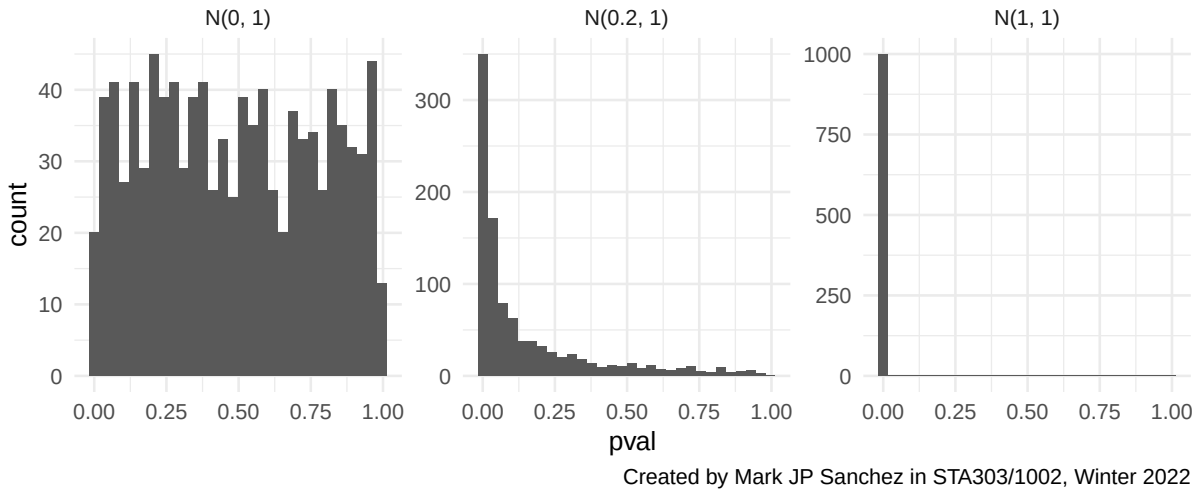

Figure 3: Distribution of p-values of T-Test (mu = 0)

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 Mark JP Sanchez in STA303/1002, Winter 2022")
```

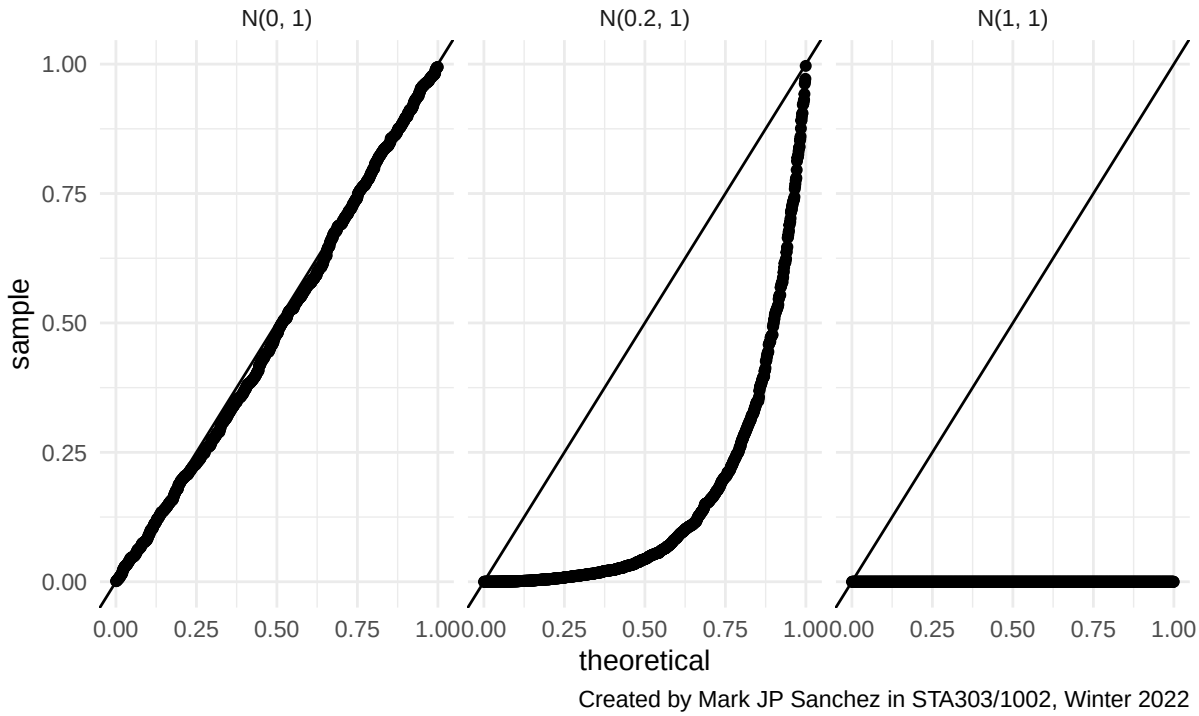

Figure 4: QQ Plot of P-values under null hypothesis

Conclusion and summary

By definition, a p-value shows us the likelihood of see a statistic as extreme as our statistic assuming the null hypothesis is true. Here our null hypothesis is that the mean of the distribution of our sample group is 0. So a p-value of 0.9 implies that if the null hypothesis were true, then performing the same experiment x number of times would imply that 90% of those statistics would be at most extreme as our or, in other words, 90% of the would have a p-value less than 0.9.

