On the comprehensibility of funtional decomposition: Analysis

James Finnie-Ansley

2023-11-21

Contents

> Let's say for example I have veeeeery long method that parses through a text (to convert it or something). Now did I understand the ReFactoring idea correctly, that it should be my goal to shorten the method? And to break it up into several smaller ones?
>
> — Wards Wiki, *How To Refactor*

1 Outline

This document provides the analysis and figures for the following hypotheses regarding the effect of functional decomposition on the comprehensibility of small-scale programs (of the order of tens of lines code):

Time:

$\mathbf{H}_0^{\text{time}}$ The null hypothesis for reading time is that the time taken by participants not depend on the version.

$\mathbf{H}_1^{\text{time}}$ The alternative hypothesis is that less time is taken by participants with the MFV version than those with the SFV version.

Explain in Plain English (EPE):

$\mathbf{H}_0^{\text{EPE}}$ The null hypothesis for the "Explanation in Plain English" is that the correctness of the explanation does not depend on the version.

$\mathbf{H}_1^{\text{EPE}}$ The alternative hypothesis is that the explanations by participants with the MFV version are more correct than those with the SFV version.

Behaviour Questions:

$\mathbf{H}_0^{\text{Beh}}$ The null hypothesis for the behaviour understanding is that the participant's scores to the behaviour questions not depend on the version.

$\mathbf{H}_1^{\text{Beh}}$ The alternative hypothesis is that the participants with the MFV version get higher score than those with the SFV version.

2 The Data

```
raw_data <- read_excel("data/data.xlsx", sheet = "ParticipantData")
```

```
data <- (
  raw_data

  # IDs
  |> rename(
        participant_id = `participant-id`,
        qualtrics_id = `qualtrics-id`
     )

  # Treatment Groups
  |> rename(treatment = Treatment)
  |> mutate(
        treatment = as.factor(case_match(
            treatment,
            "WS" ~ "WeatherSingle",
            "WM" ~ "WeatherMultiple",
            "DS" ~ "DaysSingle",
            "DM" ~ "DaysMultiple"
        )),
     )

  # Demographics

  # What is the most advanced course you have completed or are currently
  # enrolled in?
  |> rename(enrolment = Q22)
```

```
|> mutate(
       enrolment = as.factor(case_match(
           enrolment,
           "100-level COMPSCI" ~ "COMPSCI 100",
           "200-level COMPSCI" ~ "COMPSCI 200",
           "300-level COMPSCI" ~ "COMPSCI 300",
           "700-level COMPSCI" ~ "COMPSCI 700",
           "200-level SOFTENG" ~ "SOFTENG 200",
           "300-level SOFTENG" ~ "SOFTENG 300",
           "700-level SOFTENG" ~ "SOFTENG 700",
           "Other 100-level programming course" ~ "OTHER 100",
           "Other 200-level programming course" ~ "OTHER 200",
           "Other 300-level programming course" ~ "OTHER 300",
           "Other 700-level programming course" ~ "OTHER 700",
           "Other programming course" ~ "OTHER",
           "No course" ~ "NONE"
       ))
   )

# How would you describe your programming experience in any
# programming language?
|> rename(education = Q23)

# How would you rate your experience in programming with Java
|> rename(programming_experience = Q24)
|> mutate(
       programming_experience = ordered(
           case_match(
               programming_experience,
               "None" ~ "NONE",
               "Written tiny programs (up to 100 lines each)" ~ "TINY",
               "Written small programs (100-500 lines each)" ~ "SMALL",
               "Written medium programs (500-2000 lines)" ~ "MEDIUM",
               "Written programs larger than 2000 lines" ~ "LARGE",
           ),
           levels = c("NONE", "TINY", "SMALL", "MEDIUM", "LARGE")
       ),
   )

# How would you describe your confidence in programming in Java
|> rename(language_confidence = Q25)
|> mutate(
       language_confidence = ordered(
           case_when(
               startsWith(language_confidence, "Very low") ~ "VERY-LOW",
               startsWith(language_confidence, "Low")  ~ "LOW",
               startsWith(language_confidence, "Medium")  ~ "MEDIUM",
               startsWith(language_confidence, "High")  ~ "HIGH",
               startsWith(language_confidence, "Very high")  ~ "VERY-HIGH",
           ),
           levels = c("VERY-LOW", "LOW", "MEDIUM", "HIGH", "VERY-HIGH")
       ),
   )
```

```
# Pre-test
|> rename(
       pretest_q1 = Q21,
       pretest_q1_mark = Q21mark,
       pretest_q2 = Q158,
       pretest_q2_mark = Q158mark,
       pretest_q3 = Q161,
       pretest_q3_mark = Q161mark,
       pretest_q4 = Q164,
       pretest_q4_mark = Q164mark,
       pretest_q5 = Q218,
       pretest_q5_mark = Q218mark
   )

# Main Survey
|> rename(
       # Study the code presented below. You will be asked a question about
       # this code on the following page. Once you leave this page you will
       # not be able to return, so spend as much time as you need to
       # understand this code. You may assume there are no errors in the code.
       code_reading_time = `reading time`,

       # Write 2-3 sentences explaining what the code you have just studied
       # on the previous page does. Your description should be at a
       # high-level but still be complete.
       code_explanation = `EPEAns`
   )

# Coding of answers
|> mutate(across(matches("EPE\\d+"), ~replace_na(., 0)))

# Rate your confidence of the accuracy of your answer.
|> rename(code_explanation_confidence = `EPEAnal`)
|> mutate(
       code_explanation_confidence = ordered(
           toupper(code_explanation_confidence),
           levels = c("LOW", "MEDIUM", "HIGH")
       ),
   )

# Behaviour Questions
|> rename(
       code_trace_q1 = `p3-q1`,
       code_trace_q1_mark = p3mark,
       code_trace_q1_time = `p3 time`,
       code_trace_q2 = `p4-q1`,
       code_trace_q2_mark = p4mark,
       code_trace_q2_time = `p4 time`,
       code_trace_q3 = `p5-q1`,
       code_trace_q3_mark = p5mark,
       code_trace_q3_time = `p5  time`,
       code_trace_q4 = `p6-q1`,
       code_trace_q4_mark = p6mark,
```

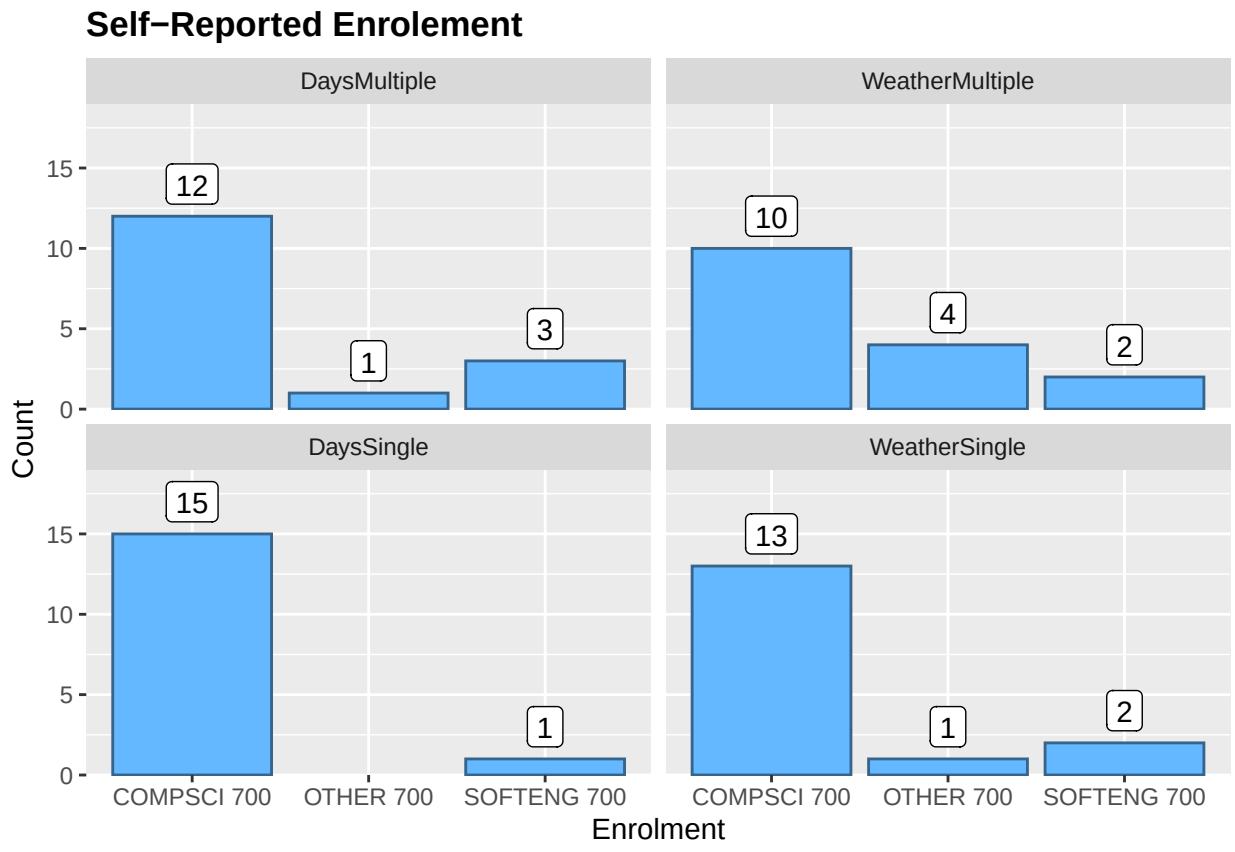
```
        code_trace_q4_time = `p6 time`,
        code_trace_q5 = `p7-q1`,
        code_trace_q5_mark = p7mark,
        code_trace_q5_time = `p7 time`,
        code_trace_q6 = `p8-q1`,
        code_trace_q6_mark = p8mark,
        code_trace_q6_time = `p8 time`
    )

# Given the same code below. How easy do you think this code is to
# understand?
|> rename(
        code_understanding = `p9-q1`,
        code_understanding_time = `p9 time`,
        code_understanding_comment = comment
    )
|> mutate(
        code_understanding = ordered(
            case_match(
                code_understanding,
                "Very easy" ~ "VERY-EASY",
                "Somewhat easy" ~ "EASY",
                "Not easy, but not difficult" ~ "NEUTRAL",
                "Difficult" ~ "DIFFICULT",
                "Very difficult" ~ "VERY-DIFFICULT"
            ),
            levels = c(
                "VERY-DIFFICULT", "DIFFICULT", "NEUTRAL", "EASY", "VERY-EASY"
            )
        ),
    )

# Given the same code and an alternative version below, which version
# do you think takes less effort to understand?
|> rename(
        code_compare_time = `compare-time`,
        code_compare_preference = `compare-q1`,
        code_compare_comments = `compare-q2`
)
|> mutate(
        code_compare_preference = factor(
            case_match(
                code_compare_preference,
                "Version 1" ~ "SINGLE",
                "Version 2" ~ "MULTIPLE",
                "Both versions take the same effort" ~ "NEITHER"
            ),
            levels = c("NEITHER", "SINGLE", "MULTIPLE")
        )
    )

|> select(-c(
        Q23_3_TEXT,  # No one filled this in
```

```
        starts_with("p2-q1...")  # redundant
    ))
)
```

```
write_csv(data, "data/clean_data.csv")
```

