

ADS2 Coding Challenge 1

Semester 1, 2023-24

Review Lecture

1. Benefits of swimming for long-distance runners

Long-distance runners don't just train for their races by running, they also use a variety of other exercises to improve their performance. One such example is the use of swimming to improve strength and muscle mass. We tested this by selecting 50 runners from a local club which has in total 500 members. In this question, you will be presented with the time taken for these 50 runners to complete a half-marathon before and after they each completed the same swimming training programme. We want to know whether this swimming improved their times.

The data can be accessed from the file 'swimming.txt'. Each row contains the results from a single athlete with the total time for the half-marathon before swimming given in minutes and seconds described in 'before_minutes' and 'before_seconds'. The time taken after swimming is described in the same way for each swimmer in 'after_minutes' and 'after_seconds'.

Questions

- What would be a suitable statistical test for these data and why? Note you may need to tidy these data before deciding on which test to use.

– The minutes and seconds need to be combined (2 points). Either way shown below is equally valid.

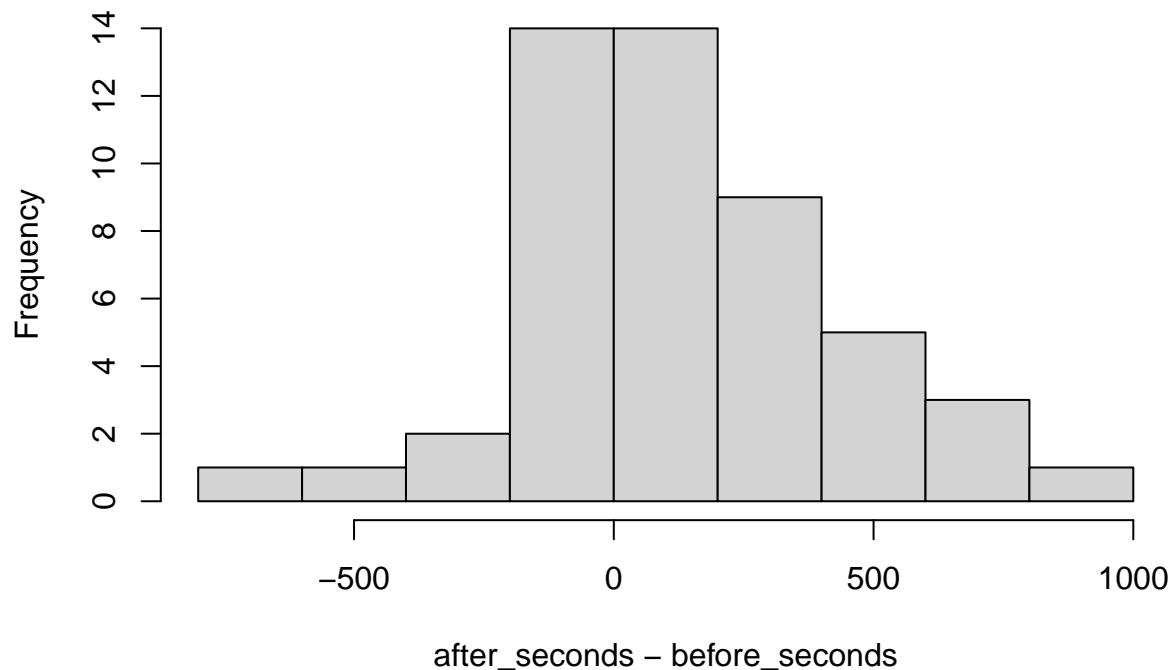
```
data<-read.table("swimming.txt", header = T)
before_seconds<-(data$before_minutes*60) + data$before_seconds
after_seconds<-(data$after_minutes*60) + data$after_seconds

before_times<-data$before_minutes + (data$before_seconds/60)
after_times<-data$after_minutes + (data$after_seconds/60)
```

