

Appendix B

Complete R Code & Annotated Output

Compiled: September 01, 2024

Contents

BACKGROUND	105
Motivation	105
Research Questions	105
Sample	106
Variables	106
PREPARATION	107
Load Packages	107
Default Settings	107
DATA WRANGLING	108
Load Data	108
Declare Factors	108
SUMMARY STATISTICS	110
Sample Size	110
Missing Data	111
Descriptive Summary	112
Correlations	114
VISUALIZATIONS	122
Univariable	122
Bivariable	146
REGRESSION ANALYSIS	149
Model Fitting	149
Final Model	157
FINAL RESULTS	162
RQ (1)	162
RQ (2)	167
RQ (3)	171
SESSION INFORMATION	178
BIBLIOGRAPHY	180

List of Tables

B.1	Illustration of the Collected Data	109
B.2	Sample Size by Program	110
B.3	Five-Number Summary for Quantitative Measures	112
B.4	Descriptive Summary of Sample, by Program	113
B.5	Correlation Matrix for Quantitative Measures, Pearson's Product Moment Correlation (r)	114
B.6	Correlations for Quantitative Measure, Pearson's Product Moment Correlation (r) and Statistical Significance (p)	115
B.7	Correlations for Quantitative Measure, Pearson's Product Moment Correlation, by Program and Gender	116
B.8	Parameter Estimates of Nested Regression Models for Weight Loss	151
B.9	Comparison of Model Fit for Nested Regression Models for Weight Loss	152
B.10	Final Regression Model for Weight Loss, Analysis of Variance and Significance of All Terms	157
B.11	Simple Slopes Analysis for Weight Loss by Time Spent in each Program by Gender, Collapsed Across Effort	159
B.12	RQ (1) Inverse Estimation of Exercise Time Required by Each Program for a Five Pound Weight Loss, Given Low Effort	163
B.13	RQ (2) Additional Weight Loss per Hour/Week Spent on Each Exercise Program with Average Effort, by Gender	167
B.14	RQ (3) Additional Weight Loss per Hour/Week Spent on Each Exercise Program with Near Maximum Effort, by Gender	172
B.15	RQ (3) Additional Weight Loss per Hour/Week Spent on Each Exercise Program with Near Max vs. Mean Effort, by Gender	174

List of Figures

B.1	Amount of Missing Data by Variable	111
B.2	Correlation Plot for Program Time, Effort Rating, and Weight Loss, Pearson's Product Moment Correlation (r) Indication Statistical Significance - Option A	117
B.3	Correlation Plot for Program Time, Effort Rating, and Weight Loss, Pearson's Product Moment Correlation (r) Indication Statistical Significance - Option B	118
B.4	Correlation Plot for Program Time, Effort Rating, and Weight Loss, Pearson's Product Moment Correlation (r) Indication Statistical Significance, for the Jogging Program	119
B.5	Correlation Plot for Program Time, Effort Rating, and Weight Loss, Pearson's Product Moment Correlation (r) Indication Statistical Significance, for the Swimming Running	120
B.6	Correlation Plot for Program Time, Effort Rating, and Weight Loss, Pearson's Product Moment Correlation (r) Indication Statistical Significance, for the Reading Running	121
B.7	Distribution of Weight Loss	122
B.8	Distribution of Weight Loss, By Program - Option A	124
B.9	Distribution of Weight Loss, By Program - Option B	125
B.10	Distribution of Weight Loss, By Program and Gender - Option A	127
B.11	Distribution of Weight Loss, By Program and Gender - Option B	129
B.12	Distribution of Weight Loss, By Program and Gender - Option C	130
B.13	Distribution of Program Time	131
B.14	Distribution of Program Time, By Program - Option A	132
B.15	Distribution of Program Time, By Program - Option B	133
B.16	Distribution of Program Time, By Program and Gender - Option A	135
B.17	Distribution of Program Time, By Program and Gender - Option B	137
B.18	Distribution of Program Time, By Program and Gender - Option C	138
B.19	Distribution of Effort Rating	139
B.20	Distribution of Effort Rating, By Program - Option A	140
B.21	Distribution of Effort Rating, By Program - Option B	141
B.22	Distribution of Effort Rating, By Program and Gender - Option A	143
B.23	Distribution of Effort Rating, By Program and Gender - Option B	144
B.24	Distribution of Effort Rating, By Program and Gender - Option C	145
B.25	Association Between Weight Loss by Program Time, by Gender and Program	146
B.26	Association Between Weight Loss by Effort Rating, by Gender and Program	147
B.27	Association Between Effort Rating by Program Time, by Gender and Program	148
B.28	Residual Diagnostics, Assess Normality of the Conditional Distribution of Weight Loss, QQ Plot	154
B.29	Residual Diagnostics, Assess Normality of the Conditional Distribution of Weight Loss, Histogram	154
B.30	Residual Diagnostics, Assess Various Assumptions of Regression, scatterplot	155

B.31 Residual Diagnostics, Identifying Influential Data Points, Cook's D Chart	156
B.32 Weight Loss Uniquely Moderated by Program Time and Effort Rating According to Gender and Program Type	161
B.33 RQ (1) Compare Exercise Time Required by Each Program for a Five Pound Weight Loss, Given Low Effort	165
B.34 RQ (2) Compare Additional Weight Loss per Hour/Week Spent on Each Exercise Program with Average Effort, by Gender	169
B.35 RQ (3) Compare Effort Discrepancies in Additional Weight Loss Associated with an Hour Increase per Week in Each Exercise Program, by Gender	176

BACKGROUND

This dataset is publicly available online as part of a seminar on “DECOMPOSING, PROBING, AND PLOTTING INTERACTIONS IN R” - UCLA Advanced Research Computing: Statistical Methods and Data Analysis

Motivation

Suppose you are doing a simple study on weight loss and notice that people who spend more time exercising lose more weight. Upon further analysis you notice that those who spend the same amount of time exercising lose more weight if they are more effortful. The more effort people put into their workouts, the less time they need to spend exercising. This is popular in workouts like high intensity interval training (HIIT).

Research Questions

You know that hours spent exercising improves weight loss, but how does it interact with effort? Here are three questions you can ask based on hypothetical scenarios.

RQ (1)

I'm just starting out and don't want to put in too much effort.

How many hours per week of exercise do I need to put in to lose 5 pounds?

RQ (2)

I'm moderately fit and can put in an average level of effort into my workout.

For every one hour increase per week in exercise, how much additional weight loss do I expect?

RQ (3)

I'm a crossfit athlete and can perform with the utmost intensity.

How much more weight loss would I expect for every one hour increase in exercise compared to the average amount of effort most people put in?

Additionally, we can visualize the interaction to help us understand these relationships.

Sample

This is a hypothetical study of weight loss for **900 participants** in a year-long study of **3 different exercise programs**, a jogging program, a swimming program, and a reading program which serves as a control activity.

Variables

- **prog**
 - exercise program
 - categorical, nominal
 - 1 = jogging
 - 2 = swimming
 - 3 = reading
- **gender**
 - participant gender, self-report
 - categorical, binary
 - 1 = Male
 - 2 = Female
- **hours**
 - hours spent in program per week
 - continuous, interval-ratio
- **effort**
 - self-report effort during program
 - continuous, interval-ratio
 - 0 = minimal physical effort
 - 50 = maximum effort
- **loss**
 - final weight loss, pounds
 - continuous, interval-ratio
 - positive = weight loss
 - negative scores = weight gain

PREPARATION

Load Packages

```
library(devtools)
devtools::install_github("SarBearSchwartz/apaSupp")
```

```
library(apaSupp)
library(tidyverse)
library(furniture)
library(psych)
library(flextable)
library(naniar)
library(rstatix)
library(ggpubr)
library(MOTE)
library(olsrr)
library(DHARMa)
library(texreg)
library(emmeans)
library(interactions)
library(performance)
```

Default Settings

```
options(dplyr.summarise.inform = FALSE)

flextable::set_flextable_defaults(theme_fun = "theme_apa",
                                  font.size = 10,
                                  layout = "autofit")

ggplot2::theme_set(theme_bw())
```


DATA WRANGLING

Load Data

This data is available online.

```
df_raw <- read.csv("https://stats.idre.ucla.edu/wp-content/uploads/2019/03/exercise.csv")
```

```
tibble::glimpse(df_raw)
```

Rows: 900

Columns: 6

```
$ id      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, ~
$ loss    <dbl> 18.022263, 10.186416, 19.747276, 1.883600, 14.242589, 19.694731~
$ hours    <dbl> 1.836704, 2.389360, 2.362117, 2.520866, 1.889828, 2.367162, 1.9~
$ effort   <dbl> 37.71218, 26.72401, 36.31657, 20.70048, 24.72712, 33.66948, 31.~
$ gender   <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
$ prog     <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
```

Declare Factors

```
df_use <- df_raw %>%
  dplyr::mutate(id = factor(id)) %>%
  dplyr::mutate(gender = factor(gender) %>%
    forcats::fct_recode("Male" = "1",
                        "Female" = "2")) %>%
  dplyr::mutate(prog = factor(prog) %>%
    forcats::fct_recode("Jogging" = "1",
                        "Swimming" = "2",
                        "Reading" = "3"))
```

```
tibble::glimpse(df_use)
```

Rows: 900

Columns: 6

```
$ id      <fct> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, ~
$ loss    <dbl> 18.022263, 10.186416, 19.747276, 1.883600, 14.242589, 19.694731~
$ hours    <dbl> 1.836704, 2.389360, 2.362117, 2.520866, 1.889828, 2.367162, 1.9~
$ effort   <dbl> 37.71218, 26.72401, 36.31657, 20.70048, 24.72712, 33.66948, 31.~
$ gender   <fct> Male, Male, Male, Male, Male, Male, Male, Male, Male, Mal~
$ prog     <fct> Jogging, Jogging, Jogging, Jogging, Jogging, Jogging, Jogging, ~
```

```
df_use %>%
  psych::headTail(top = 6) %>%
  flextable::flextable() %>%
  flextable::set_caption("Illustration of the Collected Data")
```

Table B.1*Illustration of the Collected Data*

id	loss	hours	effort	gender	prog
1	18.02	1.84	37.71	Male	Jogging
2	10.19	2.39	26.72	Male	Jogging
3	19.75	2.36	36.32	Male	Jogging
4	1.88	2.52	20.7	Male	Jogging
5	14.24	1.89	24.73	Male	Jogging
6	19.69	2.37	33.67	Male	Jogging
		
897	-2.25	2.6	28.11	Female	Reading
898	-13.37	2.84	27.91	Female	Reading
899	-2.78	1.44	29.75	Female	Reading
900	-5.6	0.86	40	Female	Reading

SUMMARY STATISTICS

Sample Size

This experiment is hypothetical and the data was simulated, thus has no inclusion or exclusion criteria.

Note: Program was randomized within gender to ensure balance.

```
df_use %>%
  furniture::table1("Gender, observed" = gender,
                    splitby = ~ prog,
                    total = TRUE,
                    na.rm = FALSE,
                    output = "markdown",
                    digits = 2,
                    caption = "Sample Size by Program",
                    export = "tab_sample_size")
```

Table B.2

Sample Size by Program

	Total	Jogging	Swimming	Reading
	n = 900	n = 300	n = 300	n = 300
Gender, observed				
Male	450 (50%)	150 (50%)	150 (50%)	150 (50%)
Female	450 (50%)	150 (50%)	150 (50%)	150 (50%)
NA	0 (0%)	0 (0%)	0 (0%)	0 (0%)

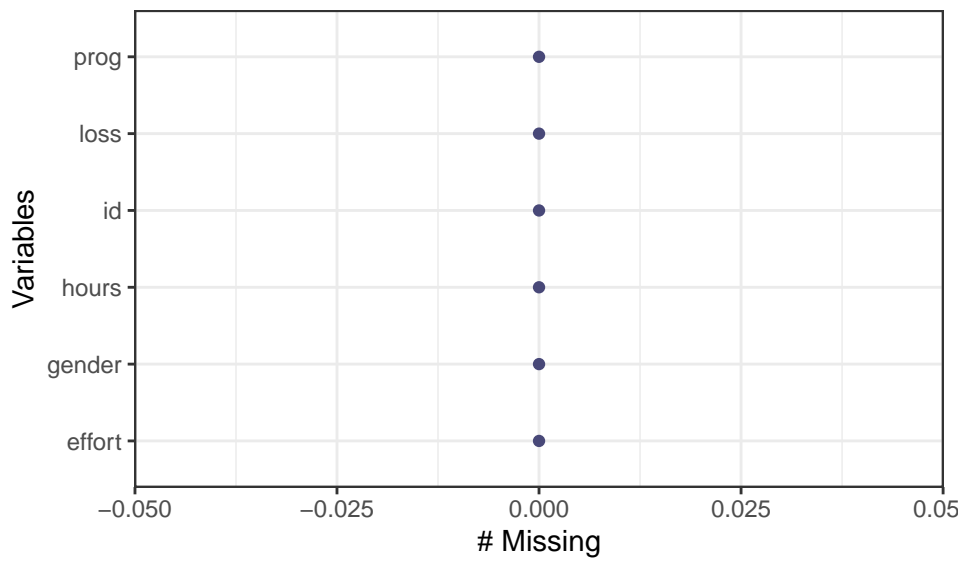
Missing Data

This experiment is hypothetical and the data was simulated, thus has no missing data.

```
df_use %>%  
  naniar::gg_miss_var() +  
  theme_bw()
```

Figure B.1

Amount of Missing Data by Variable



Descriptive Summary

```

tab_5numsum <- df_use %>%
  dplyr::select_if(is.numeric) %>%
  psych::describe(quant = c(.25, .75)) %>%
  data.frame() %>%
  dplyr::mutate(n = as.character(n)) %>%
  tibble::rownames_to_column(var = "Variable") %>%
  dplyr::mutate_if(is.numeric, apaSupp::apa2) %>%
  dplyr::select(Variable,
               "n" = n,
               "Min" = min,
               "Q1" = Q0.25,
               "Mdn" = median,
               "Q3" = Q0.75,
               "Max" = max) %>%
  flextable::flextable() %>%
  flextable::set_caption("Five-Number Summary for Quantitative Measures")

tab_5numsum

```

Table B.3

Five-Number Summary for Quantitative Measures

Variable	n	Min	Q1	Mdn	Q3	Max
loss	900	-17.14	-1.74	7.88	20.05	54.15
hours	900	0.18	1.68	2.01	2.34	4.07
effort	900	12.95	26.26	29.63	33.10	44.08

```
df_use %>%
  furniture::table1("Gender" = gender,
                    "Weight Loss, pounds" = loss,
                    "Program Use, hours" = hours,
                    "Program Effort, rating" = effort,
                    splitby = ~ prog,
                    total = TRUE,
                    test = TRUE,
                    # na.rm = FALSE,
                    output = "markdown",
                    digits = 2,
                    caption = "Descriptive Summary of Sample, by Program",
                    export = "tab_descriptives")
```

Table B.4*Descriptive Summary of Sample, by Program*

	Total	Jogging	Swimming	Reading	P-Value
	n = 900	n = 300	n = 300	n = 300	
Gender					1
Male	450 (50%)	150 (50%)	150 (50%)	150 (50%)	
Female	450 (50%)	150 (50%)	150 (50%)	150 (50%)	
Weight Loss, pounds					<.001
	10.02 (14.10)	8.03 (7.45)	25.82 (8.92)	-3.79 (4.11)	
Program Use, hours					0.469
	2.00 (0.49)	1.99 (0.47)	1.99 (0.48)	2.03 (0.53)	
Program Effort, rating					0.18
	29.66 (5.14)	30.05 (5.04)	29.65 (5.17)	29.28 (5.20)	

Correlations

```

tab_cor <- df_use %>%
  dplyr::select("Program Time" = hours,
               "Effort Rating" = effort,
               "Weight Loss" = loss) %>%
  rstatix::cor_mat() %>%
  rstatix::pull_lower_triangle() %>%
  dplyr::rename(Variables = rowname) %>%
  flextable::flextable() %>%
  flextable::width(j = 1, width = 2, unit = "in") %>%
  flextable::width(j = 2:4, width = 1, unit = "in") %>%
  flextable::colformat_double(digits = 3) %>%
  flextable::set_caption("Correlation Matrix for Quantitative Measures, Pearson's Product Moment

tab_cor

```

Table B.5

Correlation Matrix for Quantitative Measures, Pearson's Product Moment Correlation (r)

Variables	Program Time	Effort Rating	Weight Loss
Program Time			
Effort Rating	0.016		
Weight Loss	0.087	0.26	

```

tab_cor_gather <- df_use %>%
  dplyr::select("Hours" = hours,
               "Effort" = effort,
               "Loss" = loss) %>%
  rstatix::cor_mat() %>%
  rstatix::cor_gather() %>%
  dplyr::mutate(p = apaSupp::p_num(p)) %>%
  dplyr::filter(cor < 1) %>%
  dplyr::group_by(cor, p) %>%
  dplyr::slice(1) %>%
  dplyr::ungroup() %>%
  tidyr::unite(col = "Variables", var1, var2, sep = " & ") %>%
  dplyr::rename("r" = "cor") %>%
  flextable::flextable() %>%
  flextable::width(j = 1, width = 2, unit = "in") %>%
  flextable::width(j = 2:3, width = 1.5, unit = "in") %>%
  flextable::colformat_double(digits = 3) %>%
  flextable::set_caption("Correlations for Quantitative Measure, Pearson's Product Moment Correlation (r) and Statistical Significance (p)")

tab_cor_gather

```

Table B.6

Correlations for Quantitative Measure, Pearson's Product Moment Correlation (r) and Statistical Significance (p)

Variables	r	p
Effort & Hours	0.016	.626
Loss & Hours	0.087	.009 **
Loss & Effort	0.260	< .001 ***


```

tab_cor_split <- df_use %>%
  dplyr::group_by(prog, gender) %>%
  dplyr::select("Hours" = hours,
               "Effort" = effort,
               "Loss" = loss) %>%

  tidyr::nest() %>%
  dplyr::mutate(cor = map(data,
                          ~ rstatix::cor_mat(.x) %>%
                            rstatix::cor_gather())) %>%

  tidyr::unnest(cor) %>%
  dplyr::select(-data) %>%
  dplyr::mutate(p = apaSupp::p_num(p)) %>%
  dplyr::filter(cor < 1) %>%
  dplyr::group_by(prog, gender, cor, p) %>%
  dplyr::slice(1) %>%
  dplyr::ungroup() %>%
  tidyr::unite(col = "Variables", var1, var2, sep = " & ") %>%
  dplyr::mutate(text = glue::glue("{apaSupp::apa3(cor)} ({p})")) %>%
  dplyr::select(-cor, -p) %>%
  tidyr::pivot_wider(names_from = Variables,
                    values_from = text) %>%

  flextable::flextable() %>%
  flextable::width(j = 3:5, width = 1.3, unit = "in") %>%
  flextable::set_caption("Correlations for Quantitative Measure, Pearson's Product Moment Correlation")

tab_cor_split

```

Table B.7

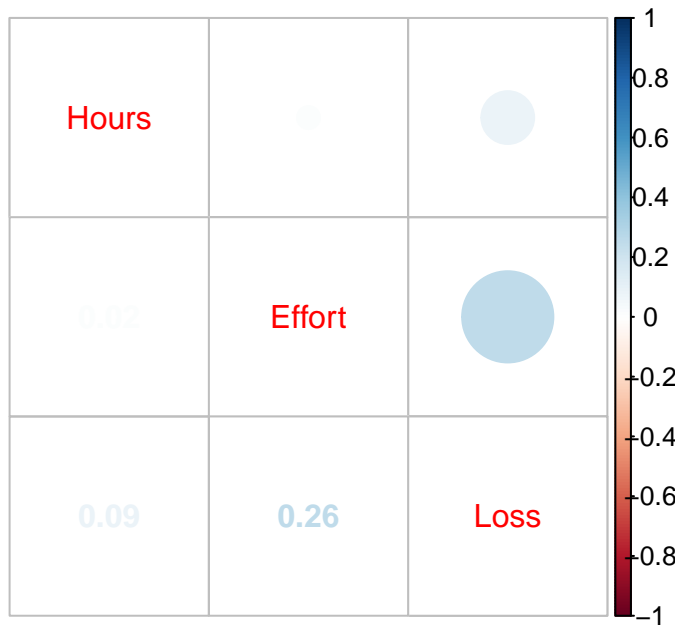
Correlations for Quantitative Measure, Pearson's Product Moment Correlation, by Program and Gender

prog	gender	Effort & Hours	Loss & Hours	Loss & Effort
Jogging	Male	0.180 (.030 *)	0.350 (< .001 ***)	0.740 (< .001 ***)
Jogging	Female	-0.041 (.616)	0.650 (< .001 ***)	0.440 (< .001 ***)
Swimming	Male	-0.049 (.550)	0.510 (< .001 ***)	0.700 (< .001 ***)
Swimming	Female	-0.059 (.473)	0.260 (.001 **)	0.780 (< .001 ***)
Reading	Male	0.036 (.661)	-0.470 (< .001 ***)	0.093 (.258)
Reading	Female	0.037 (.649)	-0.300 (< .001 ***)	0.037 (.654)

```
df_use %>%
  dplyr::select("Hours" = hours,
                "Effort" = effort,
                "Loss" = loss) %>%
  cor() %>%
  corrrplot::corrplot.mixed()
```