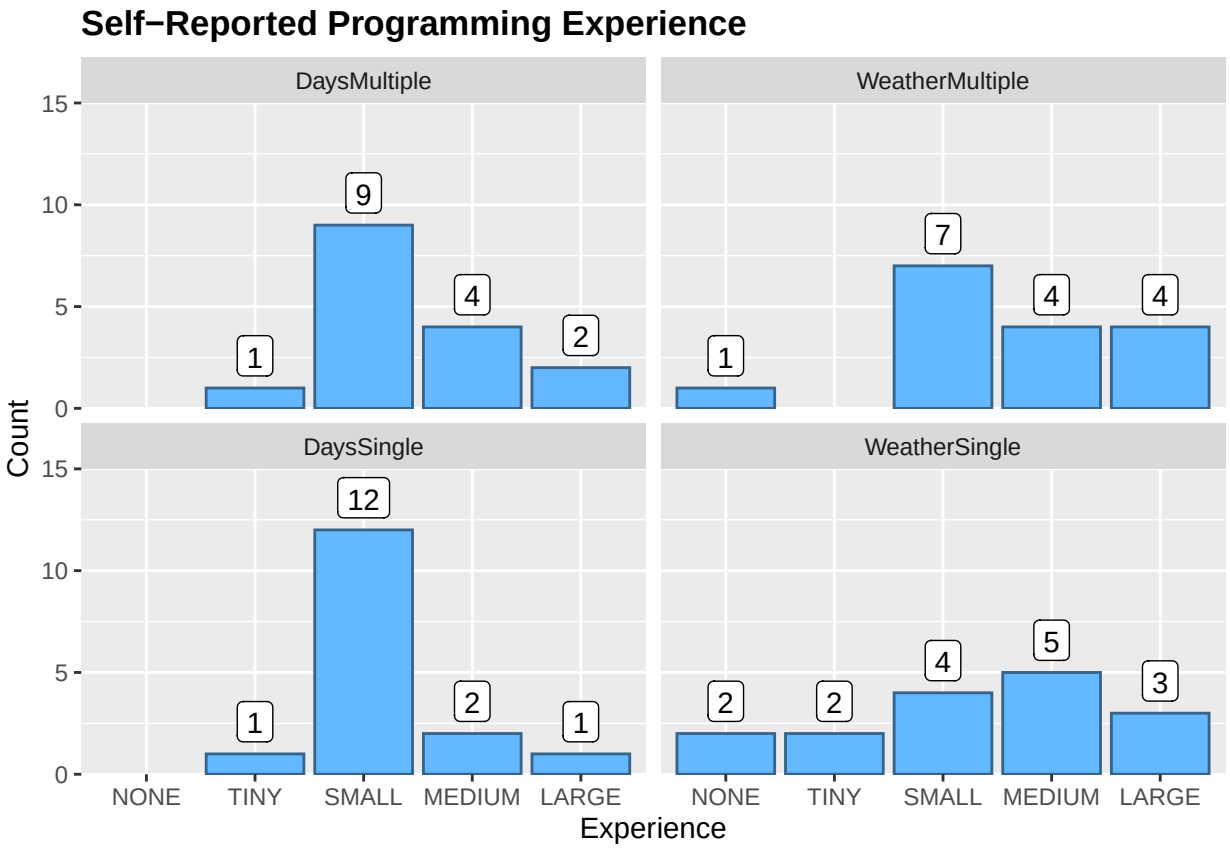
3 Demographics

```
(
    ggplot(data, aes(x = enrolment))
    + labs(title = "Self-Reported Enrolement", x = "Enrolment", y = "Count")
    + geom_bar(stat = "count", fill = "steelblue1", color = "steelblue4")
    + facet_wrap(~treatment, dir = "v")
    + geom_label(stat = "count", aes(label = after_stat(count)), vjust = -0.3)
    + scale_y_continuous(expand = expansion(add = c(0, 4)))
    + theme(plot.title = element_text(face = "bold"))
)
```

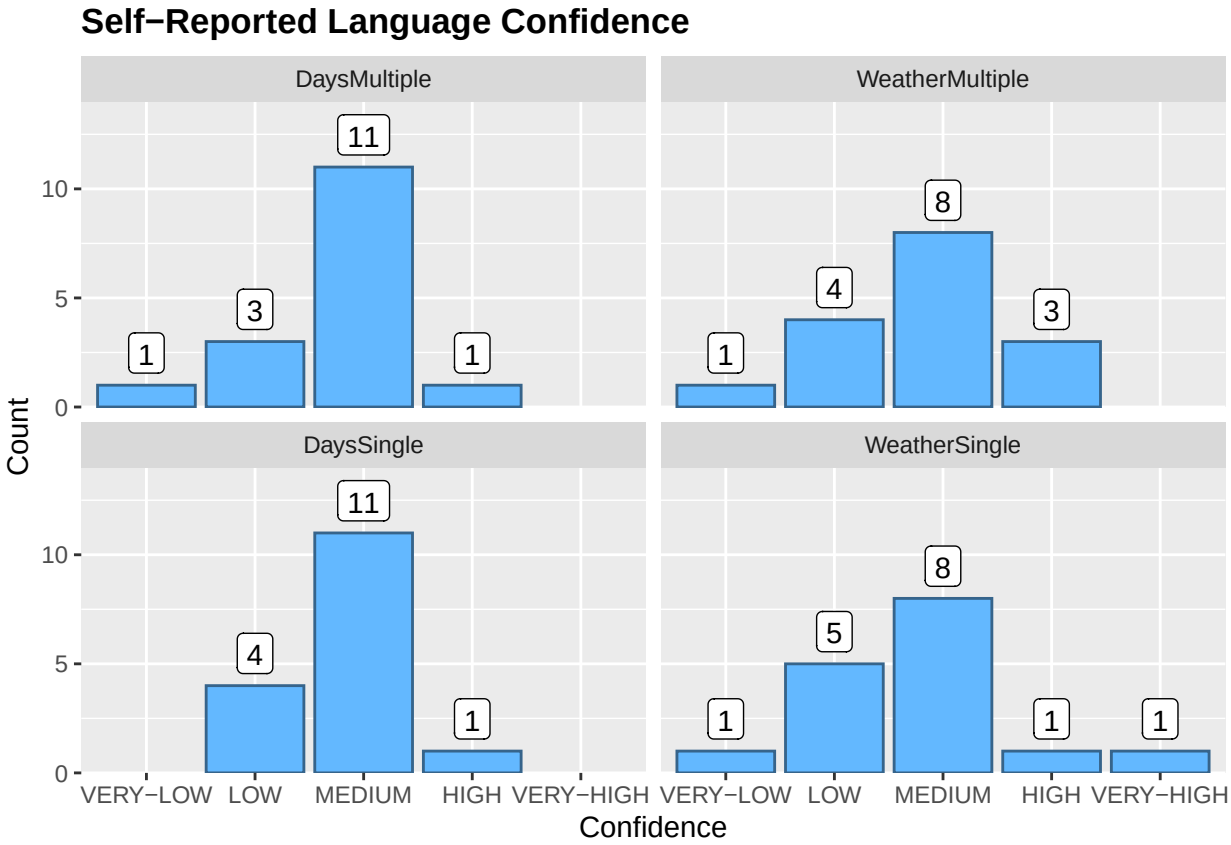



```
(
    ggplot(data, aes(x = programming_experience))
    + labs(
        title = "Self-Reported Programming Experience",
        x = "Experience", y = "Count"
      )
    + geom_bar(stat = "count", fill = "steelblue1", color = "steelblue4")
```

```
    + facet_wrap(~treatment, dir = "v")
    + geom_label(stat = "count", aes(label = after_stat(count)), vjust = -0.3)
    + scale_y_continuous(expand = expansion(add = c(0, 3)))
    + theme(plot.title = element_text(face = "bold"))
)
```




```
(
    ggplot(data, aes(x = language_confidence))
    + labs(
          title = "Self-Reported Language Confidence",
          x = "Confidence", y = "Count"
      )
    + geom_bar(stat = "count", fill = "steelblue1", color = "steelblue4")
    + facet_wrap(~treatment, dir = "v")
    + geom_label(stat = "count", aes(label = after_stat(count)), vjust = -0.3)
    + scale_y_continuous(expand = expansion(add = c(0, 3)))
    + theme(plot.title = element_text(face = "bold"))
)
```