– The main assumption that needs to be tested is that the data are normally-distributed. – You could inspect this distribution either visually with a histogram or using a Shapiro-Wilks test, either are appropriate. Note that a non-significant ($p > 0.05$) value for a Shapiro-Wilks test states that the underlying data can be assumed to be normally distributed. – As these data are in a paired format, we will (later) be using a paired test or a one-sample test on the differences between the before and after times. Therefore, we do not need to check that both distributions are normally distributed but that the difference between the times is normal.

```
hist(after_seconds-before_seconds)
```

Histogram of after_seconds – before_seconds



```
shapiro.test(after_seconds-before_seconds)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: after_seconds - before_seconds  
## W = 0.96742, p-value = 0.1814
```

– Other assumptions can be observed (data are continuous) or assumed (random sampling or independence of mean/standard error. No need to test these directly.

– You should use a T-test (2 points), with the paired setting (2 points). This test should be a one-tailed test as we are only asking if it *improves* running time

- What are your null and alternative hypotheses? (4 points)

– Null hypothesis: there is no difference in running time/speed after swimming (2 points)

– Alternative hypothesis: Running time decreases (or speed increases) after swimming (2 points).

– You could also consider an alternative hypothesis that there is just a difference in time/speed after swimming. This would be the equivalent of a two-sided test and would not have been awarded full marks. Note that my example analyses are one-tailed.

- Is there a statistically significant improvement on runners' times after swimming? (10 points)

```
t.test(before_times, after_times, paired = T, alternative = "less")
```

```
##  
## Paired t-test  
##  
## data: before_times and after_times  
## t = -2.9199, df = 49, p-value = 0.002639  
## alternative hypothesis: true mean difference is less than 0
```

```
## 95 percent confidence interval:
##      -Inf -0.8648518
## sample estimates:
## mean difference
##      -2.031

t.test(before_times, after_times, paired = F, alternative = "less")

##
## Welch Two Sample t-test
##
## data: before_times and after_times
## t = -0.95644, df = 81.458, p-value = 0.1708
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 1.502031
## sample estimates:
## mean of x mean of y
## 127.2407 129.2717
```

– Only the paired t-test show a significant result ($p < 0.05$). – Other problems came from changing the order of before_times/after_times or setting the alternative to “greater” (how might you have checked this before?). A two-tailed test would also have reported a significant decrease in time but would not have been awarded full marks.

2. Number of emergency room admissions

Dr. Horsey leads a hospital. She wants to better understand the number of people admitted to the hospital’s emergency room, in order to be able to plan staff schedules. In particular, she wants to know whether there is a difference in emergency room admissions between weekends and weekdays. For this purpose, she looks at admissions data for the afternoon shift (eight hours from 1pm to 9pm) of every Monday (weekday) and every Sunday (weekend) for an entire year. The data collected is the number of patients admitted during that hour.

The dataset is provided in file `hospital_admissions.csv`. It contains columns recording the week (numbered from 1 to 52), day (Monday or Sunday), hour (numbered 1 to 8) and number of patients admitted during that hour.

Questions

- Import the dataset and plot the data in a useful way. (10 points)

– correct data import

```
library(ggplot2)
admissions = read.csv('hospital_admissions.csv')
```

– processing the data to calculate attendances per day. Important to note that the data was recorded per hour and therefore had to be summed over each day to get a sensible value.

```
weeks = unique(admissions$week)

sundata = admissions[admissions$weekday == "Sunday",]
monddata = admissions[admissions$weekday == "Monday",]

sunsums = {}
monsums = {}
```

```

for(week in weeks) {
  sunsums[week] =
    sum(admissions$patients_per_hour[admissions$week == week & admissions$weekday == "Sunday"])
  monsums[week] =
    sum(admissions$patients_per_hour[admissions$week == week & admissions$weekday == "Monday"])
}

mean(monsums)

## [1] 33.55769

mean(sunsums)

