STA303/1002 Portfolio

An exploration of linear mixed models and common misconceptions in statistics

Tianye Wang

2022-02-17

Contents

List of Figures

Introduction

In this document, I show my statistical skills sample, writing sample and reflection sections separately.I'll explain each of them below.

In terms of statistical skills, I can preprocess the data very well, and through the processing of data visualization, I can better show the characteristics of the data to the reader. And through the analysis of the data, I can build a good model of data correlation. And the model can make good predictions about the data of the test set. In the task, I showed the interpreters of my two statistical concepts. These two interpretations were made to help people with only basic statistical knowledge better understand what the results of statistical analysis represent. In the writing example, I read and analyzed the literature. And through his own thinking, he wrote a book report. Helps readers understand the content of the article more easily and quickly. In terms of reflaction to example, I explained the feeling of doing this assignment and the significance of this assignment.

This assignment improved my overall abilities as a student in statistics. Improved in data analysis, reading literature, and writing.

Statistical skills sample

Task 1: Setting up libraries and seed value

```
# load the package "tidyverse"
library(tidyverse)
# Create an object called last3digplus
 last3digplus<-100+698
```

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

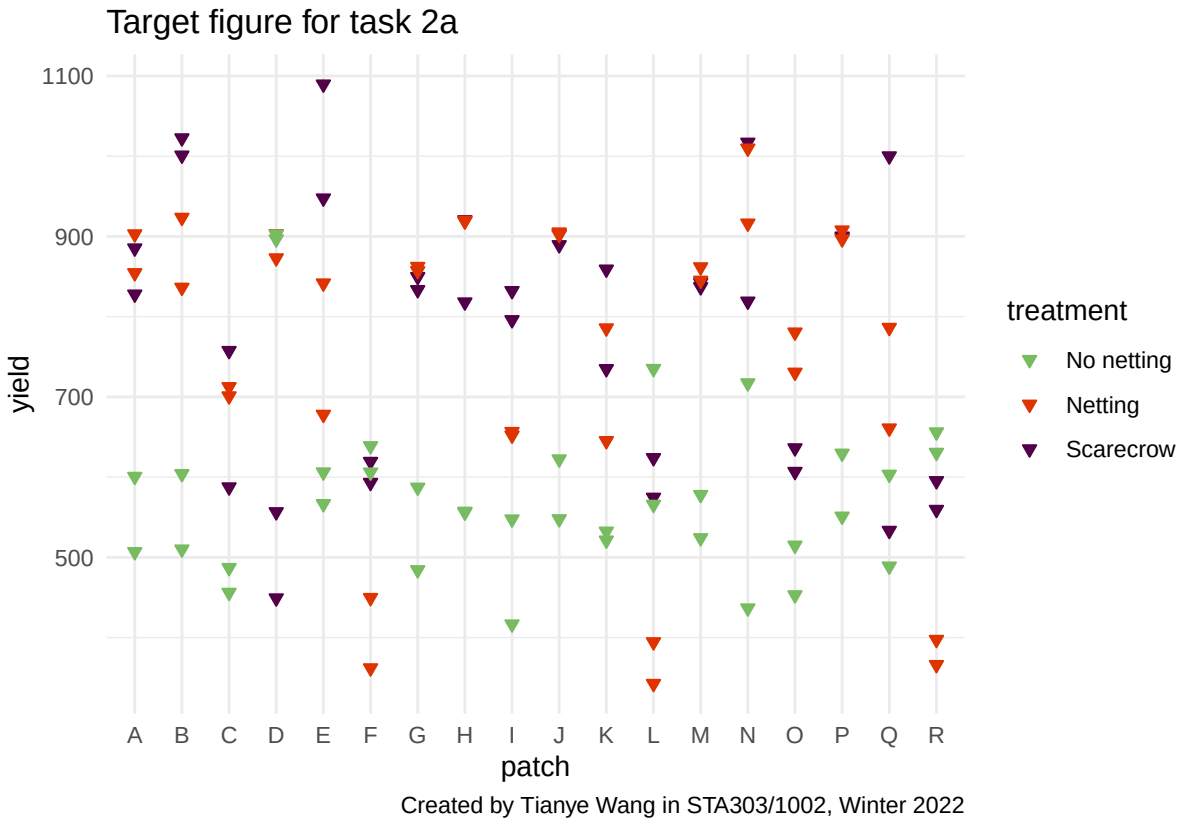
Growinng your (grandmother's) strawberry patch

```
# Sourcing it makes a function available
source("grow_my_strawberries.R")
```

```
# Run the function,and save the output
my_patch<-grow_my_strawberries(seed = last3digplus)
# Alter the my_patch data so that treatment is a factor
my_patch$treatment<-factor(my_patch$treatment)
```