Figure B.2

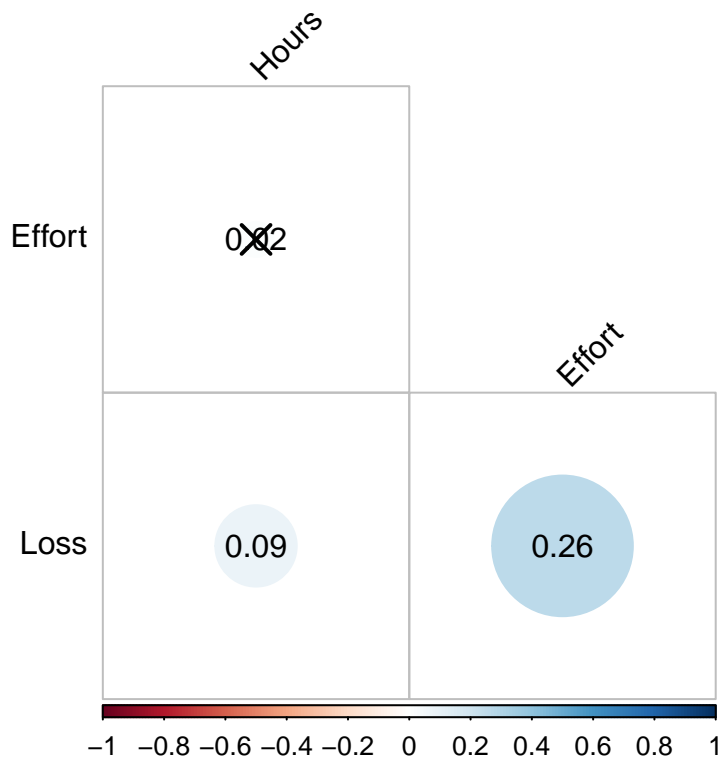
Correlation Plot for Program Time, Effort Rating, and Weight Loss, Pearson's Product Moment Correlation (r) Indication Statistical Significance - Option A



```
df_use %>%
  dplyr::select("Hours" = hours,
                "Effort" = effort,
                "Loss" = loss) %>%
  rstatix::cor_mat() %>%
  rstatix::cor_reorder() %>%
  rstatix::pull_lower_triangle() %>%
  rstatix::cor_plot(label = TRUE)
```

Figure B.3

Correlation Plot for Program Time, Effort Rating, and Weight Loss, Pearson's Product Moment Correlation (r) Indication Statistical Significance - Option B



```
df_use %>%
  dplyr::filter(prog == "Jogging") %>%
  dplyr::select("Hours" = hours,
               "Effort" = effort,
               "Loss" = loss) %>%
  rstatix::cor_mat() %>%
  rstatix::cor_reorder() %>%
  rstatix::pull_lower_triangle() %>%
  rstatix::cor_plot(label = TRUE)
```

Figure B.4

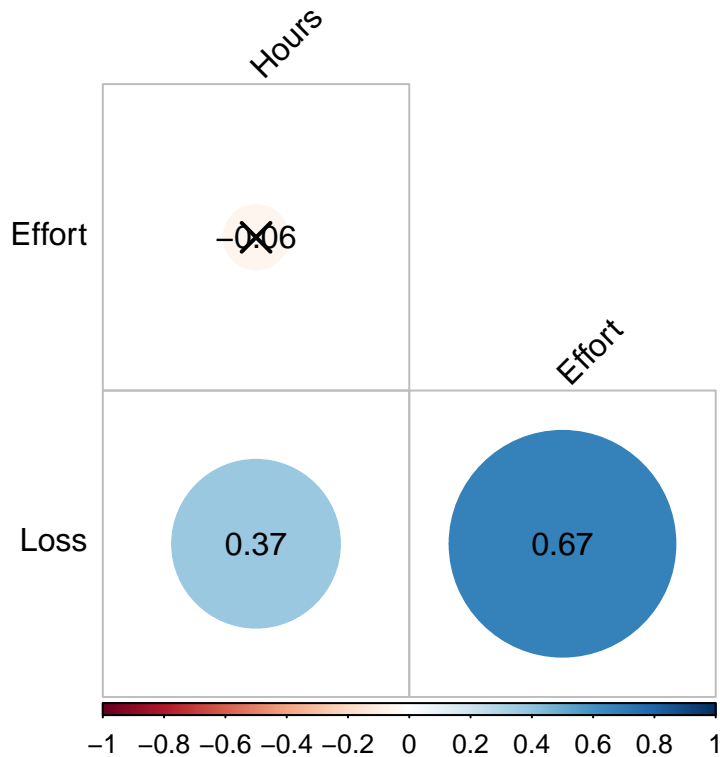
Correlation Plot for Program Time, Effort Rating, and Weight Loss, Pearson's Product Moment Correlation (r) Indication Statistical Significance, for the Jogging Program



```
df_use %>%
  dplyr::filter(prog == "Swimming") %>%
  dplyr::select("Hours" = hours,
               "Effort" = effort,
               "Loss" = loss) %>%
  rstatix::cor_mat() %>%
  rstatix::cor_reorder() %>%
  rstatix::pull_lower_triangle() %>%
  rstatix::cor_plot(label = TRUE)
```

Figure B.5

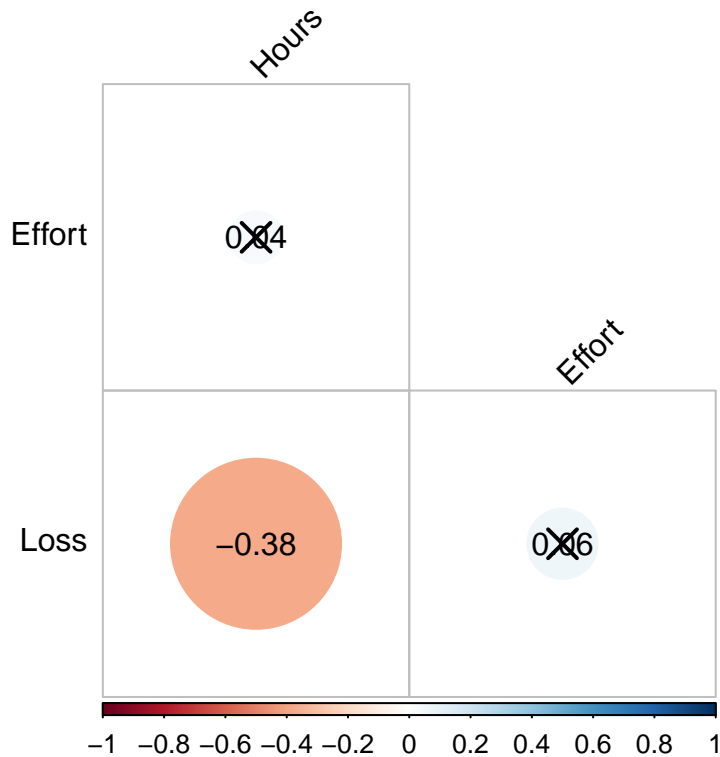
Correlation Plot for Program Time, Effort Rating, and Weight Loss, Pearson's Product Moment Correlation (r) Indication Statistical Significance, for the Swimming Running



```
df_use %>%
  dplyr::filter(prog == "Reading") %>%
  dplyr::select("Hours" = hours,
               "Effort" = effort,
               "Loss" = loss) %>%
  rstatix::cor_mat() %>%
  rstatix::cor_reorder() %>%
  rstatix::pull_lower_triangle() %>%
  rstatix::cor_plot(label = TRUE)
```

Figure B.6

Correlation Plot for Program Time, Effort Rating, and Weight Loss, Pearson's Product Moment Correlation (r) Indication Statistical Significance, for the Reading Running



VISUALIZATIONS

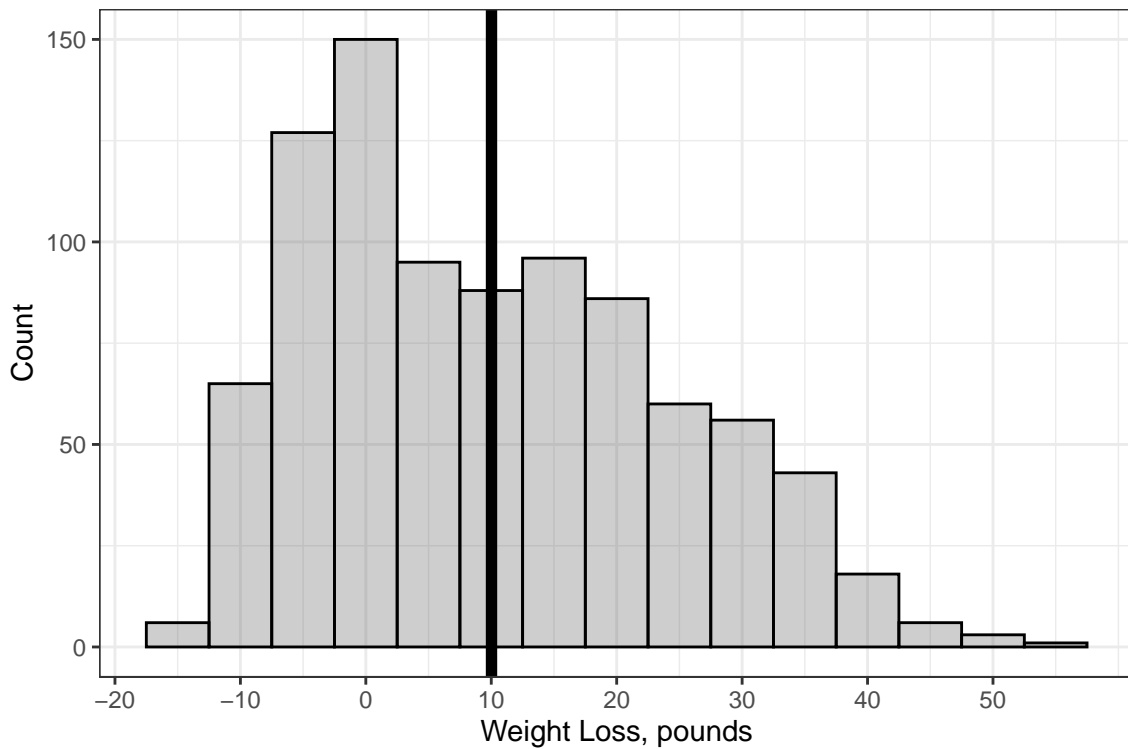
Univariable

Weight Loss

```
df_use %>%
  ggplot(aes(loss)) +
  geom_histogram(color = "black",
                 alpha = .3,
                 binwidth = 5) +
  stat_summary(aes(xintercept = ..x..,
                   y = 0),
               fun = mean,
               geom = "vline",
               orientation = "y",
               linewidth = 2) +
  labs(x = "Weight Loss, pounds",
       y = "Count") +
  scale_x_continuous(breaks = seq(from = -20, to = 50, by = 10))
```

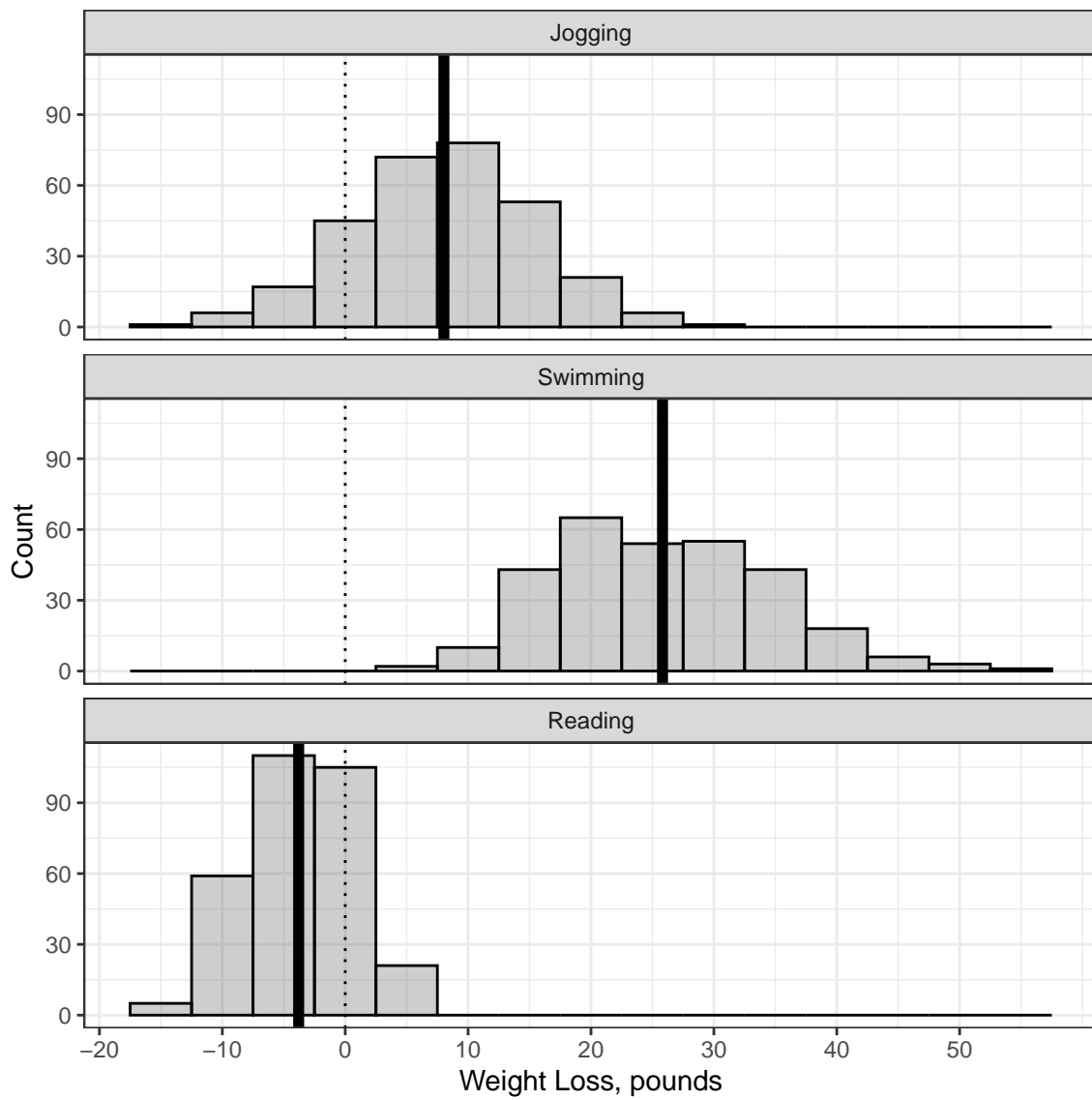
Figure B.7

Distribution of Weight Loss

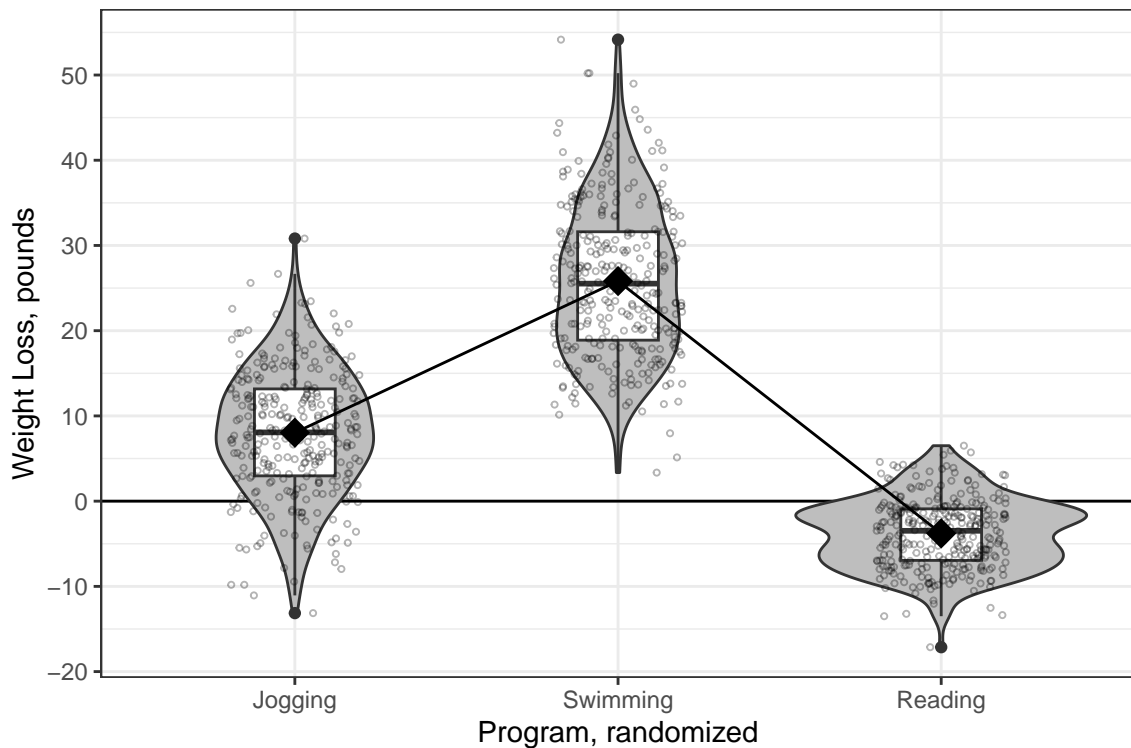


```
df_use %>%  
  ggplot(aes(loss)) +  
  geom_histogram(color = "black",  
                 alpha = .3,  
                 binwidth = 5) +  
  stat_summary(aes(xintercept = ..x.., y = 0),  
               fun = mean,  
               geom = "vline",  
               orientation = "y",  
               linewidth = 2) +  
  labs(x = "Weight Loss, pounds",  
        y = "Count") +  
  facet_wrap(~ prog, ncol = 1) +  
  geom_vline(xintercept = 0, color = "black", linetype = "dotted") +  
  scale_x_continuous(breaks = seq(from = -20, to = 50, by = 10))
```


Figure B.8
Distribution of Weight Loss, By Program - Option A



```
df_use %>%
  ggplot(aes(x = prog,
             y = loss)) +
  geom_hline(yintercept = 0, color = "black") +
  geom_violin(fill = "gray") +
  geom_boxplot(width = .25) +
  geom_jitter(size = .75,
              shape = 1,
              alpha = .3,
              width = .2,
              height = 0) +
  stat_summary(fun = "mean",
              geom = "point",
              shape = 18,
              size = 5) +
  stat_summary(aes(group = 1),
              fun = "mean",
              geom = "line") +
  labs(x = "Program, randomized",
       y = "Weight Loss, pounds") +
  scale_y_continuous(breaks = seq(from = -20, to = 50, by = 10))
```