```

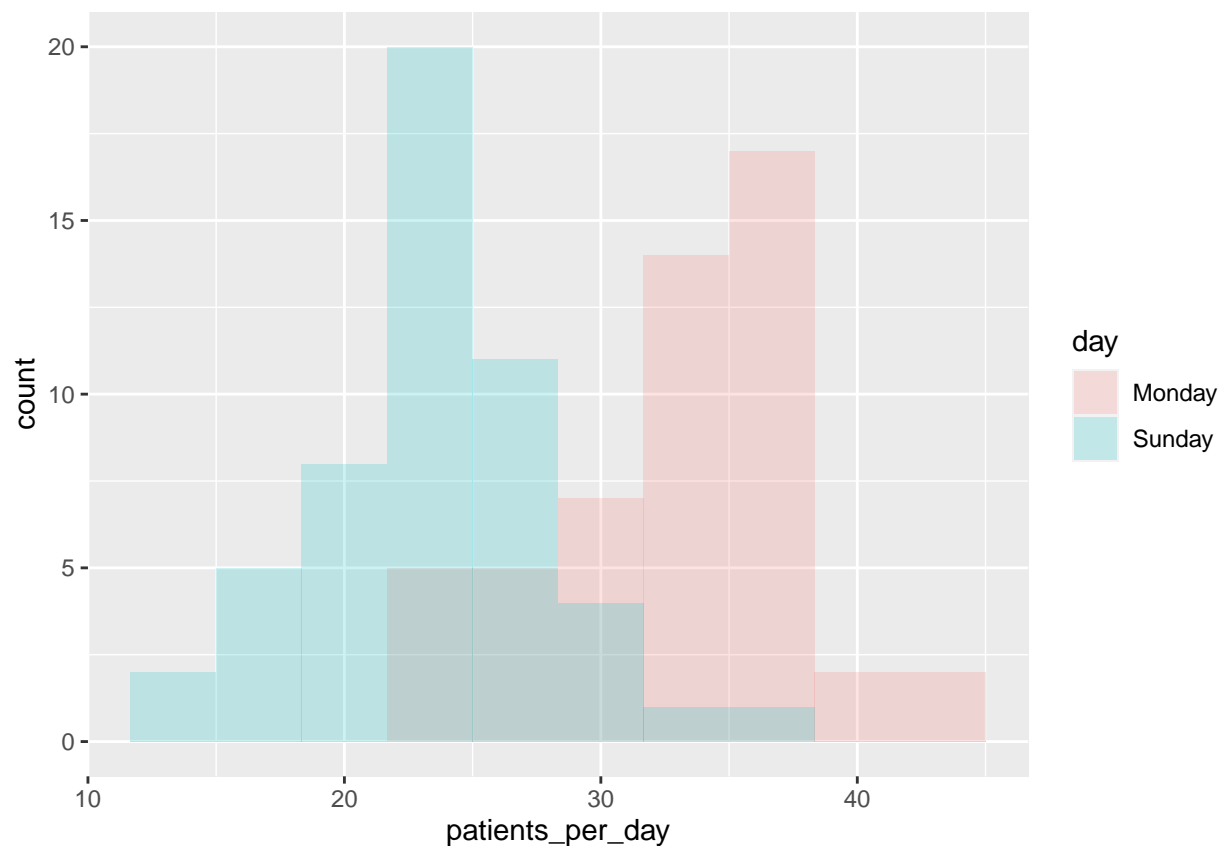
```
## [1] 23.51923
```

– useful plot of data such as overlaid histograms. There are lots of other ways this could be plotted, e.g. boxplot, CDF. Which do you prefer?

