The University of Newcastle

College of Engineering, Science and Environment

School of Information and Physical Sciences

Statistical Inference
STAT2300: Semester 2, 2024

Group Project

Author (s):
Nathan Hill, C3334136
Jacob Saunders, C3262240

October 20, 2024

STAT2300 Group Project

Introduction

Proposal Due: Electronically via Canvas by 11:59pm, Sunday, September 22.

Project Due: Electronically via Canvas by 11:59pm, Sunday, October 20.

Presentation: During the Week 12 lab.

Worth: 30% of your overall grade (20% for the group project and 10% for the group presentation)

- The purpose of this group project is to work collaboratively to apply the statistical concepts and methods of the course to a data set of your group's choosing and present your work to an audience of your peers.
- Please note that your submission should contain your own work. Please refer to the **Student Conduct Rule** for more information.
- The use of generative AI is not permitted for this assessment. **See more details.**

Forming Groups

For this project, you must work in groups of 2 or 3 students. Please formalise your groups in Canvas in the People area. If you're struggling to find someone to work with, there is a thread in the Discussion Board where you can reach out to other students to form groups.

Proposal

The purpose of the proposal is to give you a mechanism to get early feedback on your ideas for the project so that you don't pursue a project that is not viable. The proposal is unweighted, but it will give you an opportunity to outline the plan for your project and me as Course Coordinator the chance to provide guidance or steer you in the right direction as needed. For the proposal, please provide a description of the following:

- **The data:** What data are you going to use? Will you use a real data set or instead performing simulations? If using a real data set, where will you obtain the data? How big will the sample size be? If performing simulations, how many simulations do you intend to perform? What will you be computing in each simulation?
- **The methods you will use to analyse the data:** You must include up to 3 methods we have learned about in STAT2300. What will these methods be able to illuminate about the data?

After your group's proposal is approved, you can proceed with the project.

Project

With the project, you will apply 3 or more concepts from STAT2300 to your chosen data set or simulation design. The project should be a comprehensive written document and it should include the following sections:

1. **Introduction:** In this section, you will provide an overview of the application or theoretical area for your project. Through the methods you will employ, what will you explore about the particular application or theoretical domain you have chosen?
2. **Data or Simulation Design:** In this section, you will provide a full description of the data you will be analysing. More specifically,
    - If you are using a real data set, provide the source of the data set, a description of all of the variables, and the sample size.
    - If you are performing simulations, fully describe the simulation design, including the number of simulations you will be performing and the objects you will create to store the information from the simulations.
3. **Topics Covered:** In this section, you will list 3 or more methods from STAT2300 that you will apply to the data, and what you will use them for.

4. **Results:** In this section, you will apply the methods from the previous section to the data set and provide a full code or working to obtain the results. Note: your work must be completely reproducible. This means:
    - If you make use of any functions contained in add-in packages, you must list these packages and include the appropriate call to library() to load the functions within your code.
    - If you perform any random number generation or random sampling, you must set the seed of the random number generation with the set.seed() function.
5. **Conclusion:** In this section, you will summarise the results and describe what they illustrate about the application or theoretical domain you've mentioned in the introduction.

Group Member Attribution

In addition to the written project, each group must submit a brief report providing details of what each group member contributed to the project. Broadly, this can be aligned to each of the 5 sections above:

1. Introduction
    - Ideas
    - Writing
    - Reviewing
2. Data/Simulation Design
    - Ideas
    - Writing
    - Reviewing
3. Topics Covered
    - Ideas
    - Writing
    - Reviewing
4. Results
    - Writing (text)
    - Writing (code)
    - Reviewing (text and/or code)
5. Conclusion
    - Writing
    - Reviewing

This document must be signed by all group members.

Peer Review

Each member of the group must individually submit a peer reflection form. This gives you the opportunity to anonymously rate and comment on the performance of the other members of your group.