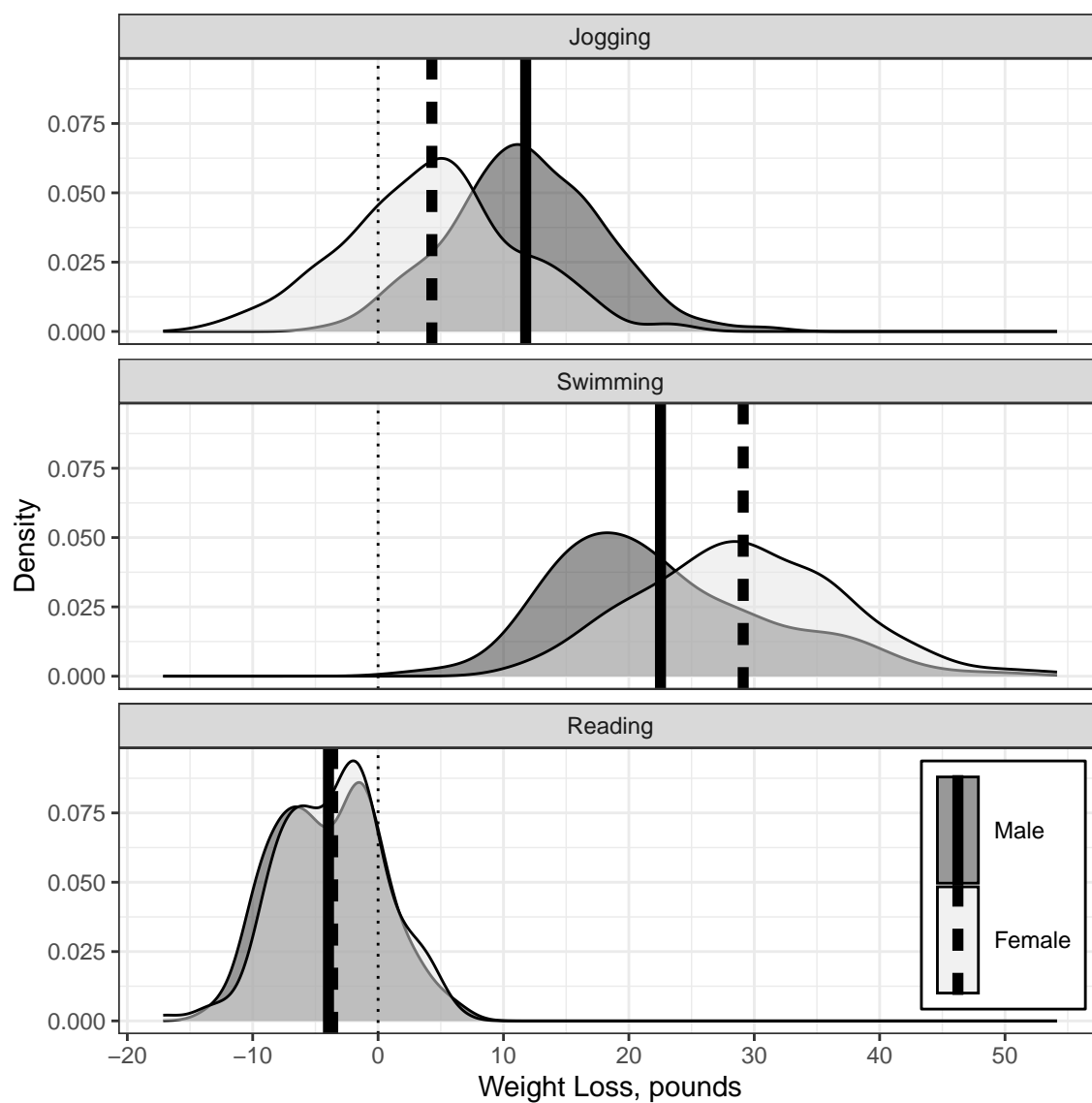
Figure B.9*Distribution of Weight Loss, By Program - Option B*

```

df_use %>%
  ggplot(aes(loss,
              fill = gender)) +
  geom_density(alpha = .5) +
  stat_summary(aes(xintercept = ..x..,
                  y = 0,
                  linetype = gender),
              fun = mean,
              geom = "vline",
              orientation = "y",
              linewidth = 2) +
  geom_vline(xintercept = 0,
            color = "black",
            linetype = "dotted") +
  labs(x = "Weight Loss, pounds",
       y = "Density",
       fill = NULL,
       color = NULL,
       linetype = NULL)+
  facet_wrap(~ prog, ncol = 1) +
  scale_fill_manual(values = c("gray20", "gray90")) +
  scale_x_continuous(breaks = seq(from = -20, to = 50, by = 10)) +
  theme(legend.position = c(1, 0),
        legend.justification = c(1.1, -.1),
        legend.background = element_rect(color = "black"),
        legend.key.height = unit(1.5, "cm"))

```

Figure B.10
Distribution of Weight Loss, By Program and Gender - Option A



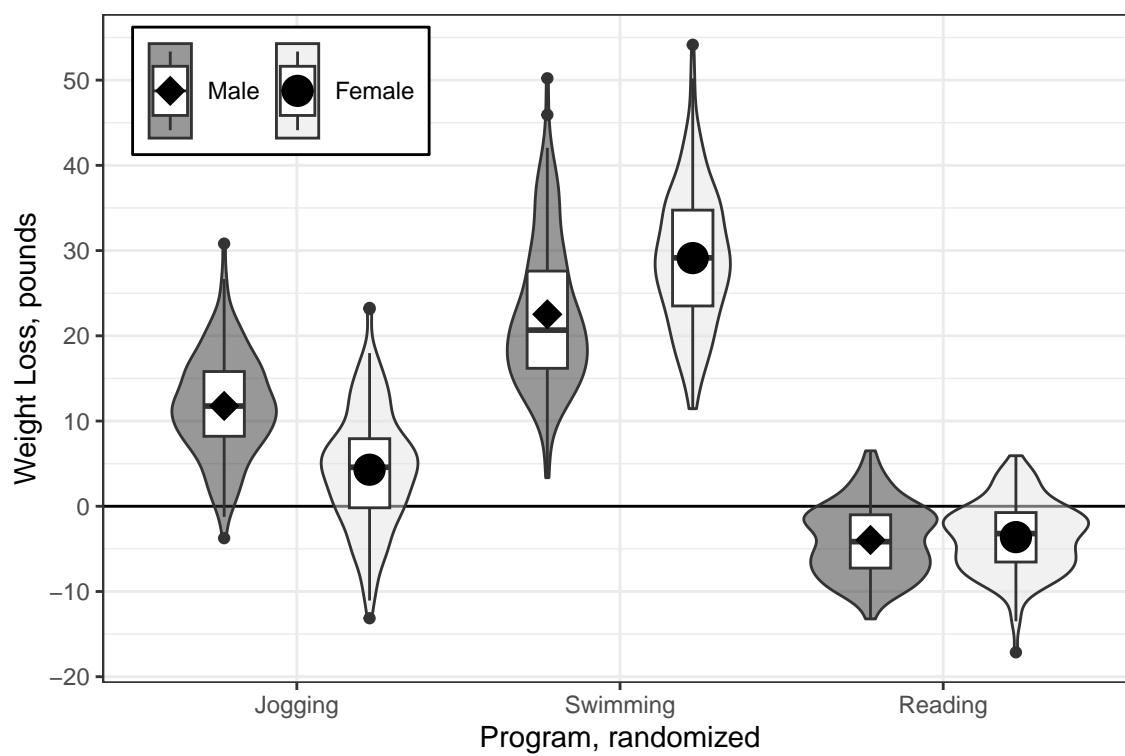
```

df_use %>%
  ggplot(aes(x = prog,
             y = loss,
             fill = gender,
             shape = gender,
             group = interaction(prog,gender))) +
  geom_hline(yintercept = 0, color = "black") +
  geom_violin(alpha = .5) +
  geom_boxplot(fill = "white",
              width = .25,
              position = position_dodge(width = .9)) +
  stat_summary(fun = mean,
              geom = "point",
              size = 5,
              position = position_dodge(width = .9)) +
  labs(x = "Program, randomized",
       y = "Weight Loss, pounds",
       fill = NULL,
       color = NULL,
       shape = NULL) +
  scale_fill_manual(values = c("gray20", "gray90")) +
  scale_shape_manual(values = c(18, 19)) +
  scale_y_continuous(breaks = seq(from = -20, to = 50, by = 10)) +
  theme(legend.position = c(0, 1),
        legend.justification = c(-.1, 1.1),
        legend.background = element_rect(color = "black"),
        legend.key.height = unit(1.3, "cm"),
        legend.direction = "horizontal")

```

Figure B.11

Distribution of Weight Loss, By Program and Gender - Option B



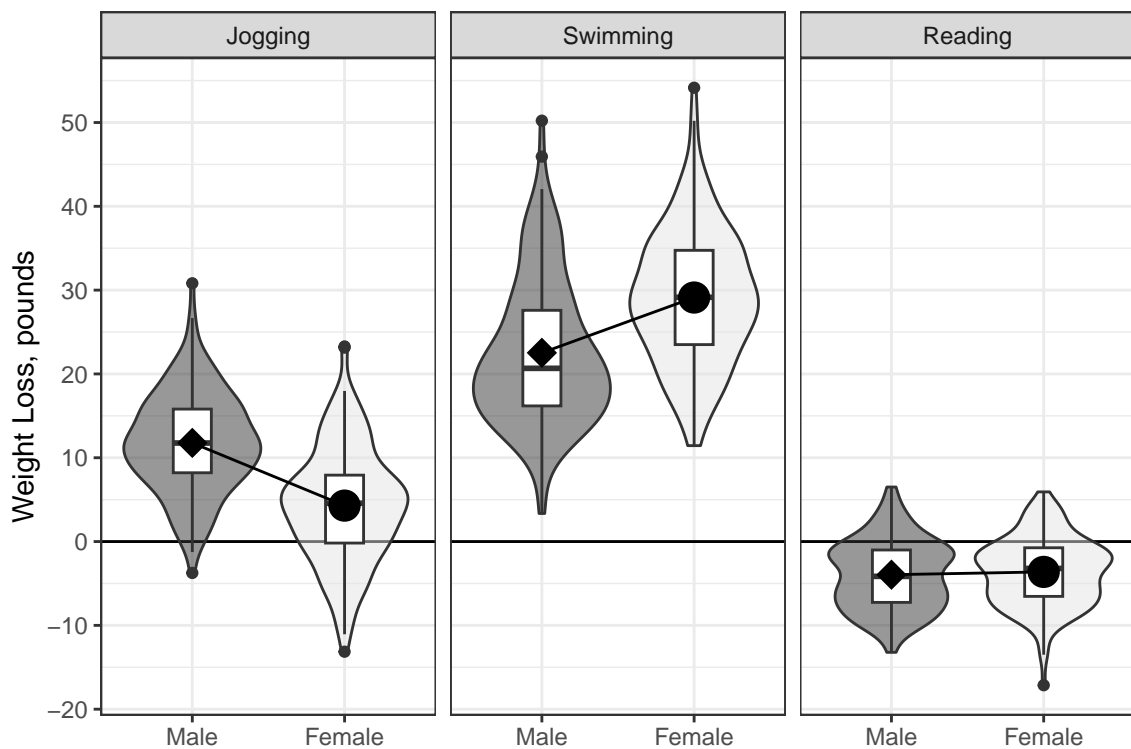
```

df_use %>%
  ggplot(aes(x = gender,
             y = loss,
             shape = gender,
             group = gender)) +
  geom_hline(yintercept = 0, color = "black") +
  geom_violin(aes(fill = gender),
             alpha = .5) +
  geom_boxplot(fill = "white",
             width = .25,
             position = position_dodge(width = .9)) +
  stat_summary(fun = mean,
             geom = "point",
             size = 5,
             position = position_dodge(width = .9)) +
  stat_summary(aes(group = 1),
             fun = "mean",
             geom = "line") +
  labs(x = NULL,
       y = "Weight Loss, pounds",
       shape = NULL) +
  scale_shape_manual(values = c(18, 19)) +
  scale_y_continuous(breaks = seq(from = -20, to = 50, by = 10)) +
  scale_fill_manual(values = c("gray20", "gray90")) +
  theme(legend.position = "none") +
  facet_grid(~ prog)

```

Figure B.12

Distribution of Weight Loss, By Program and Gender - Option C

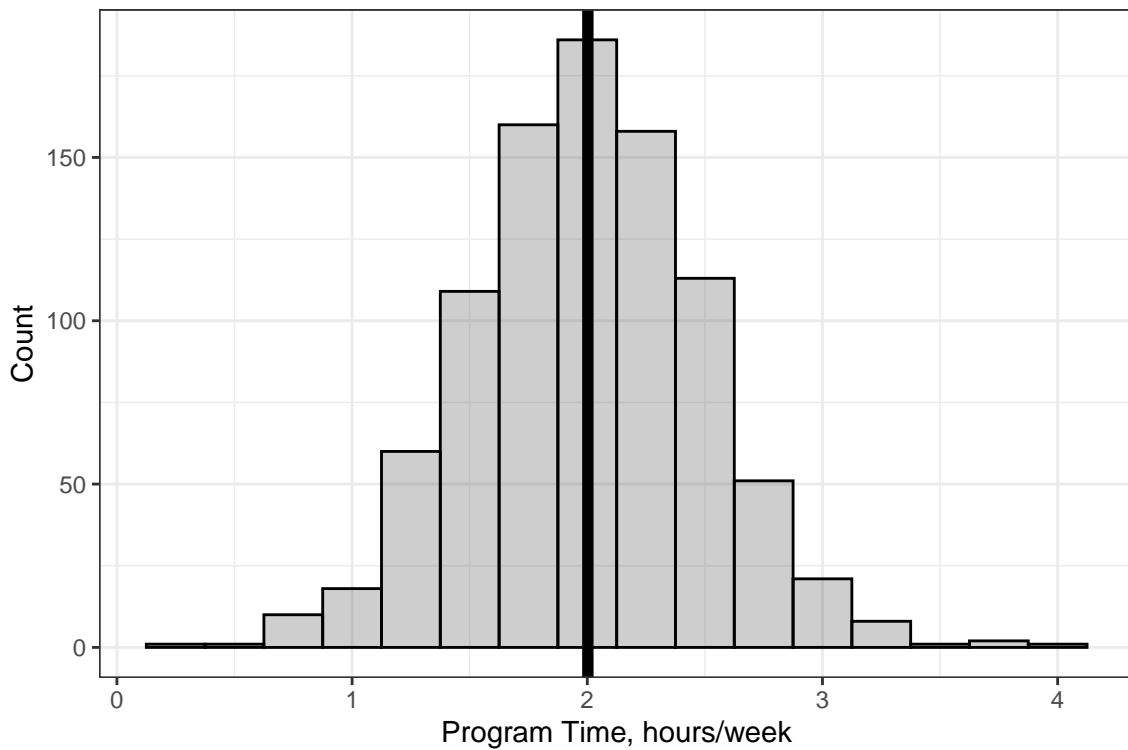


Program Time

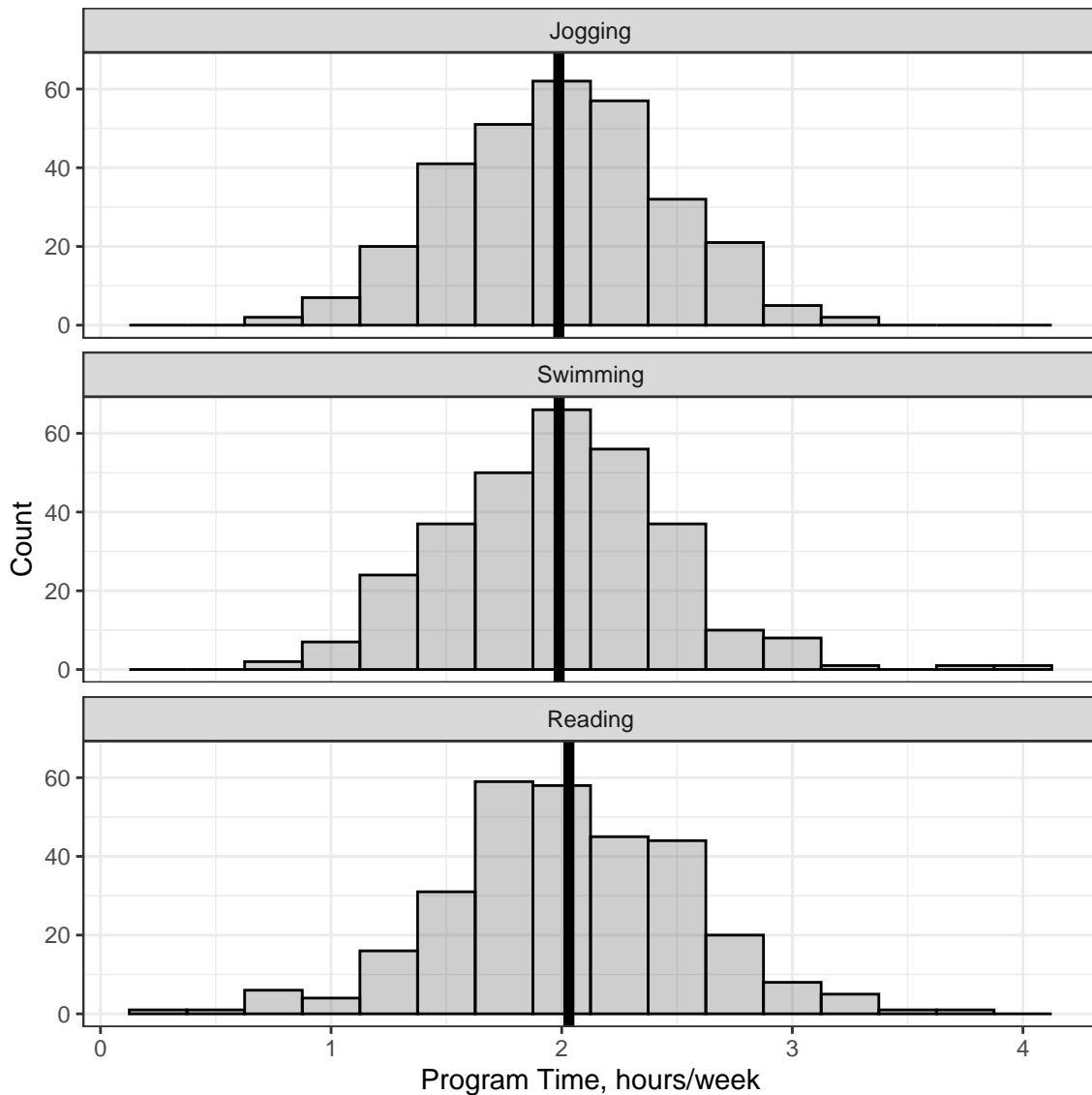
```
df_use %>%
  ggplot(aes(hours)) +
  geom_histogram(color = "black",
                 alpha = .3,
                 binwidth = .25) +
  stat_summary(aes(xintercept = ..x..,
                   y = 0),
               fun = mean,
               geom = "vline",
               orientation = "y",
               linewidth = 2) +
  labs(x = "Program Time, hours/week",
       y = "Count")
```

Figure B.13

Distribution of Program Time



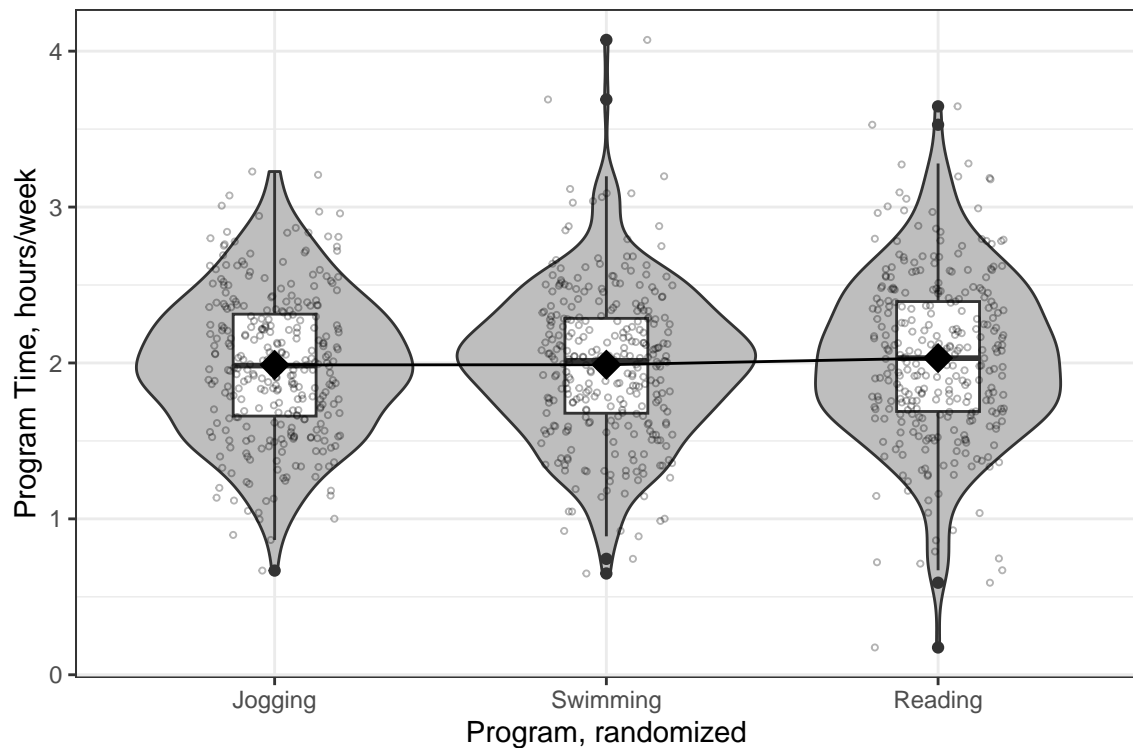

```
df_use %>%
  ggplot(aes(hours)) +
    geom_histogram(color = "black",
                  alpha = .3,
                  binwidth = .25) +
    stat_summary(aes(xintercept = ..x.., y = 0),
                fun = mean,
                geom = "vline",
                orientation = "y",
                linewidth = 2) +
    labs(x = "Program Time, hours/week",
         y = "Count") +
    facet_wrap(~ prog, ncol = 1)
```