This definition can be seen in our graphs. In our first QQ plot, the N(0,1) one, we can see that our theoretical and sample quantile have an identity relationship with each other. This makes sense because our sample is taken from the null hypothesis. For example, here the theoretical 90% quantile is the same as our sample 90% quantile. We can see that this relationship disappears as our mean deviates from 0. Looking at the N(0.2,1) graph, we can see that each theoretical quantile covers less and less of our sample distribution or in other words, we are seeing a lot more allegedly extreme values than we are supposed to. This is the most evident in our final graph which comes from a N(1,1) distribution. Here we can see that all the values from our theoretical quantile are less extreme than any of the values from our sample statistic. From this

we can see that higher p-values show evidence against null hypothesis.

Writing sample

One of the coolest things about studying statistics is thinking about all the studies you can perform with this new toolset. There are many caveats and pitfalls that will lessen the impact of one's findings either intentionally or unintentionally either through some desire to see an outcome or misconceptions.

P-hacking is a very famous and problematic practice that is used in some research studies. There are many forms of p-hacking but the general gist of it is that you are trying to lower a large p-value. Once you see a p-value, you should accept the value and not attempt to alter the conditions to try and increase it. To minimize this effect authors can state if the sample size was chosen in advance and any form of p-hacking should be informed to the reader and denoted as preliminary (Motulsky, 2014).

Aside from just p-hacking, there is also the problem of relying on the interpretation of the p-value too heavily. A p-value of 0.05 does not imply that the odds of making a type 1 error is 5%. For example, if you suspect the null hypothesis is true only 10% of the time, then using a hypothesis test with a power of 80%, a 5% significance level cut off would give us a potentially 36% chance of making a type 1 error. This result can be seen in this calculation. Let's say you do 1000 experiments on this 90% true null hypothesis. So, 100 of these experiments will have a sample such that the null hypothesis is false and due to our power of 80%, we can see that 80 of these experiments will correctly have a p-value less than 0.05. With our 900 leftover experiments where the null hypothesis is true, 45 of these will have a p-value less than 0.05. Which means 125 of these experiments have a p-value less than 0.05 but 45/125 or 36% of these experiments are false positives (Motulsky, 2014).

Lastly, there is a lot of misconception about standard errors of means. The standard error of a mean is different from the standard error of the distribution of the sample. The distribution of the mean and the distribution of the sample are 2 different distributions. The author should be aware of this and should notify the reader as well especially if they are displaying them on a graph.

The purpose of research is to know more about the subject. Results are exciting but you should always be cautious and aware of the pitfalls of your methods so you need to be aware of these pitfalls. Since at the end of the day, there is no point to publishing research if it isn't true.

Word count: 453 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?

Within this portfolio, I am proud of the strawberry patch linear mixed model experiment. The strawberry patch linear mixed model section was the most confusing concept intensive part of this portfolio for me. That section required understanding of random variables, regression, models and hypothesis testing. It served as a great review for me as well as show what everything I was learning was culminating to. Most of my statistics classes before this class covered theoretical topic while showing only minimal applications. This section here showed the actual use case of all of those theoretical equations. The section also showed how useful knowing the derivations and theoretical properties of the model as some of the applications, such as calculating the different variances, aren't pre-built into the R model. I am proud that I actually got to work on and finish this section.

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

I can apply what I learned from this assignment to future personal and industry projects that I may work on. Things such as creating and interpreting linear mixed models seem very powerful. A lot of the time you just want to figure out how a certain response variable is affected by a certain independent variable. You are aware that other factors may be affecting the response variable but understanding those isn't your goal. Linear mixed models lets you really dig deep into understanding how a certain variable affects the response variable. In our strawberry patch example, we used random variables to take care of the variables that we knew affected the yield but weren't important to us. By creating a random variable for the patch and the patch-treatment interaction, we can feel more confident that the treatment coefficient represents the relationship between yield and treatment type.

What is something I'd do differently next time?

Next time, I would perform a larger exploratory data analysis on the strawberry plot. Here we kind of just fitted the linear mixed models because we were told we were able to. If I were to ever revisit this project in the future I would make sure that making a linear mixed model is actually the right choice. I would check the distribution of each variable and ensure that all assumptions are met. I would also make sure that the treatment term is actually statistically significant as we kind of assumed that it was actually affecting the data. It is definitely possible that we are overfitting our models to our data by including some of these terms.