STA303 Portfolio

An exploration of linear mixed models and common misconceptions in statistics

Lingjun Meng

2022-02-17

Contents

List of Figures

Introduction

As a student enrolled in STA 303, I am qualified with a range of statistical skills that are required to be a Data Scientist. For example, handling data set, creating visualizations, understanding the assumptions for different models when statistical methods are involved in the data analysis, being fluent with R as well as communicating the result of the analysis to a range of different audiences.

As a third-year undergraduate, I am also prepared to start my career life after my graduation. I can identify the skills and abilities that are required by the employees, I am also able to make judgment about whether I am qualified with the skills required. For instance, I can determine if I have the transferable skills required to working with colleagues as well as the professional skills in the field of Data Science. Supporting evidences of me possessing the skills can be provided.

In this Portfolio, I will introduce 5 different kinds of statistical skills by providing four specific examples. Skills demonstrated includes fluency in R, understanding source of variance and apply liner mixed model, building functions that interpret the definition of statistical terms including CI and P-value, creating reproducible examples as well as visualizing the behaviors of P-values calculated from simulation data.

There will also be a Writing Sample regarding a scientific paper, which demonstrated 5 possible misconception about data analysis. I made comment on the intriguing ideas brought up by the author and made connection to the statistical skills that I posses. At the end of this portfolio, there will be a reflection section sharing my thoughts after finishing this portfolio. That includes the thing I am proud of for this Portfolio, how I will apply these skills demonstrated in the future as well as something I would do differently next time.

Statistical skills sample

Task 1: Setting up libraries and seed value

```
library(tidyverse)
library(lme4)
#Loading the libraries

last3digplus <- 100 + 579
#Setting up
```

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

```
source("grow_my_strawberries.R")

my_patch <- grow_my_strawberries(seed = last3digplus) %>% mutate(
  treatment = treatment %>% fct_relevel("No netting"))
#Loading the data in and changing the level of the factor variable "treatment"

levels(my_patch$treatment)
```

```
## [1] "No netting" "Netting"    "Scarecrow"
```

```
#Checking if the levels are in order.
```

Plotting the strawberry patch

```
my_patch %>%
  ggplot(aes(x = patch, y = yield, color = treatment, fill = treatment)) +
  scale_fill_manual(values = c("#78BC61", "#E03400", "#520048")) +
  scale_colour_manual(values = c("#78BC61", "#E03400", "#520048")) +
  geom_point(pch = 25) +
  theme_minimal() +
  labs(caption = "Created by LINGJUN MENG in STA303/1002, Winter 2022",
       x = "Patch",
       y = "Yield")
```