Plotting the strawberry patch

```
# Return to Statdew Valley: exploring sources of variance in a balanced experimental
↪  design
my_patch  %>%
  ggplot(aes(x = patch, y = yield)) +
  geom_point(aes(colour = treatment,fill = treatment), pch = 25) +
  scale_colour_manual(values = c("No netting" = "#78BC61","Netting" =
↪  "#E03400","Scarecrow" = "#520048" ) ) +
  scale_fill_manual(values = c("No netting" = "#78BC61","Netting" =
↪  "#E03400","Scarecrow" = "#520048" )) +
  theme_minimal() +
  labs(title = "Target figure for task 2a",caption = "Created by Tianye Wang in
↪  STA303/1002, Winter 2022")
```



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

```
# The model formulation for the model
library(lme4)
full_mod<-lmer(yield~treatment+ (1|patch)+(1|treatment:patch),data = my_patch)
summary(full_mod)
```

Model formula

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: yield ~ treatment + (1 | patch) + (1 | treatment:patch)
##    Data: my_patch
##
## REML criterion at convergence: 1323.7
##
## Scaled residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -3.04950 -0.41966  0.04706  0.40880  3.01661
##
## Random effects:
##  Groups          Name        Variance Std.Dev.
##  treatment:patch (Intercept) 16397    128.05
##  patch           (Intercept)  3355     57.92
##  Residual                     5921     76.95
## Number of obs: 108, groups:  treatment:patch, 54; patch, 18
##
## Fixed effects:
##                     Estimate Std. Error t value
## (Intercept)           751.11      35.52  21.145
## treatmentNo netting  -172.23      46.38  -3.714
## treatmentScarecrow     32.72      46.38   0.706
##
## Correlation of Fixed Effects:
##             (Intr) trtmNn
## trtmntNnttn -0.653
## trtmntScrcr -0.653  0.500
```

Linear mixed model: $yield = \beta_{intercept} + \beta_{treatment} * treatment + (effect_{patch} + effect_{treatment*patch} + \epsilon)$

where:

- $\beta_{intercept}$ is basic intercept excluded the effect of patch and the interactive, and
- $effect_{patch}$ is the random effect of patch and
- $effect_{treatment*patch}$ is the random effect of the interactive between patch and treatment

```
# Create agg_patch
agg_patch<- aggregate(my_patch$yield,by = list(patch = my_patch$patch),mean)
agg_patch$yield_avg_patch<-agg_patch$x
agg_patch<-agg_patch[,-2]
agg_patch<-tibble(agg_patch)
# Create agg_int
agg_int<- aggregate(my_patch$yield,by = list(patch = my_patch$patch,treatment =
↪  my_patch$treatment),mean)
agg_int$yield_avg_int<-agg_int$x
```

```
agg_int<-agg_int[,-3]
agg_int<-tibble(agg_int)
# Create int_mod
int_mod<-lm(data = my_patch,yield~patch*treatment)
# Create patch_mod
patch_mod<-lm(yield_avg_patch ~ 1, data = agg_patch)

# Create agg_mod
agg_mod<-lm(yield_avg_int~.,data = agg_int)
# Create var_patch
var_patch<-(summary(patch_mod)$sigma)**2
# Creat var_int
var_int<-(summary(agg_mod)$sigma)**2
# Create var_ab
var_ab<-(summary(int_mod)$sigma)**2
```

```
# Create a table
tibble(`Source of variation` = c("variance in average yield patch-to-patch",
                                 "variance after fitting the version ",
                                 "variance in yield explained by the interaction"),
       Variance = c("var_patch", "var_int", "var_ab"),
       Proportion = c(round(var_patch, 2),
                      round(var_int, 2),
                      round(var_ab,2) )) %>%
  knitr::kable(caption = "A table that compare the variances of different models
")
```

Table 1: A table that compare the variances of different models

Source of variation	Variance	Proportion
variance in average yield patch-to-patch	var_patch	9807.30
variance after fitting the version	var_int	19357.42
variance in yield explained by the interaction	var_ab	5920.62

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