Presentation

In your presentation, your group will have 8 minutes to present your project to the rest of the class during the lab in Week 12. Your presentation should be accompanied by some visual content, such as slides. There are various ways you might format your presentation, but one structure which might work well:

1. **Introduction:** introduce the subject area of your project and why it is interesting. Perhaps briefly indicate the techniques your group applied. ($\sim$2 minutes)
2. **Results:** provide an overview of the results of your project. You don't need to showcase all of the code, but report the main results and how they are meaningful to the subject area. ($\sim$5 minutes)
3. **Conclusion:** provide a summary of your findings and what your group learned about the subject area. ($\sim$1 minute)

1 Maximum Likelihood Estimation (MLE)

We will use MLE to estimate the parameters of a statistical model that describes the relationship between BMI and diabetes progression. This will help us understand the strength and nature of BMI effect on the population.

```
head(ddat)
```

```
##            age         sex         bmi           bp           s1          s2
## 1  0.038075906  0.05068012  0.06169621  0.021872386 -0.044223498 -0.03482076
## 2 -0.001882017 -0.04464164 -0.05147406 -0.026327528 -0.008448724 -0.01916334
## 3  0.085298906  0.05068012  0.04445121 -0.005670422 -0.045599451 -0.03419447
## 4 -0.089062939 -0.04464164 -0.01159501 -0.036656081  0.012190569  0.02499059
## 5  0.005383060 -0.04464164 -0.03638469  0.021872386  0.003934852  0.01559614
## 6 -0.092695478 -0.04464164 -0.04069594 -0.019441826 -0.068990650 -0.07928784
##             s3          s4          s5          s6 target
## 1 -0.043400846 -0.002592262  0.019907486 -0.017646125    151
## 2  0.074411564 -0.039493383 -0.068331547 -0.092204050     75
## 3 -0.032355932 -0.002592262  0.002861309 -0.025930339    141
## 4 -0.036037570  0.034308859  0.022687745 -0.009361911    206
## 5  0.008142084 -0.002592262 -0.031987639 -0.046640874    135
## 6  0.041276824 -0.076394504 -0.041176167 -0.096346157     97
```

```
neg_log_likelihood <- function(params, data) {
  # Extract the parameters
  beta0 <- params[1]        # Intercept
  beta1 <- params[2]        # Slope
  sigma <- params[3]        # Standard Deviation

  # Extract dependent and independent variables
  x <- ddat$bmi
  y <- ddat$target

  # Predicted value of a linear model
  y_pred <- beta0 + beta1 * x

  # Log-likelihood for normal distribution
  norm_log_likelihood <- -sum(dnorm(y, mean = y_pred, sd = sigma, log = TRUE))

  return(norm_log_likelihood)
}
# Now to use optim() to minimise the negative log-likelihood and estimate params
int_params <- c(beta0 = 0, beta1 = 0, sigma = 1)

mle <- optim(
  par = int_params,           # Initial guess
  fn = neg_log_likelihood,    # The function being minimised
  data = ddat,                # Data to be used
  method = "BFGS",            # Optimisation method (less fragile and more common)
  hessian = TRUE              # Return Hessian for variance estimation
)

# Results
mle$par
```

