ADS2 Mock Coding Challenge 2

Semester 2, 2023-24

Technical Instructions

You have 3 hours to complete this assignment. There are **three** questions, all of which need to be completed. The instructions and data sets (.csv files) can be downloaded from Blackboard Learn.

Please make an R Markdown file for your response. A template is provided. Please follow the structure set out in the template. Please remember to include your roll number (but not your name) in the author field, as well as in the name of the final document.

The final submission is a pdf knitted from the R Markdown file (if you cannot knit to pdf directly, then knit to Word and convert the outcome to a pdf file using the "Export" function in Word or another text editor).

The submission should contain explanatory text, answers to questions, all results, and all the code used to generate the results. There is one exception: When you read a .csv file and if your name is in the file path, you are allowed to hide that code chunk so that your anonymity is maintained.

You will be graded not only on your answers to the questions but also on your ability to compile a well-formatted and readable R Markdown document. It is therefore advisable to knit early and often and check that your document can be knitted without errors and that the result is in line with your expectations. If you have code chunks that take a long time to run, use the code chunk option `cache = TRUE`. This means that the results of the code chunk get saved and will be used in the next knit, instead of being computed again (provided the code chunk has not changed).

Please upload your pdf file to the assessment dropbox at the end of the assignment. We are aware that due to increased traffic when everybody uploads their file, your upload may be a few minutes past the deadline. In such cases, we will consult the time at which the pdf document was produced and use this to determine whether or not your submission counts as a late submission. If so, the same penalties apply as for other in-course assessments.

Honour Code

This is an open-book assessment. This means you are allowed to consult your previous notes and use your previous code. You are also allowed to look up commands online if you need to (though the assessment is designed in such a way that you should not need commands or methods beyond what has been taught in this class). If you use code from an online source, please state what the source is (name of site, author if possible, url, date accessed).

You are **not** allowed to work with other students on this assessment. This is why we do not allow mobile phones. Of course, because we are allowing internet access, we cannot completely rule out the possibility of you working together. But we ask that you don't.

We appeal to your sense of honour and integrity. It is wrong to cheat, so don't do it.

By submitting this assignment, you declare that this is the result of your own work and that you did not either get help from or help, other students.

If, in marking the finished work, we find evidence that students have colluded, this will be treated as a potential violation of academic integrity and brought before the ZAMO.

Treatment for type 1 diabetes (T1D)

Your team developed an insulinomimetic that can be used to substitute insulin in patients with T1D. You induced T1D in mice by injecting them with streptozotocin (this drug destroys pancreatic β-cells, which abrogates insulin production and leads to the loss of control over glucose metabolism). Subsequently, mice were injected with vehicle or this novel drug in 2 possible concentrations. Their blood glucose level (mM) was measured before the injection and 1 hour after.

Answer the questions below and provide clear and reproducible code as well as your comments.

Questions

- Import, check, and organize the data appropriately. Treat data and reformat columns if needed.
- Plot the data in a useful way.
- Choose, justify, state the statistical hypotheses, and carry out an appropriate test to answer whether the drug is useful.
- Present and discuss your results. Is this novel drug useful? What would you suggest doing next?

Marking (25 points total)

Import, check, and organize the data appropriately. Treat data and reformat columns if needed.

Import, check, and organise the data appropriately. (3 points)

- The dataset is imported and briefly checked = 1 points.
- Rearrange and reformat columns, clean data = 2 points.

The dataset is imported –
1 points

The data are loaded and quickly checked:

```
t1d_drug <- read.csv(file = "t1d_drug.csv")
head(t1d_drug)
```

```
##   ID Treatment    Measurement Glucose Comment
## 1  1   Vehicle Glucose_before   11.51
## 2  2   Vehicle Glucose_before   10.95    Died
## 3  3   Vehicle Glucose_before    9.54
## 4  4   Vehicle Glucose_before   11.87
## 5  5   Vehicle Glucose_before    9.89
## 6  6   Vehicle Glucose_before   12.10
```

```
anyNA(t1d_drug)
```

```
## [1] TRUE
```

```
anyDuplicated(t1d_drug)
```

```
## [1] 0
```

By a brief examination, you may see the following features:

- there are *some* NAs;
- there are no duplicated rows;
- the data structure is not bad, but not exactly good:
  - Likely, the measurements are paired;
  - It is clearly **tidy**. But not exactly convenient. It may be better to put the *before* and *after* measurements along each other;
  - the dependent variable, `Glucose`, has some `NAs`;
  - There is a column called `Comment`. It is not empty.