```
# Fit 3 models
mod0<-lm(data = my_patch,yield~treatment)
mod1<-lm(data = my_patch,yield~.)
mod2<-lm(data = my_patch,yield~patch*treatment)
library(lmtest)
lrtest(mod0,mod1)
```

```
## Likelihood ratio test
##
## Model 1: yield ~ treatment
## Model 2: yield ~ patch + treatment
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   4 -698.80
## 2  21 -673.03 17 51.531  2.437e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lrtest(mod0,mod2)
```

```
## Likelihood ratio test
##
## Model 1: yield ~ treatment
## Model 2: yield ~ patch * treatment
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   4 -698.80
## 2  55 -584.87 51 227.85  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lrtest(mod1,mod2)
```

```
## Likelihood ratio test
##
## Model 1: yield ~ patch + treatment
```

```
## Model 2: yield ~ patch * treatment
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1  21 -673.03
## 2  55 -584.87 34 176.32  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

REML / ML

If your random effects are nested, or you have only one random effect, and if your data are balanced (i.e., similar sample sizes in each factor group) don't use REML, because you can use maximum likelihood. If your random effects are crossed, don't set the REML argument because it defaults to TRUE anyway. you can use REML (or ML) whenever you want (regardless of the random effects structure - single vs. multiple, balanced vs. unbalanced, crossed vs. nested) in simple cases (balanced/nested/etc.) REML can be proven to provide unbiased estimates of variance components (but not unbiased estimates of e.g. standard deviation or log standard deviation) you cannot compare models that differ in fixed effects if they are fitted by REML rather than ML.

Justification and interpreation

Compared mod0 with mod1: From the output we can see that the p-value of the likelihood ratio test is 2.437e-05. Since this is less than .05, we would reject the null hypothesis. Thus, we would conclude that the mod1 offers a significant improvement in fit over the mod0. Compared mod0 with mod2: From the output we can see that the p-value of the likelihood ratio test is 2.2e-16. Since this is less than .05, we would reject the null hypothesis. Thus, we would conclude that the mod2 offers a significant improvement in fit over the mod0. Compared mod1 with mod2: From the output we can see that the p-value of the likelihood ratio test is 2.2e-16. Since this is less than .05, we would reject the null hypothesis. Thus, we would conclude that the mod2 offers a significant improvement in fit over the mod1. Therefore we choose mod2 as the final model.

Task 3a: Building a confidence interval interpreter

```r
# 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: NOPE! This error message should be improved.")
  } else if(!is.numeric(upper)) {
    # produce a warning if upper isn't numeric
    warning("Warning: NOPE! This error message should be improved.")
  } else if(!is.numeric(ci_level) | ci_level < 0 | ci_level > 100) {
    # produce a warning if ci_level isn't appropriate
    warning("Warning: Your ci_level is wrong. This error message should be improved.")
  } else{
    # print interpretation

  str_c("We have ", ci_level,
        "% confidence that ", stat,
        "is bewteen ", lower, " and ", upper,
        ". " )
  }
}

# Test 1
ci_test1 <- interpret_ci(10, 20, 99, "mean number of shoes owned by students")
str_c("CI function test 1: ",ci_test1)
```

```
## [1] "CI function test 1: We have 99% confidence that mean number of shoes owned by studer
```

```
# Test 2
ci_test2 <- interpret_ci(10, 20, -1, "mean number of shoes owned by students")
str_c("CI function test 2: ",ci_test2)
```

```
## [1] "CI function test 2: Warning: Your ci_level is wrong. This error message should be im
```

```
# Test 3
ci_test3 <- interpret_ci(10, 20, 95, 99)
str_c("CI function test 3: ",ci_test3)
```

```
## [1] "CI function test 3: Warning:stat should be a character string that describes the sta
```

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
    ↪  statistics of interest.")
  } else if(!is.numeric(pval)) {
    warning("You p value should be a number.")
  }  else if(pval > 1) {
    warning("Warning: The range of p value should be between 0 and 1.")
  } else if(pval < 0){
    warning("Warning:he range of p value should be between 0 and 1.")
  } else if(pval > 0.05){
    str_c("The p value is ", round(pval, 3),
                ", we fail to reject the hyothesis that ", nullhyp)
  } else if(pval <= 0.05){
    str_c("The p value is ", round(pval, 3),
                ", we reject the hypothesis that ", nullhyp, ".")
  }
}

pval_test1 <- interpret_pval(0.0000000003,
                             "the mean grade for statistics students is the same as
                             ↪  for non-stats students")
str_c("p value function test 1: ",pval_test1)
```