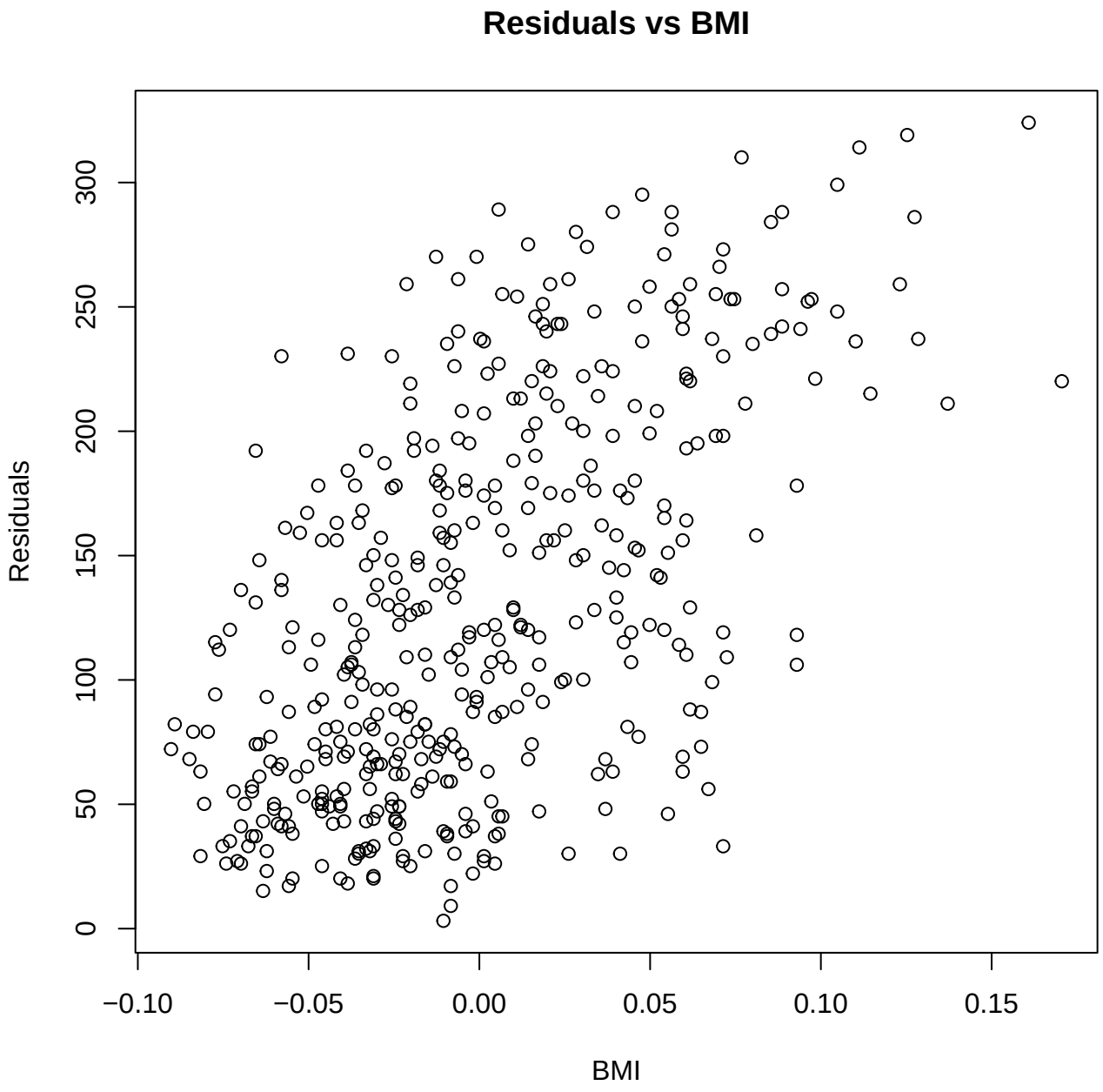
```
##        beta0        beta1        sigma
##   21.8565673    0.3093952 4102.6313545
```

```
# Predicted values from the model put back into likelihood function for LP
```

```r
y_pred <- mle$par[1] + mle$par[2] * ddat$bmi

# Residuals
residuals <- ddat$target - y_pred

# Plot residuals to check for patterns
plot(ddat$bmi, residuals, main = "Residuals vs BMI", xlab = "BMI", ylab = "Residuals")
```



2 Likelihood Ratio Test

We will perform a likelihood ratio test to compare two nested models: one that includes only BMI as a predictor of diabetes progression, and another that includes both BMI and age. This will help us determine whether adding age significantly improves our understanding of diabetes progression.