Figure B.14*Distribution of Program Time, By Program - Option A*

```
df_use %>%
  ggplot(aes(x = prog,
             y = hours)) +
  geom_violin(fill = "gray") +
  geom_boxplot(width = .25) +
  geom_jitter(size = .75,
             shape = 1,
             alpha = .3,
             width = .2,
             height = 0) +
  stat_summary(fun = "mean",
             geom = "point",
             shape = 18,
             size = 5) +
  stat_summary(aes(group = 1),
             fun = "mean",
             geom = "line") +
  labs(x = "Program, randomized",
       y = "Program Time, hours/week")
```

Figure B.15

Distribution of Program Time, By Program - Option B



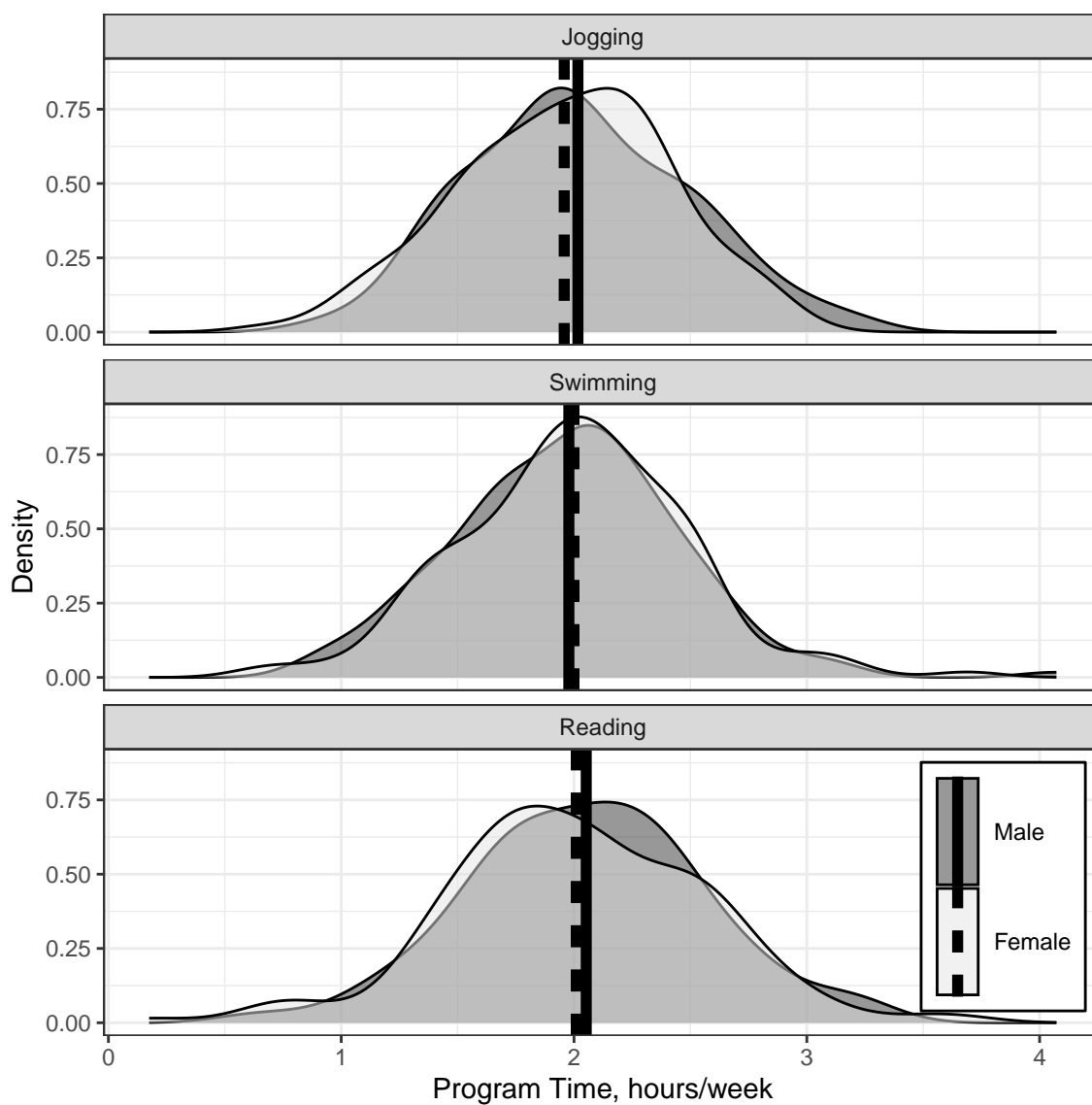
```

df_use %>%
  ggplot(aes(hours,
              fill = gender)) +
  geom_density(alpha = .5) +
  stat_summary(aes(xintercept = ..x..,
                  y = 0,
                  linetype = gender),
              fun = mean,
              geom = "vline",
              orientation = "y",
              linewidth = 2) +
  labs(x = "Program Time, hours/week",
       y = "Density",
       fill = NULL,
       color = NULL,
       linetype = NULL)+
  facet_wrap(~ prog, ncol = 1) +
  scale_fill_manual(values = c("gray20", "gray90")) +
  theme(legend.position = c(1, 0),
        legend.justification = c(1.1, -.1),
        legend.background = element_rect(color = "black"),
        legend.key.height = unit(1.5, "cm"))

```

Figure B.16

Distribution of Program Time, By Program and Gender - Option A



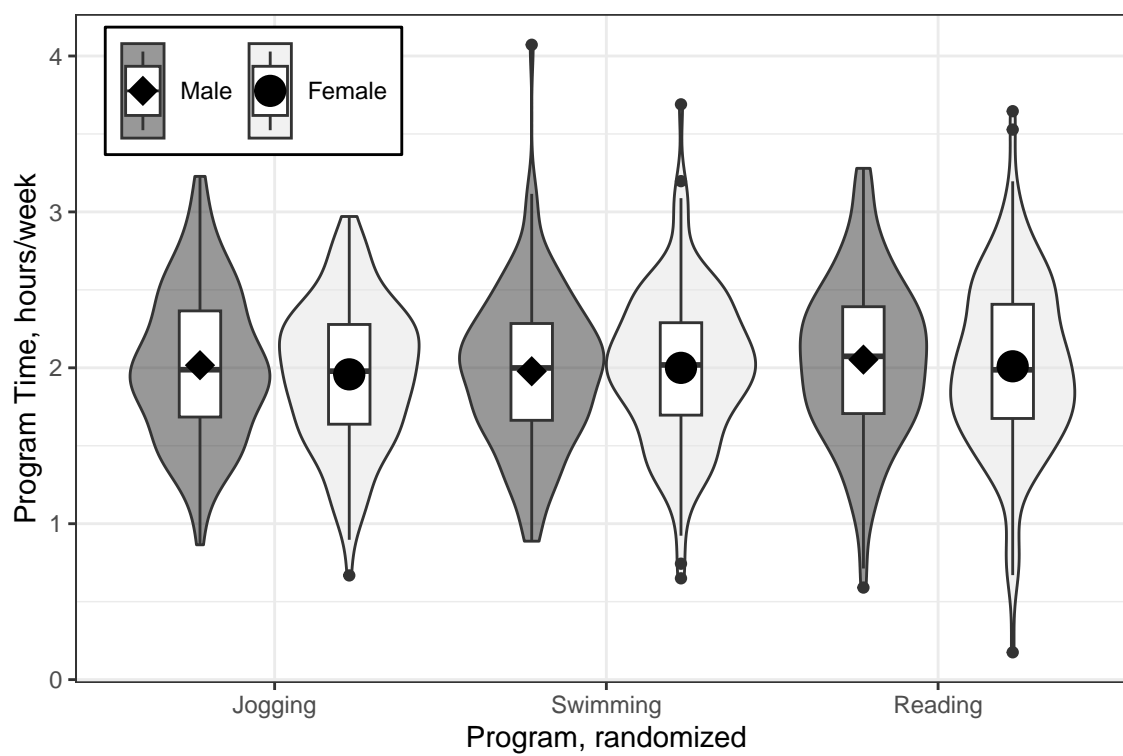
```

df_use %>%
  ggplot(aes(x = prog,
             y = hours,
             fill = gender,
             shape = gender,
             group = interaction(prog,gender))) +
  geom_violin(alpha = .5) +
  geom_boxplot(fill = "white",
              width = .25,
              position = position_dodge(width = .9)) +
  stat_summary(fun = mean,
              geom = "point",
              size = 5,
              position = position_dodge(width = .9)) +
  labs(x = "Program, randomized",
       y = "Program Time, hours/week",
       fill = NULL,
       color = NULL,
       shape = NULL) +
  scale_fill_manual(values = c("gray20", "gray90")) +
  scale_shape_manual(values = c(18, 19)) +
  theme(legend.position = c(0, 1),
        legend.justification = c(-.1, 1.1),
        legend.background = element_rect(color = "black"),
        legend.key.height = unit(1.3, "cm"),
        legend.direction = "horizontal")

```

Figure B.17

Distribution of Program Time, By Program and Gender - Option B



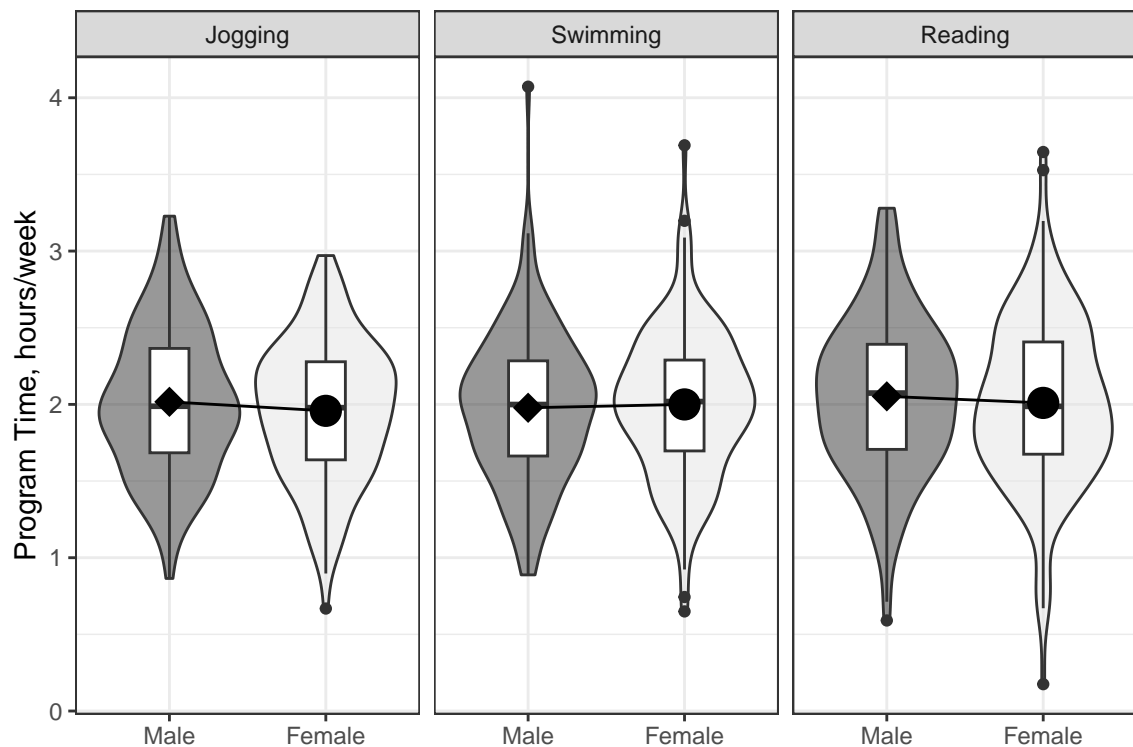
```

df_use %>%
  ggplot(aes(x = gender,
             y = hours,
             shape = gender,
             group = gender)) +
  geom_violin(aes(fill = gender),
             alpha = .5) +
  geom_boxplot(fill = "white",
             width = .25,
             position = position_dodge(width = .9)) +
  stat_summary(fun = mean,
             geom = "point",
             size = 5,
             position = position_dodge(width = .9)) +
  stat_summary(aes(group = 1),
             fun = "mean",
             geom = "line") +
  scale_fill_manual(values = c("gray20", "gray90")) +
  labs(x = NULL,
       y = "Program Time, hours/week",
       shape = NULL) +
  scale_shape_manual(values = c(18, 19)) +
  theme(legend.position = "none") +
  facet_grid(~ prog)

```

Figure B.18

Distribution of Program Time, By Program and Gender - Option C

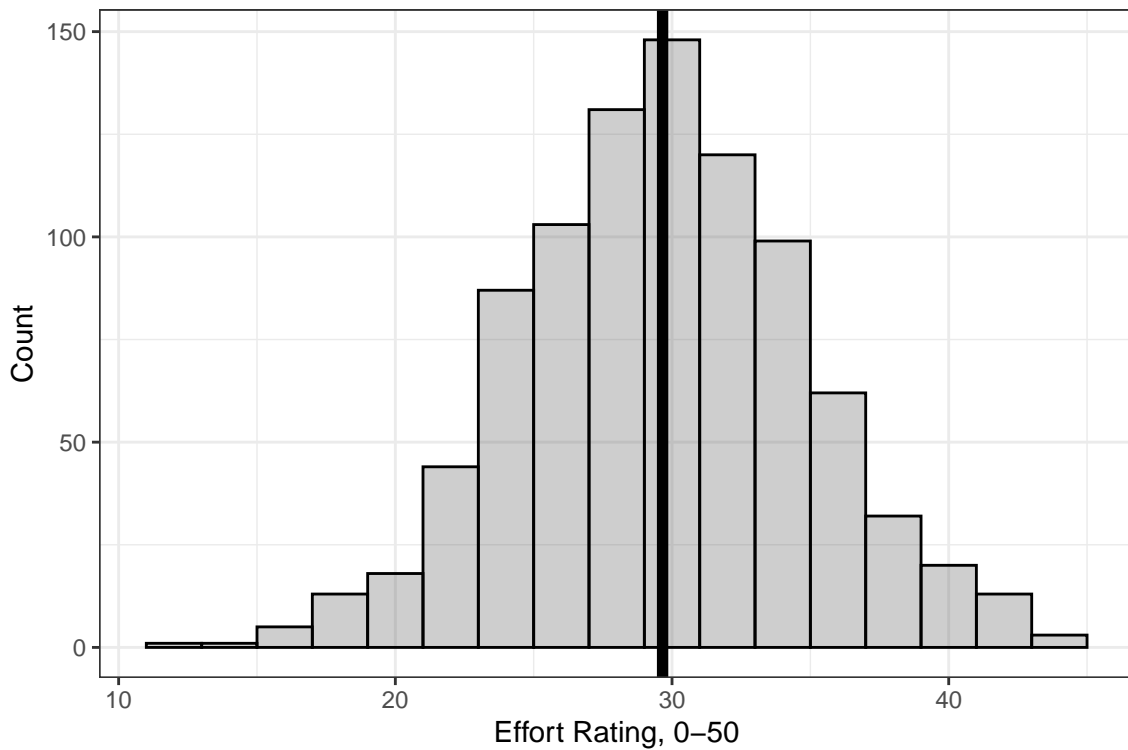


Effort Rating

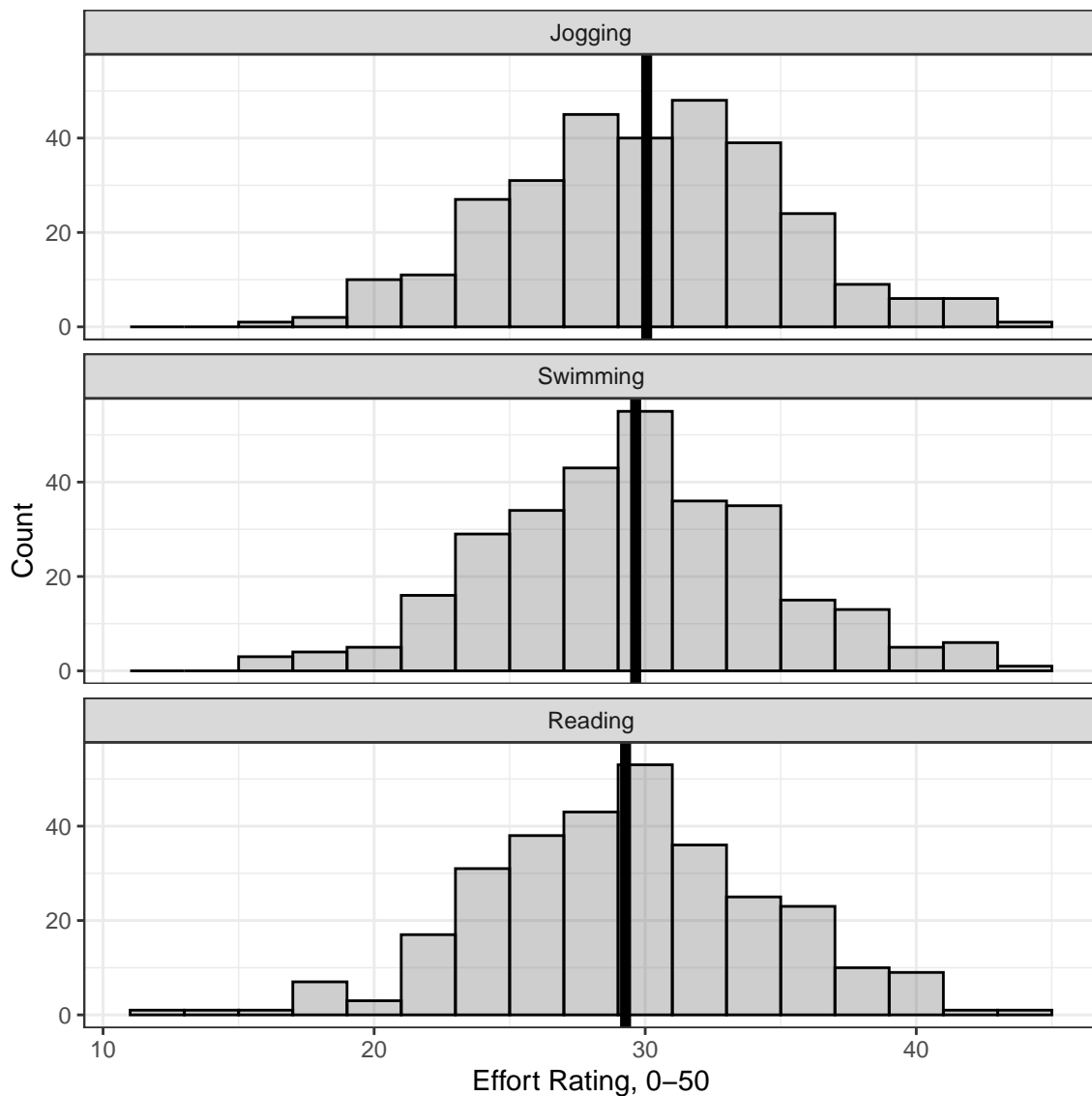
```
df_use %>%
  ggplot(aes(effort)) +
  geom_histogram(color = "black",
                 alpha = .3,
                 binwidth = 2) +
  stat_summary(aes(xintercept = ..x..,
                   y = 0),
               fun = mean,
               geom = "vline",
               orientation = "y",
               linewidth = 2) +
  labs(x = "Effort Rating, 0-50",
       y = "Count")
```

