STA303/1002 Portfolio

An exploration of linear mixed models and common misconceptions in statistics

Yu-Chun Chien

2022-02-17

Contents

List of Figures

Introduction

This is the portfolio assignment for the method of data analysis course. In the course, we are taught the ability to wrangle and explore datasets, create appropriate data visualizations, write R code for data analysis and visualization, understand the assumption of different statistical models, and interpret the results of the analysis and communicate with different audiences. In this assignment, I have the opportunity to demonstrate my data wrangling, analysis, and visualization skills, programming skills using R, as well as my communication skills.

In the Statstical skills section, I have the the chance to explore the sources of variance in a balanced experimental design by first visualizing the data, modeling the data, and comparing different sources of variance. I also applied linear mixed models to the data and interpreted the data. In addition, I explored the meaning of p-value by simulating data from different normal distributions, conducting one sample two sided t-test, plotting the distribution of p-value to visualize the distribution, and related the results to the interpretation of p-value. To practice my statistical communication skills, I have also explained and produced a reprex, a reproducible example.

In both the Statistical Skills and Writing section, I have demonstrate my ability to communicate with different audience. In the Statistical Skills part, I implemented functions to interpret confidence interval and p-value to audience for researcher that do not have much statistical background, explained the usage of the function, and introduce confidence interval and p-value. In the Writing part, I wrote a letter to myself in the future about the misconceptions of statistics including p-hacking and the over-reliance on significant level so that I could recall what it is and how to avoid it.

Statistical skills sample

Task 1: Setting up libraries and seed value

```r
# run package
library(tidyverse)
library(dplyr)
#install.packages("lme4")
library(lme4)
```

```r
# 100 + student ID
last3digplus <- 100 + 380
```

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

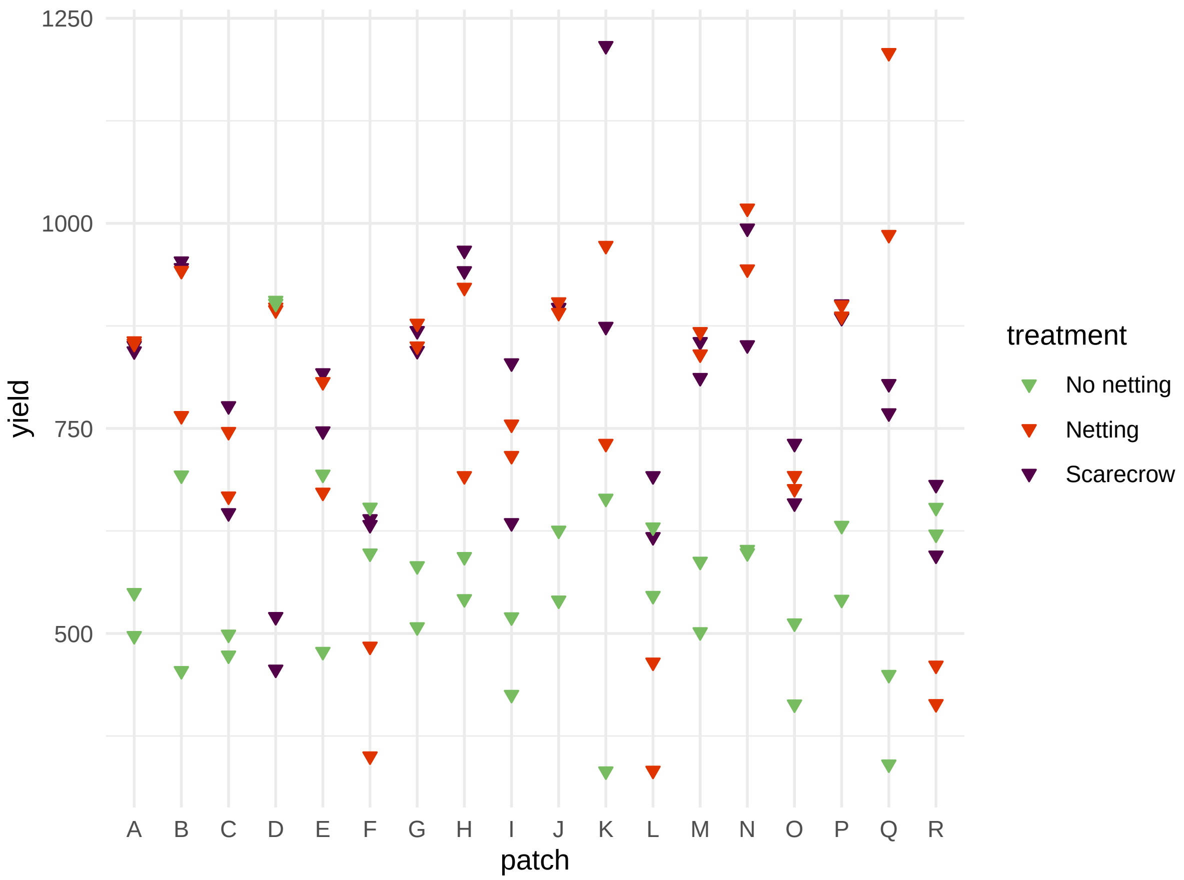
```r
# Don't edit this file
# Sourcing it makes a function available
source("grow_my_strawberries.R")
```

```r
# run strawberry
my_patch <- grow_my_strawberries(seed = last3digplus)

my_patch <- my_patch %>%
  mutate(treatment = as_factor(treatment)) %>%
  mutate(treatment = fct_relevel(treatment, "Scarecrow", after = Inf)) %>%
 mutate(treatment = fct_relevel(treatment, "No netting", after = 0))
```