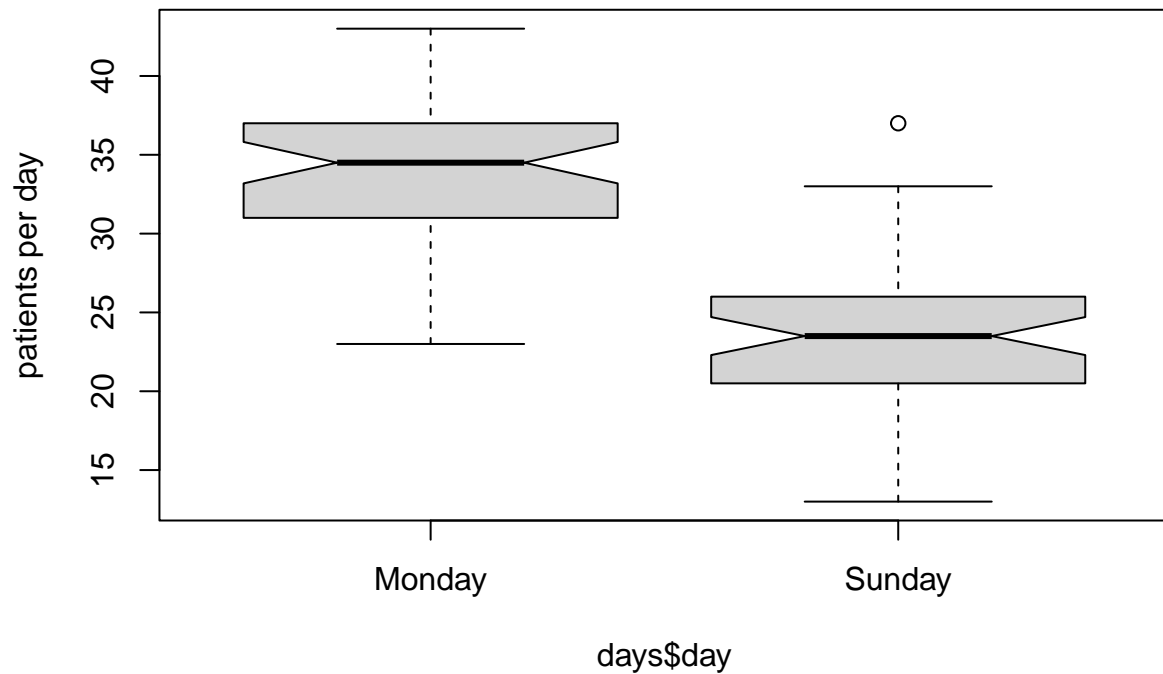
```

days = data.frame(patients_per_day = c(monsums, sunsums),
  day = c(rep("Monday", length(weeks)), rep("Sunday", length(weeks))))
ggplot(days, aes(x = patients_per_day, fill = day)) +
  geom_histogram(position = "identity", alpha = 0.2, bins = 10)

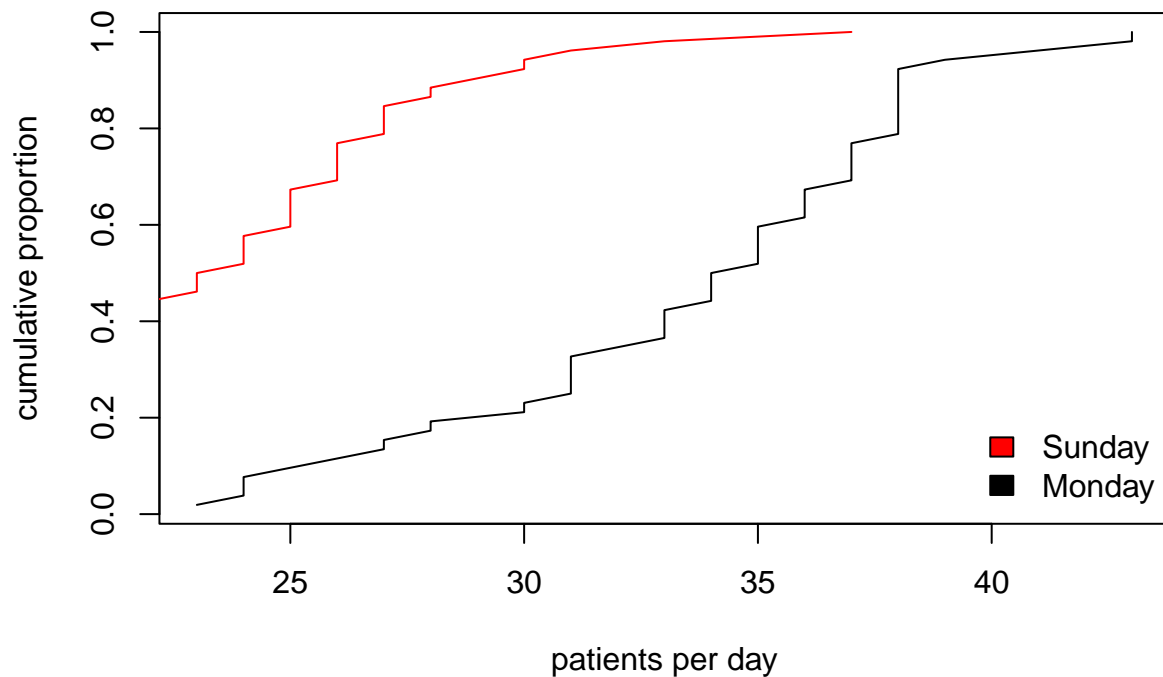
```



```
boxplot(days$patients_per_day~days$day, notch = T, ylab = "patients per day")
```



```
plot(sort(monsums), 1:length(monsums)/length(monsums), type = 'l', ylab = "cumulative proportion", xlab = "patients per day", col = 'black')
lines(sort(sunsums), 1:length(sunsums)/length(sunsums), col = 'red')
legend('bottomright', legend = c("Sunday", "Monday"), fill = c('red', 'black'), bty = 'n')
```