```
## [1] "p value function test 1: The p value is 0, we reject the hypothesis that the mean gr
```

```
pval_test2 <- interpret_pval(0.0499999,
                             "the mean grade for statistics students is the same as
                             ↪  for non-stats students")
str_c("p value function test 2: ",pval_test2)
```

```
## [1] "p value function test 2: The p value is 0.05, we reject the hypothesis that the mean
```

```
pval_test3 <- interpret_pval(0.050001,
                             "the mean grade for statistics students is the same as
                             ↪  for non-stats students")
str_c("p value function test 3: ",pval_test3)
```

```
## [1] "p value function test 3: The p value is 0.05, we fail to reject the hyothesis that t
```

```
pval_test4 <- interpret_pval("0.05", 7)
str_c("p value function test 4: ",pval_test4)
```

```
## [1] "p value function test 4: Warning: nullhyp should be a character string that describe
```

Task 3c: User instructions and disclaimer

Instructions

A confidence interval interpreter is a function that helps you interpret a frequentist confidence interval with the appropriate language. A confidence interval displays the probability that a parameter will fall between a pair of values around the mean. Confidence intervals measure the degree of uncertainty or certainty in a sampling method. They are most often constructed using confidence levels of 95% or 99%. You can use some equation to compute confidence interval. Then enter the results into the the confidence interval interpreter.The output is the interpretion of a frequentist confidence interval. The function is interpret_ci(lower, upper, ci_level, stat) The meaning of 4 arguments: – lower, the lower bound of the confidence interval (numeric) – upper, the upper bound of the confidence interval (numeric) – ci_level, the confidence level this interval was calculated at, e.g. 99 or 95 (numeric) Confidence level refers to the percentage of probability, or certainty, that the confidence interval would contain the true population parameter when you draw a random sample many times. ci_level must be an numeric value between 0 and 100. – stat a description of the statistic of interest. There are some examples: ci_1 <- interpret_ci(10, 20, 99, “mean number of shoes owned by students”) ci_2 <- interpret_ci(1, 50, 95, “mean number of books bought by students”) The expected output is : ci_1: We have 99% confidence that mean number of shoes owned by students is bewteen 10 and 20. " ci_2:We have 95% confidence that mean number of books bought by students is bewteen 1 and 50.

A p value interpreter is a function that will interpret p values based on strength of evidence. The function is interpret_pval(pval, nullhyp) The meaning of 4 arguments: -pval, it must be a numeric. A p-value is a measure of the probability that an observed difference could have occurred just by random chance. The lower the p-value, the greater the statistical significance of the observed difference. -nullhyp, a description of the statistic of interest. The null hypothesis is a typical statistical theory which suggests that no statistical relationship and significance exists in a set of given single observed variable, between two sets of observed data and measured phenomena. There are some examples: pval_1 <- interpret_pval(0.03, “the mean grade for statistics students is the same as for non-stats students”) pval_2 <- interpret_pval(0.6, “the

mean grade for statistics students is the same as for non-stats students") The expected output: The p value is 0.03, we reject the hypothesis that the mean grade for statistics students is the same as for non-stats students. The p value is 0.6, we fail to reject the hypothesis that the mean grade for statistics students is the same as for non-stats students.

Disclaimer

If you require any more information or have any questions about our site's disclaimer, please feel free to contact us by email at 123456789@gmail.com.

All the information on this interpreter is published in good faith and for general information purpose only. The interpreter is not allowed for illegal or commercial use. And the person using the interpreter needs to have a certain understanding of the basic concepts of statistics.Any inappropriate content generated by the interpreter is the sole responsibility of the user. The person using the interpreter already knows by default all the rules and contents of the interpreter.

Consent By using our interpreter, you hereby consent to our disclaimer and agree to its terms.

Update Should we update, amend or make any changes to this document, those changes will be prominently posted here.

Task 4: Creating a reproducible example (reprex)

If we need to upload our code online for display, or seek help from others to find mistakes. We need a better representation of our code to help others be able to better understand our code. We should use the simplest and build- in data possible. That way, you don't let others misunderstand the message your code is sending. Reduce extraneous information in your code and keep only the code that needs to be run. Use good coding style.This way others will help you faster and more comfortably. You are asking people to run this code. So,don't use code that can no longer run on other people's computers, or that harms someone else's workspace.