Plotting the strawberry patch

```r
# Visualize strawberry patch
my_patch %>%
  ggplot(aes(x=patch, y=yield, fill = treatment, color = 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 Yu-Chun in STA303/1002, Winter 2022")
```



Created by Yu−Chun in STA303/1002, Winter 2022

Figure 1: Yield of Each Strawberry Patch Across Three 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 amount of yield of strawberry produced (in kgs) in the kth harvesting time

- μ is the grand mean of strawberry weight yielded
- α_i are the I fixed effects for treatment
- b_j are the random effects for patch j
- $(\alpha b)_{ij}$ are the IJ interaction terms for the interaction between the treatment and the patch
- $(\alpha b)_{ij} \sim N(0, \sigma^2_{\alpha b})$
- $b_j \sim N(0, \sigma^2_b)$
- $\epsilon_{ijk} \sim N(0, \sigma^2)$
- All the random effects are mutually independent random variables

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

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

```
# interaction model
int_mod <- lm(yield~patch*treatment, data = my_patch)
#summary(int_mod)
```

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

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

```
# patch to patch variance
var_patch <- summary(patch_mod)$sigma^2 - (summary(agg_mod)$sigma^2)/3
```

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

```
# interaction variance
var_ab <- summary(agg_mod)$sigma^2 - var_int/2
```

```
# proportion explained by patch to patch variance
var_patch_pro <- round(var_patch/(var_patch+var_int+var_ab), 2)

# proportion of residual variance
var_int_pro <- round(var_int/(var_patch+var_int+var_ab), 2)

# proportion of interaction variance
var_ab_pro <- round(var_ab/(var_patch+var_int+var_ab),2)

tibble(`Source of variation` = c("patch",
                                 "residual",
                                 "patch:treatment"),
       Variance = c("var_patch", "var_int", "var_ab"),
       Proportion = c(var_patch_pro,
                      var_int_pro,
var_ab_pro)) %>%
  knitr::kable(caption = "Source of Variation")
```

Table 1: Source of Variation

Source of variation	Variance	Proportion
patch	var_patch	0.06
residual	var_int	0.27
patch:treatment	var_ab	0.67