- Is there a difference in patient admission rates between Mondays and Sundays?
- state hypotheses for testing: H_0 there is no difference between the number of patients per day on Mondays and Sundays. H_a there is a difference between the number of patients per day on Mondays and Sundays.
- justify choice of test method

```
shapiro.test(monsums)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: monsums  
## W = 0.95206, p-value = 0.03564
```

```
shapiro.test(sunsums)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: sunsums  
## W = 0.987, p-value = 0.838
```

Data doesn't meet conditions for t-test, could use a Mann-Whitney or Wilcoxon rank sum test (its the same test but with different names). Alternatively (but nobody did it) you could use simulation to compare means.

– correctly implement a valid test

```
wilcox.test(monsums, sunsums, paired = F)
```

```
##  
## Wilcoxon rank sum test with continuity correction  
##  
## data: monsums and sunsums  
## W = 2492.5, p-value = 1.159e-13  
## alternative hypothesis: true location shift is not equal to 0
```

– present correct answer. There is a clear statistically significant result (at $p < 0.05$) where there is a significant difference between the number of patients per day on Mondays and Sundays. Monday sees a significantly greater number of patients.

- Based on your findings, what advice would you give Dr. Horsey?

– Due to clear statistical significance, Dr Horsey should increase the number of staff on weekdays while also decreasing the number of staff at the weekend (when there are fewer patients and therefore fewer staff required).

3. Spinal cord injury and novel biomaterials

Complete spinal cord injury (SCI) is a severe condition caused by the trauma of the spinal cord that leads to its breakage. Depending on the site and type of injury, it may lead to the loss of motor and sensory functions and disability. The common prognosis is bad, with less than 30% of patients being able to regain at least some functions over the affected body parts spontaneously within 1 year. The commonly accepted clinical scale to describe the patient condition is American Spinal Injury Association Impairment Scale (AIS scale), which has 5 grades: 'A' (the complete motor and sensory functional impairment, complete disability), 'B', 'C', 'D', and 'E' (the above-mentioned functions are normal, healthy subject).

Your team has developed a novel biomaterial that may help to regenerate injured nervous tissue after SCI. Studies on animals showed promising results, and now you perform Phase I-II trial on patients with complete chronic SCI. Your team recruited patients with chronic SCI (at least 1 year since the trauma passed with no signs of recovery), recorded their condition upon admission, and installed implants with the novel material. 15-18 months later, you recorded patient AIS scores again.

The data is in the `SCI_before.csv` and `SCI_after.csv` files. Merge both tables appropriately, analyse the data, draw conclusions, and give further suggestions.

- Import, arrange the data (merge both pieces of data and make the data possible to analyse), and make it suitable for analysis, e.g. the values. You should perform all the manipulations in R and provide the code.