```
# load the packages needed
library(tidyverse)
library(reprex)
# Creat the data
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))
head(my_data)
#[1]1   16
#[2]1   18
#[3]1   19
#[4]1   15
#[5]1   15
#[6]1   23
# summarize the dataset my_data into a new dataset my_summary and find the mean value
↪  for each group
my_summary <- my_data %>%
summarize(group_by = group, mean_val = mean(value))
head(my_summary)
#   group_by  mean_va<dbl>
#[1]   1           19.67
#[2]   1           19.67
#[3]   1           19.67
#[4]   1           19.67
#[5]   1           19.67
#[6]   1           19.67
# Check the data whether has one row per group
glimpse(my_summary) )
#Rows: 100
#Columns: 2
#$ group_by <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, ..
#$ mean_val <dbl> 19.67, 19.67, 19.67, 19.67,..
```

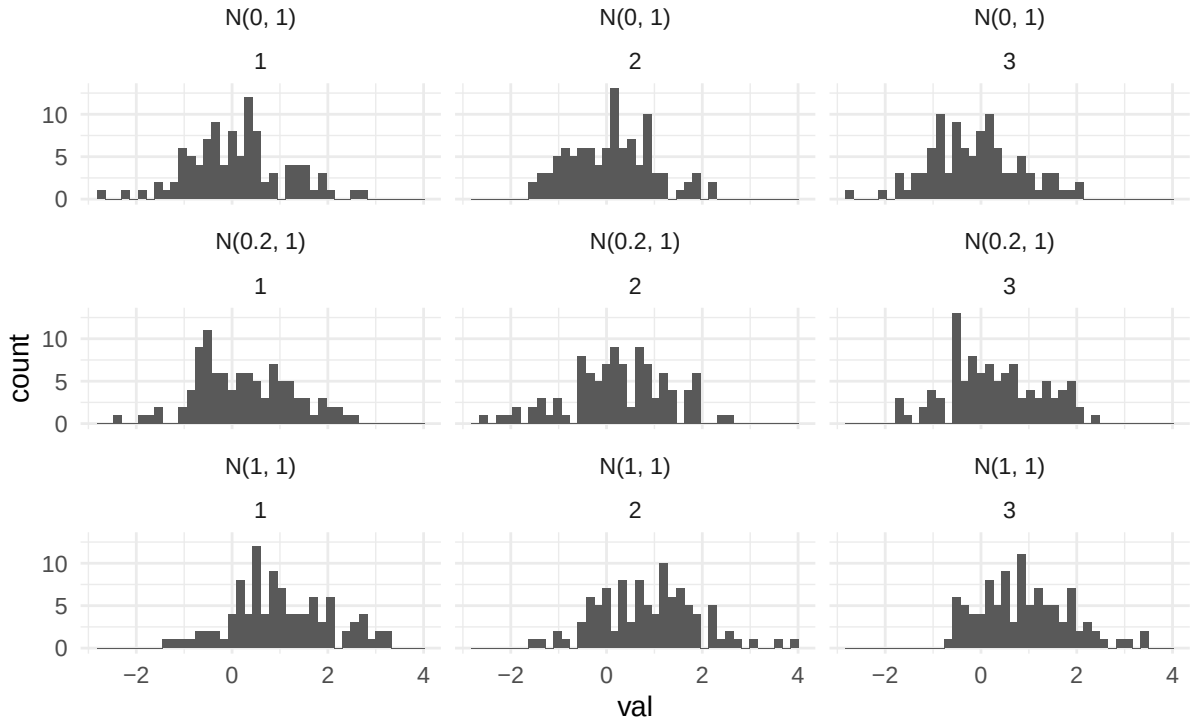
Task 5: Simulating p-values

Setting up simulated data

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