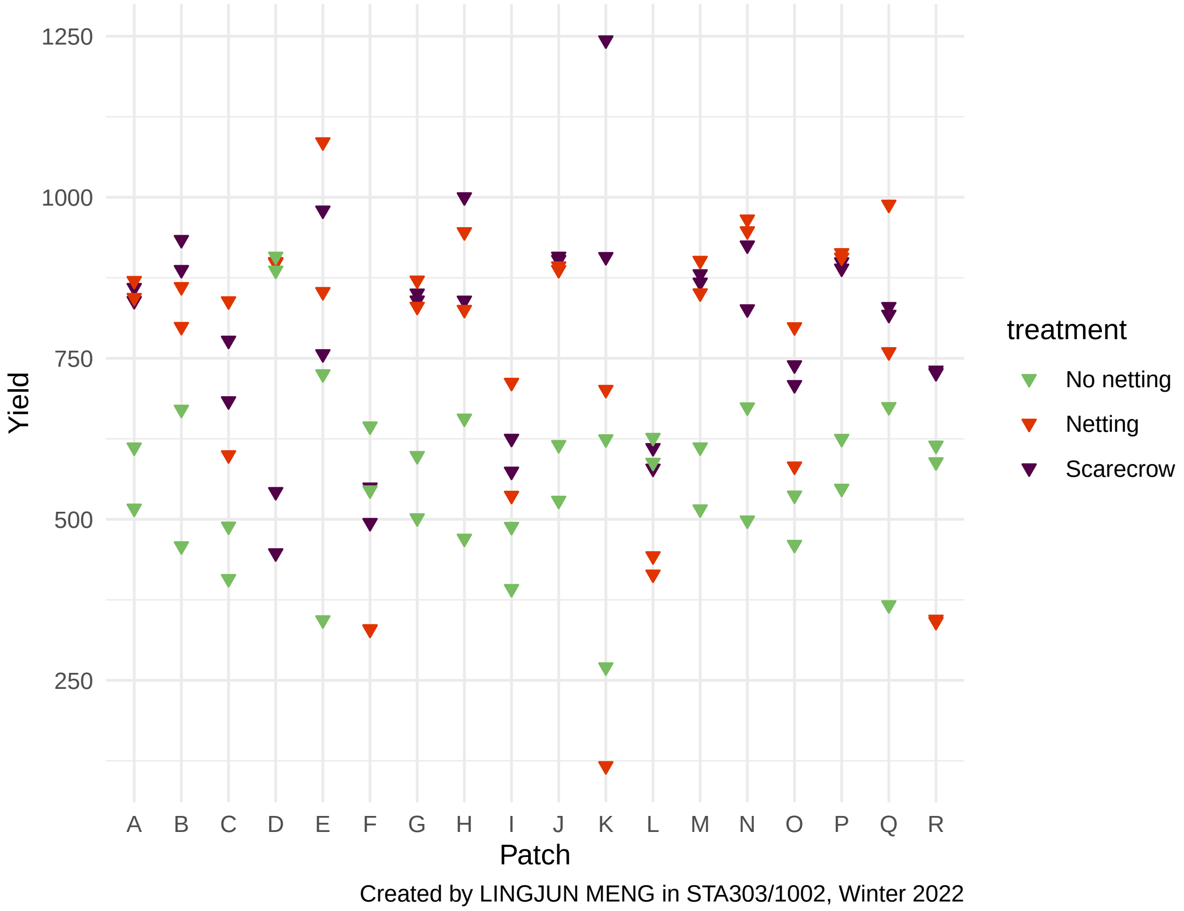


Figure 1: Yield of the Straberries in different Patch with different Treatment

```
#Creating the plot.
```

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

Model formula

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

where:

- y_{ij} are the yield of the strawberries grew in the j^{th} Patch with Treatment i.
- μ is the grand mean of all the strawberries.
- α_i are the fixed effects of the Treatment i on the yield.
- b_j are the random effects of the j^{th} Patch on the yield, which is a normal random variable that follows $N(0, \sigma_b^2)$
- $(\alpha b)_{ij}$ are the random effects of the interaction of Treatment i and the j^{th} Patch on the yield, which is a normal random variable that follows $N(0, \sigma_{ab}^2)$
- $\epsilon_i j$ are referring to the errors between the fitted values and the true target, which is also a normal random variable that follows $N(0, \sigma^2)$

```r
# Lots and lots of code! Woo!
int_mod <- lm(yield ~ treatment * patch, data = my_patch)
#Creating int_model using interaction term

agg_patch <- my_patch %>%
  group_by(patch) %>%
  summarise(yield_avg_patch = mean(yield))
#Calculating average yield per patch using aggregated data.

patch_mod <- lm(yield_avg_patch ~ 1, data=agg_patch)
#Creating patch_model using only intercept.

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

agg_mod <- lm(yield_avg_int~patch + treatment, data = agg_int)
#Creating the agg_model using two predictors.

#Calculating the variance of each model.
var_int <- summary(int_mod)$sigma^2

var_ab <- summary(agg_mod)$sigma^2 - var_int/2

var_patch <- summary(patch_mod)$sigma^2 - (summary(agg_mod)$sigma^2)/3

var_total <- var_int + var_ab + var_patch
```

```
tibble(`Source of variation` = c("treatment:patch",
                                  "patch",
                                  "residual"),
       Variance = c(var_ab, var_patch, var_int),
       Proportion = c(var_ab/var_total,
                      var_patch/var_total,
                      var_int/var_total)) %>%
  knitr::kable(caption = "Variance of different Sources and their Proportion of
↪   Variance Explained")
```

Table 1: Variance of different Sources and their Proportion of Variance Explained