– The datasets are imported

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr    1.5.1
## v ggplot2    3.4.4      v tibble     3.2.1
## v lubridate  1.9.3      v tidyr      1.3.1
## v purrr      1.0.2
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()     masks stats::lag()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(dplyr)
```

```
# Import the data
```

```
SCI_data <- dir(pattern = "SCI+.*.csv")
```

```
SCI_task <- lapply(X = SCI_data, read.csv)
```

```
# Diagnose it
```

```
for(i in 1:length(SCI_task)){
  print(paste(c("SCI_task_", i), collapse = ""))
  print(head(SCI_task[[i]]))
  print("Any NA?")
  anyNA(SCI_task[[i]]) %>% print()
  print("Any duplicated rows?")
  sum(duplicated(SCI_task[[i]])) %>% print()
}
```

```
## [1] "SCI_task_1"
```

```
##   patient_ID AIS_after
```

```
## 1         13         A
```

```
## 2          9         C
```

```
## 3         24         A
```

```
## 4          4         A
```

```
## 5         11         B
```

```
## 6         16         A
```

```
## [1] "Any NA?"
```

```
## [1] FALSE
```

```
## [1] "Any duplicated rows?"
```

```
## [1] 8
```

```
## [1] "SCI_task_2"
```

```
##   patient_ID AIS_before
```

```
## 1          6         A
```

```
## 2          1         A
```

```
## 3         10         B
```

```
## 4         12         A
```

```
## 5         21         A
```

```
## 6         18         A
```

```
## [1] "Any NA?"
```

```
## [1] FALSE
## [1] "Any duplicated rows?"
## [1] 8
```

We check how the data looks like and observe that some entries are clearly duplicated. Our outcome variable is the AIS score, and the values are coded as letters. From the task description, we can see that it must be an ordinal scale that describes the disease severity with 5 ranks: $A > B > C > D > E$.

The data must be reformatted to have 3-4 columns: patient ID (*optional in this case*), clinical scores before and after the treatment (*also optional*), and the difference between both scores; or patient ID, measurement (before and after the treatment) and the respective clinical scores.

- The datasets are merged, ordinal values of the AIS score are converted to numeric values, and duplicated values are removed

```
# Merge data
# this line tells you what IDs are the same
SCI_task[[1]] <- SCI_task[[1]][match(SCI_task[[2]]$patient_ID,
                                     SCI_task[[1]]$patient_ID),]
# this line does column binding using the patient_ID from above, and then selects relative output
SCI_task <- do.call(cbind, SCI_task) %>%
  select(1,4,2) %>%
  arrange(patient_ID)
rownames(SCI_task) <- NULL
```

The duplicated rows are related to the same patients in both data frames, so we may just exclude them. Clinical scores can be converted as follows (for example):

Table 1: AIS scores and the disease severity.

AIS score	A	B	C	D
Disease severity	5	4	3	3

The data set may be in either format.

```
# remove duplicated
SCI_task <- SCI_task[!duplicated(SCI_task), ]

# convert scores
score2number <- function(x){
  if(x == "A"){
    output <- 5
  }else{
    if(x == "B"){
      output <- 4
    }else{
      output <- 3
    }
  }
  return(output)
}