3 Confidence Intervals using the Newton-Raphson Method

We will compute 95% confidence intervals for the parameters estimated in our MLE model using the likelihood interval obtained through the Newton-Raphson method. This will provide a range of plausible values for each parameter in the population, helping us understand the uncertainty in our estimates.

```
# Log-likelihood function for the normal distribution
log_likely <- function(col, mu) return(- (1 / (2 * var(col))) * sum((col - mu)^2))

# First derivative of a of the log-likelihood function
score <- function(col, mu) return(sum((col - mu) / var(col)))

# Second derivative of a of the log-likelihood function
hessian <- function(col) return(-length(col) / var(col))

newton_raphson_mle <- function(col, θ = 1, tolerance = 1e-8, max_iter = 1000) {
    h <- hessian(col)
    for (i in 1:max_iter) {
        s <- score(col, θ)  # Call the score function
        theta_new <- θ - solve(h, s) # Update θ
        if (max(abs(theta_new - θ)) < tolerance) return(theta_new)
        θ <- theta_new
    }
    warning("Newton-Raphson algorithm did not converge")
    return(θ)
}

conf_int <- function(col, mle, conf_level = 0.95, range = c(-1, 1), tolerance = 1e-8) {

    h <- hessian(col)
    alpha <- 1 - conf_level

    # Normal-based confidence interval
    se <- sqrt(-1 / h)
    z <- qnorm(1 - alpha / 2)
    nm_min <- mle - z * se
    nm_max <- mle + z * se

    # Likelihood-based confidence interval
    max_ll <- log_likely(col, mle)
    crit_value <- qchisq(conf_level, df = 1) / 2
    root_find <- function(theta) log_likely(col, theta) - (max_ll - crit_value)
    ll_min <- uniroot(root_find, c(range[1], mle), tol = tolerance)$root
    ll_max <- uniroot(root_find, c(mle, range[2]), tol = tolerance)$root

    return(list(nm = c(min = nm_min, max = nm_max), ll = c(min = ll_min, max = ll_max)))

}

for (name in names(ddat)) {

    if (name == "target") next
    col <- ddat[[name]]
    mle <- newton_raphson_mle(col)
    ci <- conf_int(col, mle)

    cat(
```

```
        "\n\nColumn:", name, "\n",
        "Newton Raphson MLE: ", mle, "\n",
        "Normal CI: [", ci$nm[1], ", ", ci$nm[2], "]\n",
        "Likelihood CI: [", ci$ll[1], ", ", ci$ll[2], "]\n",
        sep = ""

    )
}
```

```
##
##
## Column:age
## Newton Raphson MLE: 6.720961e-20
## Normal CI: [-0.004439332, 0.004439332]
## Likelihood CI: [-0.004439332, 0.004439332]
##
##
## Column:sex
## Newton Raphson MLE: 1.030358e-17
## Normal CI: [-0.004439332, 0.004439332]
## Likelihood CI: [-0.004439332, 0.004439332]
##
##
## Column:bmi
## Newton Raphson MLE: -2.245954e-16
## Normal CI: [-0.004439332, 0.004439332]
## Likelihood CI: [-0.004439332, 0.004439332]
##
##
## Column:bp
## Newton Raphson MLE: -4.765418e-17
## Normal CI: [-0.004439332, 0.004439332]
## Likelihood CI: [-0.004439332, 0.004439332]
##
##
## Column:s1
## Newton Raphson MLE: -1.412826e-17
## Normal CI: [-0.004439332, 0.004439332]
## Likelihood CI: [-0.004439332, 0.004439332]
##
##
## Column:s2
## Newton Raphson MLE: 3.994985e-17
## Normal CI: [-0.004439332, 0.004439332]
## Likelihood CI: [-0.004439332, 0.004439332]
##
##
## Column:s3
## Newton Raphson MLE: -5.466477e-18
## Normal CI: [-0.004439332, 0.004439332]
## Likelihood CI: [-0.004439332, 0.004439332]
##
##
## Column:s4
## Newton Raphson MLE: -7.489457e-18
## Normal CI: [-0.004439332, 0.004439332]
```

```
## Likelihood CI: [-0.004439332, 0.004439332]
##
##
## Column:s5
## Newton Raphson MLE: 9.281476e-17
## Normal CI: [-0.004439332, 0.004439332]
## Likelihood CI: [-0.004439332, 0.004439332]
##
##
## Column:s6
## Newton Raphson MLE: 1.124024e-17
## Normal CI: [-0.004439332, 0.004439332]
## Likelihood CI: [-0.004439332, 0.004439332]
```