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

```
# mod0, a linear model with only treatment

mod0 <- lm(yield~treatment, data = my_patch)
#summary(mod0)
```

```
#mod1, a linear mixed model with treatment and patch (appropriate choices about what
↪  is a fixed vs random effect should be made)
mod1 <- lmer(yield~treatment + (1|patch), data = my_patch)
#summary(mod1)
```

```
# mod2, a linear mixed model with treatment, patch and the interaction of treatment
↪  and patch (appropriate choices about what is a fixed vs random effect should be
↪  made)
mod2 <- lmer(yield~treatment + (1|patch) + (1|patch:treatment), data = my_patch)
#summary(mod2)
```

```
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 -683.83
## 2   6 -667.37  1 32.931  9.549e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
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 -699.92
## 2   5 -683.83  1 32.171  1.412e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

When comparing both mod0 vs.mod1 and mod1 vs. mod2, we are using REML and we can use likelihood ratio test with models fit since the fix effect is same, which is treatment. We are only comparing the nested random effects here. Also, since our main goal is to estimate the fixed and random model parameter, REML is more suitable in this situation.

Justification and interpreation

Based on the two comparisons (mod0 vs. mod1 and mod1 vs. mod2), the p-value is small enough and there is evidence that mod0 is different from mod1 and mod1 is different from mod2. This means that patch does influences the result and the interaction between patch and treatment also influences the result.

According to the linear mixed model, when applying the "Net" treatment, the yield of the strawberry will increase, on average, by 210 kg. When applying the "Scarecrow" treatment, the yield of the strawberry will increase, on average, by 230 kg. Furthermore, from part 2a, the interaction between patch and treatment account for the most variability for the model, which correspond to our model comparison here.

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
    # 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 numeric and is the lower bound of the confidence
    ↪  interval")
  } else if(!is.numeric(upper)) {
    # produce a warning if upper isn't numeric
    warning("Warning: upper should be numeric and is the 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: Incorrect Confidence interval level. A correct level should take
    ↪  numeric value between 0 and 100.")
  } else{
    # print interpretation
    # this is the main skill I want to see, writing a good CI interpretation.
  str_c("The confidence level is ", ci_level,
        "%. And the text fed to stat is ", stat,
        ". There is also the lower and upper bounds: ", lower, " and ", upper,
        ". According to the result, we are confident that ", ci_level, " out of 100
        ↪  times our estimate will fall between upper and lower bound (", lower, ",",
        ↪  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: The confidence level is 99%. And the text fed to stat is mean number of shoes owned by students. There is also the lower and upper bounds: 10 and 20. According to the result, we are confident that 99 out of 100 times our estimate will fall between upper and lower bound (10,20).

CI function test 2: Warning: Incorrect Confidence interval level. A correct level should take numeric value between 0 and 100.

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
# This is just an example! MODIFY THIS CODE
interpret_pval <- function(pval, nullhyp){
  if(!is.character(nullhyp)) {
    warning("
            Warning: nullyhyp should be a character string.")
  } else if(!is.numeric(pval)) {
    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 negative")
  } else if(pval >= 0.1){
    str_c("The p value is ", round(pval, 3),
                ", we have no evidence against the hyothesis that ", nullhyp)
  }else if(0.1 > pval & pval >= 0.05){
    str_c("The p value is ", round(pval, 3),
                ", we have weak evidence against the hyothesis that ", nullhyp)
  } else if(0.05 > pval & pval >= 0.01){
    str_c("The p value is ", round(pval, 3),
                ", we have some evidence against the hyothesis that ", nullhyp)
  }else if(0.01 >pval & pval >= 0.001){
    str_c("The p value is ", round(pval, 3),