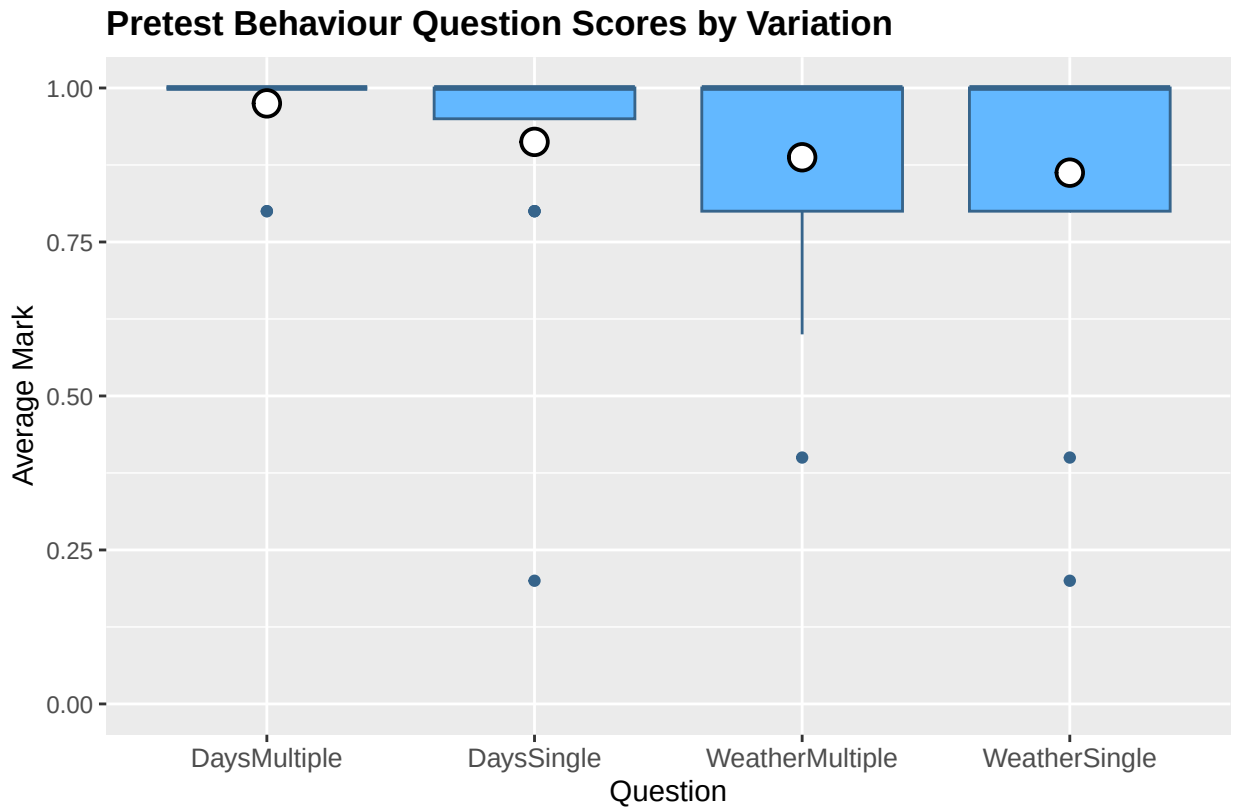
4 Pretest

Pretest scores for behaviour questions are averaged across each participant.

```
pretest_behaviour_marks <- (
  data
  |> select(c(participant_id, treatment, matches("^pretest.*mark$")))
  |> mutate(
        grade = rowMeans(
            across(pretest_q1_mark:pretest_q5_mark),
            na.rm = TRUE
        )
    )
)
```

```
(
    ggplot(pretest_behaviour_marks, aes(x = treatment, y = grade))
    + labs(
          title = "Pretest Behaviour Question Scores by Variation",
          caption = "Each Question Marked out of 1",
          x = "Question", y = "Average Mark"
      )
    + geom_boxplot(color = "steelblue4", fill = "steelblue1")
    + ylim(0, 1)
    + stat_summary(
          fun = mean, geom = "point",
```

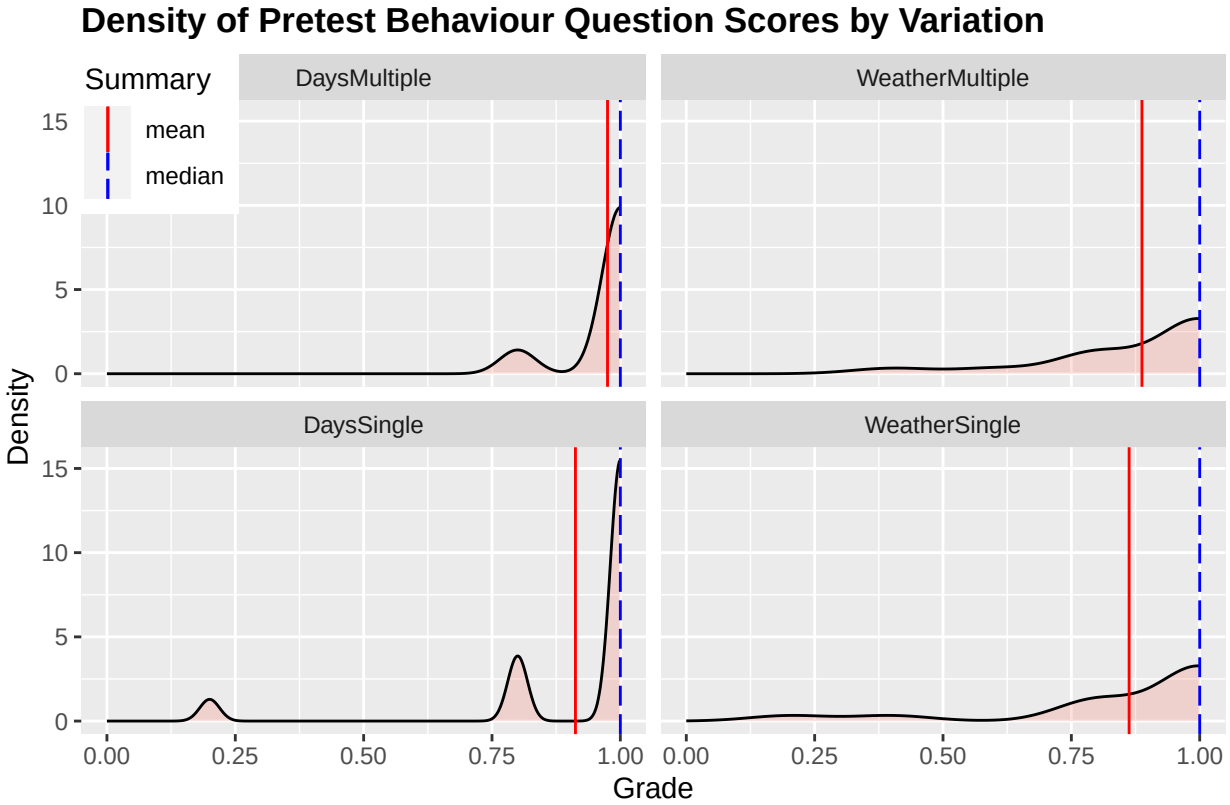
```
        shape = 21, size = 4, stroke = 1,
        color = "black", fill = "white"
    )
  + theme(
        plot.title = element_text(face = "bold"),
        axis.text.x = element_text(size = 10)
    )
)
```



Each Question Marked out of 1

```
averages <- (
    pretest_behaviour_marks
    |> group_by(treatment)
    |> summarise(
          mean = mean(grade),
          median = median(grade)
      )
)

(
    ggplot(pretest_behaviour_marks, aes(x = grade))
    + labs(
        title = "Density of Pretest Behaviour Question Scores by Variation",
        caption = "Each Question Marked out of 1",
        x = "Grade", y = "Density",
        color = "Summary", linetype = "Summary",
    )
```

```
    + geom_density(alpha = 0.25, fill = "salmon")
    + xlim(0, 1)
    + geom_vline(
          aes(xintercept = mean, color = "mean", linetype = "mean"),
          data = averages,
      )
    + geom_vline(
          aes(xintercept = median, color = "median", linetype = "median"),
          data = averages,
      )
    + facet_wrap(~ treatment, dir = "v")
    + scale_colour_manual(
          name = "Summary", values = c(mean = "red", median = "blue")
      )
    + scale_linetype_manual(
          name = "Summary", values = c(mean = "solid", median = "longdash")
      )
    + theme(
          plot.title = element_text(face = "bold"),
          legend.position = c(0.064, 0.95)
      )
)
```



Each Question Marked out of 1

5 Data Selection: Pretest Grades

Participants scoring less than 50% on the pretest behaviour questions are excluded.

```
passing_paticipants <- (
    pretest_behaviour_marks
    |> filter(grade >= 0.5)
    |> pull(participant_id)
)

data <- data |> filter(participant_id %in% passing_paticipants)
```