4 Bootstrap Confidence Intervals

We will use the bootstrap method to construct confidence intervals for the correlation coefficient between blood pressure and diabetes progression. This non-parametric approach will allow us to make inferences about the strength of this relationship in the population without making strong distributional assumptions.

```
bootstrap <- function(data, lvls, α = 0.05, studentized = FALSE, debug = FALSE) {

    par_mean <- mean(data)
    lvls_len <- length(lvls)

    recurse <- function(data, lvls, α, par_mean) {

        len <- length(lvls)
        if (len == 0) return(list(μ = mean(data), std_err = sd(data)/sqrt(length(data))))

        n <- length(data)
        iter <- lvls[1]
        boot <- data.frame(
            μ = numeric(iter),
            se = numeric(iter),
            t_stat = numeric(iter)
        )

        for (i in 1:iter) {# Loop through each iteration at the current depth

            # Bootstrap resample the data
            bsample <- sample(data, n, replace = TRUE)

            # Recurse to the next depth, passing the reduced slice of levels
            result <- recurse(bsample, lvls[-1], α, par_mean)

            # Store the mean and standard error
            boot$μ[i] <- result$μ
            boot$se[i] <- result$std_err

            # Store the studentized t-statistic if requested
            if (studentized) boot$t_stat[i] <- (boot$μ[i] - par_mean) / boot$se[i]

            if (debug) cat("Iteration: ", i, "\n", sep = "")
```

```
        }

        # This only happens once at the top level
        if (len == lvls_len) return(list(μ = mean(boot$μ), lvls = lvls, α = α, par_mean = par_mean))
        # This happens at every other level
        else return(list(μ = mean(boot$μ), std_err = sd(boot$μ)))

    }

    return(recurse(data, lvls, α, par_mean))

}

for (name in names(ddat)) {

    if (name == "target") next
    col <- ddat[[name]]
    boot <- bootstrap(ddat$age, c(100, 100, 5), studentized = TRUE)
    mle <- boot$μ
    ci <- conf_int(col, mle)

    cat(

        "\n\nColumn:", name, "\n",
        "Bootstrap MLE: ", mle, "\n",
        "Normal CI: [", ci$nm[1], ", ", ci$nm[2], "]\n",
        "Likelihood CI: [", ci$ll[1], ", ", ci$ll[2], "]\n",
        sep = ""

    )
}
```

```
##
##
## Column:age
## Bootstrap MLE: 0.0003634305
## Normal CI: [-0.004075902, 0.004802763]
## Likelihood CI: [-0.004454184, 0.004454184]
##
##
## Column:sex
## Bootstrap MLE: 0.0001142359
## Normal CI: [-0.004325096, 0.004553568]
## Likelihood CI: [-0.004440802, 0.004440802]
##
##
## Column:bmi
## Bootstrap MLE: -9.9833e-05
## Normal CI: [-0.004539165, 0.004339499]
## Likelihood CI: [-0.004440455, 0.004440455]
##
##
## Column:bp
## Bootstrap MLE: 7.176762e-05
## Normal CI: [-0.004367565, 0.0045111]
## Likelihood CI: [-0.004439912, 0.004439912]
```

```
##
##
## Column:s1
## Bootstrap MLE: 0.0004560851
## Normal CI: [-0.003983247, 0.004895417]
## Likelihood CI: [-0.004462699, 0.004462699]
##
##
## Column:s2
## Bootstrap MLE: 4.830931e-05
## Normal CI: [-0.004391023, 0.004487642]
## Likelihood CI: [-0.004439595, 0.004439595]
##
##
## Column:s3
## Bootstrap MLE: -1.649518e-05
## Normal CI: [-0.004455828, 0.004422837]
## Likelihood CI: [-0.004439363, 0.004439363]
##
##
## Column:s4
## Bootstrap MLE: 5.255363e-05
## Normal CI: [-0.004386779, 0.004491886]
## Likelihood CI: [-0.004439643, 0.004439643]
##
##
## Column:s5
## Bootstrap MLE: -0.0003804972
## Normal CI: [-0.00481983, 0.004058835]
## Likelihood CI: [-0.004455609, 0.004455609]
##
##
## Column:s6
## Bootstrap MLE: 0.0001553199
## Normal CI: [-0.004284012, 0.004594652]
## Likelihood CI: [-0.004442048, 0.004442049]
```