                ", we have strong evidence against the hyothesis that ", nullhyp)
  }else if(pval < 0.001){
    str_c("The p value is <.001, we have 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, we have 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, we have some evidence against the hyothesis 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 hyothesis that the mean grade for statistics students is the same as for non-stats students

p value function test 4: Warning: nullyhyp should be a character string.

Task 3c: User instructions and disclaimer

Instructions

The function interpret_ci checks whether your input of lower and upper bound is in the correct data type (numeric), whether your confidence level is appropriate (a numeric value between 0 and 100), and whether your description of the statistics of interest is in the correct data type (character string). It takes four parameters: lower, upper, ci_level, stat. To use this function, provide your lower and upper bound of the confidence interval and the confidence level and the statistics of interest, then the function will tell you the meaning of the confidence interval. An important term related to confidence interval is the population parameter, which is used to describe the entire population. For instance, if we want to estimate the mean of the population, μ will be the population parameter, while our sample statistics will be the mean of the sample that we take from the population. Confidence interval provides the percentage of intervals that contains the true parameter if you sample the same data set repeatedly. In other words, the meaning of 95% confidence interval is that, if we were to sample 100 times repeatedly with replacements from a population and calculate its confidence interval each time, then approximately 95 of the confidence interval will contain the true population parameter.

The function interpret_pval takes two parameters: pval and nullhyp. It checks whether your nullhyp parameter is a character string and whether your pval is acceptable (numeric and between 0 and 1). To use this function, input your p-value and your null hypothesis, then the function will interpret the meaning of the p-value under your null hypothesis. Null hypothesis is the hypothesis that there is no difference between some characteristics or between different population. By knowing the null hypothesis, we can calculate the p-value of the hypothesis test, which is the probability that we can observe the data that we have currently, given that the null hypothesis is true. In general, if the p-value is greater than 0.1, we have no evidence against the null. If the p-value is between 0.05 and 0.1, then we have weak evidence against the null. If the p-value is between 0.01 and 0.05, we have some evidence against the null. If the p-value is between 0.001

and 0.01, we have strong evidene against the null. Lastly, if the p-value is less than 0.001, we have very strong evidence against the null.

Disclaimer

You should be mindful when you are using the p-value interpreter. The cut-off point used in the interpreter is just a common standard. However, depends on your subject of interest, the p-value might be in general higher or lower compared to this interpreter due to the nature of you study. This is to say that the interpreter is only for you to have a sense of how your data is behaving and you should take it with a grain of salt. Simply because your p-value is small does not mean that your research is successful, you should still focus on your analysis and null hypothesis to see if that make sense in the context. In addition, you should assess the p-value after you are done with the research and data collection process, after you finished your analysis. You should not modify your analysis or data after you find out that your p-value is not low enough.

Task 4: Creating a reproducible example (reprex)

A reprex is a reproducible example that is minimal. It is needed when you are stuck when writing code or you could not find the correct ways to write code to achieve your goal. When creating reprex, you will have to make sure to make code reproducible by including every library and objects you used. It is also important to make it readable and minimal so that people would understand your code better and thus would get better help.