Remember, we want you to show us your data cleaning maneuvers! You will be scored less if you do not show it.

As the data are paired, we may calculate the difference between both measurements. Also, we may convert `Treatment` to the ordered factor. Also, we need to check what rows contain `NAs`. It may be reasonable to exclude these rows, but we need to check first. For instance, it may be interesting to find out *which* columns include `NAs` and what else is in the respective part of the table.

```
##   ID Treatment Glucose_before Glucose_after Comment
## 1  1   Vehicle          11.51         12.24
## 2  2   Vehicle          10.95            NA    Died
## 3  3   Vehicle           9.54          7.98
## 4  4   Vehicle          11.87         11.60
## 5  5   Vehicle           9.89          9.09
## 6  6   Vehicle          12.10         12.15
```

R has many interesting functions. Apart from iterating through rows, you can use some of the `apply()` commands. These are often overlooked, and it is a shame. They can be useful when you need to run a certain function (usually, a non-standard function) over your table and get a quick response.

Anyway, after you check your table, you may see something interesting:

```
##   ID Treatment Glucose_before Glucose_after Comment
## 1  2   Vehicle          10.95          <NA>    Died
## 2  7   Vehicle          12.49          <NA>    Died
## 3  8   Vehicle          10.00          <NA>    Died
## 4  9   Vehicle          10.33          <NA>    Died
## 5 10   Vehicle           9.50          <NA>    Died
## 6 17   1 mg/ml           8.50          <NA>    Died
## 7 18   1 mg/ml          10.86          <NA>    Died
## 8 22   5 mg/ml          12.95          <NA>    Died
## 9 23   5 mg/ml           8.42          <NA>    Died
```

Seems, `NAs` are associated with the mice that died. Also, all of these dead mice are in the treatment groups. Itself, it is something worth discussing. But for now, we can drop these animals as they are not useful for the further analysis – we cannot pair them to anything.

```
t1d_drug <- t1d_drug %>%
  drop_na()
```

Finally:

- The data are imported and checked.

Rearrange and reformat columns –
2 point

- `NAs` are cleared;
- Grouping columns are set as factors;
- The data are converted to the structure that is suitable for analysis. Here, I decided to shift from the tidy format to the *convenient* one – it is wider, but it will make my life easier. See below how.

```
## 'data.frame':    21 obs. of  6 variables:
##  $ ID            : int  1 3 4 5 6 11 12 13 14 15 ...
##  $ Treatment     : Ord.factor w/ 3 levels "Vehicle"<"1 mg/ml"<..: 1 1 1 1 1 2 2 2 2 2 ...
##  $ Glucose_before: num  11.51 9.54 11.87 9.89 12.1 ...
##  $ Glucose_after : num  12.24 7.98 11.6 9.09 12.15 ...
##  $ Comment       : chr  "" "" "" "" ...
##  $ Difference    : num  -0.73 1.56 0.27 0.8 -0.05 ...
```