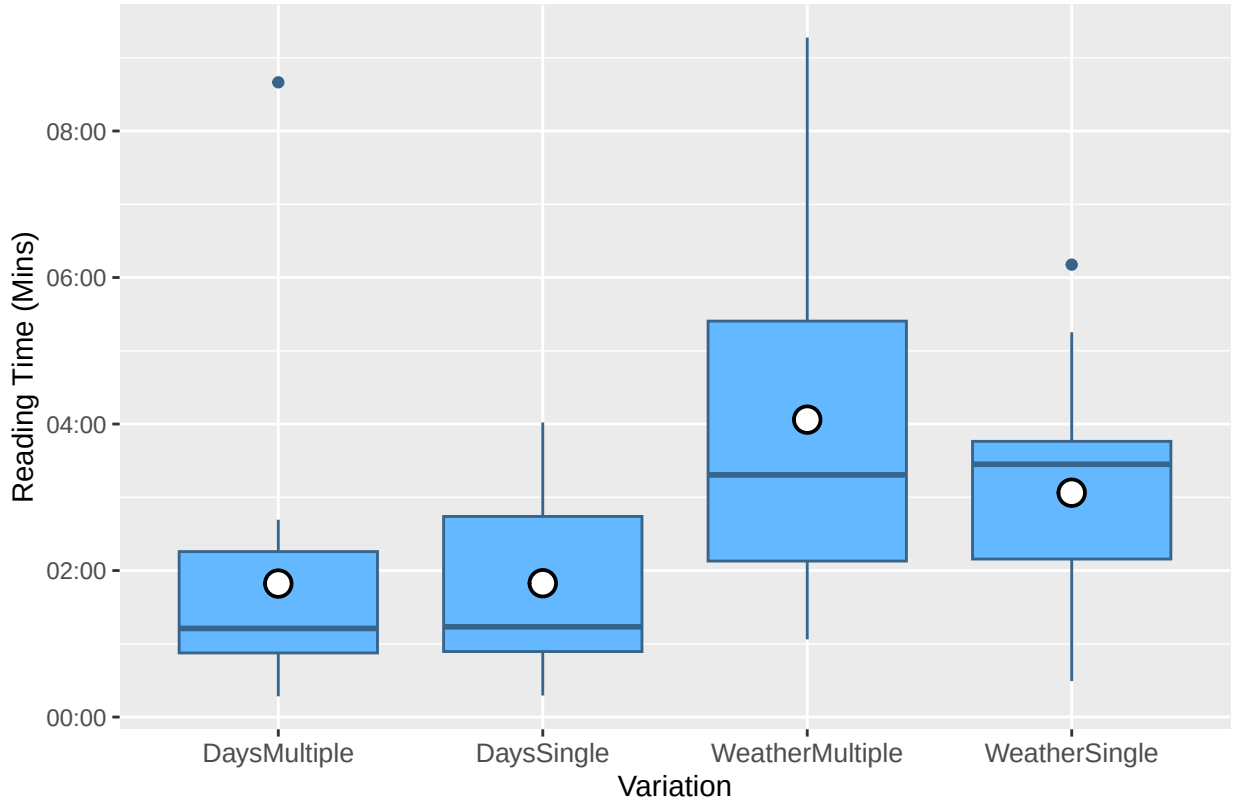
6 Hypothesis 1: Reading Time

$\mathbf{H}_0^{\text{time}}$ The null hypothesis for reading time is that the time taken by participants not depend on the version.

$\mathbf{H}_1^{\text{time}}$ The alternative hypothesis is that less time is taken by participants with the MFV version than those with the SFV version.

6.1 Figures

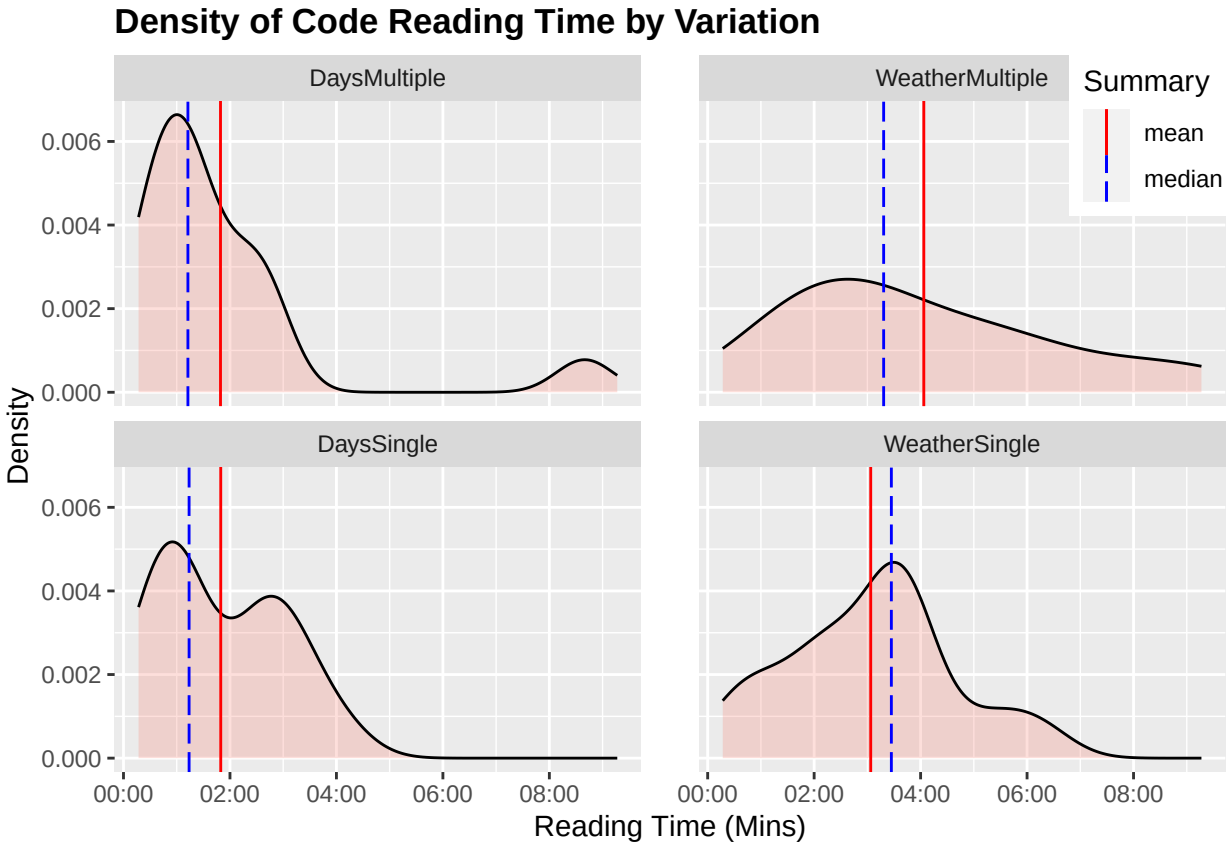
```
box_plot_time_reading_by_treatment <- (
    ggplot(data, aes(x = treatment, y = code_reading_time))
    + labs(
          caption = "White dots show means",
          x = "Variation", y = "Reading Time (Mins)"
      )
    + geom_boxplot(fill = "steelblue1", color = "steelblue4")
    + scale_y_time(labels = ~strftime(.x, "%M:%S"))
    + stat_summary(
          fun = mean, geom = "point",
          shape = 21, size = 4, stroke = 1,
          color = "black", fill = "white"
      )
    + theme(
          plot.title = element_text(face = "bold"),
          axis.text.x = element_text(size = 10)
      )
)

ggsave(
    "figures/box_plot_reading_time.png",
    box_plot_time_reading_by_treatment,
    width = 6, height = 2.75, units = "in"
)

box_plot_time_reading_by_treatment
```



White dots show means

```
averages <- (
    data
    |> group_by(treatment)
    |> summarise(
           mean = mean(code_reading_time),
           median = median(code_reading_time)
       )
)

density_plot_time_reading_by_treatment <- (
    ggplot(data, aes(x = code_reading_time))
    + labs(
          title = "Density of Code Reading Time by Variation",
          x = "Reading Time (Mins)", y = "Density",
          color = "Summary", linetype = "Summary",
      )
    + geom_density(alpha = 0.25, fill = "salmon")
    + scale_x_time(labels = ~strftime(.x, "%M:%S"))
    + geom_vline(
          aes(xintercept = mean, color = "mean", linetype = "mean"),
          data = averages,
      )
    + geom_vline(
          aes(xintercept = median, color = "median", linetype = "median"),
          data = averages,
      )
```

```
    + facet_wrap(~treatment, dir = "v")
    + scale_colour_manual(
        name = "Summary", values = c(mean = "red", median = "blue")
      )
    + scale_linetype_manual(
        name = "Summary", values = c(mean = "solid", median = "longdash")
      )
    + theme(
        plot.title = element_text(face = "bold"),
        legend.position = c(0.935, 0.95),
        panel.spacing.x = unit(1.5, "lines")
      )
)

ggsave(
    "figures/density_plot_reading_time.png",
    density_plot_time_reading_by_treatment,
    width = 6, height = 4, units = "in"
)

density_plot_time_reading_by_treatment
```



6.2 Analysis

6.2.1 Summary Statistics

```
(
    data
    |> group_by(treatment)
    |> summarise(
           mean = mean(code_reading_time),
           median = median(code_reading_time)
       )
)
```

```
# A tibble: 4 x 3
  treatment        mean median
  <fct>           <dbl>  <dbl>
1 DaysMultiple     109.   72.6
2 DaysSingle       110.   73.9
3 WeatherMultiple  244.  198.
4 WeatherSingle    184.  207.
```

6.2.2 Significance

```
(
  data
  |> group_by(treatment)
  |> summarise(out = tidy(shapiro.test(code_reading_time)), .groups = 'drop')
  |> unnest(c(out))
)
```

6.2.2.1 Normality

```
# A tibble: 4 x 4
  treatment       statistic   p.value method
  <fct>               <dbl>     <dbl> <chr>
1 DaysMultiple        0.642 0.0000415 Shapiro-Wilk normality test
2 DaysSingle          0.913 0.148     Shapiro-Wilk normality test
3 WeatherMultiple     0.920 0.195     Shapiro-Wilk normality test
4 WeatherSingle       0.963 0.764     Shapiro-Wilk normality test
```

Groups are not all normal, Wilcoxon rank sum test is performed to check significance.

```
days_multiple_code_reaing_time <- (
    data |> filter(treatment == "DaysMultiple") |> pull(code_reading_time)
)
days_single_code_reaing_time <- (
    data |> filter(treatment == "DaysSingle") |> pull(code_reading_time)
)

weather_multiple_code_reaing_time <- (
    data |> filter(treatment == "WeatherMultiple") |> pull(code_reading_time)
)
weather_single_code_reaing_time <- (
    data |> filter(treatment == "WeatherSingle") |> pull(code_reading_time)
```

```
)
```

```
wilcox.test(
    days_multiple_code_reaing_time,
    days_single_code_reaing_time,
    alternative = "less"
)
```

6.2.2.2 Wilcox

```
    Wilcoxon rank sum exact test

data:  days_multiple_code_reaing_time and days_single_code_reaing_time
W = 107, p-value = 0.313
alternative hypothesis: true location shift is less than 0
```

```
wilcox.test(
    weather_multiple_code_reaing_time,
    weather_single_code_reaing_time,
    alternative = "less"
)
```

```
    Wilcoxon rank sum exact test

data:  weather_multiple_code_reaing_time and weather_single_code_reaing_time
W = 120, p-value = 0.7477
alternative hypothesis: true location shift is less than 0
```

6.3 Interpretation

There is no significant reduction in time for either of the many function variants — the null hypothesis cannot be rejected.