Source of variation	Variance	Proportion
treatment:patch	17740.358	0.5258631
patch	3163.842	0.0937832
residual	12831.494	0.3803536

```
#Creating the table with the variance of each term as well as their Proportion of
↪   variance explained.
```

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

```
#Creating required 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)

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

```
## Likelihood ratio test
##
## Model 1: yield ~ treatment + (1 | patch) + (1 | patch:treatment)
## Model 2: yield ~ treatment + (1 | patch)
## Model 3: yield ~ treatment
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   6 -687.62
## 2   5 -697.06 -1 18.877  1.394e-05 ***
## 3   4 -713.78 -1 33.443  7.338e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Making comparison between the fitted models.
```

Here we are using REML as all the models have the same fixed effect and our goal is the estimates of both the fixed and random parameters. This is consistent with what we are doing: assessing the model quality with the same fixed effect and different random effects.

Justification and interpreation

From the output generated by the Likelihood ratio test, the P-values calculated for both of model 1 and model 2 are extremely small. That provides strong evidence against the null hypothesis that the reduced models are just as good as the full model. (model 2) Therefore, mod2 is the most appropriate 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: 1375.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.3214 -0.4929  0.0792  0.3765  2.1882
##
## Random effects:
##  Groups          Name        Variance Std.Dev.
##  patch:treatment (Intercept) 17740    133.19
##  patch           (Intercept)  3164     56.25
##  Residual                    12831    113.28
## Number of obs: 108, groups:  patch:treatment, 54; patch, 18
##
## Fixed effects:
##                     Estimate Std. Error t value
## (Intercept)           563.51      38.96  14.464
## treatmentNetting      177.81      51.81   3.432
## treatmentScarecrow    227.58      51.81   4.393
##
## Correlation of Fixed Effects:
##             (Intr) trtmnN
## trtmntNttng -0.665
## trtmntScrcr -0.665  0.500
```

From the summary of mod2 we can see the coefficients for fixed effect are 563.51, 177.81 and 227.58 respectively. That means the strawberries with No netting treatment are expected to have a yield of 563.51, the strawberries with Netting treatment are expected to have a yield that is 177.81 higher than the base(No netting treatment) and the strawberries with Scarecrow treatment are expected to have a yield that is 227.58 higher than the base(No netting treatment). In addition, note that the variances that are explained by each source of the random effects exactly match the table generated above at Task 2a.

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("lower can only take numerical values as it is the lower bond of the CI")
  } else if(!is.numeric(upper)) {
    # produce a warning if upper isn't numeric
    warning("upper can only take numerical values as it is the upper bond 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("ci_level can only take numerical values between 0 and 100")
  } else{
    # print interpretation
    # this is the main skill I want to see, writing a good CI interpretation.
  str_c("This is the confidence level: ", ci_level,
        "%. And the text fed to stat is ", stat,
        ". There is also the lower and upper bounds: ", lower, " and ", upper,
        ". meaning that we are ", ci_level,"% confident that the true population
        mean nuber of shoes owned by students 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: This is the confidence level: 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. meaning that we are 99% confident that the true population mean nuber of shoes owned by students is between 10 and 20