```r
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
set.seed(last3digplus)
```

```
# Simulations!
# set group
group <- rep(1:1000, each = 100)

# sim 1 N(0, 1)
val <- rnorm(100000, mean = 0, sd = 1)
sim1 <- tibble(group, val)

# sim 2 N(0.2, 1)
val <- rnorm(100000, mean = 0.2, sd = 1)
sim2 <- tibble(group, val)
# sim 3 N(1, 1)
val <- rnorm(100000, mean = 1, sd = 1)
sim3 <- tibble(group, val)
```

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

```
# join sim_description
# 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)"))

all_sim$sim <- as.numeric(all_sim$sim)

# join
all_sim <- all_sim %>%
  left_join(sim_description, by = "sim")
```

```
# histogram of first three groups

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 Yu-Chun in STA303/1002, Winter 2022")
```

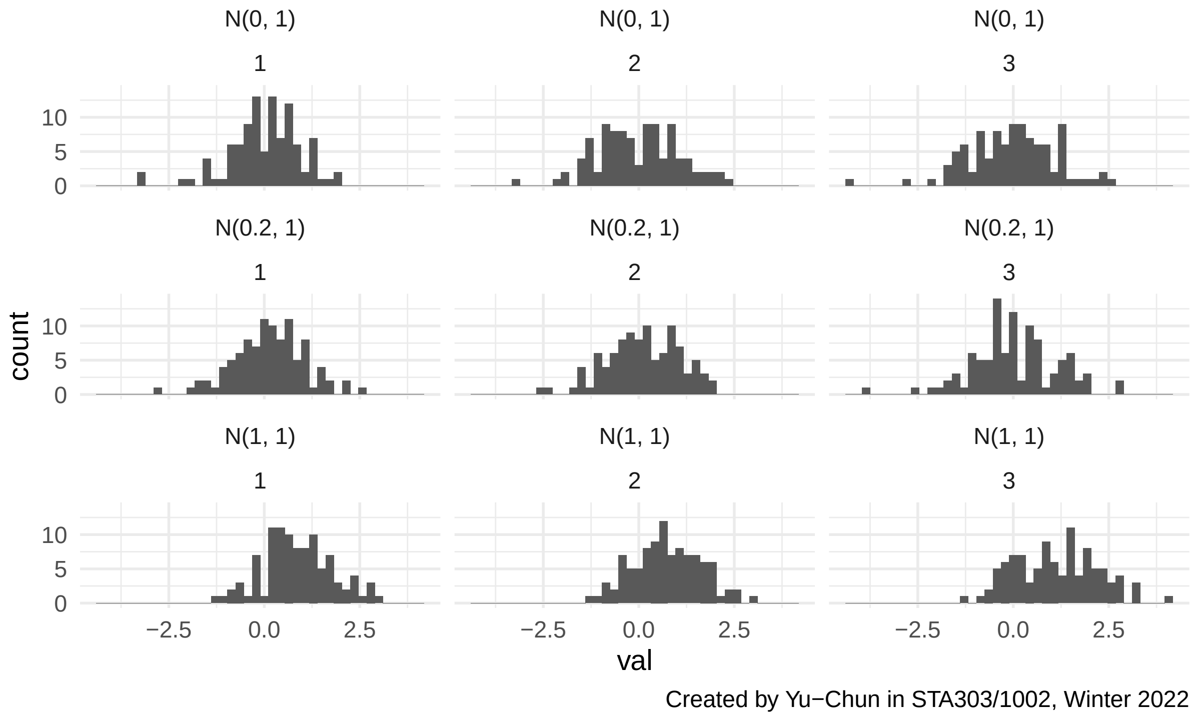


Figure 2: Histogram of Three Simulated Normal Distribution with Three Groups Each

Calculating *p* values

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

```r
# Visualize histogram

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

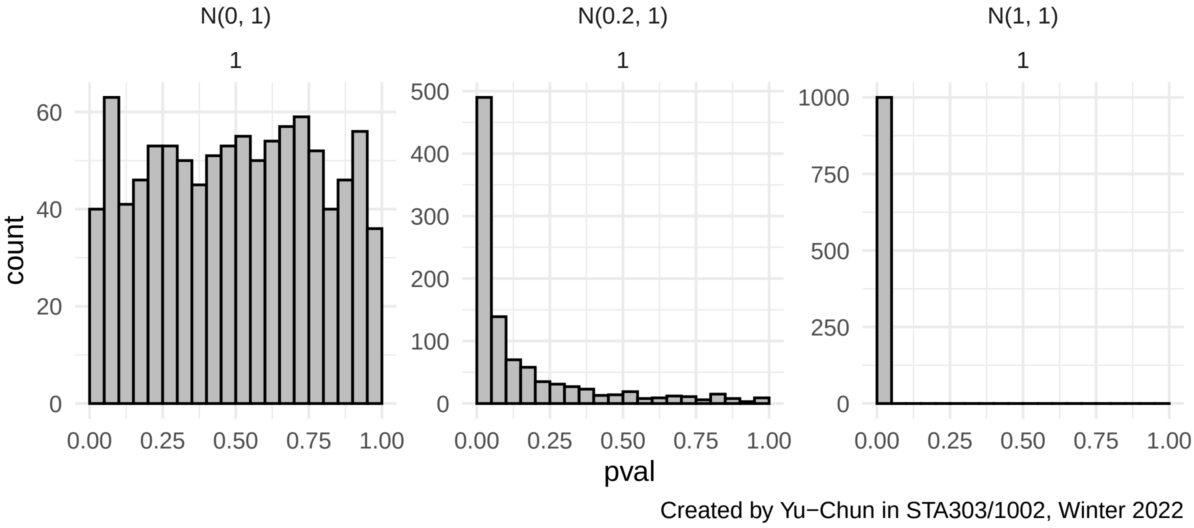


Figure 3: Distribution of P-value of Different Normal Distribution

Drawing Q-Q plots

```r
# Visualize qqplot
pvals %>%
  ggplot(aes(sample = pval)) +
  geom_qq(distribution = stats::qunif) +
  geom_abline(intercept = 0, slope = 1) +
  facet_wrap(~desc) +
  theme_minimal() +
  labs(caption = "Created by Yu-Chun in STA303/1002, Winter 2022")
```