7 Hypothesis 2: Explain in Plain English

$\mathbf{H}_0^{\text{EPE}}$ The null hypothesis for the "Explanation in Plain English" is that the correctness of the explanation does not depend on the version.

$\mathbf{H}_1^{\text{EPE}}$ The alternative hypothesis is that the explanations by participants with the MFV version are more correct than those with the SFV version.

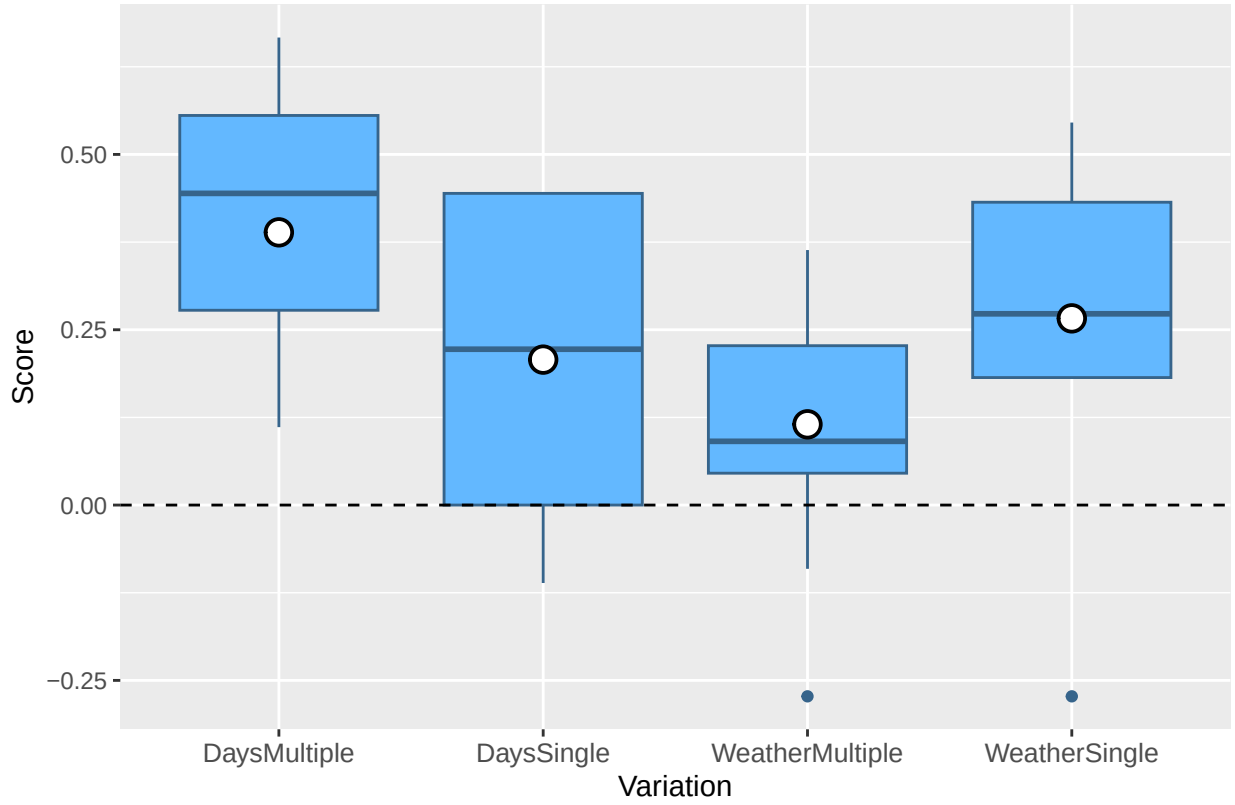
Scores are calculated as the row averages of the codes (+1 for correct code, -1 for incorrect code, and 0 otherwise).

```
weather_data <- (
  data
  |> select(c(participant_id, treatment, EPE1:EPE11))
  |> filter(treatment == "WeatherMultiple" | treatment == "WeatherSingle")
  |> mutate(across(EPE9:EPE11, ~-.x))
  |> mutate(EPE_grade = rowMeans(across(EPE1:EPE11)))
)

days_data <- (
  data
```

```
  |> select(c(participant_id, treatment, EPE1:EPE9))
  |> filter(treatment == "DaysMultiple" | treatment == "DaysSingle")
  |> mutate(across(EPE7:EPE9, ~-.x))
  |> mutate(EPE_grade = rowMeans(across(EPE1:EPE9)))
)

EPE_data <- bind_rows(weather_data, days_data)
```

7.1 Figures

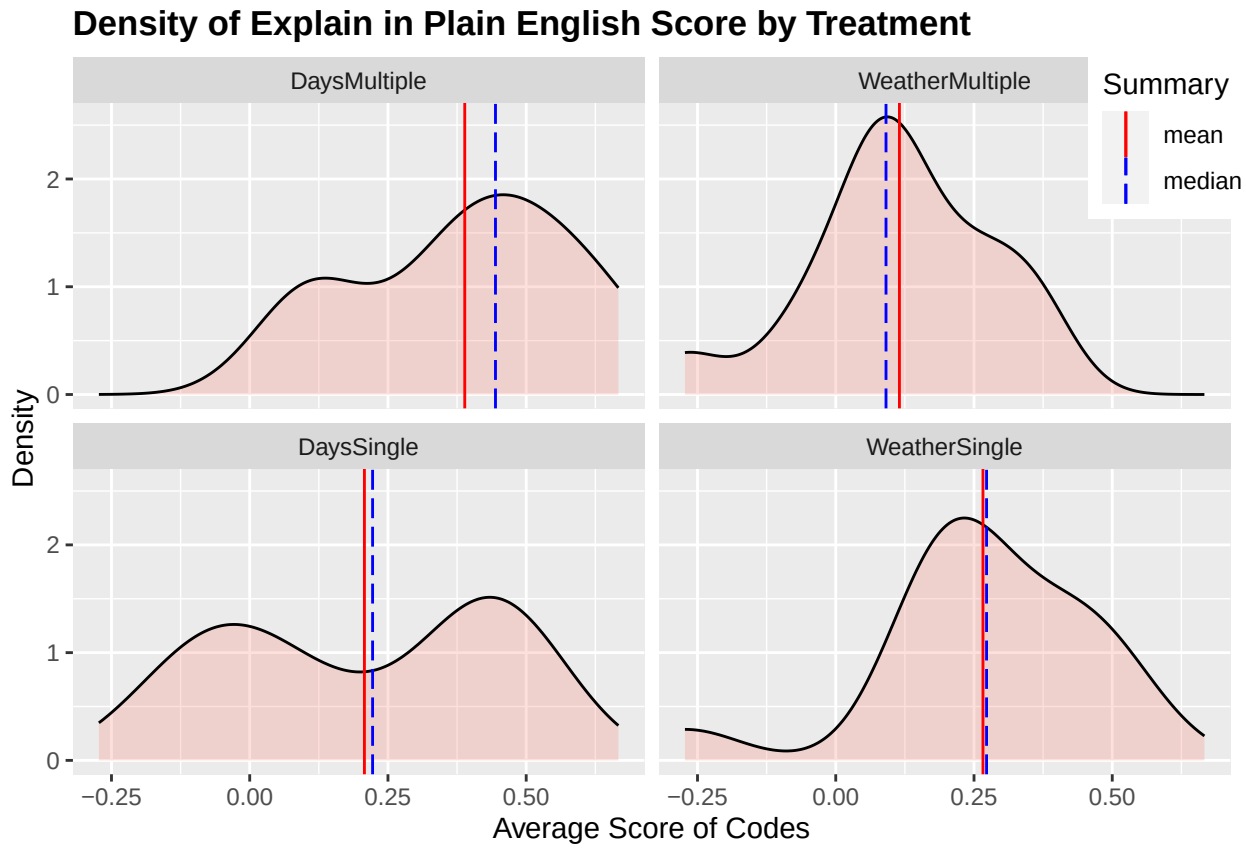
```
box_plot_explain_in_plain_english_scores_by_treatment <- (
    ggplot(EPE_data, aes(x = treatment, y = EPE_grade))
    + labs(
          caption = "White dots show means; Dashed line shows y = 0",
          x = "Variation", y = "Score"
      )
    + geom_boxplot(fill = "steelblue1", color = "steelblue4")
    + geom_hline(yintercept = 0, color = "black", linetype = "dashed")
    + stat_summary(
          fun = mean, geom = "point",
          shape = 21, size = 4, stroke = 1,
          color = "black", fill = "white"
      )
    + theme(
          plot.title = element_text(face = "bold"),
          axis.text.x = element_text(size = 10)
      )
)

ggsave(
    "figures/box_plot_explain_in_plain_english.png",
    box_plot_explain_in_plain_english_scores_by_treatment,
    width = 6, height = 2.75, units = "in"
)

box_plot_explain_in_plain_english_scores_by_treatment
```



White dots show means; Dashed line shows y = 0

```
averages <- (
    EPE_data
    |> group_by(treatment)
    |> summarise(
           mean = mean(EPE_grade),
           median = median(EPE_grade)
       )
)

density_plot_explain_in_plain_english_scores_by_treatment <- (
    ggplot(EPE_data, aes(x = EPE_grade))
    + labs(
          title = "Density of Explain in Plain English Score by Treatment",
          x = "Average Score of Codes", y = "Density",
          color = "Summary", linetype = "Summary",
      )
    + geom_density(alpha = 0.25, fill = "salmon")
    + geom_vline(
          aes(xintercept = mean, color = "mean", linetype = "mean"),
          data = averages,
      )
    + geom_vline(
          aes(xintercept = median, color = "median", linetype = "median"),
          data = averages,
      )
    + facet_wrap(~treatment, dir = "v")
```

```
    + scale_colour_manual(
          name = "Summary", values = c(mean = "red", median = "blue")
      )
    + scale_linetype_manual(
          name = "Summary", values = c(mean = "solid", median = "longdash")
      )
    + theme(
          plot.title = element_text(face = "bold"),
          legend.position = c(0.95, 0.95)
      )
)

ggsave(
    "figures/density_plot_explain_in_plain_english.png",
    density_plot_explain_in_plain_english_scores_by_treatment,
    width = 6, height = 4, units = "in"
)

density_plot_explain_in_plain_english_scores_by_treatment
```