CI function test 2: ci_level can only take numerical values 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

```
interpret_pval <- function(pval, nullhyp){
  if(!is.character(nullhyp)) {
    warning("Invalid nullhyp, the null hypothesis can't take Non-character string
    ↪  values")
    #Raise warning if the input of nullhyp is invalid
  } else if(!is.numeric(pval)) {
    warning("Invalid pval, P-values should be numerical values")
    #Raise warning if the input of pval is invalid
  }  else if(pval > 1) {
    warning("Invalid pval, P-values can not be greater than 1")
    #Raise warning if the input of pval is invalid
  } else if(pval < 0){
    warning("Invalid pval, P-values can not be smaller than 0")
    #Raise warning if the input of pval is invalid
  } else if(pval < 0.001){
    str_c("The p value is <0.001,
          we have very strong evidence against the null hyothesis that ", nullhyp,
↪   ".")
  } else if(pval <= 0.01){
    str_c("The p value is ", round(pval, 3),
          ", we have strong evidence against the null hypothesis that ", nullhyp, ".")
  } else if(pval <= 0.05){
    str_c("The p value is ", round(pval, 3),
          ", we have moderate/some evidence against the null hypothesis that ",
          ↪  nullhyp, ".")
  } else if(pval <= 0.1){
    str_c("The p value is ", round(pval, 3),
          ", we have weak evidence against the null hypothesis that ", nullhyp, ".")
  } else if(pval > 0.1){
    str_c("The p value is ", round(pval, 3),
          ", we have no 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 <0.001, we have very strong evidence against the null hyothesis 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 moderate/some 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, we have 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: Invalid nullhyp, the null hypothesis can't take Non-character string values

Task 3c: User instructions and disclaimer

Instructions

The Confidence Interval Interpreter function takes four values: lower bond, upper bond, Confidence level and the statement of the parameter. The function will explain the definition of the CI with given parameters, which is making inference about the true population parameter. Normally the population parameter is unknown, especially when the population is large. For instance, it is impossible for us to find the true average height of human by measuring the height of every individual on Earth. However, you can make inference about the population parameter using Confidence Intervals. And that is exactly what the function will do with given lower bond, upper bond, Confidence level and the statement of the parameter.

The P-value Interpreter function takes the P-value calculated and the statement of null hypothesis. The P-value is calculated under the condition that we assume the null hypothesis to be true. Thus, if the P-values are small, there could be a chance that we are making the wrong assumption. The smaller the P-value is, the stronger the evidence is against the null hypothesis. A null hypothesis is usually something we want to test. For example, a null hypothesis can be there are no difference between the average temperature in winter and in summer in Toronto. The function will comment on the strength of the evidence that we have against the null hypothesis with valid P-value calculated and the statement of null hypothesis provided.

Disclaimer

Note that when inputting without variable names for the CI interpreter function, make sure the lower bond is in the front and the upper bond is behind the lower bond. The function will still work even with lower bond being greater than the upper bond. Also note that the lower and upper bonds as well as the Confidence level should be numerical and Confidence level should be between 0 and 100. And statement should be a character string that describes the statistics of interest, don't forget the quotation marks.

For the P-value Interpreter function, just make sure the input is valid. The null hypothesis also should be a character string, and the P-value should be numerical and between 0 and 1.

Task 4: Creating a reproducible example (reprex)

It is very easy to understand what a reproducible example(reprex) is, it is just something others can reproduce. With the lowest effort and highest efficiency, others can understand what your problem is by showing them a reprex. That is especially helpful when you encounter a problem with the R code and try to seek for help online. With a reprex, you are helping others helping you as your situation will be addressed clearly.

Below is an example of a reprex.

```
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...
```

Created by the reprex package (v2.0.1)

Together with the reprex, you can also describe what you are trying to do in words. In this case the description will be:

I am trying to get a new summarized dataset where the mean value of each group will be displayed, and I only want one row for each individual group. I know that I should use summarize and group_by function, but the output is not the way I expected. What should I do?

Task 5: Simulating p-values

Setting up simulated data

```r
# Simulations
set.seed(last3digplus)
#Setting seed so we get the same result for each simulation

group = rep(1:1000, each = 100)
#setting group id for each simulation

val <- rep(0,100000)
#Creating a null vector
sim1 <- tibble(group,val)
sim2 <- tibble(group,val)
sim3 <- tibble(group,val)
#Creating three simulation datasets.

sim1$val <- rnorm(100000, mean = 0, sd = 1)
sim2$val <- rnorm(100000, mean = 0.2, sd = 1)
sim3$val <- rnorm(100000, mean = 1, sd = 1)
#doing simulations for different simulation datasets.

all_sim <- bind_rows(sim1, sim2, sim3, .id = "sim")
all_sim$desc <- rep(c("N(0, 1)", "N(0.2, 1)", "N(1, 1)"), each = 100000)
#Stacking all datasets into one.

desc = c("N(0, 1)", "N(0.2, 1)", "N(1, 1)")
sim_description <- tibble(sim = 1:3, desc)
#Creating the the description of the simulations dataset.
```