Marking Guide

Project

Introduction

- [**4 marks**] Provides a clear, concise, and comprehensive overview of the application or theoretical area. Clearly explains the significance and relevance of the project. States the methods to be employed and the specific objectives, demonstrating a thorough understanding of how the methods will explore the chosen domain.
- [**3 marks**] Provides a clear overview of the application or theoretical area. Explains the significance and relevance of the project. States the methods to be employed and the specific objectives, demonstrating a good understanding of how the methods will explore the chosen domain.
- [**2 marks**] Provides a basic overview of the application or theoretical area. Provides a basic explanation of the significance and relevance of the project. States the methods and objectives, demonstrating a satisfactory understanding of how the methods will explore the chosen domain.
- [**1 mark**] Provides an unclear or incomplete overview of the application or theoretical area. Fails to explain the significance and relevance of the project. Fails to clearly state the methods and objectives, demonstrating a limited

understanding of how the methods will explore the chosen domain.

Data/Simulation Design

- [**4 marks**] Provides a comprehensive and detailed description of the data or simulation design. For real data, includes the source, a thorough description of all variables, and the sample size. For simulations, fully describes the design, number of simulations, and objects created to store information.
- [**3 marks**] Provides a clear description of the data or simulation design. For real data, includes the source, a good description of most variables, and the sample size. For simulations, describes the design, number of simulations, and objects created to store information.
- [**2 marks**] Provides a basic description of the data or simulation design. For real data, includes the source and a basic description of some variables, and the sample size. For simulations, provides a basic description of the design and number of simulations, with limited detail on objects created to store information.
- [**1 mark**] Provides an incomplete or unclear description of the data or simulation design. For real data, lacks the source, description of variables, or sample size. For simulations, provides an unclear or incomplete description of the design, number of simulations, and objects created to store information.

Topics Covered

- [**3 marks**] Provides a clear, detailed explanation for each method and its application to the data. Demonstrates a thorough understanding of how each method will be used and its relevance to the project.
- [**2 marks**] Provides a clear, detailed explanation for two methods and their application to the data. Demonstrates a thorough understanding of how these two methods will be used and their relevance to the project.
- [**1 mark**] Provides a clear, detailed explanation for one method and its application to the data. Demonstrates a thorough understanding of how this method will be used and its relevance to the project.

Results: Method, Code, Working, and Appropriateness for the Task

- [**13–15 marks**] The methods applied are highly appropriate for the task. The R code is well-written, clear, and includes detailed comments. Any mathematical working is thorough and accurate, leading to correct results. The conclusions for each method are insightful and relevant to the data set.
- [**10–12 marks**] The methods applied are appropriate for the task. The R code is mostly clear and includes some comments. Any mathematical working is mostly accurate, with minor errors that do not significantly affect the results. The conclusions are relevant but may lack some insight.
- [**7–9 marks**] The methods applied are somewhat appropriate for the task. The R code is functional but lacks clarity and comments. Any mathematical working has some errors that affect the results but are still understandable. The conclusions are somewhat relevant but may be incomplete or lack depth.
- [**4–6 marks**] The methods applied are minimally appropriate for the task. The R code is unclear and lacks comments. Any mathematical working has significant errors that affect the results and make them difficult to interpret. The conclusions are minimally relevant and lack insight.
- [**1–3 marks**] The methods applied are inappropriate for the task. The R code is poorly written and lacks comments. Any mathematical working is inaccurate and leads to incorrect results. The conclusions are irrelevant or incorrect.