Figure B.19

Distribution of Effort Rating




```
df_use %>%
  ggplot(aes(effort)) +
    geom_histogram(color = "black",
                  alpha = .3,
                  binwidth = 2) +
    stat_summary(aes(xintercept = ..x.., y = 0),
                fun = mean,
                geom = "vline",
                orientation = "y",
                linewidth = 2) +
    labs(x = "Effort Rating, 0-50",
         y = "Count") +
    facet_wrap(~ prog, ncol = 1)
```

Figure B.20*Distribution of Effort Rating, By Program - Option A*

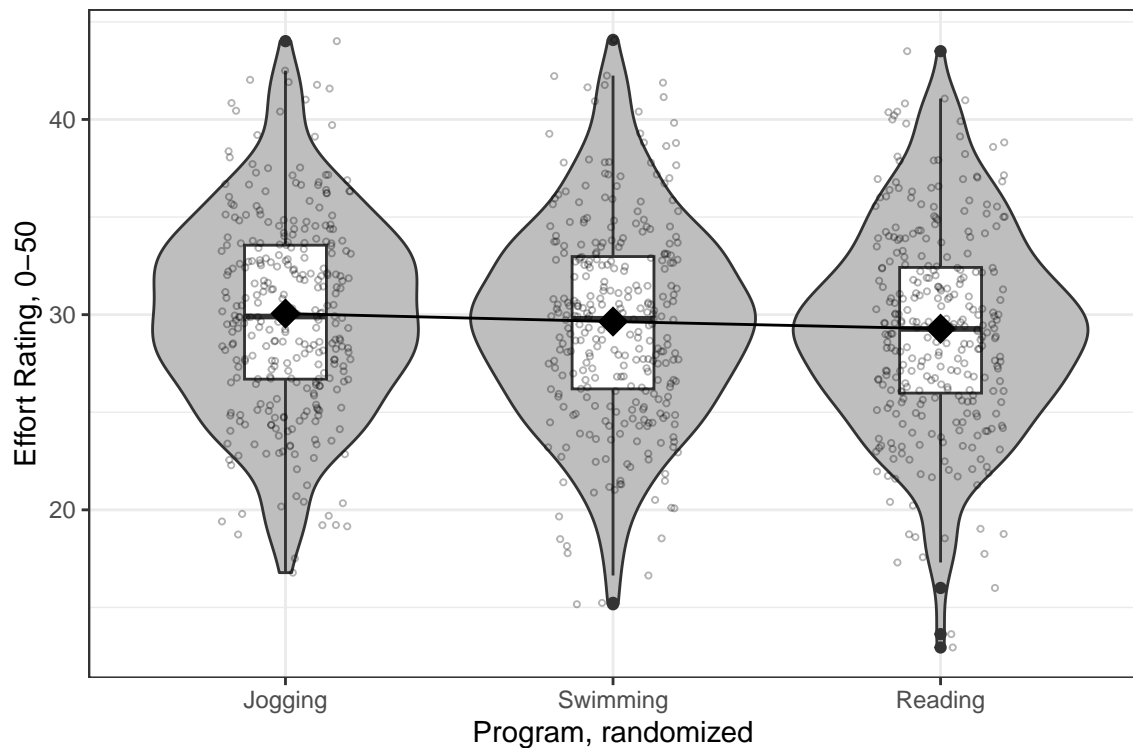
```

df_use %>%
  ggplot(aes(x = prog,
             y = effort)) +
  geom_violin(fill = "gray") +
  geom_boxplot(width = .25) +
  geom_jitter(size = .75,
              shape = 1,
              alpha = .3,
              width = .2,
              height = 0) +
  stat_summary(fun = "mean",
              geom = "point",
              shape = 18,
              size = 5) +
  stat_summary(aes(group = 1),
              fun = "mean",
              geom = "line") +
  labs(x = "Program, randomized",
       y = "Effort Rating, 0-50")

```

Figure B.21

Distribution of Effort Rating, By Program - Option B



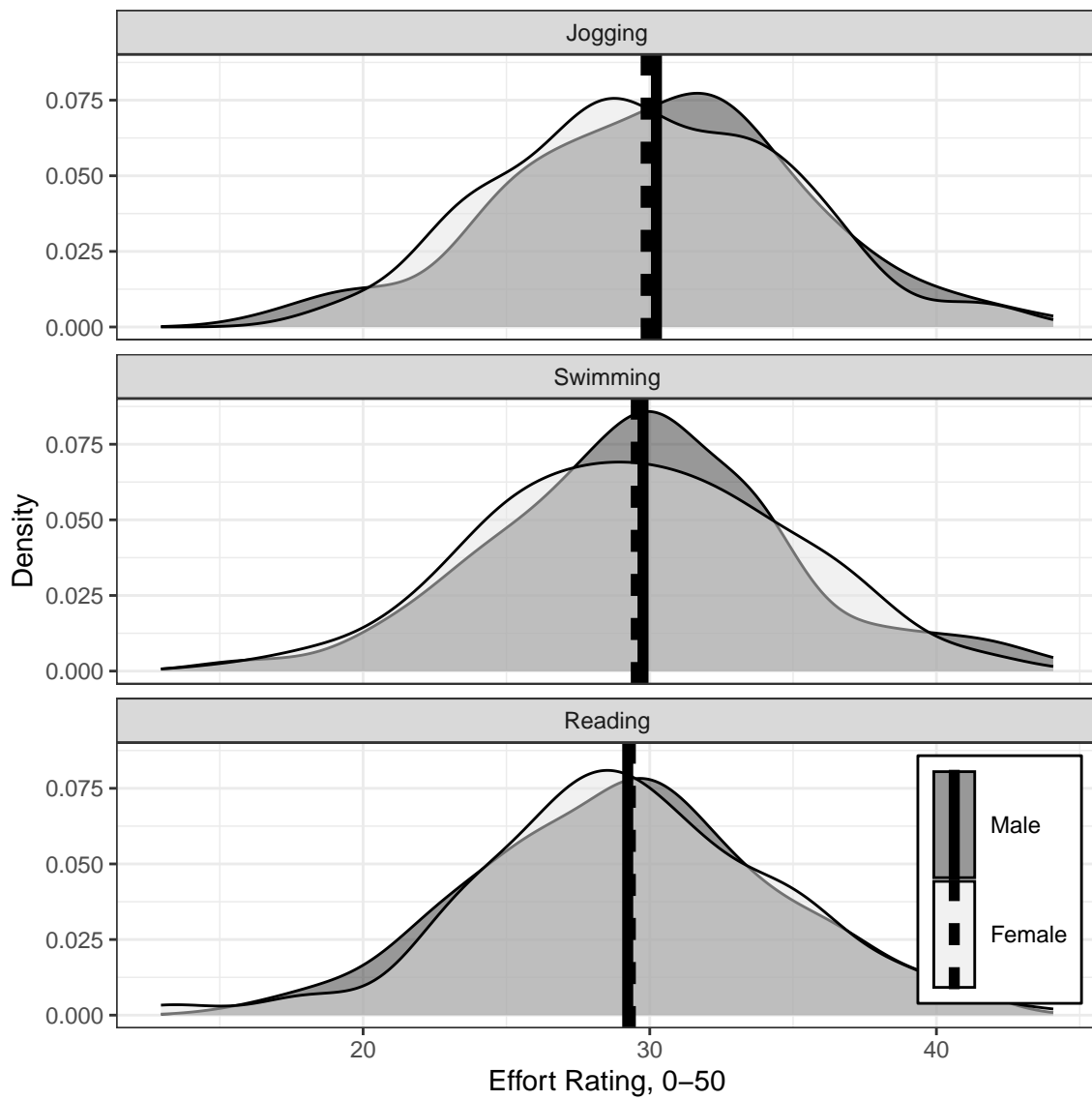
```

df_use %>%
  ggplot(aes(effort,
              fill = gender)) +
  geom_density(alpha = .5) +
  stat_summary(aes(xintercept = ..x..,
                    y = 0,
                    linetype = gender),
               fun = mean,
               geom = "vline",
               orientation = "y",
               linewidth = 2) +
  labs(x = "Effort Rating, 0-50",
       y = "Density",
       fill = NULL,
       linetype = NULL)+
  facet_wrap(~ prog, ncol = 1) +
  scale_fill_manual(values = c("gray20", "gray90")) +
  theme(legend.position = c(1, 0),
        legend.justification = c(1.1, -.1),
        legend.background = element_rect(color = "black"),
        legend.key.height = unit(1.5, "cm"))

```

Figure B.22

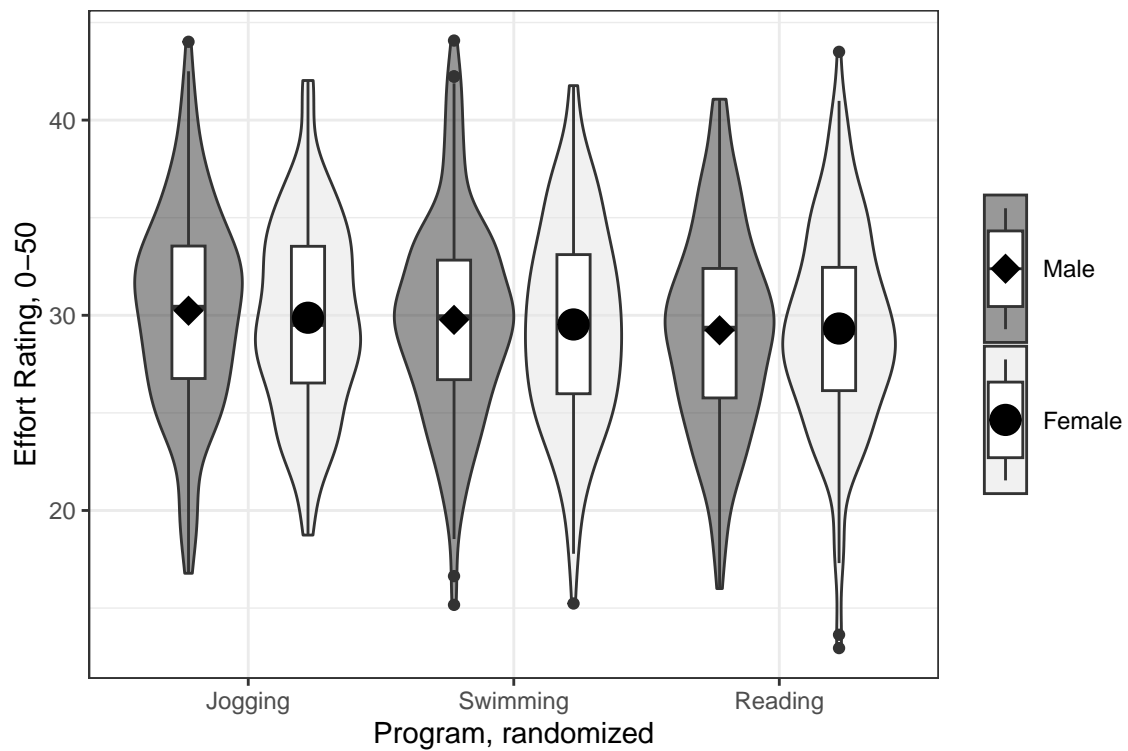
Distribution of Effort Rating, By Program and Gender - Option A



```
df_use %>%
  ggplot(aes(x = prog,
             y = effort,
             fill = gender,
             shape = gender,
             group = interaction(prog,gender))) +
  geom_violin(alpha = .5) +
  geom_boxplot(fill = "white",
              width = .25,
              position = position_dodge(width = .9)) +
  stat_summary(fun = mean,
              geom = "point",
              size = 5,
              position = position_dodge(width = .9)) +
  labs(x = "Program, randomized",
       y = "Effort Rating, 0-50",
       fill = NULL,
       color = NULL,
       shape = NULL) +
  scale_fill_manual(values = c("gray20", "gray90")) +
  scale_shape_manual(values = c(18, 19)) +
  theme(legend.key.height = unit(2, "cm"))
```

Figure B.23

Distribution of Effort Rating, By Program and Gender - Option B



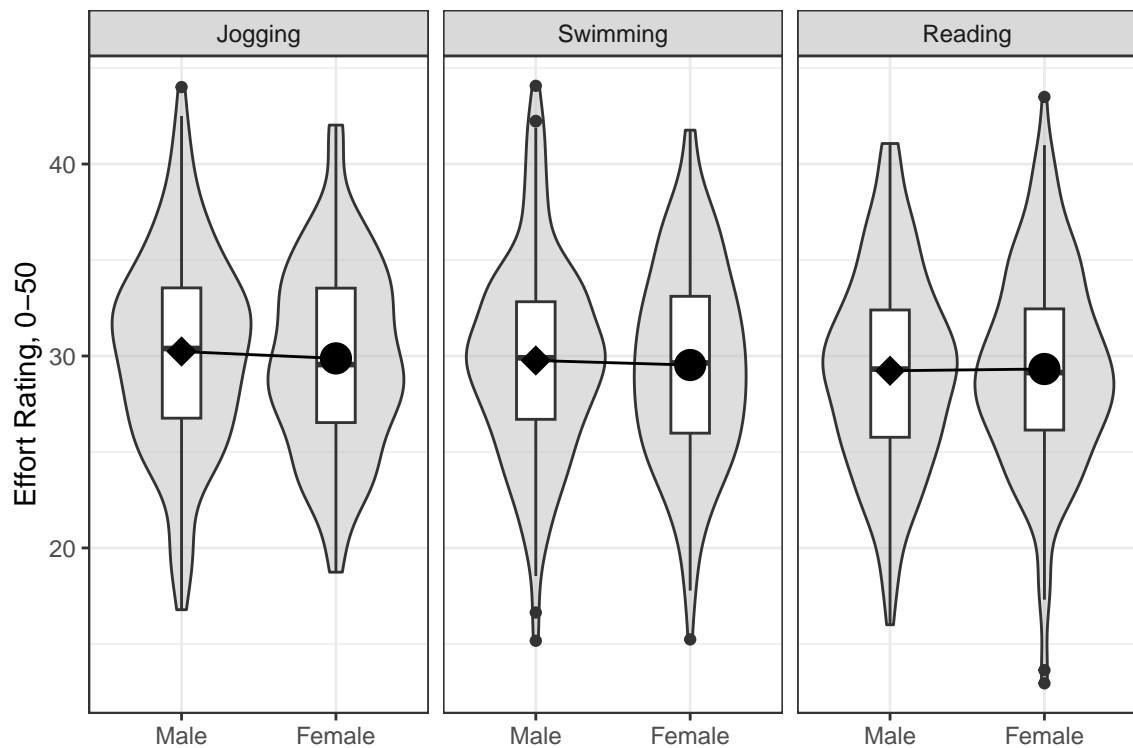
```

df_use %>%
  ggplot(aes(x = gender,
             y = effort,
             shape = gender,
             group = gender)) +
  geom_violin(fill = "gray",
             alpha = .5) +
  geom_boxplot(fill = "white",
             width = .25,
             position = position_dodge(width = .9)) +
  stat_summary(fun = mean,
             geom = "point",
             size = 5,
             position = position_dodge(width = .9)) +
  stat_summary(aes(group = 1),
             fun = "mean",
             geom = "line") +
  labs(x = NULL,
       y = "Effort Rating, 0-50",
       shape = NULL) +
  scale_shape_manual(values = c(18, 19)) +
  theme(legend.position = "none") +
  facet_grid(~ prog)

```

Figure B.24

Distribution of Effort Rating, By Program and Gender - Option C



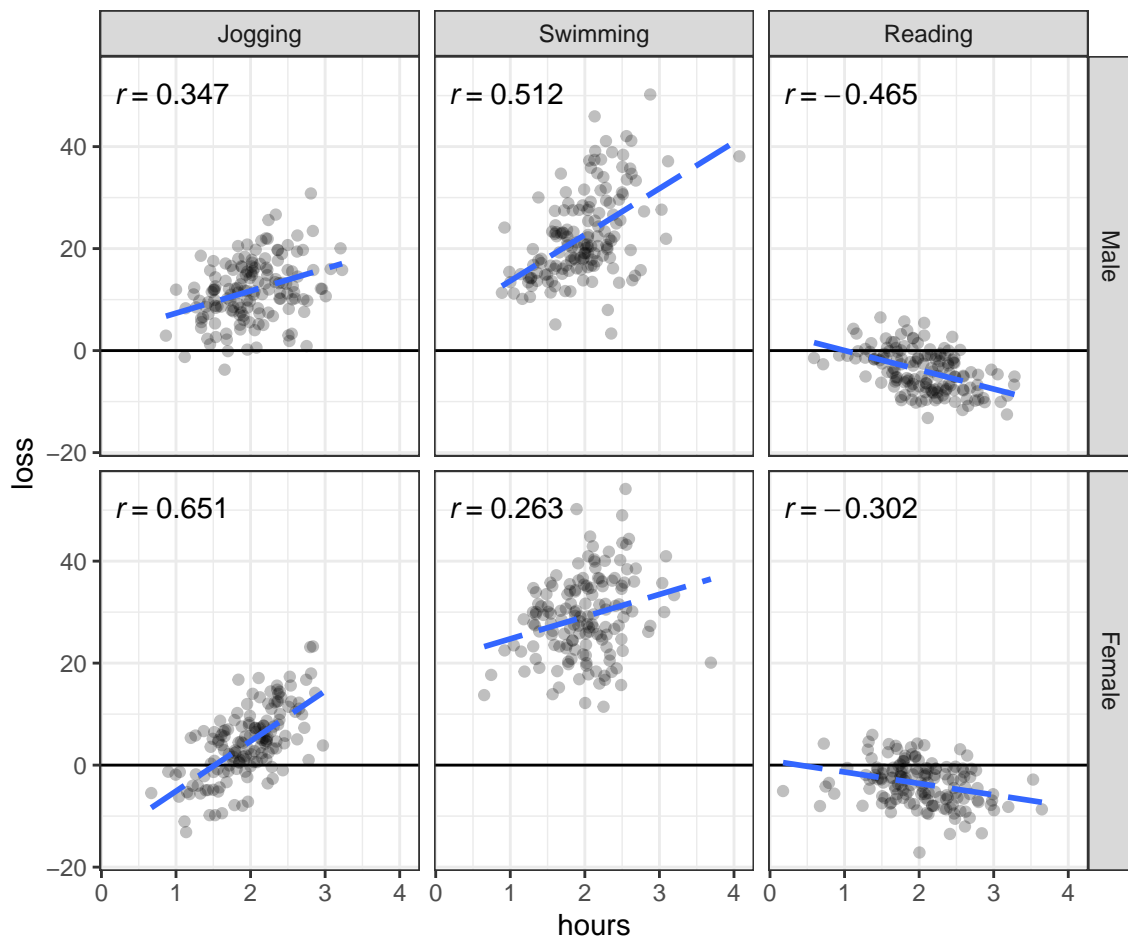
Bivariable

Weight Loss by Program Time

```
df_use %>%
  ggplot(aes(x = hours,
             y = loss)) +
  geom_hline(yintercept = 0, color = "black") +
  geom_point(alpha = .25) +
  theme_bw() +
  facet_grid(gender ~ prog) +
  geom_smooth(method = "lm",
             formula = y ~ x,
             se = FALSE,
             linetype = "longdash") +
  ggpubr::stat_cor(aes(label = ..r.label..),
                  cor.coef.name = "r",
                  digits = 3)
```