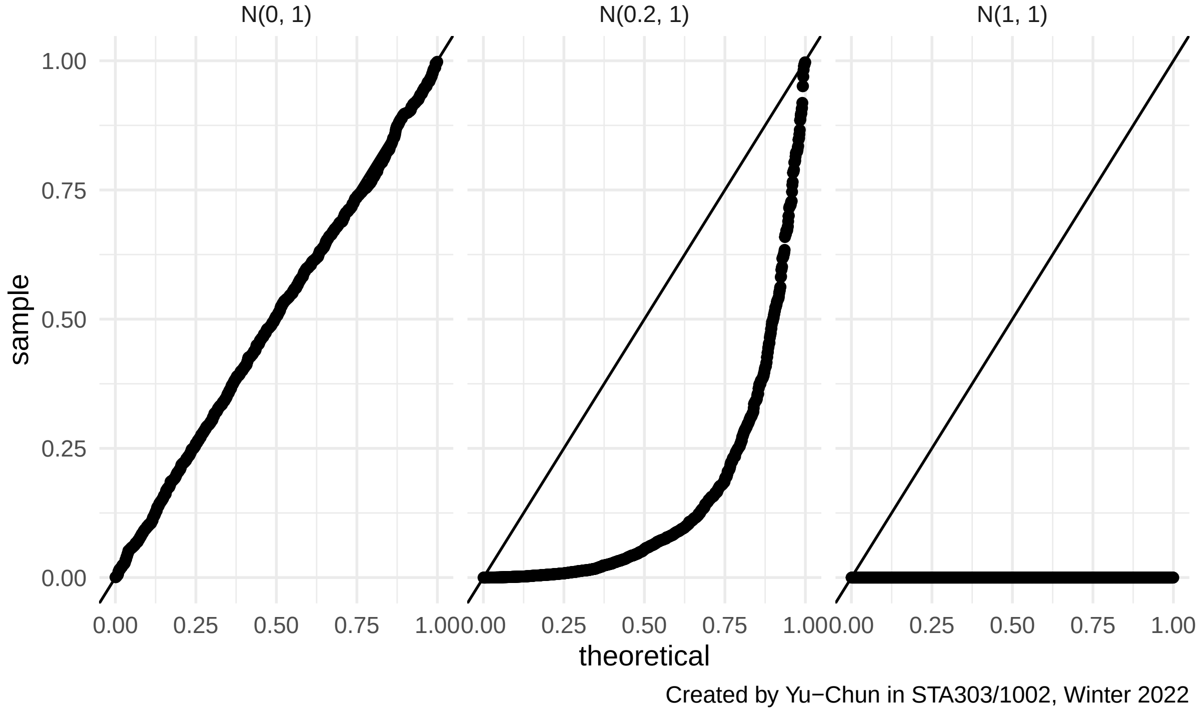


Figure 4: Q-Q Plots of Three Normal Distribution

Conclusion and summary

P-value is the probability that we can observe our observed data under the null hypothesis. Here, we simulate data from three normal distributions having same variance but different mean and conducted one sample two sided t-test. The null hypothesis is that the mean of the distribution is 0. Looking at Figure 3, the distribution of the p-value could be used to explain the meaning of p-value. Here, the p-value of the normal distribution with mean 0 is uniformly distributed, meaning that under the null hypothesis that the mean of the distribution of zero, it is sometimes likely to observe the data we obtained while sometimes unlikely. This means that we do not have the evidence to reject the null hypothesis. As the mean goes from 0 to 0.2 to 1, the p-value is more clustered on 0, meaning that most of the times we are not likely to see the observed data if the mean is 0.

This is connected to question 16 in that we can observe from the histogram that we just plot in Figure 3 that the distribution of the p-value of $N(0,1)$ is $Unif(0,1)$. Thus, approximately 10% of the p-values is between 0.9 and 1, and the fourth option of question 16 is correct.