Results – Reproducibility

- [**7–9 marks**] The results are fully reproducible. All necessary packages are listed and loaded with library(). Any random number generation or sampling includes a set.seed() call. The code runs without errors and produces the expected results.
- [**4–6 marks**] The results are mostly reproducible. Most necessary packages are listed and loaded, but one or two might be missing. Most random number generation or sampling includes a set.seed() call. The code runs with minor issues that do not significantly affect the results.

- **[1–3 marks]** Some of the results are reproducible. Some necessary packages are listed and loaded, but many others may be missing. Some number generation or sampling includes a set.seed() call. The code runs with minor issues that do not significantly affect the results.

Conclusion

- **[5 marks]** The conclusion is exceptionally clear and insightful. It effectively summarises the results and provides a deep understanding of what they illustrate about the application or theoretical domain. The connections to the introduction are strong and well-articulated.
- **[4 marks]** The conclusion is clear and insightful. It summarises the results well and provides a good understanding of what they illustrate about the application or theoretical domain. The connections to the introduction are clear but may lack some depth.
- **[3 marks]** The conclusion is somewhat clear and insightful. It summarises the results adequately but may miss some key points about what they illustrate about the application or theoretical domain. The connections to the introduction are present but not strongly articulated.
- **[2 marks]** The conclusion is minimally clear and insightful. It provides a basic summary of the results but lacks depth in explaining what they illustrate about the application or theoretical domain. The connections to the introduction are weak.
- **[1 mark]** The conclusion is unclear and lacks insight. It fails to effectively summarise the results or explain what they illustrate about the application or theoretical domain. The connections to the introduction are missing or poorly articulated.

Note: The default position is that each student in the group will receive the same mark. However, the peer reflection form may be used to adjust marks individually if appropriate.

Presentation

During your presentation, your fellow classmates and I will provide an assessment based on the following:

- **Introduction:** Did the group clearly introduce the subject area and explain why it is interesting?
- **Techniques:** Were the techniques applied appropriate and relevant for the application?
- **Presentation of results:** Were the main results of the project clearly presented and explained?
- **Visual content:** Were the slides or other visual aids clear, relevant, and effectively used?
- **Engagement:** Did the presentation keep the audience engaged and interested?
- **Conclusion:** Did the group provide a clear summary of their findings and what they learned?
- **Teamwork:** Was there good coordination among group members during the presentation?
- **Structure:** Was the presentation well-organised (including timing), and easy to follow?

The group's mark will be an aggregate of my reflection (50% weight) and the collective reflection of your peers (50% weight).

Data Sources

If you're not sure where to look for sources of data, here are some potential places that might inspire an idea:

- **Our World in Data:** A large repository of global data across many different topics. A list of topics is available here.
- **Bureau of Meteorology:** A source of weather, water, and climate data in Australia.

- **Australian Bureau of Statistics:** The ABS is Australia's national statistical agency, providing official statistics on a wide range of economic, social, population and environmental matters of importance to Australia.
- **Sports-Reference:** A network of websites containing detailed historical statistics for (mostly) North American sports leagues.
- **HURDAT 2:** A historical database of tropical cyclones. There are databases for the Atlantic Tropical Basin and the Northeast Pacific Tropical Basin.
- **HYG:** A database with a wide variety of variables measured on stars.
- **SIMBAD:** Another astronomical database, which includes other celestial objects in addition to stars.
- **Awesome Public Datasets:** A Github repository of publicly available (and presumably awesome!) datasets.

These sources offer a wide range of data across various fields and topics. When choosing a dataset, consider its relevance to your project objectives, the quality and completeness of the data, and any potential ethical considerations in its use.