7.2 Analysis

7.2.1 Summary Statistics

```
(
    EPE_data
    |> group_by(treatment)
    |> summarise(
           mean = mean(EPE_grade),
           median = median(EPE_grade)
       )
)
```

```
# A tibble: 4 x 3
  treatment        mean median
  <fct>           <dbl>  <dbl>
1 DaysMultiple    0.389 0.444
2 DaysSingle      0.207 0.222
3 WeatherMultiple 0.115 0.0909
4 WeatherSingle   0.266 0.273
```

7.2.2 Significance

```
(
  EPE_data
  |> group_by(treatment)
  |> summarise(out = tidy(shapiro.test(EPE_grade)), .groups = 'drop')
  |> unnest(c(out))
)
```

7.2.2.1 Normality

```
# A tibble: 4 x 4
  treatment       statistic p.value method
  <fct>               <dbl>   <dbl> <chr>
1 DaysMultiple        0.892 0.0603  Shapiro-Wilk normality test
2 DaysSingle          0.776 0.00185 Shapiro-Wilk normality test
3 WeatherMultiple     0.943 0.424   Shapiro-Wilk normality test
4 WeatherSingle       0.851 0.0233  Shapiro-Wilk normality test
```

Groups are not all normal, Wilcoxon rank sum test is performed to check significance.

```
days_multiple_explain_in_plain_english_scores <- (
    EPE_data |> filter(treatment == "DaysMultiple") |> pull(EPE_grade)
)
days_single_explain_in_plain_english_scores <- (
    EPE_data |> filter(treatment == "DaysSingle") |> pull(EPE_grade)
)

weather_multiple_explain_in_plain_english_scores <- (
    EPE_data |> filter(treatment == "WeatherMultiple") |> pull(EPE_grade)
)
weather_single_explain_in_plain_english_scores <- (
    EPE_data |> filter(treatment == "WeatherSingle") |> pull(EPE_grade)
)
```

```
wilcox.test(
```

```
    days_multiple_explain_in_plain_english_scores,
    days_single_explain_in_plain_english_scores,
    alternative = "greater"
)
```

7.2.2.2 Wilcox

```
Warning in wilcox.test.default(days_multiple_explain_in_plain_english_scores, :
cannot compute exact p-value with ties
```

```
    Wilcoxon rank sum test with continuity correction

data:  days_multiple_explain_in_plain_english_scores and days_single_explain_in_plain_english_scores
W = 171, p-value = 0.02018
alternative hypothesis: true location shift is greater than 0
```

```
wilcox.test(
    weather_multiple_explain_in_plain_english_scores,
    weather_single_explain_in_plain_english_scores,
    alternative = "greater"
)
```

```
Warning in
wilcox.test.default(weather_multiple_explain_in_plain_english_scores, : cannot
compute exact p-value with ties
```

```
    Wilcoxon rank sum test with continuity correction

data:  weather_multiple_explain_in_plain_english_scores and weather_single_explain_in_plain_english_sco
W = 49.5, p-value = 0.9934
alternative hypothesis: true location shift is greater than 0
```

7.3 Interpretation

There is a significant increase in the correctness of explain in plain English responses (according to the aggregate code scores) for the days + multiple functions variant, but no significant increase for the weather + multiple functions variant.

There is not enough evidence that functional decomposition improves the ability to comprehend code at small scales — the null hypothesis cannot be rejected.

8 Hypothesis 3: Behaviour Questions

$\mathbf{H}_0^{\text{Beh}}$ The null hypothesis for the behaviour understanding is that the participant's scores to the behaviour questions not depend on the version.

$\mathbf{H}_1^{\text{Beh}}$ The alternative hypothesis is that the participants with the MFV version get higher score than those with the SFV version.

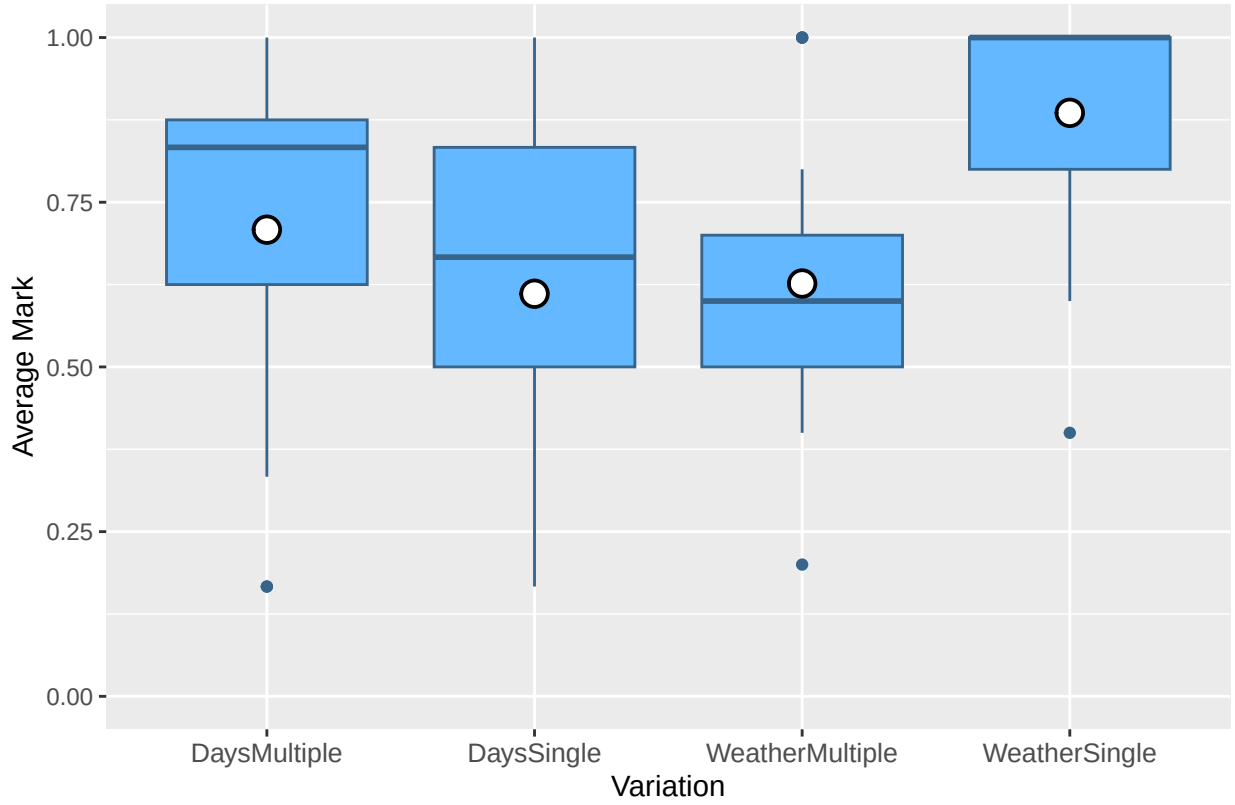
Marks are calculated as the row averages of the scores for behaviour questions

```
behaviour_marks <- (
  data
  |> select(c(participant_id, treatment, matches("^code_trace.*mark$")))
  |> mutate(
        grade = rowMeans(
```

```
            across(code_trace_q1_mark:code_trace_q6_mark),
            na.rm = TRUE
        )
    )
)
```

8.1 Figures

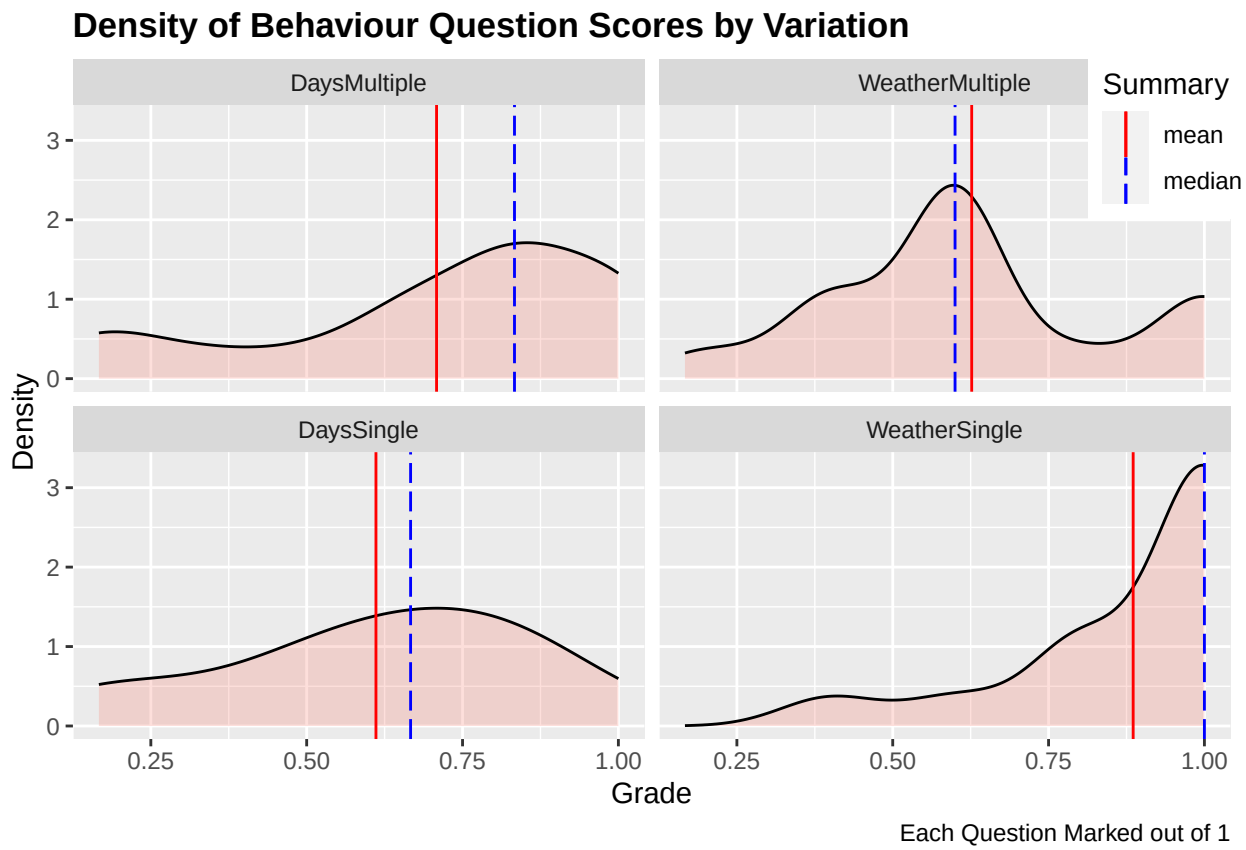
```
box_plot_average_over_behaviour_questions <- (
    ggplot(behaviour_marks, aes(x = treatment, y = grade))
    + labs(
          caption = "White dots show means; Each Question Marked out of 1",
          x = "Variation", y = "Average Mark"
      )
    + geom_boxplot(color = "steelblue4", fill = "steelblue1")
    + ylim(0, 1)
    + stat_summary(
          fun = mean, geom = "point",
          shape = 21, size = 4, stroke = 1,
          color = "black", fill = "white"
      )
    + theme(
          plot.title = element_text(face = "bold"),
          axis.text.x = element_text(size = 10)
      )
)

ggsave(
    "figures/box_plot_behaviour_questions.png",
    box_plot_average_over_behaviour_questions,
    width = 6, height = 2.75, units = "in"
)

box_plot_average_over_behaviour_questions
```