```
# Creat sim1,sim2,sim3
sim1<-tibble(group = rep(1:1000, each = 100),val = rnorm(100000,0,1))
sim2<-tibble(group = rep(1:1000, each = 100),val = rnorm(100000,0.2,1))
sim3<-tibble(group = rep(1:1000, each = 100),val = rnorm(100000,1,1))
#Stack all 4 datasets into one new dataset
all_sim<-bind_rows(sim1, sim2, sim3, .id = "sim")
#  Create sim_description
sim_description <- tibble(sim = 1:3,desc = c("N(0, 1)","N(0.2, 1)","N(1, 1)"))
# all_sim joining on the dataset sim_description
all_sim$sim<-as.numeric(all_sim$sim)
all_sim<-inner_join(all_sim,sim_description)
# Plot histograms for the first three groups for each simulated dataset in one plot
all_sim %>% filter(group <= 3) %>% ggplot(aes(x = val)) + geom_histogram(bins = 40) +
↪  facet_wrap(desc~group, nrow = 3,ncol = 3) + theme_minimal() + labs(title = "Figure
↪  2: Target first visualisation for task 5",caption = "Created by Tianye Wang in
↪  STA303/1002, Winter 2022")
```

Figure 2: Target first visualisation for task 5



Created by Tianye Wang in STA303/1002, Winter 2022

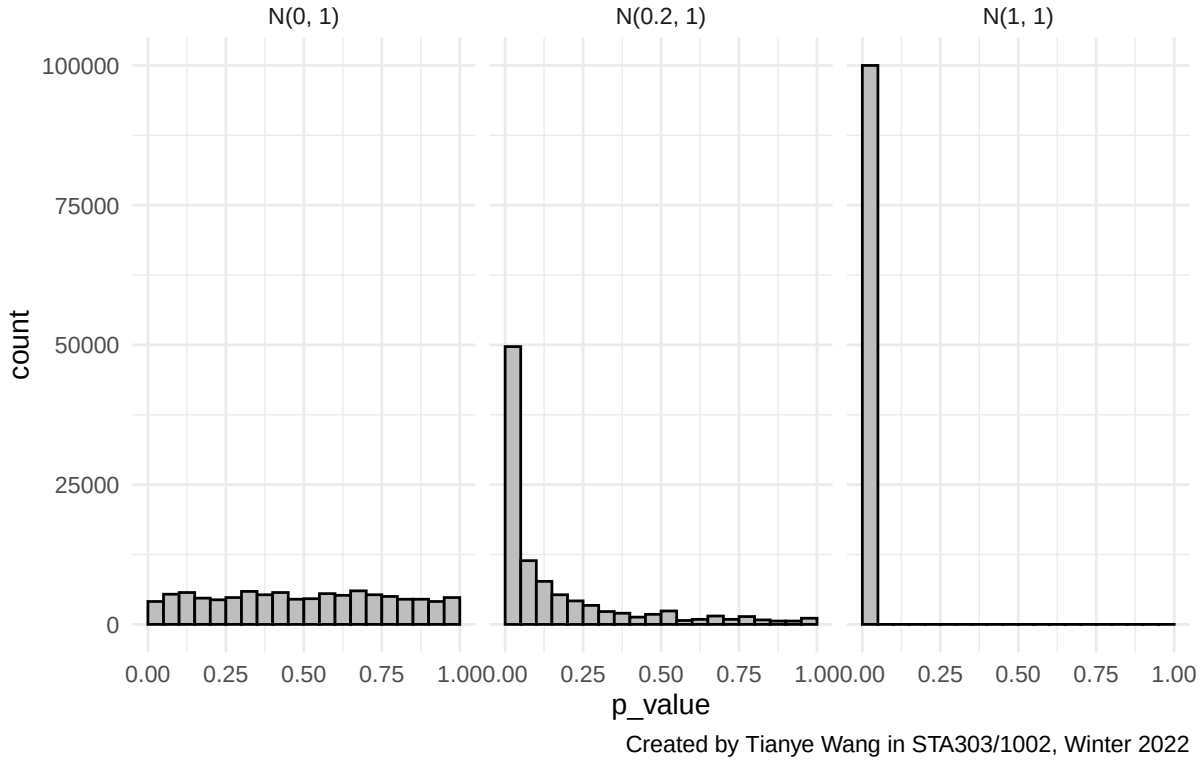
Calculating *p* values

```
# Creat p_value table
library(dplyr)
pval<- all_sim %>% group_by(desc,group)
pval$group<-as.numeric(pval$group)
t_test_desc1<-1:1000
for (i in 1:1000){
  df<-pval %>% filter(group == i,desc == "N(0, 1)")
  t_test1<-t.test(df$val, mu = 0)
  t_test_desc1[i]<-t_test1$p.value
}
t_test_desc2<-1:1000
for (i in 1:1000){
  df<-pval %>% filter(group == i,desc == "N(0.2, 1)")
  t_test1<-t.test(df$val, mu = 0)
  t_test_desc2[i]<-t_test1$p.value
}
```