```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 LINGJUN MENG in STA303/1002, Winter 2022",
       y = "Frequency",
       x = "Values")
```

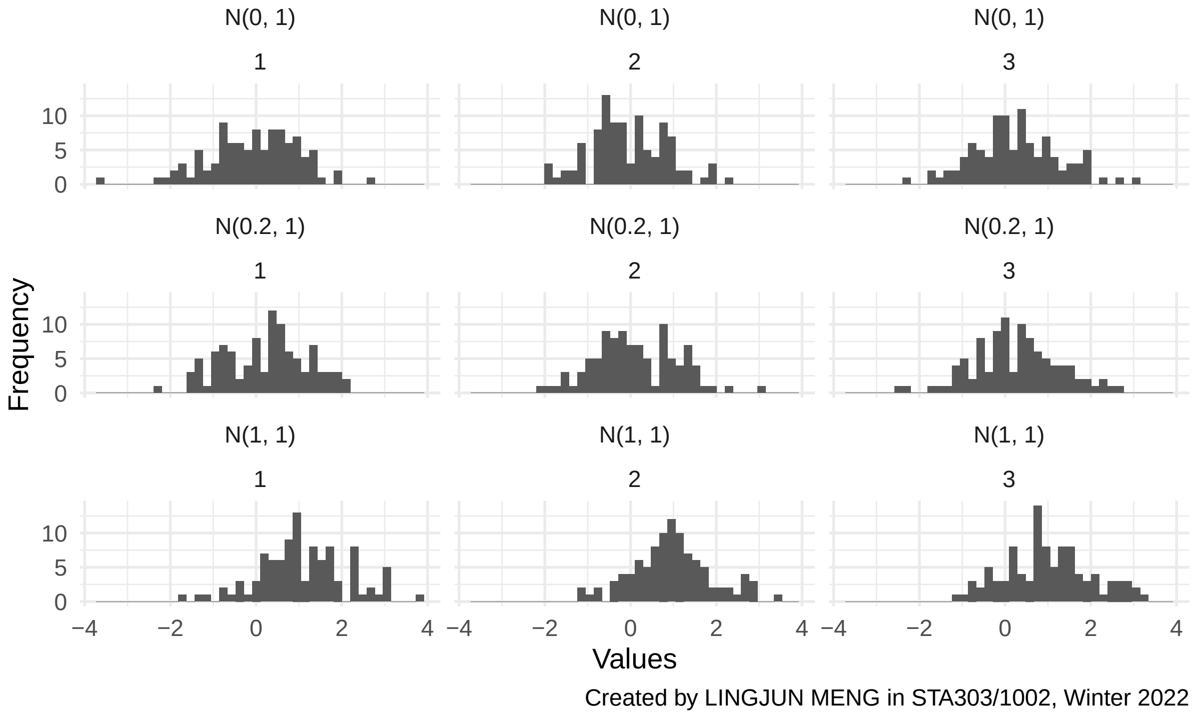


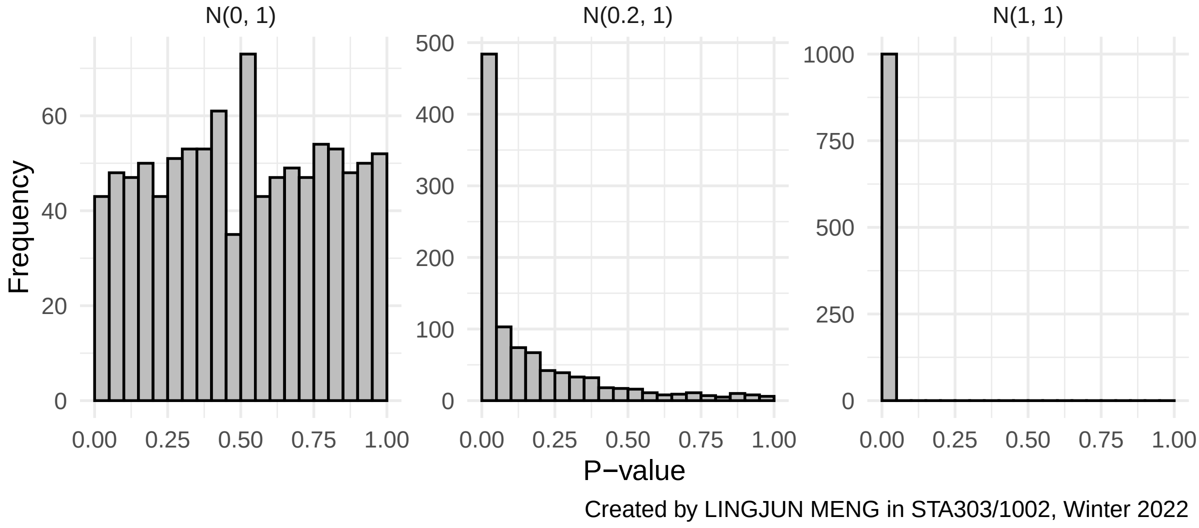
Figure 2: Histograms of the Values simulated in the first three groups for each simulation