Figure B.25

Association Between Weight Loss by Program Time, by Gender and Program

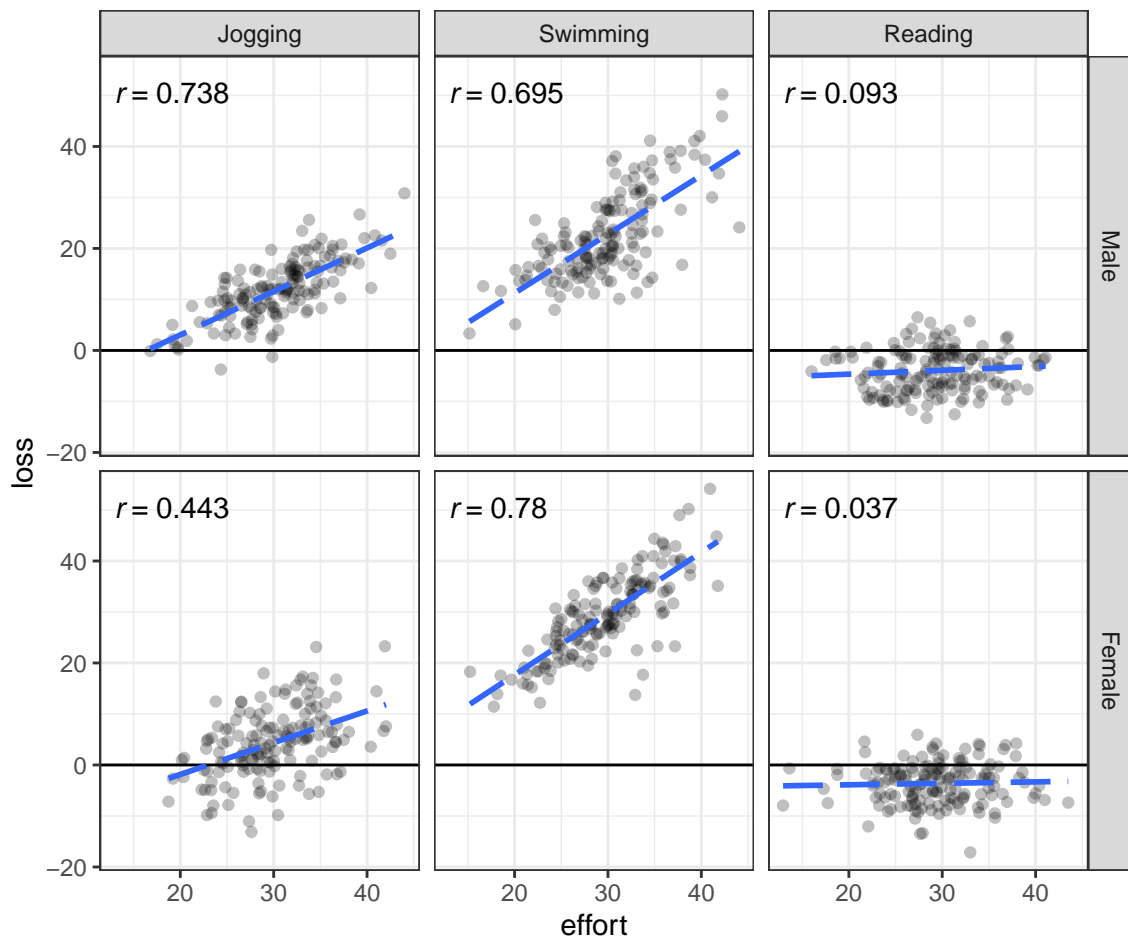


Weight Loss by Effort Rating

```
df_use %>%
  ggplot(aes(x = effort,
             y = loss)) +
  geom_hline(yintercept = 0, color = "black") +
  geom_point(alpha = .25) +
  theme_bw() +
  facet_grid(gender ~ prog) +
  geom_smooth(method = "lm",
             formula = y ~ x,
             se = FALSE,
             linetype = "longdash") +
  ggpubr::stat_cor(aes(label = ..r.label..),
                  cor.coef.name = "r",
                  digits = 3,
                  r.accuracy = .001)
```

Figure B.26

Association Between Weight Loss by Effort Rating, by Gender and Program

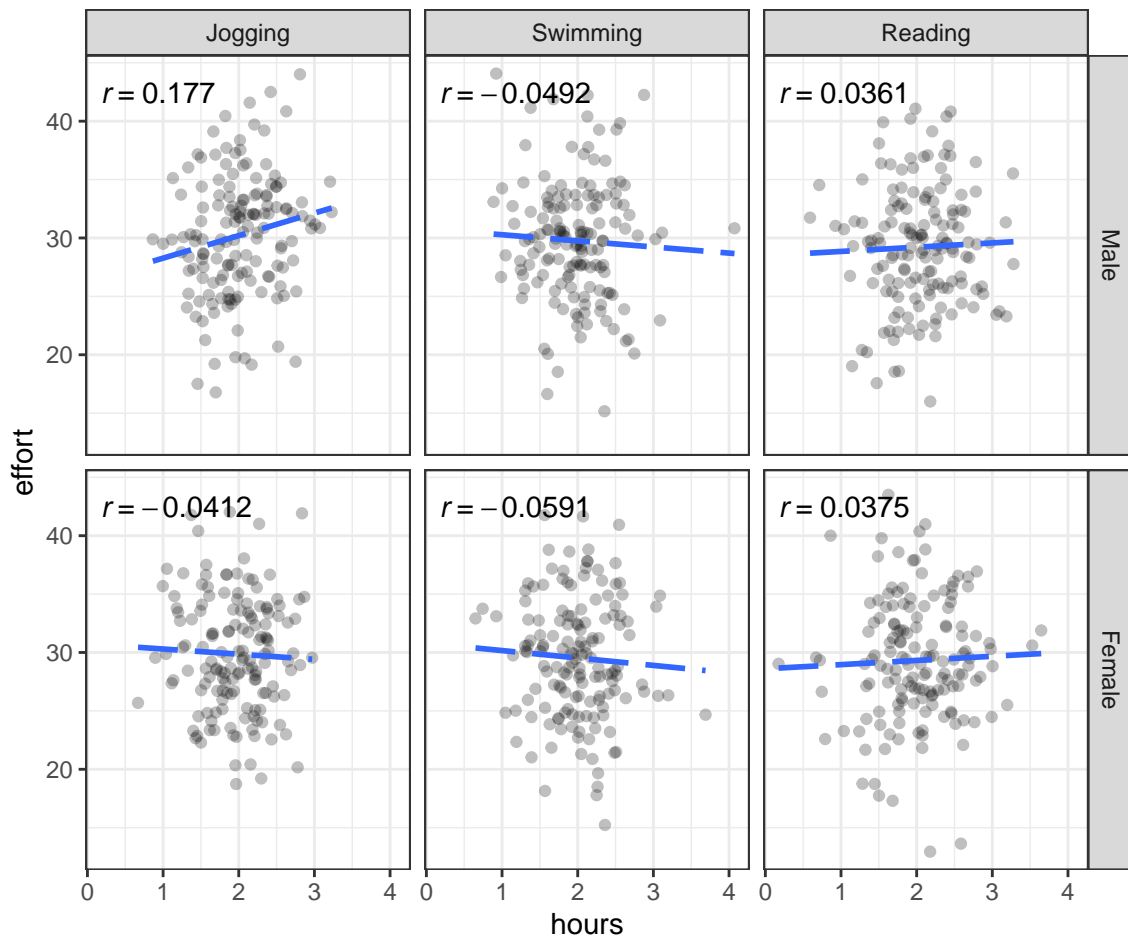


Effort Rating by Program Time

```
df_use %>%
  ggplot(aes(x = hours,
             y = effort)) +
  geom_point(alpha = .25) +
  theme_bw() +
  facet_grid(gender ~ prog) +
  geom_smooth(method = "lm",
             formula = y ~ x,
             se = FALSE,
             linetype = "longdash") +
  ggpubr::stat_cor(aes(label = ..r.label..),
                  cor.coef.name = "r",
                  digits = 3)
```

Figure B.27

Association Between Effort Rating by Program Time, by Gender and Program



REGRESSION ANALYSIS

Multip

Model Fitting

Fit Nested Models

```
fit_1 <- lm(loss ~ prog + hours + gender + effort,  
            data = df_use)  
  
fit_2 <- lm(loss ~ prog*hours*gender +  
            prog*hours*effort,  
            data = df_use)  
  
fit_3 <- lm(loss ~ prog*hours*gender +  
            prog*gender*effort +  
            prog*hours*effort,  
            data = df_use)  
fit_4 <- lm(loss ~ prog*hours*gender*effort,  
            data = df_use)
```

Parameter Estimate Table

```
texreg::knitreg(list(fit_1, fit_2, fit_3, fit_4),  
  custom.model.names = c("fit 1", "fit 2", "fit 3", "fit 4"),  
  caption = "Parameter Estimates of Nested Regression Models for Weight Loss")
```

Table B.8*Parameter Estimates of Nested Regression Models for Weight Loss*

	fit 1	fit 2	fit 3	fit 4
(Intercept)	−17.80*** (1.49)	2.18 (6.39)	−0.03 (6.58)	7.42 (9.15)
progSwimming	18.05*** (0.50)	8.30 (8.76)	10.80 (8.99)	−1.74 (12.05)
progReading	−11.45*** (0.50)	6.08 (8.38)	7.46 (8.68)	2.91 (12.11)
hours	3.14*** (0.41)	−6.46* (3.17)	−6.27* (3.17)	−9.98* (4.48)
genderFemale	0.01 (0.41)	−20.95*** (1.97)	−17.34*** (3.24)	−31.80* (12.75)
effort	0.65*** (0.04)	0.12 (0.21)	0.20 (0.22)	−0.04 (0.30)
progSwimming:hours		−5.15 (4.28)	−5.34 (4.28)	0.89 (5.84)
progReading:hours		−0.52 (4.15)	−0.62 (4.16)	1.63 (5.92)
progSwimming:genderFemale		35.59*** (2.73)	31.35*** (4.60)	57.63*** (17.38)
progReading:genderFemale		18.00*** (2.65)	15.62*** (4.44)	24.74 (16.66)
hours:genderFemale		7.18*** (0.97)	7.27*** (0.97)	14.55* (6.28)
progSwimming:effort		−0.33 (0.28)	−0.42 (0.29)	−0.00 (0.39)
progReading:effort		−0.27 (0.28)	−0.33 (0.29)	−0.18 (0.40)
hours:effort		0.31** (0.10)	0.30** (0.10)	0.42** (0.15)
progSwimming:hours:genderFemale		−11.15*** (1.34)	−11.23*** (1.34)	−24.32** (8.45)
progReading:hours:genderFemale		−5.63*** (1.28)	−5.70*** (1.29)	−10.25 (8.22)
progSwimming:hours:effort		0.39** (0.14)	0.40** (0.14)	0.20 (0.19)
progReading:hours:effort		−0.20 (0.14)	−0.19 (0.14)	−0.26 (0.20)
genderFemale:effort			−0.13 (0.09)	0.35 (0.42)
progSwimming:genderFemale:effort			0.15 (0.12)	−0.72 (0.57)
progReading:genderFemale:effort			0.08 (0.12)	−0.21 (0.55)
hours:genderFemale:effort				−0.24 (0.20)
progSwimming:hours:genderFemale:effort				0.43 (0.28)
progReading:hours:genderFemale:effort				0.14 (0.27)
R ²	0.82	0.93	0.93	0.93
Adj. R ²	0.81	0.92	0.92	0.92
Num. obs.	900	900	900	900

*** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$

Compare Model Fit

```
tab_lm_fits <- performance::compare_performance(fit_1, fit_2, fit_3, fit_4) %>%
  data.frame() %>%
  dplyr::select(Name, AIC, BIC, R2, R2_adjusted, RMSE, Sigma) %>%
  flextable::flextable() %>%
  flextable::set_caption("Comparison of Model Fit for Nested Regression Models for Weight Loss")

tab_lm_fits
```

Table B.9

Comparison of Model Fit for Nested Regression Models for Weight Loss

Name	AIC	BIC	R2	R2_adjusted	RMSE	Sigma
fit_1	5,811.56	5,845.18	0.82	0.81	6.06	6.08
fit_2	5,006.15	5,097.40	0.93	0.92	3.82	3.86
fit_3	5,009.82	5,115.47	0.93	0.92	3.82	3.86
fit_4	5,013.01	5,133.07	0.93	0.92	3.81	3.86

Residual Diagnostics

The standard regression assumptions include the following about residuals/errors:

- The error has a normal distribution (normality assumption).
- The errors have mean zero.
- The errors have same but unknown variance (homoscedasticity assumption).
- The error are independent of each other (independent errors assumption).

Normality Test:

- Shapiro-Wilk (W): The recommended sample size for this test ranges from 7 to 2000. Origin allows sample sizes from 3 to 5000. However, when sample size is relatively large, D'Agostino K-squared or Lilliefors are generally preferred over Shapiro-Wilk.
- Kolmogorov-Smirnov (D): The K-S test, though known to be less powerful, is widely used. Generally, it requires **large sample sizes**.
- Cramer-von Mises (CH): The Cramer-von Mises goodness-of-fit test is based on the empirical distribution and an ordered statistic. It is distribution-free (can be used for other distributions as well) omnibus test alternative to the Kolmogorov-Smirnov test (also ecdf-based). The p-value is based on the largest discrepancy between the empirical distribution and the hypothetical (normal, in this case) distribution. In terms of power against commonly-encountered alternatives it doesn't shine compared to the rest of the test in our goodness-of-fit calculator, but it is still widely used.
- Anderson-Darling (A^2): The Anderson-Darling normality test is a modification of the Cramer-von Mises approach and is thus a distance-test based on the empirical cumulative distribution function and distribution-free in its generic form. Compared with the Cramer-von Mises distance, the Anderson-Darling distance places more weight on observations in the tails of the distribution. It shows decent sensitivity against a variety of distributions, most notably the Laplace and Uniform distribution. Sample size of less than 26 is recommended, but industrial data with 200 and more might pass A-D. The p-value of the A-D test depends on simulation algorithms.

```
olsrr::ols_test_normality(fit_2)
```

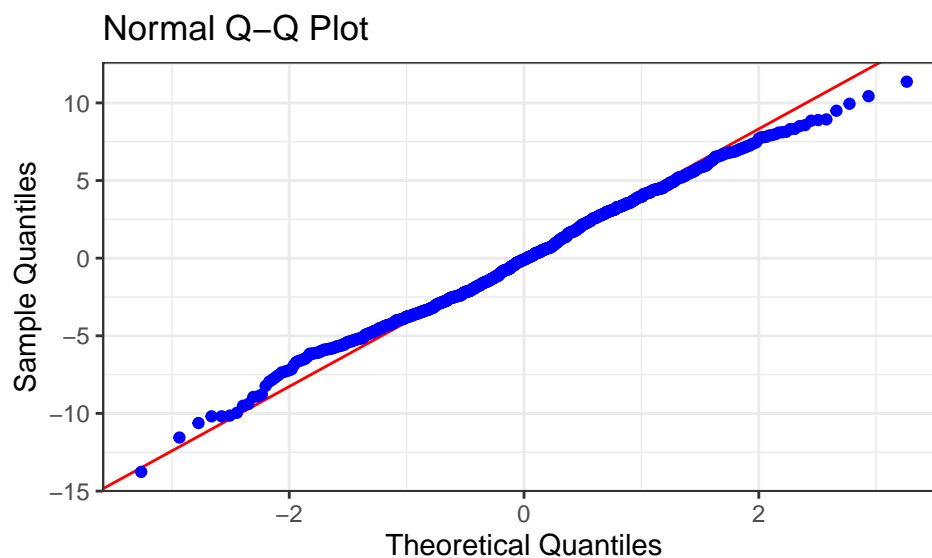
Test	Statistic	pvalue
Shapiro-Wilk	0.9962	0.0270
Kolmogorov-Smirnov	0.0326	0.2927
Cramer-von Mises	63.5595	0.0000
Anderson-Darling	1.2164	0.0036

```
olsrr::ols_plot_resid_qq(fit_2)
```

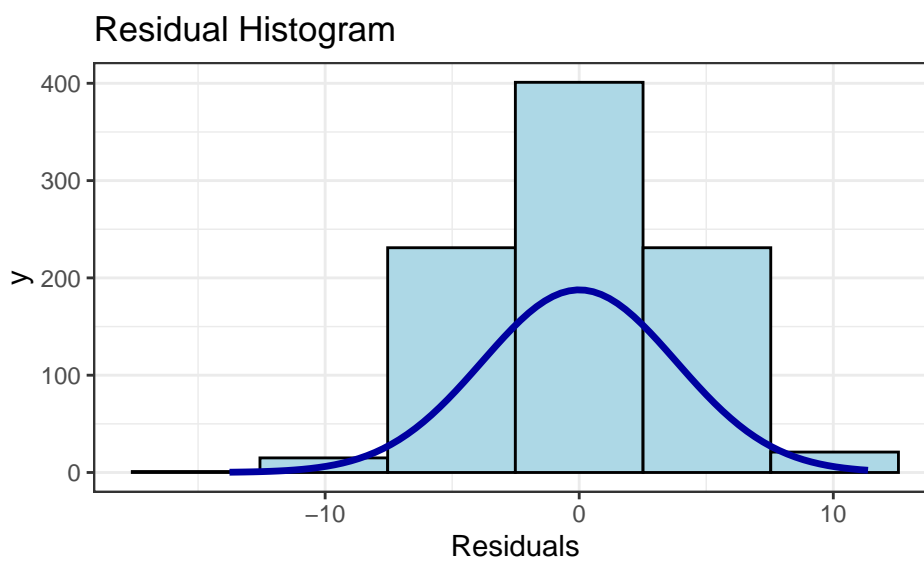
```
olsrr::ols_plot_resid_hist(fit_2)
```

Figure B.28

Residual Diagnostics, Assess Normality of the Conditional Distribution of Weight Loss, QQ Plot

**Figure B.29**

Residual Diagnostics, Assess Normality of the Conditional Distribution of Weight Loss, Histogram



It is a scatter plot of residuals on the y axis and fitted values on the x axis to detect non-linearity, unequal error variances, and outliers.

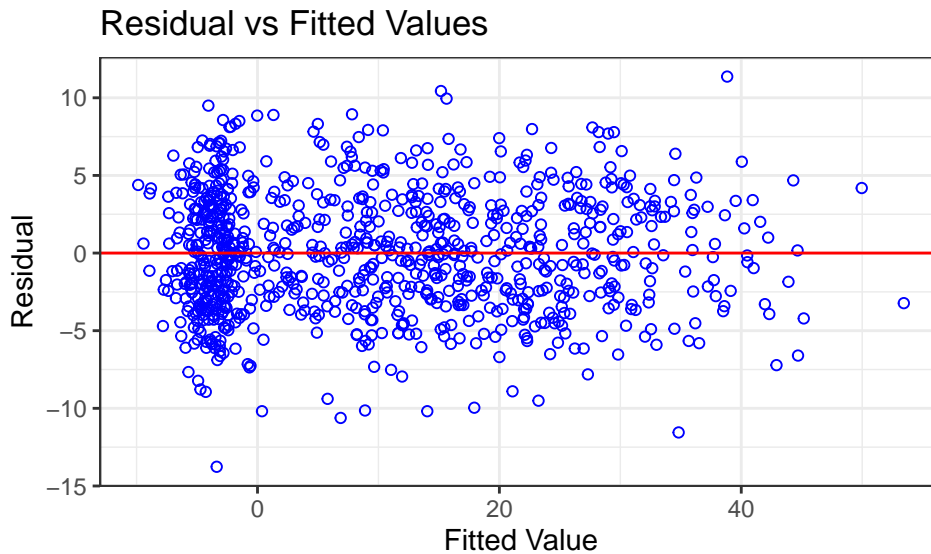
Characteristics of a well behaved residual vs fitted plot:

- The residuals **spread randomly** around the 0 line indicating that the relationship is linear.
- The residuals form an **approximate horizontal band** around the 0 line indicating homogeneity of error variance.
- No one residual is **visibly away from** the random pattern of the residuals indicating that there are no outliers.

```
olsrrr::ols_plot_resid_fit(fit_2)
```