```
t_test_desc3<-1:1000
for (i in 1:1000){
  df<-pval %>% filter(group == i,desc == "N(1, 1)")
  t_test1<-t.test(df$val, mu = 0)
  t_test_desc3[i]<-t_test1$p.value
}
table1<-tibble(group = 1:1000,desc = "N(0, 1)",p_value = t_test_desc1)
table2<-tibble(group = 1:1000,desc = "N(0.2, 1)",p_value = t_test_desc2)
table3<-tibble(group = 1:1000,desc = "N(1, 1)",p_value = t_test_desc3)
all_table<-bind_rows(table1, table2, table3, .id = "table")
df<-inner_join(pval,all_table)
df<-df[,-5]
```

```
# Provide an appropriate figure caption
df %>% ggplot(aes(x = p_value)) + geom_histogram(boundary = 0, binwidth = 0.05, fill =
↪  "grey", color = "black") + facet_wrap(~desc) + theme_minimal() + labs(title =
↪  "Figure 3: Target second visualisation for task 5",caption = "Created by Tianye
↪  Wang in STA303/1002, Winter 2022")
```

Figure 3: Target second visualisation for task 5



Drawing Q-Q plots

```
# Create one final plot that creates a 2x2 figure with QQ plots for each simulation
df %>%
ggplot(aes(sample = p_value)) +
geom_qq(distribution = qunif) +
geom_abline(intercept = 0, slope = 1) +
facet_wrap(~desc) +
theme_minimal() +
labs(title = "Figure 4: Target final visualisation for task 5",caption = "Created by
↪  Tianye Wang in STA303/1002, Winter 2022")
```

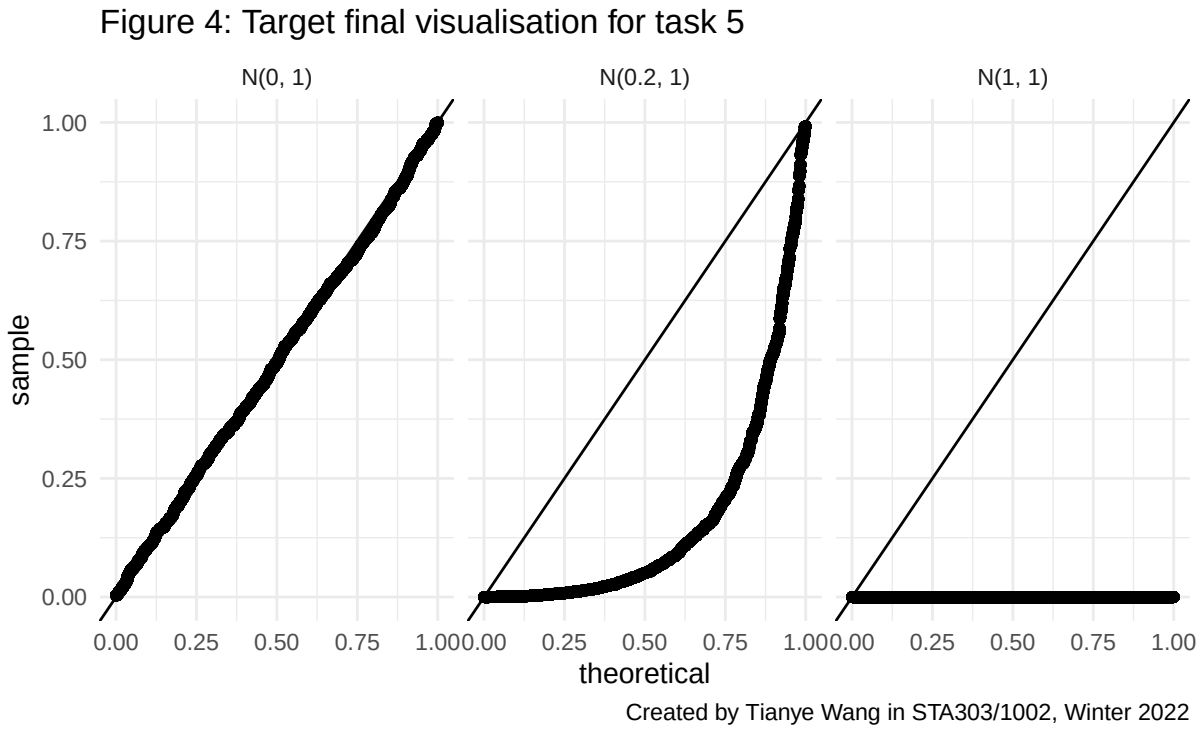


Figure 1: This is a figure caption

Conclusion and summary

The definition of a p value In statistics, the p-value is the probability of obtaining results at least as extreme as the observed results of a statistical hypothesis test, assuming that the null hypothesis is correct. The p-value is used as an alternative to rejection points to provide the

smallest level of significance at which the null hypothesis would be rejected. A smaller p-value means that there is stronger evidence in favor of the alternative hypothesis.

My task I Create a new dataset called pvals that starts with all_sim, groups by both desc AND group, and then use "for" function to summarizes to find the p value for each group and each desc. Creat table1,table2,table3 for each desc and group.Then I combine the row of three tables. Creat a new table called all_table. Then I inner join the all_table and pval. Creat a new table called df. Then use the data in df to have a test based on a one sample, 2-sided t.test, where the null hypothesis is the the population mean is 0. is zero. Then I use ggplots to have a visualisation of the distribution of p value for each desc. It gives me three histograms. After that I also use the data to make a qq plot. And I find that the distribution of desc ="N(0,1)" conforms to the uniform distribution in qq polt. Therefore we can answer pre-knowledge check. The answer is that approximately 10% pf the p-values will be between 0.9 and 1 because the distribution of p value is uniform.

Writing sample

Introduction This article raises the question that only a small fraction of most published articles can be reproduced.The authors believe that one of the most likely reasons is that most of the authors of the article are not very clear about many statistical concepts.The author proposes five common misconceptions about statistics and data analysis.

P hacking Many people increase the proportion of data test results less than 0.05 by increasing the size of the data set, or by testing multiple hypothesis tests. This proves that the results are significant. However, the results obtained through such behavior are biased, and do not prove that the results obtained by the data are scientific.So the author makes two suggestions to make the experiment more scientific. The author states that For each figure or table, clearly state whether or not the sample size was chosen in advance, and whether every step used to process and analyze the data was planned as part of the experimental protocol. If you use any form of P-hacking, label the conclusions as "preliminary."(Harvey J. Motulsky 2014).