# perform score2number on columns 2 and 3
# MARGIN = c(1,2) means apply to rows and columns)
SCI_task[, 2:3] <- apply(SCI_task[, 2:3], MARGIN = c(1,2), score2number)
SCI_task[, 2:3] <- apply(SCI_task[, 2:3], MARGIN = c(1,2), as.integer)
```



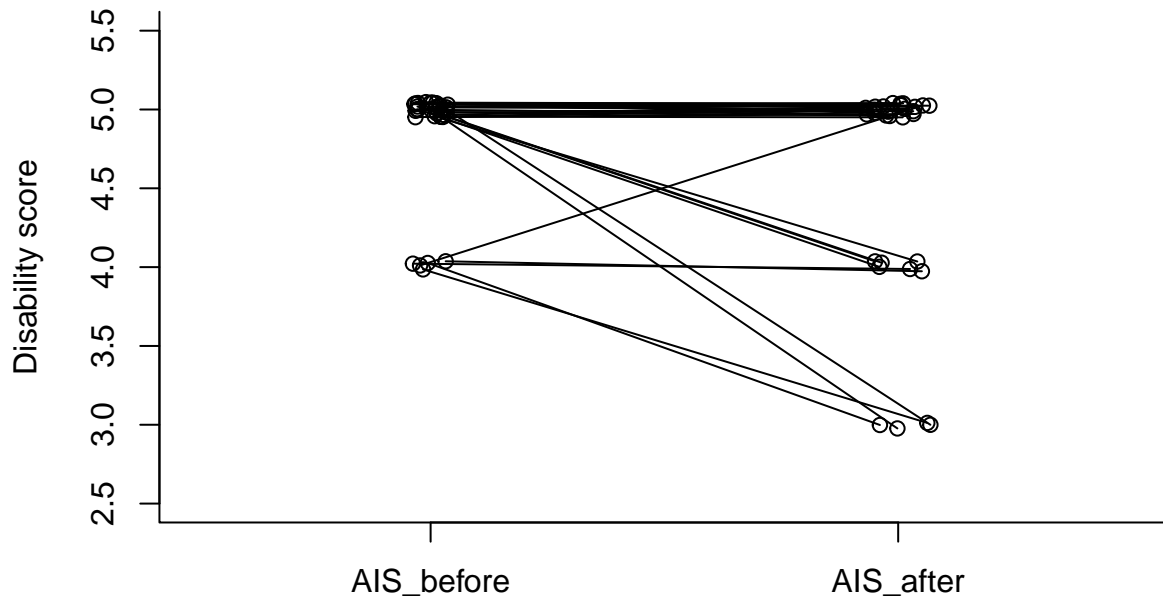
```
# Reshape
SCI_task <- SCI_task %>%
  gather(key = "Measurement", value = "AIS", 2,3) %>%
  mutate(Measurement = as.factor(Measurement))
```

- Check your data carefully. Identify features of the data and discuss your conclusions. Make illustrative plots. (6 points)

The data is ordinal, discrete, and paired. The data therefore also lacks normality. Let's make a graph:

```
# generate a nice data frame with x0 and x1 coordinates
data_to_plot <- spread(SCI_task, key = "Measurement", value = "AIS") %>%
  select(-1) %>%
  apply(MARGIN = c(1,2),
        FUN = jitter, factor = 0.5) %>%
  as.data.frame() %>%
  select(2,1) %>%
  mutate(x0 = 1, x1 = 2)
# add the jitter points to the x0 and x1 coordinates
colnames(data_to_plot)
```