Writing sample

As a statistics and data analyst professional, I hope you have successfully used data and statistical models to generate insights and help solve important problems by the insights of the data. I also hope that you utilized what you have learned in your undergraduate studies and avoid making mistakes that many data analysts or statisticians make, which includes p-hacking and relying on statistical hypothesis testing too much (Motulsky, 2014).

Firstly, p-hacking might sound scary, but it actually often happens. P-hacking happens when you attempt to keep the p-value under the significant level after you have already conducted the research and analysis (Motulsky, 2014). Example of p-hacking is when you alter the sample size by including more data or by only analyzing a subset of your data to reach a p-value less then the significant level. Another example stated by Motulsky (2014) is when you determine your hypothesis after you conducted the research by analyzing your data in multiple ways then settle down for a hypothesis that could lead to a p-value under 0.05. To avoid p-hacking, Motulsky (2014) suggested to let readers know whether the sample size is choose before conducting research and whether you alter anything after research is conducted. If you did, then be sure to mention that the conclusion is "preliminary".

Another misconception or mistake is to believe that statistical hypothesis testing and stating whether the test is statistically significant is needed in every research. In statistics, the most agreed cut off point of "statistical significance" is 0.05. However, this cut off point hugely depends on situation, such as the subject matters or the field of the study. In some cases this need to be higher while in some this need to be lower (Motulsky, 2014). Furthermore, just because the p-value of your analysis is not small enough does not mean that your analysis is without worth. For a p-value slightly higher than the cut-off point, it might be due to the sample of the research, or many other reasons that could potentially influence the p-value. In a statistical hypothesis testing, the more important thing is to report how likely will we observe the data under the null hypothesis instead of a binary decision (Motulsky, 2014). Since the word "significant" is often confusing and have multiple meaning, Motulsky (2014) suggested to not include "significant" but include the p-value if you are using hypothesis testing in your research.

I hope this reminds you of what you have been learning in your undergraduate studies. Statistics and data analysis is powerful and useful when used properly and correctly, but it might become disastrous when you misuse it since it might cause decision-makers to make wrong decision that potentially is harmful to the public. Thus, to use statistics to help make informed-decisions, make sure to keep in mind these misconceptions and avoid it in your research.

Word count: 477 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?

This is my first time creating a reprex and I am really proud to learn how to create a reprex since it is such a useful tool when I need to reach out for help when I have any programming problems. Before this course, I have seen reprex multiple time without even knowing that it is called reprex, but I do not know how to produce one. Although the actual steps of producing a reprex is easier than I thought, it is still a valuable thing to know so that in the future whenever I encountered any problem, I could create a reprex and make a post on Stack Overflow. It not only is easier for people to read, but it also increase my chance of getting the answer that I wanted.

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

I might apply my statistical communication skills, modeling skills, and data visualization skills in future work and study. As a future biostatistician wannabe, it is likely that I will have to work with people who do not necessary have enough statistical background to know the statistical concepts. Thus, statistical communication skills is important so that I could interpret the terms correctly and explain to my colleagues in an easy to know fashion. Moreover, since I might encounter many experimental designs as a biostatistician, I might want to apply different kind of models, for instance linear mixed models, and distinguish random and mixed effects. Lastly, to clearly communicate the results and to clearly understand the trends of the data, I would apply my data visualization skill to communicate my findings effectively.

What is something I'd do differently next time?

In general I think that I am doing pretty good in this portfolio and have corrected the mistakes from the mini-portfolio. The one thing that I think bothers me the most is that I should knit more often, panic less when seeing errors, and go look for help when error occur. For instance, in order to skip a line, I first put the wrong latex. I did not realize that this is an issue and did not spend time looking at the error message. I go through the coding part multiple times but still could not found the problem. I then go and search the error message, then found out the line that was causing error. This takes me more than ten minutes, and I think if I calm down at first and check the error message or search it online, then I might be able to spend less time on it. I also realized how common and often it is to make minor error that prevents knitting to pdf, so I do not have to panic each time I encountered an error.