White dots show means; Each Question Marked out of 1

```
averages <- (
    behaviour_marks
    |> group_by(treatment)
    |> summarise(
           mean = mean(grade),
           median = median(grade)
       )
)

density_plot_behaviour_grades_by_treatment <- (
    ggplot(behaviour_marks, aes(x = grade))
    + labs(
          title = "Density of Behaviour Question Scores by Variation",
          caption = "Each Question Marked out of 1",
          x = "Grade", y = "Density",
          color = "Summary", linetype = "Summary",
      )
    + geom_density(alpha = 0.25, fill = "salmon")
    + geom_vline(
          aes(xintercept = mean, color = "mean", linetype = "mean"),
          data = averages,
      )
    + geom_vline(
          aes(xintercept = median, color = "median", linetype = "median"),
          data = averages,
      )
```

```
    + facet_wrap(~treatment, dir = "v")
    + scale_colour_manual(
        name = "Summary", values = c(mean = "red", median = "blue")
      )
    + scale_linetype_manual(
        name = "Summary", values = c(mean = "solid", median = "longdash")
      )
    + theme(
        plot.title = element_text(face = "bold"),
        legend.position = c(0.95, 0.95)
      )
)

ggsave(
    "figures/density_plot_behaviour_questions.png",
    density_plot_behaviour_grades_by_treatment,
    width = 6, height = 4, units = "in"
)

density_plot_behaviour_grades_by_treatment
```



Each Question Marked out of 1

8.2 Analysis

8.2.1 Summary Statistics

```
(
    behaviour_marks
    |> group_by(treatment)
    |> summarise(
           mean = mean(grade),
           median = median(grade)
       )
)
```

```
# A tibble: 4 x 3
  treatment        mean median
  <fct>           <dbl>  <dbl>
1 DaysMultiple    0.708  0.833
2 DaysSingle      0.611  0.667
3 WeatherMultiple 0.627  0.6
4 WeatherSingle   0.886  1
```

8.2.2 Significance

```
(
  behaviour_marks
  |> group_by(treatment)
  |> summarise(out = tidy(shapiro.test(grade)), .groups = 'drop')
  |> unnest(c(out))
)
```

8.2.2.1 Normality

```
# A tibble: 4 x 4
  treatment       statistic  p.value method
  <fct>               <dbl>    <dbl> <chr>
1 DaysMultiple        0.856 0.0168   Shapiro-Wilk normality test
2 DaysSingle          0.924 0.221    Shapiro-Wilk normality test
3 WeatherMultiple     0.883 0.0533   Shapiro-Wilk normality test
4 WeatherSingle       0.681 0.000235 Shapiro-Wilk normality test
```

Groups are not all normal, Wilcoxon rank sum test is performed to check significance.

```
days_multiple_behaviour_scores <- (
    behaviour_marks |> filter(treatment == "DaysMultiple") |> pull(grade)
)
days_single_behaviour_scores <- (
    behaviour_marks |> filter(treatment == "DaysSingle") |> pull(grade)
)

weather_multiple_behaviour_scores <- (
    behaviour_marks |> filter(treatment == "WeatherMultiple") |> pull(grade)
)
weather_single_behaviour_scores <- (
    behaviour_marks |> filter(treatment == "WeatherSingle") |> pull(grade)
```

```
)
```

```
wilcox.test(
    days_multiple_behaviour_scores,
    days_single_behaviour_scores,
    alternative = "greater"
)
```

8.2.2.2 Wilcox

```
Warning in wilcox.test.default(days_multiple_behaviour_scores,
days_single_behaviour_scores, : cannot compute exact p-value with ties


    Wilcoxon rank sum test with continuity correction

data:  days_multiple_behaviour_scores and days_single_behaviour_scores
W = 151, p-value = 0.1088
alternative hypothesis: true location shift is greater than 0
```

```
wilcox.test(
    weather_multiple_behaviour_scores,
    weather_single_behaviour_scores,
    alternative = "greater"
)
```

```
Warning in wilcox.test.default(weather_multiple_behaviour_scores,
weather_single_behaviour_scores, : cannot compute exact p-value with ties


    Wilcoxon rank sum test with continuity correction

data:  weather_multiple_behaviour_scores and weather_single_behaviour_scores
W = 44, p-value = 0.9976
alternative hypothesis: true location shift is greater than 0
```

8.3 Interpretation

There is no significant increase in the correctness of behaviour questions for either of the many function variants — the null hypothesis cannot be rejected.

9 Other Results

Below are charts and summaries of other results from this study which are not directly related to our hypotheses. These are not considered for the discussion or as part of the results, but, regardless, were explored to see if there were any interesting relationships between the treatments.

9.1 Confidence for Code Explanations

Speculative Hypothesis: Participants will be more confidant on the Multiple versions compared to the Single versions.

9.1.1 Figures

```
bar_chart_confidence <- (
    ggplot(data, aes(x = treatment, fill = code_explanation_confidence))
    + labs(x = "Version", y = "Proportion")
    + geom_bar(stat = "count", position = "fill")
    + scale_fill_YlOrBr(name = "Key", discrete = TRUE)
    + theme(plot.title = element_text(face = "bold"))
)

ggsave(
    "figures/bar_chart_confidence.png",
    bar_chart_confidence,
    width = 6, height = 2.5, units = "in"
)
```

9.1.2 Analysis

```
days_multiple_confidence <- (
    data
    |> filter(treatment == "DaysMultiple")
    |> pull(code_explanation_confidence)
)
days_single_confidence <- (
    data
    |> filter(treatment == "DaysSingle")
    |> pull(code_explanation_confidence)
)

weather_multiple_confidence <- (
    data
    |> filter(treatment == "WeatherMultiple")
    |> pull(code_explanation_confidence)
)
weather_single_confidence <- (
    data
    |> filter(treatment == "WeatherSingle")
    |> pull(code_explanation_confidence)
)
```

```
wilcox.test(
    as.numeric(days_multiple_confidence),
    as.numeric(days_single_confidence),
    alternative = "greater"
)
```

```
Warning in wilcox.test.default(as.numeric(days_multiple_confidence),
as.numeric(days_single_confidence), : cannot compute exact p-value with ties


    Wilcoxon rank sum test with continuity correction

data:  as.numeric(days_multiple_confidence) and as.numeric(days_single_confidence)
W = 100, p-value = 0.8156
alternative hypothesis: true location shift is greater than 0
```

```
wilcox.test(
    as.numeric(weather_multiple_confidence),
    as.numeric(weather_single_confidence),
    alternative = "greater"
)
```

```
Warning in wilcox.test.default(as.numeric(weather_multiple_confidence), :
cannot compute exact p-value with ties
```

```

    Wilcoxon rank sum test with continuity correction

data:  as.numeric(weather_multiple_confidence) and as.numeric(weather_single_confidence)
W = 53.5, p-value = 0.9946
alternative hypothesis: true location shift is greater than 0
```