```
#Creating histogram for each distribution sample in the first 3 groups
```

Calculating *p* values

```
#Calculating P-values for each group using two sided t test
pvals <- all_sim %>%
  group_by(desc, group) %>%
  summarise(p_value = t.test(val, mu = 0)$p.value,.groups = "drop")
```

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



Created by LINGJUN MENG in STA303/1002, Winter 2022

Figure 3: P-values from 2-sided t test of the simulated value with Null Hypothesis that population mean being equal to 0

```
#Plotting the P-values for each sample ditribution
```

Drawing Q-Q plots

```
pvals %>%
  ggplot(aes(sample = p_value)) +
  geom_qq(distribution = qunif) +
  geom_abline(intercept = 0, slope = 1) +
  facet_wrap(~desc) +
  theme_minimal() +
  labs(caption = "Created by LINGJUN MENG in STA303/1002, Winter 2022",
       x = "Theoretical Quantiles",
       y = "Sample Quantiles")
```

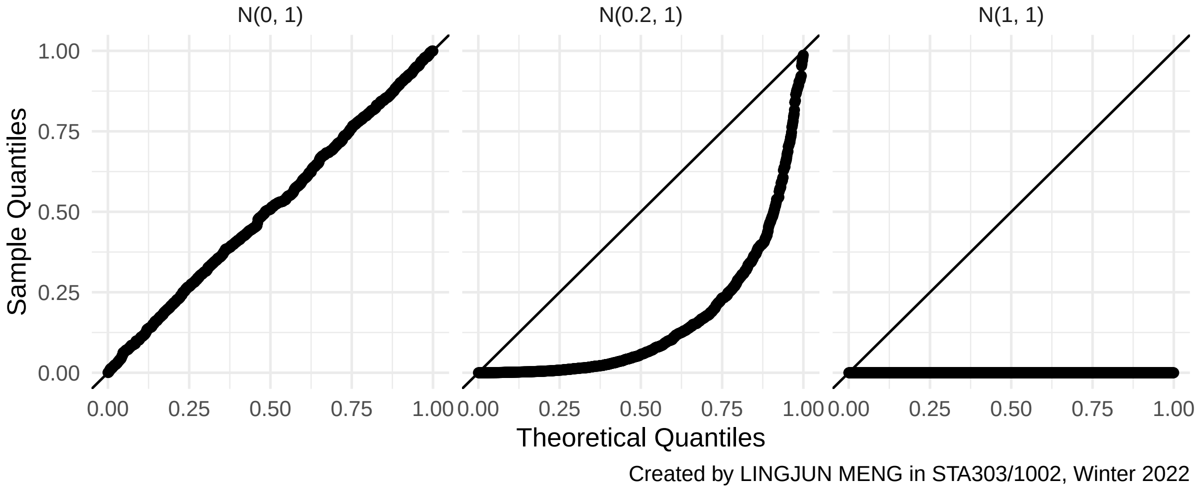


Figure 4: Q-Q plots of the P-values for three simulated values assuming the Null