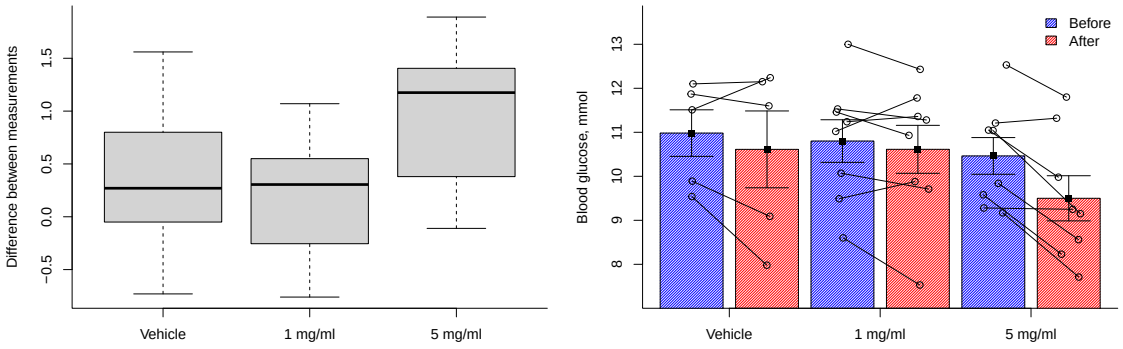
Now, it is fine. Relocation of the columns is optional, but it would be better to recode the independent variables.

Plot the data in a useful way. (5 points)

It is a pure place for all the possible creativity: you can use boxplots, whisker plots with mean or median +/- SE/SD/CI or IQR, or the primary points. The plot must:

- be available at all – +1 point.
- be informative: see above – +2 points.
- nicely formatted: clear labels, nice colors – +1 point.
- with the primary data points or exceptionally good formatting – +1 point.

Here is a reasonably good plot that may expect to get 4 points *at least* (Figure 1A) and a very sophisticated-to-draw plot that may expect to get 5 points (Figure 1B):



(a) A decent boxplot

(b) A nice plot, but unreasonably difficult to draw

Figure 1: Suitable plots

Another 1 point can be granted if the primary points are plotted. There is no need to make these boxes filled with different colors. In this case, these colors will not add you any particular information. One more issue why the plot in Figure 1B is a bit inferior choice is that if we analyze *the difference between measurements*, Figure 1B does not display it. But that is a detail.

Alternatives:

- Stripchart with the primary points and the mean or median value;
- Plot with the mean/median and SD/SE/95% CI or IQR, respectively;
- Plot both measurements and pair them. But it is really cumbersome.

And Figure 2 shows graph with bad labels. It is not *terribly* bad, but I would rather give it 3 points.

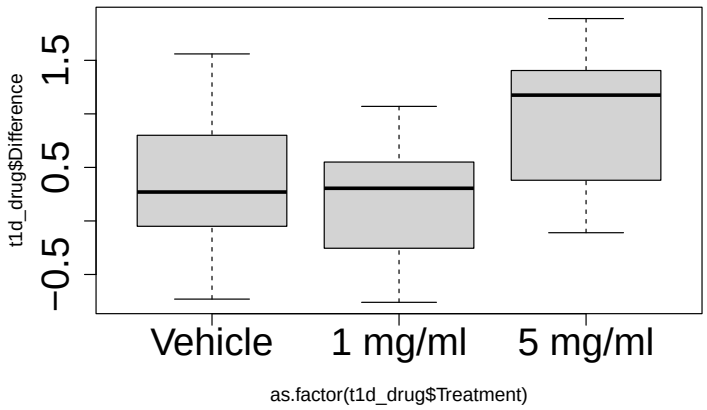


Figure 2: An *acceptable* plot

You can try other reasonable options as well!

Choose, justify, state the statistical hypotheses, and carry out an appropriate test to answer whether the drug is useful. (13 points)

Choose the appropriate test – 2 points

We have 3 levels of `Treatment` → 3 groups (6 groups if we decide to *not* calculate the difference between measurements). It means we need to use ANOVA. As we have 1 factor, we should try to use a 1-way ANOVA if the data fit the requirements. Thus, the method of choice is a 1-way ANOVA or (if cannot run it) the Kruskal-Wallis test. Even a better choice is to use a mixed-effects 2-way ANOVA (`Treatment` × `Measurement`), but this approach was not covered. So, we will rank both approaches equally if they are performed *correctly*. Guess which approach is easier to run.