P values convey information about effect size The p-value varies with the sample size. This situation causes two kinds of problems: A large P value is not proof of no (or little) effect;A small P value is not proof of a large effect(Harvey J. Motulsky 2014) For the same experimental data, the way the data is processed is different, resulting in a very large range of P-value changes. So neither a large P-value nor a very small P-value can reflect any problem. Therefore the author states that Always show and emphasize the effect size (as difference, percent difference, ratio, or correlation coefficient) along with its confidence interval.Consider omitting reporting of P values. This approach can help authors make their papers more rigorous. For third misconception authors who write papers always emphasize that the data must be significant, so the authors have a certain bias against significance.The author hopes not to always emphasize that the data is significant, and it is easy to cause psychological implications to the reader. For fourth misconception,the authors call for not always focusing on the standard deviation of the data. For fifth misconception,the authors believe that too much experimental detail should not be elaborated in the article.

Conclusion

After reading this article, I learned that some common sense things are not necessarily accurate. And I learned how to make my experimental data more accurate. The author also taught me some good ways to write a report. These methods will make the reader more comfortable with the entire experimental process of my report.

Word count: 433 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?

A lot of the code used in R is unfamiliar to me, and I need to consult a lot of information to understand the purpose of this code. There are also some models of data that I am not familiar with, and I also need to read a lot of literature to understand the model and the purpose of the model.After this learning process, I felt very proud when I typed out the code through my own understanding. I learned something that worked for me. And research knowledge makes me happy.And every time I change my code over and over again, and then my result is closer to the teacher's result, until I get the same result as the teacher, it feels very good.

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

When making an interpreter, I couldn't help but enter the appropriate code and the appropriate interpretation statement. I also need to write some notes on the interpreter for a novice in statistics, as well as a disclaimer.After doing these things, I clearly realized that many of the things I did not only need to be understood by myself, but also output corresponding results. I also need to try to make people understand what I'm doing and be able to really help some beginners. I need to think about it all from the perspective of the user of what I'm making.I should also have this awareness when I am engaged in work in the future. As far as possible, let people without statistical background understand the meaning of some statistics, and the meaning of the corresponding professional results.

What is something I'd do differently next time?

On some visual output images, I should try to make the pictures look as beautiful as possible. And the information shown in the pictures is more explicit. In terms of language, it should be more professional and make it easier for people without a statistical foundation to understand.And I'm not particularly familiar with reporting. The code in R is still very rusty. I need to strengthen my ability to write code and write reports.