Figure B.30

Residual Diagnostics, Assess Various Assumptions of Regression, scatterplot

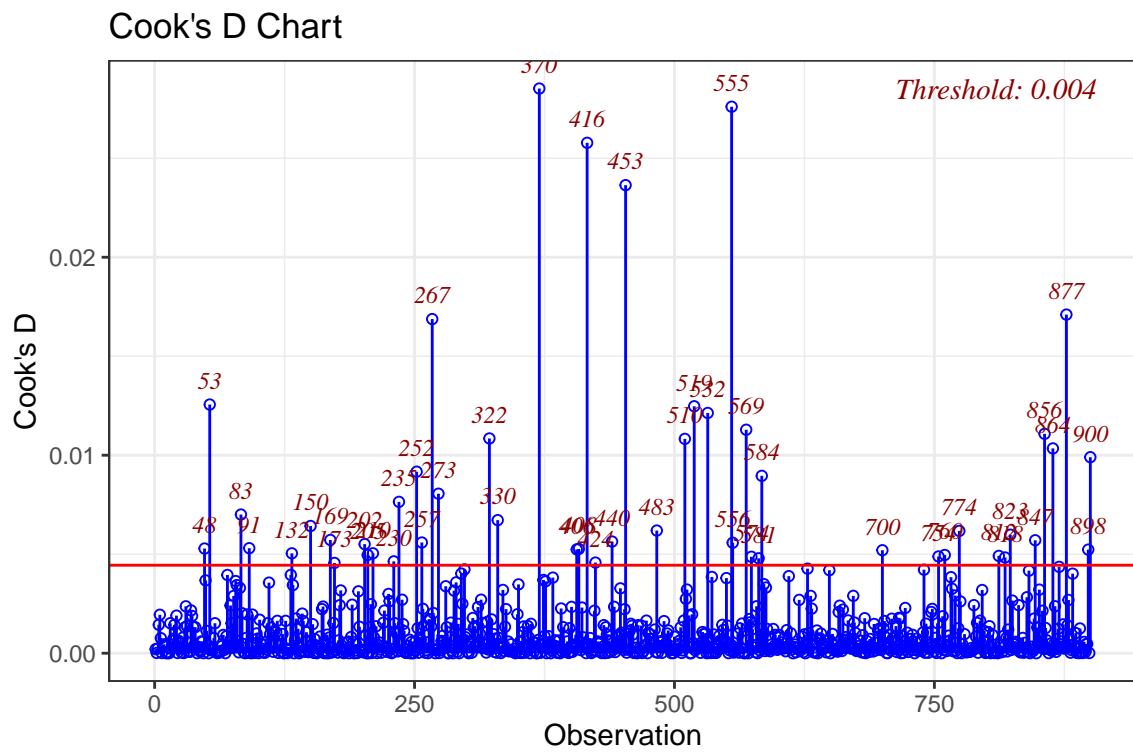


Cook's distance was introduced by American statistician R Dennis Cook in 1977. It is used to **identify influential data points**. It depends on both the residual and leverage i.e it takes it account both the x value and y value of the observation. A data point having a large cook's d indicates that the data point strongly influences the fitted values.

```
olsrr::ols_plot_cooksd_chart(fit_2)
```

Figure B.31

Residual Diagnostics, Identiry Influential Data Points, Cook's D Chart



Final Model

Anlaysis of Variance

```

tab_lm_aov <- anova(fit_2) %>%
  data.frame() %>%
  tibble::rownames_to_column() %>%
  dplyr::select("Model Term" = rowname,
               MS = Mean.Sq,
               df = Df,
               F = F.value,
               p = Pr..F.) %>%
  dplyr::mutate(p = apaSupp::p_num(p)) %>%
  dplyr::mutate_if(is.double, apaSupp::apa2) %>%
  flextable::flextable() %>%
  flextable::set_caption("Final Regression Model for Weight Loss, Analysis of Variance and Significance")

tab_lm_aov

```

Table B.10

Final Regression Model for Weight Loss, Analysis of Variance and Significance of All Terms

Model Term	MS	df	F	p
prog	66638.60	2	4467.31	< .001 ***
hours	2344.49	1	157.17	< .001 ***
gender	2.14	1	0.14	.705
effort	10046.72	1	673.51	< .001 ***
prog:hours	2748.38	2	184.25	< .001 ***
prog:gender	3392.35	2	227.42	< .001 ***
hours:gender	77.31	1	5.18	.023 *
prog:effort	2684.32	2	179.95	< .001 ***
hours:effort	766.94	1	51.41	< .001 ***
prog:hours:gender	545.54	2	36.57	< .001 ***
prog:hours:effort	161.74	2	10.84	< .001 ***
Residuals	14.92	882	NA	

Parameter Estimates

```
# texreg::knitreg(fit_2,  
#                 custom.model.names = "Weight Loss, pounds",  
#                 single.row = TRUE,  
#                 caption = "Parameter Estimates of Final Regression Models for Weight Loss")
```

Simple Slopes Analysis

```

tab_simpslopes <- interactions::sim_slopes(model = fit_2,
                                          pred = hours,
                                          modx = prog,
                                          mod2 = gender) %>%

  broom::tidy() %>%
  dplyr::select(Gender = mod2.value,
                Program = modx.value,
                b = estimate,
                SE = std.error,
                CI_lo = conf.low,
                CI_hi = conf.high,
                p = p.value) %>%
  dplyr::mutate(across(c(b, SE, p, CI_lo, CI_hi),
                      as.numeric)) %>%
  dplyr::mutate(p = apaSupp::p_num(p)) %>%
  dplyr::mutate_if(is.numeric, apaSupp::apa2) %>%
  flextable::flextable() %>%
  flextable::set_caption("Simple Slopes Analysis for Weight Loss by Time Spent in each Program by Gender")

tab_simpslopes

```

Table B.11

Simple Slopes Analysis for Weight Loss by Time Spent in each Program by Gender, Collapsed Across Effort

Gender	Program	b	SE	CI_lo	CI_hi	p
Male	Jogging	2.68	0.67	1.36	4.01	< .001 ***
Male	Swimming	9.13	0.66	7.84	10.41	< .001 ***
Male	Reading	-3.74	0.62	-4.96	-2.52	< .001 ***
Female	Jogging	9.86	0.70	8.49	11.24	< .001 ***
Female	Swimming	5.15	0.66	3.86	6.44	< .001 ***
Female	Reading	-2.19	0.58	-3.32	-1.06	< .001 ***

Visualization

```

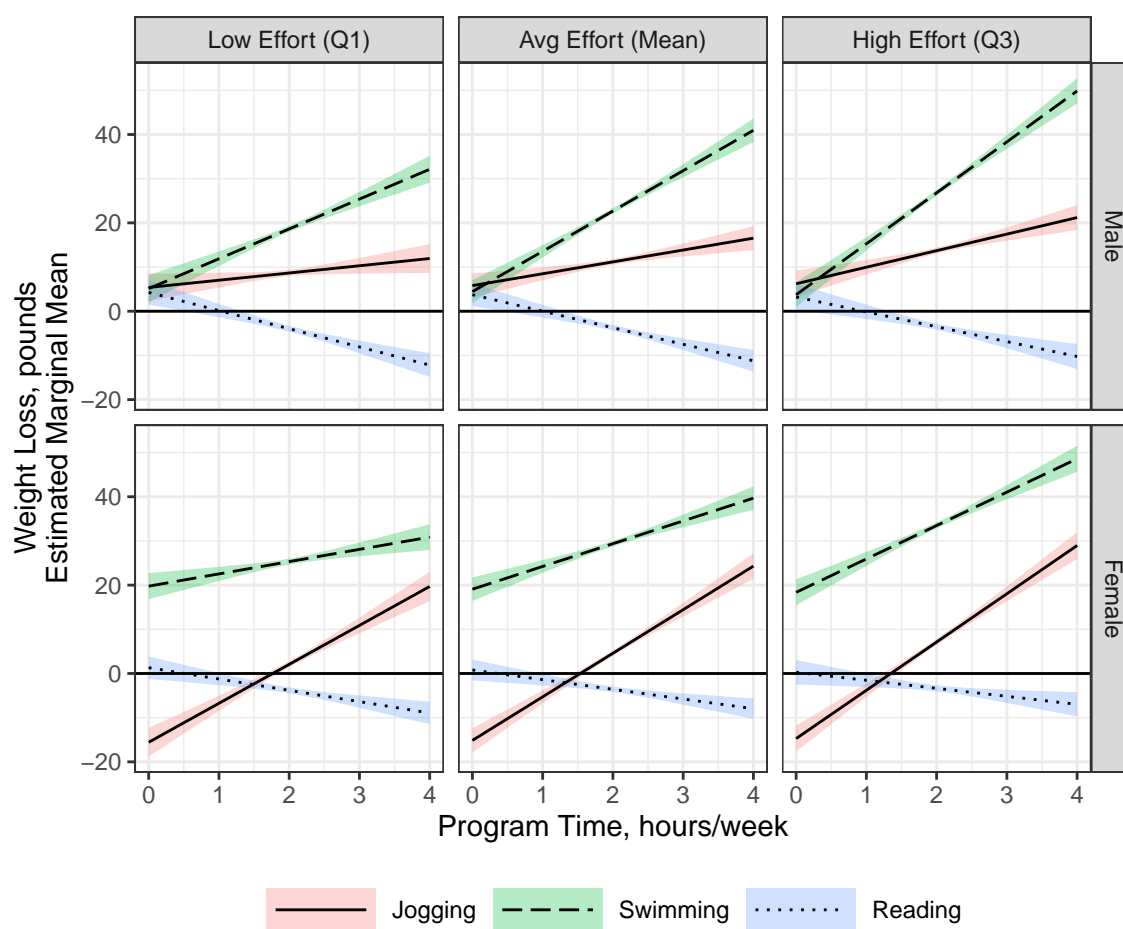
effects::Effect(mod = fit_2,
  focal.predictors = c("prog", "hours",
    "gender", "effort"),
  xlevels = list(hours = seq(from = 0,
    to = 4,
    by = .1),
    effort = c(26.26, 29.66, 33.10))) %>%
data.frame() %>%
dplyr::mutate(effort = effort %>%
  factor(levels = c(26.26, 29.66, 33.10),
    labels = c("Low Effort (Q1)",
      "Avg Effort (Mean)",
      "High Effort (Q3)"))) %>%

ggplot(aes(x = hours,
  y = fit,
  group = prog)) +
geom_ribbon(aes(ymin = lower,
  ymax = upper,
  fill = prog),
  alpha = .3) +
geom_line(aes(linetype = prog)) +
theme_bw() +
facet_grid(gender ~ effort) +
geom_hline(yintercept = 0, color = "black") +
scale_linetype_manual(values = c("solid", "longdash", "dotted")) +
labs(x = "Program Time, hours/week",
  y = "Weight Loss, pounds\nEstimated Marginal Mean",
  linetype = NULL,
  color = NULL,
  fill = NULL) +
theme(legend.position = "bottom",
  legend.key.width = unit(1.5, "cm"))

```

Figure B.32

Weight Loss Uniquely Moderated by Program Time and Effort Rating According to Gender and Program Type



FINAL RESULTS

RQ (1)

I'm just starting out and don't want to put in too much effort.

How many hours per week of exercise do I need to put in to lose 5 pounds?

Constraints

- Exclude the **Reading** program
- Set **effort** to a low value of 26.26 out of 50 which is the 25th percentile (Q1)
- Invert Predictions

Inverse Estimation

```

tab_rq1 <- fit_2 %>%
  effects::Effect(mod = .,
                  focal.predictors = c("prog", "hours",
                                       "gender", "effort"),
                  xlevels = list(hours = seq(from = 0,
                                             to = 4,
                                             by = .1),
                                effort = c(26.26, 33.10))) %>%

  data.frame() %>%
  dplyr::filter(prog != "Reading") %>%
  dplyr::filter(effort == 26.26) %>%
  dplyr::filter(fit > 0) %>%
  dplyr::mutate(dist5 = abs(fit - 5)) %>%
  dplyr::group_by(prog, gender) %>%
  dplyr::mutate(dist5_min = min(dist5)) %>%
  dplyr::filter(dist5 == dist5_min) %>%
  dplyr::ungroup() %>%
  dplyr::arrange(prog, gender) %>%
  dplyr::select("Program" = prog,
               "Gender" = gender,
               "Exercise Time, hours/week" = hours,
               "Weight Loss, pounds" = fit) %>%
  flextable::flextable() %>%
  flextable::set_caption(caption = "RQ (1) Inverse Estimation of Exercise Time Required by Each I

tab_rq1

```

Table B.12

RQ (1) Inverse Estimation of Exercise Time Required by Each Program for a Five Pound Weight Loss, Given Low Effort

Program	Gender	Exercise Time, hours/week	Weight Loss, pounds
Jogging	Male	0.00	5.38
Jogging	Female	2.30	4.71
Swimming	Male	0.00	5.12
Swimming	Female	0.00	19.77

Visualization

```

effects::Effect(mod = fit_2,
                 focal.predictors = c("prog", "hours",
                                     "gender", "effort"),
                 xlevels = list(hours = seq(from = 0,
                                             to = 4,
                                             by = .1),
                                effort = c(26.26, 29.66, 33.10))) %>%

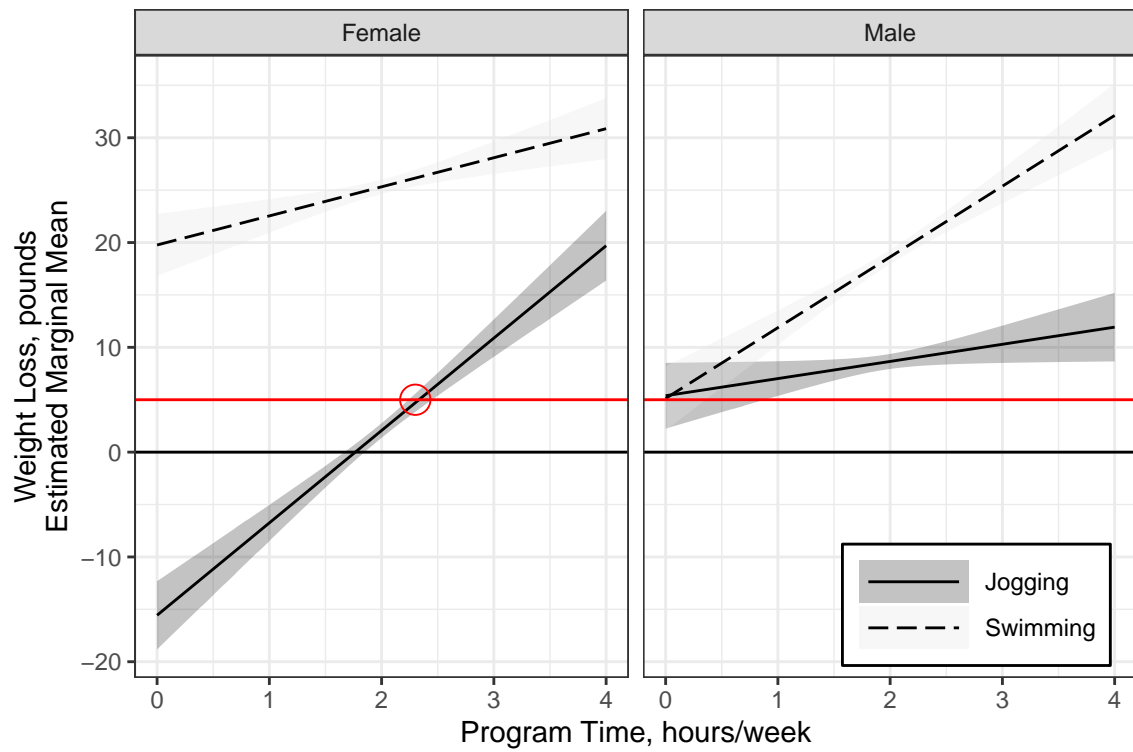
data.frame() %>%
dplyr::filter(prog != "Reading") %>%
dplyr::filter(effort == 26.26) %>%
ggplot(aes(x = hours,
            y = fit,
            group = prog)) +
geom_ribbon(aes(ymin = lower,
               ymax = upper,
               fill = prog),
           alpha = .3) +
geom_line(aes(linetype = prog)) +
theme_bw() +
facet_grid(~ gender) +
geom_hline(yintercept = 0,
           color = "black")+
geom_hline(yintercept = 5,
           color = "red")+
scale_linetype_manual(values = c("solid", "longdash", "dotted")) +
scale_fill_manual(values = c("gray20", "gray90")) +
scale_y_continuous(breaks = seq(from = -20, to = 50, by = 10)) +
labs(x = "Program Time, hours/week",
     y = "Weight Loss, pounds\nEstimated Marginal Mean",
     linetype = NULL,
     fill = NULL) +
theme(legend.position = c(1, 0),
      legend.justification = c(1.1, -.1),
      legend.background = element_rect(color = "black"),
      legend.key.width = unit(1.5, "cm")) +
geom_point(data = data.frame(gender = "Female",
                             prog = "Jogging",
                             hours = 2.3,
                             fit = 5),

           color = "red",
           shape = 1,
           size = 5)

```

Figure B.33

RQ (1) Compare Exercise Time Required by Each Program for a Five Pound Weight Loss, Given Low Effort



Conclusion

- Even when putting in little effort, both men and women on the swimming program should expect at least a 5 pound weight loss, irrespective of weekly time spent.
- While men should also expect a 5+ pound weight loss on the jogging program, irrespective of weekly time spent, women will need to plan on spending at least 2 hours and 20 min per week running.

RQ (2)

I'm moderately fit and can put in an average level of effort into my workout.

For every one hour increase per week in exercise, how much additional weight loss do I expect?

Constraints

- Exclude the Reading program
- Set `effort` to the mean value
- Interpret the slopes, `b`

Simple Slopes Analysis

```
sim_slope_M <- fit_2 %>%
  emmeans::emtrends(var = "hours",
                    ~ prog | gender) %>%
  data.frame() %>%
  dplyr::filter(prog != "Reading")
```

```
tab_rq2 <- sim_slope_M %>%
  dplyr::select(Program = prog,
                Gender = gender,
                Slope = hours.trend,
                SE, lower.CL, upper.CL) %>%
  dplyr::mutate_if(is.numeric, apaSupp::apa2) %>%
  flextable::flextable() %>%
  flextable::set_caption("RQ (2) Additional Weight Loss per Hour/Week Spent on Each Exercise Program with Average Effort, by Gender")

tab_rq2
```

Table B.13

RQ (2) Additional Weight Loss per Hour/Week Spent on Each Exercise Program with Average Effort, by Gender

Program	Gender	Slope	SE	lower.CL	upper.CL
Jogging	Male	2.68	0.67	1.36	4.01
Swimming	Male	9.13	0.66	7.84	10.41
Jogging	Female	9.86	0.70	8.49	11.24
Swimming	Female	5.15	0.66	3.86	6.44

Visualize

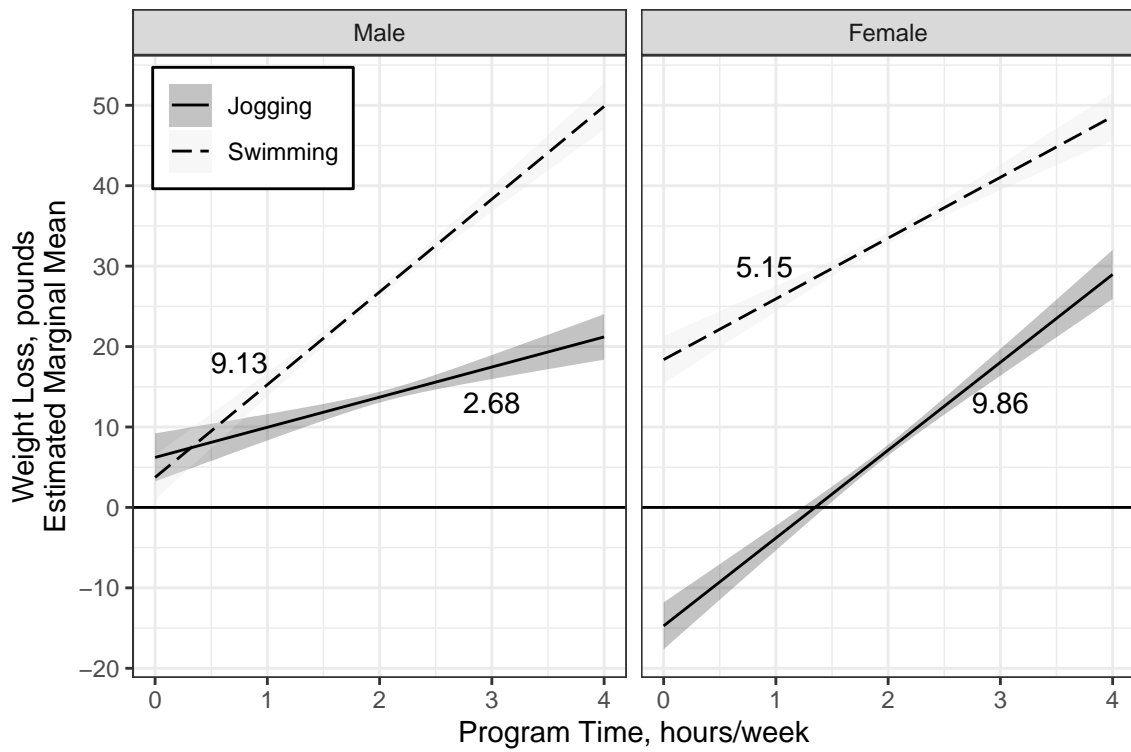
```

effects::Effect(mod = fit_2,
  focal.predictors = c("prog", "hours",
    "gender", "effort"),
  xlevels = list(hours = seq(from = 0,
    to = 4,
    by = .1),
    effort = c( 26.26, 29.63, 33.10))) %>%
  data.frame() %>%
  dplyr::filter(prog != "Reading") %>%
  dplyr::filter(effort == 33.10) %>%
  ggplot(aes(x = hours,
    y = fit,
    group = prog)) +
  geom_ribbon(aes(ymin = lower,
    ymax = upper,
    fill = prog),
    alpha = .3) +
  geom_line(aes(linetype = prog)) +
  theme_bw() +
  facet_grid(~ gender) +
  geom_hline(yintercept = 0,
    color = "black")+
  scale_linetype_manual(values = c("solid", "longdash", "dotted")) +
  scale_y_continuous(breaks = seq(from = -20, to = 50, by = 10)) +
  scale_fill_manual(values = c("gray20", "gray90")) +
  labs(x = "Program Time, hours/week",
    y = "Weight Loss, pounds\nEstimated Marginal Mean",
    linetype = NULL,
    fill = NULL) +
  theme(legend.position = c(0, 1),
    legend.justification = c(-.1, 1.1),
    legend.background = element_rect(color = "black")) +
  geom_text(data = sim_slope_M %>%
    dplyr::mutate(hours = c(3, .75, 3, 0.9),
      fit = c(13, 18, 13, 30)),
    aes(label = glue::glue("{apaSupp::apa2(hours.trend)}"))))

```

Figure B.34

RQ (2) Compare Additional Weight Loss per Hour/Week Spent on Each Exercise Program with Average Effort, by Gender



Conclusion

An hour increase per week in exercise, will result in a different amount of additional total weight loss, depending on gender and program type.