If you choose the Kruskal-Wallis test or a median test, you will get 1 point (these choices are reasonable but inferior) unless you explain your choice (which is difficult). Other tests (simple t-test or WMW U-test) are wrong and will give 0 points.

Justify your test choice – 2 points

Assumptions for a 1-way ANOVA test are:

1. Independence of observations.
    - We can assume it at once.
2. Normality of residuals:
    - Can be checked visually (`plot(anova_model, 2)` or by plotting residuals using a histogram) or by running a suitable hypothesis test

3. Equality of variance:
    - Can be checked visually (`plot(anova_model, 1)`) or by running a suitable hypothesis test
4. Equal group size (have to use different types of SS calculation for the ANOVA table if this requirement is violated):
    - The group size can be noticed in the data diagnosis step
    - This requirement is irrelevant if you run a 1-way ANOVA test. (This is the aspect why the 1-way ANOVA is superior to the factorial ANOVA.)

Altogether, if everything is done correctly, you get 2 points. If something on the list is missing, you get 1 point.

Only one approach must be used! Why? What will you do if you get conflicting results? If you present both approaches, you will get 1 point.

Let's see the result of hypothesis testing:

```
##
##  Bartlett test of homogeneity of variances
##
## data:  Difference by Treatment
## Bartlett's K-squared = 0.75358, df = 2, p-value = 0.6861
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(anova_model)
## W = 0.95615, p-value = 0.4422
```

Perfect hit! **Parametric one-way ANOVA can be used here**

State the statistical hypotheses – 2 points

The correct form – 2 points

It is quite difficult to do something wrong here.

- H_0: means of different `Treatment` groups are the same
- H_1: means of different `Treatment` groups are **NOT** the same

The same for the other model elements.

The not completely correct form (still available) – 1 point

Hypothesis like:

- H_0: `Treatment` does not help to reduce blood glucose level
- H_1: `Treatment` helps to reduce blood glucose level

will be marked less. At least, you specified them and they make sense.

A clearly wrong form – 0 points

Hypothesis like these will yield no points:

- H_0: `Treatment` is useful to treat diabetes
- H_1: `Treatment` is not useful.

Also, it is a double hit – H_0 and H_1 are mixed.

Carry out an appropriate test to answer whether the drug is useful. – 6 points

You run the test you chose before. In our case, it will be ANOVA with post-hoc tests. However, you may treat `Treatment` as a continuous quantitative predictor. Linear regression could have been used (you can transform treatments from a quanlitative variable to the quantitative one), but it is a suboptimal choice

If you chose the wrong test, you will get fewer points *in the step above*. But if you run it adequately and the results are reasonable, you may get full points here as well.

1-way ANOVA – 3 points

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Treatment    2  2.556  1.2779   2.601  0.102
## Residuals   18  8.843  0.4913
```

The difference between the groups is insignificant – 3 points

You *must* report what you have and stop your analysis here. Or run post-hocs to get descriptive statistics (difference between means):

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Difference ~ Treatment, data = t1d_drug)
##
## $Treatment
##                       diff        lwr       upr     p adj
## 1 mg/ml-Vehicle -0.18125 -1.2010721 0.8385721 0.8934458
## 5 mg/ml-Vehicle  0.59250 -0.4273221 1.6123221 0.3222395
## 5 mg/ml-1 mg/ml  0.77375 -0.1206931 1.6681931 0.0968779
```

You may try to run a 2-way ANOVA mentioned above, but I **highly** recommend you do not. This type of models is difficult to specify *without mistakes* and it is best if you have groups of the equal size. Because factorial ANOVA requires special approaches to calculate SS values for each factor if the group sizes are not equal.

Present and discuss your results. Is this novel drug useful? What would you suggest doing next? (5 points)

Present and discuss your results. – 2 points

Diabetic mice treated with the drug at 5 mg/ml showed reduction in blood glucose level (0.84 mM less than before treatment), but the effect did not reach the desired level of statistical significance. There is totally no effect at the lower concentration of the drug compared with vehicle.