```
## [1] "AIS_before" "AIS_after" "x0" "x1"
data_to_plot[, 3:4] <- apply(data_to_plot[, 3:4],
                             MARGIN = c(1,2),
                             FUN = jitter, factor = 2)
plot(x = data_to_plot[,3], y = data_to_plot[,1],
     xlim = c(0.5,2.5), ylim = c(2.5, 5.5),
     axes = F, xlab = "", ylab = "Disability score")
points(x = data_to_plot[,4], y = data_to_plot[,2])
segments(x0 = data_to_plot[,3],
         y0 = data_to_plot[,1],
         x1 = data_to_plot[,4],
         y1 = data_to_plot[,2])
axis(2)
axis(1, at = 1:2, labels = unique(SCI_task$Measurement))
box(bty = "l")
```



- Formulate the correct statistical hypothesis, choose the appropriate statistical test, and check assumptions for this test. Explain your choice briefly. Then, perform this test and identify whether the difference between the experimental groups is statistically significant.

H_0 is that the true median (or average) score change is 0, and H_1 is that it is not 0.

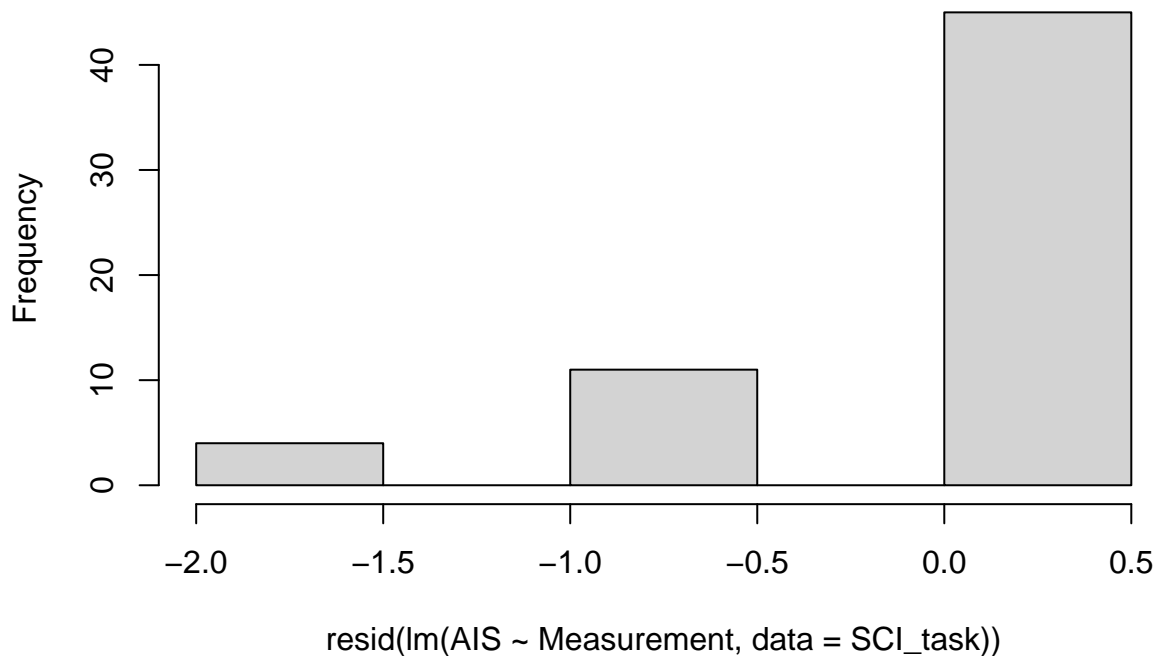
Normality check shows that the residuals from a linear regression of the data are not normally distributed:

```
shapiro.test(resid(lm(AIS ~ Measurement, data = SCI_task)))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(lm(AIS ~ Measurement, data = SCI_task))
## W = 0.74088, p-value = 5.939e-09
```

```
hist(resid(lm(AIS ~ Measurement, data = SCI_task)))
```

Histogram of resid(lm(AIS ~ Measurement, data = SCI_task))



The student should choose the paired U-test, which tests the equity of ranks between both groups. Paired Student's t-test is a suboptimal choice here (lack of normality). Other tests (unpaired) should yield 0 points unless explained.

The hypothesis test shows that the change in the disease severity is not equal to 0:

```
wilcox.test(AIS ~ Measurement, data = SCI_task, paired = T)
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with zeroes

##
## Wilcoxon signed rank test with continuity correction
##
## data: AIS by Measurement
## V = 4, p-value = 0.02459
## alternative hypothesis: true location shift is not equal to 0
```

- Discuss the data you got. What did you obtain? Are there any flaws in the experimental design and what would you suggest to your colleagues? Support your statements with the appropriate statistics and/or effect size estimates.

There are several flaws in this experiment that may be pointed out: 1. there is **no control group** (it would not be possible to add **sham-operated** persons anyway); 2. the information about the patient is **incomplete** as it **does not include details about their clinical picture** and **possible important clinical covariates** (age, the location of the trauma, their detailed case description upon admission, etc). You may comment about the need to **add other clinical covariates to predict the clinical outcome**. You may mention that the **condition of several persons** not only did not improve but even **deteriorated further**, and **this issue must be addressed**. A suggestion to increase the **sample size** will yield points only if it is supported by **power analysis**.