```
#Plotting the Q-Q plots for each sample distribution
```

Conclusion and summary

In this task, We simulated 1000 groups of 100 normally distributed data points from N(0,1), N(0.2,1) and N(1,1) respectively. Then the P-values for H_0 that each group of data points have $\mu = 0$ were calculated and plotted for each group. The Q-Q plots for the P-values for each distribution were also generated.

By looking the histogram and Q-Q plot for the first group that is actually drawn from N(0,1), we can expect the P-values to be uniformly distributed if the true population mean is equal to be mean set in the Null Hypothesis. That is also consistent with the Q-Q plot generated, where few to no dot is off the straight line. In addition, we can also observe that the bigger the difference between the true population parameter and the one set in H_0, the more P-values will be centered close to 0. Discrepancies in the Q-Q plots will also be more notable with increased difference.

Writing sample

I believe every aspiring data scientist knows the significance of creating a reproducible work. That is the substantial factor for other experienced investigators to be able to reproduce your work if published in a journal. However, Motulsky found two examples that only a very small proportion of the investigator can actually reproduce the work. (Motulsky,2014) And he concluded that one reason for the investigators of being not able to reproduce the work could be the poor understanding of statistical concepts. Motulsky proposed 5 possible misconceptions, and 2 of the ideas were found to be very interesting.

The first one is about P-hacking, where you change the process of an analysis until you get the P-value that you like. Among the different type of P-hacking, there was one of Hypothesizing after you know the result. Motulsky provided a illustration of analyzing the association between jelly bean and acne. The investigator knew the outcome, and he was trying to find out which color of jelly bean will cause acne. So he tested 20 colors until he found the P-value of green bean is smaller than 0.5. I didn't expect this to be P-hacking until I saw the paper, but it did come to me that this is reasonable. Note the distributions of P-values in Task 5, where it will be uniformly distributed when the H_0 set μ to be the true population mean. In that case, there still can be 5% of the P-values being less than 0.05 as it follows Unif(0,1). That means even H_0 is true, there can still be a chance we get a P-value that is smaller than 0.05. Thus, hypothesizing after the outcome is known can be considered as P-hacking. An interesting point of how to understand P-hacking was raised by Motulsky, where you will add in bias if you change the analysis procedure after you get the P-value that you don't like. This converges well to the idea of cross-validation of a model. No matter it is a linear, logistic or Bayesian model, We can only test the performance of the model using unseen data. The test result will be biased if we include any data that is shown before, no matter it is in the training process or optimization process.

The second intriguing misunderstanding introduced by Motulsky was that P-values actually don't convey any information about the effective size of the analysis. He provided examples with sample P-value calculated but with different effective size, the conclusion of the analysis could be totally different regarding their sizes.(Motulsky, 2014) Thus, it will be meaningless if the conclusion was only made based on the P-value alone. I have always been obsessed with concluding with P-value and ignoring the size of the analysis. Therefore, always remember to include the effective size of the analysis when drawing a conclusion.

In conclusion, to become a qualified data scientist, one should be committed to the ethics and statistical methods, and build a solid foundation by understanding the substantial statistical concept. When doing your own analysis, you should do it "seriously, carefully and completely"(Motulsky, 2014)

Word count: 506 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?

One thing I am proud of in this portfolio is being able to identify the distribution of the P-values of the values simulated from normal distribution. If H_0 is set to be equal to the true population parameter, then the P-values will be uniformly distributed. I am proud of this as I found this out using the knowledge I learned, and I was able to create Q-Q plots based on their distribution.

Another thing that I am proud of is that I was able to make connections between what I see in a scientific paper and the knowledge that I acquired. Especially the part where I was talking about the relationship between how P-hacking make the analysis biased and cross validation. I am proud that I can apply my knowledge to the ideas that are brought up by other researchers.

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

Many of the skills demonstrated in this portfolio are very practical. For instance, making a Reproducible example.(reprex) For the work or study in the future, when I encounter a problem of the coding and try to seek for help, I will be benefited by possessing this skill.

Other skills including statistical skills, plotting skills and coding skills will build a substantial foundation for my future work or study. As Motulsky(2014) mentioned in his paper, we should not fool ourselves with a weak understanding of the concepts about the statistical methods. I will pursue a career in the field of Data Science, and these skills are the building blocks of my future success.

What is something I'd do differently next time?

I made a plan for the portfolio and I attended the office hours for help with the problem I encountered. I did much better compared to the mini-portfolio. However, it would be better if I did not split the workload to be that spread out.

I was planning to finish the portfolio within 10 hours, and I spent 2 hours each day for 5 days to finish it. However, these 5 days are not consecutive, and it took me some time to get familiar with the last steps because the time in between was more than 48 hours. This reduced my efficiency and productivity, and it would be much better if I spend 3 hours a day and finish the portfolio within 3 or 4 days.