Still, there is quite expressed effect: $\eta^2 = 0.224$ and Cohen's f = 0.537.

The drug showed clear signs of toxicity (some of the drug-treated mice died).

Altogether, the drug may be useful, but only at the highest dose (that is toxic as well).

What would you suggest doing next? – 3 points

The drug seem to be able to reduce glucose, but we cannot reject H_0. It is legal to ask for the increased sample size here, but you must justify your request. How? Provide some argumentation:

- power estimation. I will use the `pwr` library because it is a bit easier to use with ANOVA:

```
##
##      Balanced one-way analysis of variance power calculation
##
##               k = 3
##               n = 7
##               f = 0.537
##       sig.level = 0.05
##           power = 0.5136221
##
## NOTE: n is number in each group
```

The current study has not enough power – it can detect differences between groups with the probability of ~0.5. Means, in these conditions, your every second experiment will fail to detect differences even if they *are available*. If we want to achieve at least 0.8 power, we need to have that many animals in each group, and the groups must be *balanced* (see `n`):

```
##
##      Balanced one-way analysis of variance power calculation
##
##               k = 3
##               n = 12.19326
##               f = 0.537
##       sig.level = 0.05
##           power = 0.8
##
## NOTE: n is number in each group
```

Also, we can say that the drug seem to work, but the vehicle or the drug are quite toxic. Thus, we need to modify the way of administration. It is reasonable to run toxicological tests to find out the most affected organs and check what causes this toxicity.

Invent even better ideas!

Overall presentation and R Markdown (25 points in total)

For this last part, check the following

- Was R Markdown used to knit a pdf file? (6 points)
  - The R Markdown file was knitted to a PDF or DOCX (DOCX → PDF) – 6 points.
  - The R Markdown file was created, but the report *was not* knitted. I needed to correct it a bit (a few minor LaTeX errors) – 5 points.
  - The R Markdown file was created, but the report *was not* knitted. I needed to correct it substantially (I needed to change the code slightly or some code was not closed properly) – 4 points.
  - The code was submitted as a well-formatted R script. Or R markdown required significant work – 3 points.
  - The code was submitted as an R script, but the formatting was not good. Or R markdown was just terrible. – 1-2 points.
- Is all code provided? The only exception we make is file import commands if the file path would reveal your identity. (4 points)
- Is all code provided *completely*? (4 points)

Check especially for lines of code that are too long to show up in the knit, like this one:

```
ggplot(hamster, aes(x=lifespan, fill=group)) + facet_grid(cols=vars(group)) + geom_histogram(binwidth =
```

This is a problem, because it means that the code is no longer reproducible to the reader of the knitted document. For full points, longer lines of code should be broken down, for instance like this:

```
p <- ggplot(hamster, aes(x=lifespan, fill=group))
p <- p+ facet_grid(cols=vars(group))
p <- p+ geom_histogram(binwidth = 5)
p <- p + xlab("life span (weeks)")
p <- p + scale_fill_manual(values=c("dodgerblue", "tomato1"))
p
```

- Is the code reproducible? To assess this, markers should select one of the three questions at random and try to reproduce the results. Note that here, we don't care if you chose the right analysis approach, or if your code is correct; all we care about is whether it's reproducible. (4 points)
- Is the file nicely formatted? It doesn't have to be a work of art, but it's nice if there are section headers for each of the questions (like there are in this document), so that it's clear where one question ends and the next begins. (4 points)

Such output is not welcomed:

3. ### HEADER

Text

```
# Also, the advantage of R Markdown it that allows for text,
# code, and results to nicely coexist in a file. Therefore,
# longer chunks of text (such as explanations) should be
# written as actual text, not as a comment in a code chunk
# (like this one!) - Actual text outside of code chunks is
# much easier to read.
```

- Is the writing clear? We don't care so much about smaller typos or perfect English. But the writing should be clear and understandable. It is evaluated by living persons. We still can understand only what we can read. (3 points)