Math650 Homework 5

Yu Huang

October 31, 2023

Abstract

1 Introduction

Use rank-sum test to study the effect of group therapy on survival of breast cancer patients (chapter 4, question 31).

2 Materials and Methods

Data comes from question 31 of chapter 4. Methods: Wilcoxon rank-sum test, permutation. R codes are available in appendix 5.

3 Results

Result table.

W(rank sum statistic for the therapy group)	479
p-value of the rank sum statistic	0.265
0.95 confidence interval for the difference	-4 to 28
W calculated manually	1074
z-statistic for the normal approximation	1.1239
p-value for the z-statistic(two-sided)	0.262
p-value of W after 10,000 permutations(one-sided)	0.131

Based on the big p-value (0.265), the effect of the group therapy treatment on survival time is NOT statistically significant. We can't for sure there's indeed evidence. We have 95% confidence that a patient can be expected to live from 4 months shorter to 28 months longer though it's useless.

Problem here is that manual calculation based on page 93 gives slightly different result(the middle part in the table), especially the rank sum statistic.

The permutation test gives a very close one-sided p-value, 0.131.

4 Conclusion and Discussion

The slight difference regarding the wilcoxon test p-value is due to the way R handles ties. R has a different way to do that from the book on page 93.

5 Appendix

```
#2006-09-24, chapter 4, question 31

wilcox_test_on_data = function(data)
{
wilcox.test(SURVIVAL~GROUP, data=data, conf.int=TRUE)

control_d = data[data$GROUP=="CONTROL",]$SURVIVAL
therapy_d = data[data$GROUP=="THERAPY",]$SURVIVAL

wilcox.test(control_d, therapy_d, conf.int=TRUE)
w_result = wilcox.test(therapy_d, control_d, conf.int=TRUE)
cat("wilcox test p-value:", w_result$p.value, "\n")
cat("wilcox test rank sum statistic:", w_result$statistic, "\n")
cat("wilcox test confidence interval:", w_result$conf.int, "\n")

#calculate the z_stat
rank_l = rank(c(control_d, therapy_d))
no_of_controls = length(control_d)
no_of_therapies = length(therapy_d)

w = sum(rank_l[(no_of_controls+1):(no_of_controls+no_of_therapies)])
cat("w calculated manually:", w, "\n")
avg_rank = mean(rank_l)
sd_rank = sd(rank_l)
z_stat = (w-avg_rank*no_of_therapies)/(sd_rank*sqrt(no_of_controls*no_of_therapies/(no_of_
cat("z_stat:", z_stat, "\n")
#through z_stat's normal approx. to manually calculate wilcoxon test p-value
z_stat_normal_p_value = pnorm(z_stat, lower.tail=FALSE)*2
cat("z_stat_normal_p_value:", z_stat_normal_p_value, "\n")
}

permutation_test = function(input_data, no_of_controls, no_of_samplings=10000)
{
stat_list = rep(1, no_of_samplings)
no_of_data = length(input_data)
for (i in seq(no_of_samplings))
{
sampled_data = sample(input_data)
rank_l = rank(sampled_data)
w = sum(rank_l[(no_of_controls+1):no_of_data])
stat_list[i] = w
}
return (stat_list)
}

get_confidence_interval = function(therapy_d, control_d, list_of_values_to_be_tested, trun
{
no_of_controls = length(control_d)
```

```
no_of_therapies = length(therapy_d)
for (i in list_of_values_to_be_tested)
{
new_therapy_d = therapy_d + i
for (j in seq(length(new_therapy_d)))
{
if (new_therapy_d[j]>truncate_limit)
{
new_therapy_d[j] = 122
}
}
rank_l = rank(c(control_d, new_therapy_d))
w = sum(rank_l[(no_of_controls+1):(no_of_controls+no_of_therapies)])
avg_rank = mean(rank_l)
sd_rank = sd(rank_l)
z_stat = (w-avg_rank*no_of_therapies)/(sd_rank*sqrt(no_of_controls*no_of_therapies/(no_of_
z_stat_normal_p_value = pnorm(z_stat, lower.tail=FALSE)
cat(i, "\t", z_stat_normal_p_value, "\n")
}
}


data = read.csv("/usr/local/doc/statistical_sleuth/ASCII/ex0431.csv")
wilcox_test_on_data(data)

#get the p-value from permutation
no_of_samplings = 1e4
stat_list = permutation_test(data$SURVIVAL, no_of_controls, no_of_samplings)
perm_p_value= sum(stat_list>=w)/no_of_samplings
cat("permutation p-value:", perm_p_value, "\n")

get_confidence_interval(therapy_d, control_d,seq(-30,20))
```