9.1.3 Intepretation

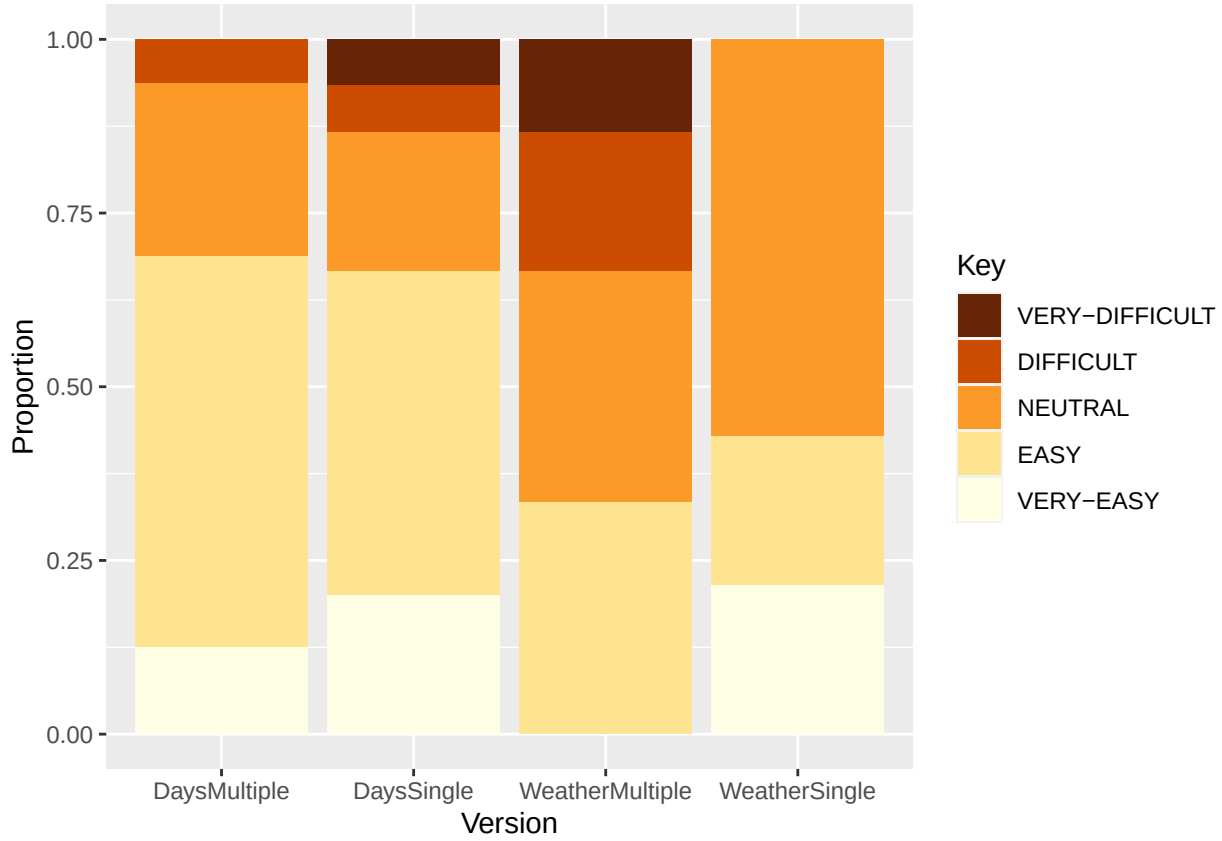
Participants are not more confidant on the Multiple versions compared to the Single versions

9.2 Reported Comprehensibility

Speculative Hypothesis: Students will find the Multiple function variants easier to understand compared to the Single function variants.

9.2.1 Figures

```
bar_chart_comprehensibility <- (
    ggplot(data, aes(x = treatment, fill = code_understanding))
    + labs(x = "Version", y = "Proportion")
    + geom_bar(stat = "count", position = "fill")
    + scale_fill_YlOrBr(name = "Key", discrete = TRUE, reverse = TRUE)
)

ggsave(
    "figures/bar_chart_comprehensibility.png",
    bar_chart_comprehensibility,
    width = 6, height = 2.5, units = "in"
)

bar_chart_comprehensibility
```



9.2.2 Analysis

```
days_multiple_understanding <- (
    data |> filter(treatment == "DaysMultiple") |> pull(code_understanding)
)
days_single_understanding <- (
    data |> filter(treatment == "DaysSingle") |> pull(code_understanding)
)

weather_multiple_understanding <- (
    data |> filter(treatment == "WeatherMultiple") |> pull(code_understanding)
)
weather_single_understanding <- (
    data |> filter(treatment == "WeatherSingle") |> pull(code_understanding)
)
```

```
wilcox.test(
    as.numeric(days_multiple_understanding),
    as.numeric(days_single_understanding),
    alternative = "greater"
)
```

```
Warning in wilcox.test.default(as.numeric(days_multiple_understanding), :
cannot compute exact p-value with ties


    Wilcoxon rank sum test with continuity correction
```

```
data:  as.numeric(days_multiple_understanding) and as.numeric(days_single_understanding)
W = 119, p-value = 0.5257
alternative hypothesis: true location shift is greater than 0
```

```
wilcox.test(
    as.numeric(weather_multiple_understanding),
    as.numeric(weather_single_understanding),
    alternative = "greater"
)
```

```
Warning in wilcox.test.default(as.numeric(weather_multiple_understanding), :
cannot compute exact p-value with ties


    Wilcoxon rank sum test with continuity correction

data:  as.numeric(weather_multiple_understanding) and as.numeric(weather_single_understanding)
W = 67.5, p-value = 0.9608
alternative hypothesis: true location shift is greater than 0
```

9.2.3 Interpretation

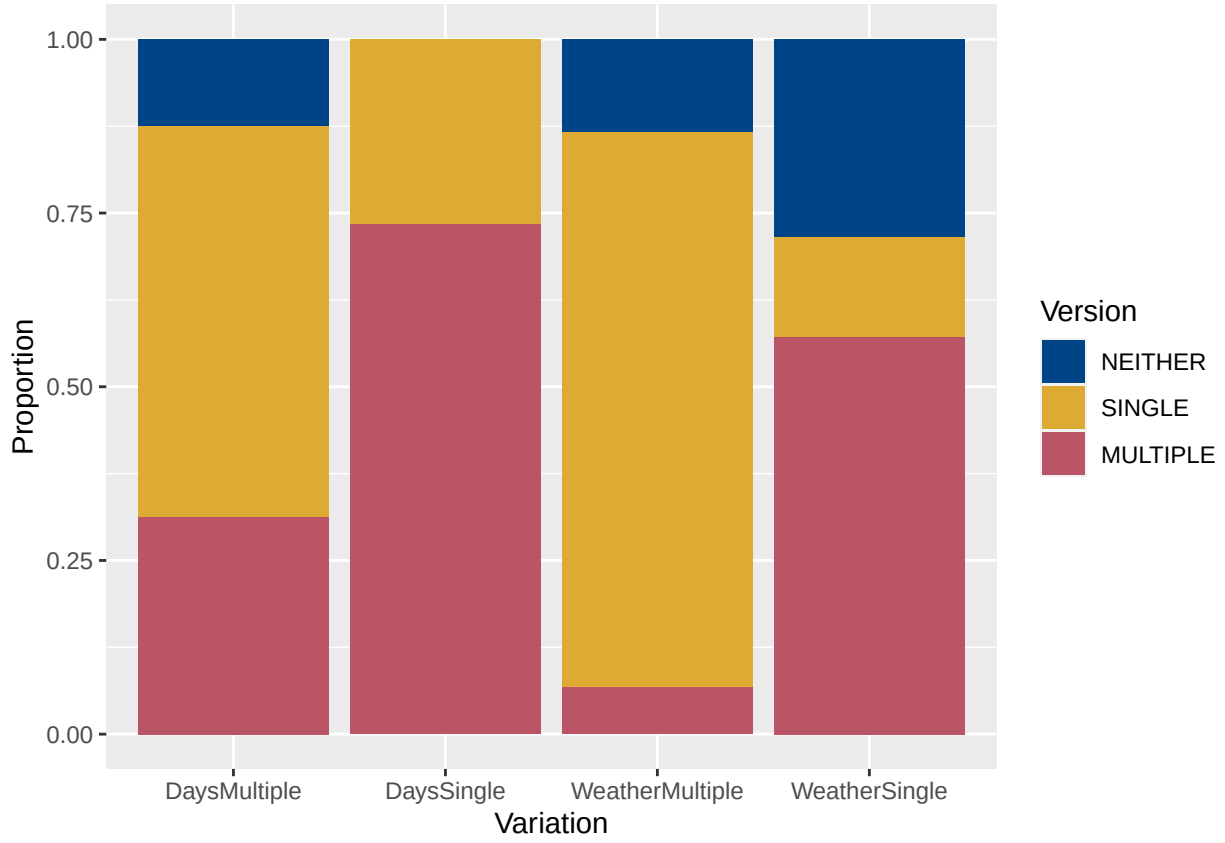
Participants do not find it easier to read the Multiple function versions compared to the Single function versions

9.3 After-the-fact Preference Between Versions

Speculative Hypothesis: Students will prefer the Multiple functions versions over the Single functions versions regardless of treatment group.

9.3.1 Figures

```
bar_chart_version <- (
    ggplot(data, aes(x = treatment, fill = code_compare_preference))
    + labs(x = "Variation", y = "Proportion")
    + geom_bar(stat = "count", position = "fill")
    + scale_fill_highcontrast(name = "Version")
)

ggsave(
    "figures/bar_chart_version.png",
    bar_chart_version,
    width = 6, height = 2.5, units = "in"
)

bar_chart_version
```



9.3.2 Interpretation

Students appear to prefer the opposite treatment to the treatment they were in.

10 Conclusions

At small scales (of the order of tens of lines of code), there appears to be no improvement in the comprehensibility of code when it it decomposed into multiple functions. We have seen no reduction in code reading time, no consistent improvement in participants' ability to explain code, and no improvement in the correctness of their answers to behaviour questions (tracing, reverse tracing, speculative code modification, etc.).

This has implications for educators who typically use small examples to demonstrate programming principles and best practice (such as modularization). It might be that at small scales there are no obvious benefits to such practices; however, in educational contexts it is not feasible to use large-scale examples where such practices are likely to benefit comprehensibility, as this is likely to overload students. New methods of teaching the effectiveness of decomposition may be needed. Further studies exploring the effect of functional decomposition at larger scales are needed to fully understand its relationship to code comprehension.

> For practical reasons, the demonstration programs must be small, many times smaller than the "life-size programs" I have in mind. My basic problem is that precisely this difference in scale is one of the major sources of our difficulties in programming!
>
> It would be very nice if I could illustrate the various techniques with small demonstration programs and could conclude with "... and when faced with a program a thousand times as large, you compose it in the same way." This common educational device, however, would be self-defeating as one of my central themes will be that any two things that differ in some respect by a factor of already a hundred or more, are utterly incomparable.

— Dijkstra (1970), *On our inability to do much*