For women, an hour/week increase in exercise will translate to 9.86 more pounds loss if jogging and 5.15 if swimming.

For men, an hour/week increase in exercise will translate to 2.68 more pounds loss if jogging and 9.13 if swimming.

RQ (3)

I'm a crossfit athlete and can perform with the utmost intensity.

How much more weight loss would I expect for every one hour increase in exercise compared to the average amount of effort most people put in?

Additionally, we can visualize the interaction to help us understand these relationships.

Constraints

- Exclude the **Reading** program
- Set **effort** to near the max: 45
- Interpret the slopes, **b**

Simple Slopes Analysis

```
sim_slope_Max <- fit_2 %>%
  emmeans::emtrends(var = "hours",
                    ~ prog | gender,
                    at = list(effort = 45)) %>%
  data.frame() %>%
  dplyr::filter(prog != "Reading")

tab_rq3a <- sim_slope_Max %>%
  dplyr::select(Program = prog,
                Gender = gender,
                Slope = hours.trend,
                SE, lower.CL, upper.CL) %>%
  dplyr::mutate_if(is.numeric, apaSupp::apa2) %>%
  flextable::flextable() %>%
  flextable::set_caption("RQ (3) Additional Weight Loss per Hour/Week Spent on Each Exercise Program with Near Maximum Effort, by Gender")

tab_rq3a
```

Table B.14

RQ (3) Additional Weight Loss per Hour/Week Spent on Each Exercise Program with Near Maximum Effort, by Gender

Program	Gender	Slope	SE	lower.CL	upper.CL
Jogging	Male	7.41	1.62	4.24	10.59
Swimming	Male	19.85	1.49	16.94	22.77
Jogging	Female	14.59	1.62	11.41	17.77
Swimming	Female	15.88	1.56	12.82	18.93

```

sim_slopes_cmp <- fit_2 %>%
  emmeans::emtrends(var = "hours",
                    ~ effort | prog + gender,
                    at = list(effort = c(29.63, 45))) %>%

  pairs() %>%
  data.frame() %>%
  dplyr::filter(prog != "Reading") %>%
  dplyr::mutate(Difference = abs(estimate)) %>%
  dplyr::mutate(p = apaSupp::p_num(p.value)) %>%
  dplyr::mutate_if(is.numeric, apaSupp::apa2) %>%
  dplyr::select(prog, gender, Difference, SE, p)

```

```

sim_slopes_M45 <- fit_2 %>%
  emmeans::emtrends(var = "hours",
                    ~ effort | prog + gender,
                    at = list(effort = c(29.63, 45))) %>%

  data.frame() %>%
  dplyr::select(prog, gender, effort, hours.trend) %>%
  dplyr::filter(prog != "Reading")

```

```

tab_rq3b <- sim_slopes_M45 %>%
  tidyr::pivot_wider(names_from = effort,
                     names_prefix = "b_",
                     values_from = hours.trend) %>%
  dplyr::rename("b,\nAverage" = b_29.63,
               "b,\nNear Max" = b_45) %>%
  dplyr::mutate_if(is.numeric, apaSupp::apa2) %>%
  dplyr::full_join(sim_slopes_cmp, by = c("prog", "gender")) %>%
  dplyr::rename(Program = prog,
                Gender = gender,
                "b, Difference" = "Difference") %>%
  flextable::flextable() %>%
  flextable::set_caption("RQ (3) Additional Weight Loss per Hour/Week Spent on Each Exercise Program with Near Max vs. Mean Effort, by Gender")

tab_rq3b

```

Table B.15

RQ (3) Additional Weight Loss per Hour/Week Spent on Each Exercise Program with Near Max vs. Mean Effort, by Gender

Program	Gender	b, Average	b, Near Max	b, Difference	SE	p
Jogging	Male	2.68	7.41	4.74	1.56	.002 **
Swimming	Male	9.11	19.85	10.75	1.41	< .001 ***
Jogging	Female	9.85	14.59	4.74	1.56	.002 **
Swimming	Female	5.13	15.88	10.75	1.41	< .001 ***

Visualize

```

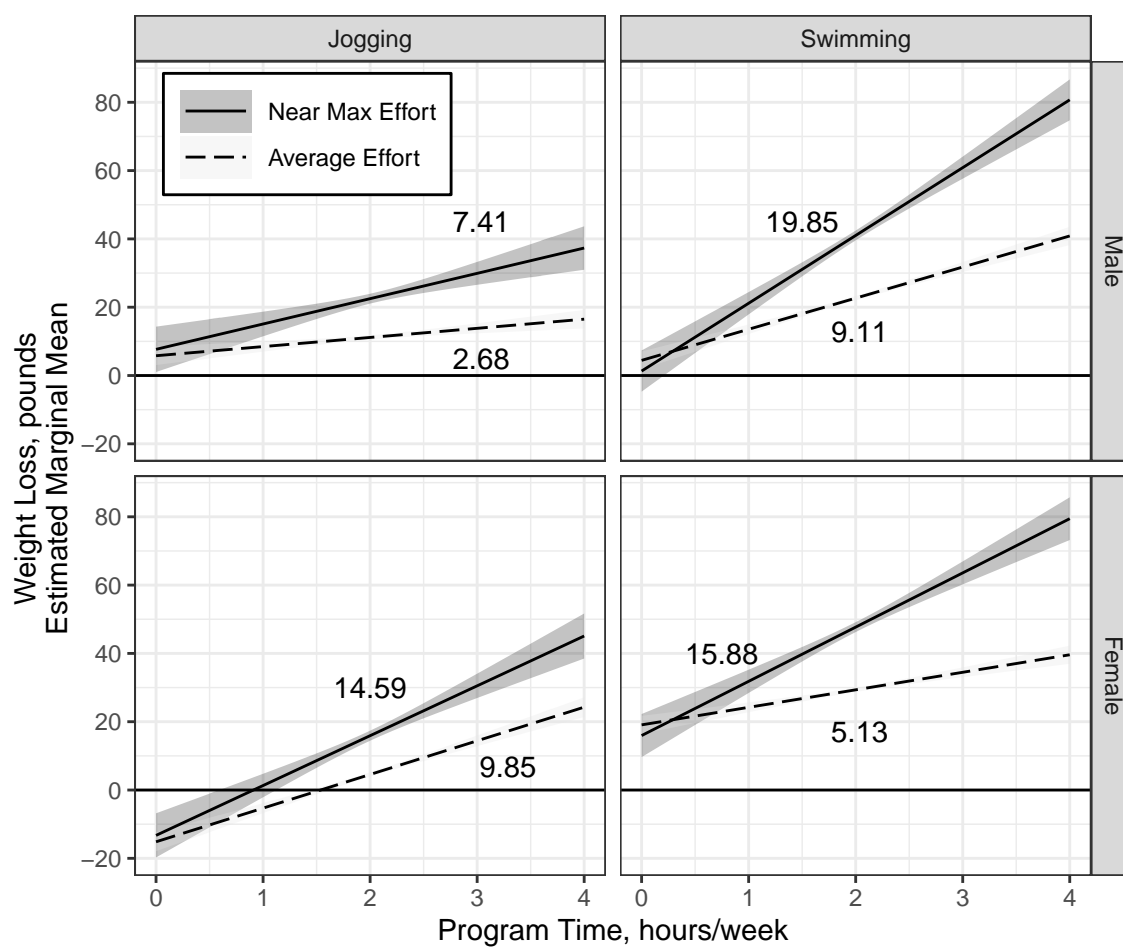
effects::Effect(mod = fit_2,
  focal.predictors = c("prog", "hours",
    "gender", "effort"),
  xlevels = list(hours = seq(from = 0,
    to = 4,
    by = .1),
    effort = c(29.63, 45))) %>%
  data.frame() %>%
  dplyr::filter(prog != "Reading") %>%
  dplyr::mutate(effort = effort %>%
    factor(levels = c(45, 29.63),
      labels = c("Near Max Effort",
        "Average Effort")))) %>%

  ggplot(aes(x = hours,
    y = fit,
    group = effort)) +
  geom_ribbon(aes(ymin = lower,
    ymax = upper,
    fill = effort),
    alpha = .3) +
  geom_line(aes(linetype = effort)) +
  theme_bw() +
  facet_grid(gender ~ prog) +
  geom_hline(yintercept = 0,
    color = "black") +
  scale_linetype_manual(values = c("solid", "longdash", "dotted")) +
  labs(x = "Program Time, hours/week",
    y = "Weight Loss, pounds\nEstimated Marginal Mean",
    linetype = NULL,
    fill = NULL) +
  scale_fill_manual(values = c("gray20", "gray90")) +
  scale_y_continuous(breaks = seq(from = -20, to = 100, by = 20)) +
  theme(legend.position = c(0, 1),
    legend.justification = c(-.1, 1.1),
    legend.background = element_rect(color = "black"),
    legend.key.width = unit(1, "cm")) +
  geom_text(data = sim_slopes_M45 %>%
    dplyr::mutate(hours = c(3, 3, 2, 1.5,
      3.25, 2, 2, 0.75),
      fit = c(5, 45, 13, 45,
        7, 30, 17, 40)) %>%
    dplyr::mutate(effort = effort %>%
      factor(levels = c(45, 29.63),
        labels = c("Near Max",
          "Average")))),
    aes(label = glue::glue("{apaSupp::apa2(hours.trend)}"))))

```

Figure B.35

RQ (3) Compare Effort Discrepancies in Additional Weight Loss Associated with an Hour Increase per Week in Each Exercise Program, by Gender



Conclusion

For the jogging program, both males and females with high effort can expect to loose an additional 1.07 pound on average for every one hour increase in exercise compared to people who put in average effort. This discrepancy is 2.43 for the swimming program.

SESSION INFORMATION

```
sessionInfo()
```

```
R version 4.3.1 (2023-06-16 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 11 x64 (build 22631)
```

```
Matrix products: default
```

```
locale:
[1] LC_COLLATE=English_United States.utf8
[2] LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8
```

```
time zone: America/Denver
tzcode source: internal
```

```
attached base packages:
[1] stats      graphics  grDevices  utils      datasets  methods    base
```

```
other attached packages:
[1] performance_0.12.0 interactions_1.1.5 emmeans_1.10.2    texreg_1.39.3
[5] DHARMA_0.4.6      olsrr_0.6.0      MOTE_1.0.2        ggpubr_0.6.0
[9] rstatix_0.7.2     naniar_1.1.0     flextable_0.9.6   psych_2.4.3
[13] furniture_1.9.14  lubridate_1.9.3  forcats_1.0.0     stringr_1.5.1
[17] dplyr_1.1.4       purrr_1.0.2      readr_2.1.5       tidyr_1.3.1
[21] tibble_3.2.1      ggplot2_3.5.1    tidyverse_2.0.0   apaSupp_0.0.0.9000
```

```
loaded via a namespace (and not attached):
[1] rstudioapi_0.16.0      jsonlite_1.8.8        magrittr_2.0.3
[4] TH.data_1.1-2          estimability_1.5.1    corrplot_0.92
[7] farver_2.1.2           nloptr_2.0.3          rmarkdown_2.27
[10] ragg_1.3.2             vctrs_0.6.5           minqa_1.2.7
[13] askpass_1.2.0          tinytex_0.51          htmltools_0.5.8.1
[16] curl_5.2.1             survey_4.4-2          broom_1.0.6
[19] sass_0.4.9             bslib_0.7.0           plyr_1.8.9
[22] sandwich_3.1-0         cachem_1.1.0          zoo_1.8-12
[25] uuid_1.2-0             mime_0.12             lifecycle_1.0.4
[28] pkgconfig_2.0.3        Matrix_1.6-5          R6_2.5.1
[31] fastmap_1.2.0          shiny_1.8.1.1         digest_0.6.35
[34] colorspace_2.1-0       reshape_0.8.9         textshaping_0.4.0
[37] labeling_0.4.3         effects_4.2-2         fansi_1.0.6
[40] timechange_0.3.0       httr_1.4.7            abind_1.4-5
[43] mgcv_1.9-1             compiler_4.3.1        fontquiver_0.2.1
[46] withr_3.0.0            pander_0.6.5          backports_1.5.0
[49] DBI_1.2.3              carData_3.0-5         ggsignif_0.6.4
[52] MASS_7.3-60           openssl_2.2.0         gfonts_0.2.0
[55] tools_4.3.1           pbivnorm_0.6.0        zip_2.3.1
```

[58] httpuv_1.6.15	visdat_0.6.0	nnet_7.3-19
[61] goftest_1.2-3	glue_1.7.0	quadprog_1.5-8
[64] nlme_3.1-165	promises_1.3.0	grid_4.3.1
[67] reshape2_1.4.4	generics_0.1.3	gtable_0.3.5
[70] MBESS_4.9.3	nortest_1.0-4	tzdb_0.4.0
[73] data.table_1.15.4	hms_1.1.3	xml2_1.3.6
[76] car_3.1-2	utf8_1.2.4	pillar_1.9.0
[79] later_1.3.2	mitools_2.4	splines_4.3.1
[82] lattice_0.21-8	survival_3.7-0	tidyselect_1.2.1
[85] fontLiberation_0.1.0	knitr_1.48	fontBitstreamVera_0.1.1
[88] gridExtra_2.3	crul_1.4.2	stats4_4.3.1
[91] xfun_0.44	stringi_1.8.4	yaml_2.3.8
[94] boot_1.3-28.1	evaluate_0.24.0	codetools_0.2-19
[97] httpcode_0.3.0	officer_0.6.6	gdtools_0.3.7
[100] cli_3.6.2	xtable_1.8-4	systemfonts_1.1.0
[103] jquerylib_0.1.4	munsell_0.5.1	lavaan_0.6-18
[106] Rcpp_1.0.12	coda_0.19-4.1	parallel_4.3.1
[109] lme4_1.1-35.3	mvtnorm_1.2-5	scales_1.3.0
[112] insight_0.20.0	ez_4.4-0	crayon_1.5.3
[115] rlang_1.1.4	multcomp_1.4-25	mnormt_2.1.1
[118] jtools_2.2.2		

BIBLIOGRAPHY

- Barrett, T. S., & Brignone, E. (2017). Furniture for quantitative scientists. *The R Journal*, 9(2), 142–148. <https://doi.org/10.32614/RJ-2017-037>
- Barrett, T. S., Brignone, E., & Laxman, D. J. (2023). *Furniture: Furniture for quantitative scientists*. <https://CRAN.R-project.org/package=furniture>
- Buchanan, E. M., Gillenwaters, A. M., Scofield, J. E., & Valentine, K. D. (2019). *MOTE: Effect size and confidence interval calculator*. <https://CRAN.R-project.org/package=MOTE>
- Gohel, D., & Skintzos, P. (2024). *Flextable: Functions for tabular reporting*. <https://ardata-fr.github.io/flextable-book/>
- Grolemund, G., & Wickham, H. (2011). Dates and times made easy with lubridate. *Journal of Statistical Software*, 40(3), 1–25. <https://www.jstatsoft.org/v40/i03/>
- Hartig, F. (2022). *DHARMA: Residual diagnostics for hierarchical (multi-level / mixed) regression models*. <http://florianhartig.github.io/DHARMA/>
- Hebbali, A. (2024). *Olsrr: Tools for building OLS regression models*. <https://olsrr.rsquaredacademy.com/>
- Kassambara, A. (2023a). *Ggpubr: ggplot2 based publication ready plots*. <https://rpkgs.datanovia.com/ggpubr/>
- Kassambara, A. (2023b). *Rstatix: Pipe-friendly framework for basic statistical tests*. <https://rpkgs.datanovia.com/rstatix/>
- Leifeld, P. (2013). texreg: Conversion of statistical model output in R to LaTeX and HTML tables. *Journal of Statistical Software*, 55(8), 1–24. <https://doi.org/10.18637/jss.v055.i08>
- Leifeld, P. (2023). *Texreg: Conversion of r regression output to LaTeX or HTML tables*. <https://github.com/leifeld/texreg/>
- Lenth, R. V. (2024). *Emmeans: Estimated marginal means, aka least-squares means*. <https://rvinth.github.io/emmeans/>
- Long, J. A. (2021). *Interactions: Comprehensive, user-friendly toolkit for probing interactions*. <https://interactions.jacob-long.com>
- Lüdecke, D., Ben-Shachar, M. S., Patil, I., Waggoner, P., & Makowski, D. (2021). performance: An R package for assessment, comparison and testing of statistical models. *Journal of Open Source Software*, 6(60), 3139. <https://doi.org/10.21105/joss.03139>
- Lüdecke, D., Makowski, D., Ben-Shachar, M. S., Patil, I., Waggoner, P., Wiernik, B. M., & Thériault, R. (2024). *Performance: Assessment of regression models performance*. <https://easystats.github.io/>

- ub.io/performance/
- Müller, K., & Wickham, H. (2023). *Tibble: Simple data frames*. <https://tibble.tidyverse.org/>
- R Core Team. (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. <https://www.R-project.org/>
- Revelle, W. (2024). *Psych: Procedures for psychological, psychometric, and personality research*. <https://personality-project.org/r/psych/>
- Schwartz, S. (2024). *apaSupp: APA formatted supplemental materials*. <https://github.com/SarBearSchwartz/apaSupp>
- Schwartz, S. E. (2024). *apaSup: An r package for preparing supplemental material, APA 7th ed. Open Science Framework*. <https://doi.org/10.17605/OSF.IO/G38JN>
- Spinu, V., Grolemund, G., & Wickham, H. (2023). *Lubridate: Make dealing with dates a little easier*. <https://lubridate.tidyverse.org>
- Tierney, N., & Cook, D. (2023). Expanding tidy data principles to facilitate missing data exploration, visualization and assessment of imputations. *Journal of Statistical Software*, 105(7), 1–31. <https://doi.org/10.18637/jss.v105.i07>
- Tierney, N., Cook, D., McBain, M., & Fay, C. (2024). *Naniar: Data structures, summaries, and visualisations for missing data*. <https://github.com/njtierney/naniar>
- Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. <https://ggplot2.tidyverse.org>
- Wickham, H. (2023a). *Forcats: Tools for working with categorical variables (factors)*. <https://forcats.tidyverse.org/>
- Wickham, H. (2023b). *Stringr: Simple, consistent wrappers for common string operations*. <https://stringr.tidyverse.org>
- Wickham, H. (2023c). *Tidyverse: Easily install and load the tidyverse*. <https://tidyverse.tidyverse.org>
- Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., ... Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43), 1686. <https://doi.org/10.21105/joss.01686>
- Wickham, H., Chang, W., Henry, L., Pedersen, T. L., Takahashi, K., Wilke, C., Woo, K., Yutani, H., Dunnington, D., & van den Brand, T. (2024). *ggplot2: Create elegant data visualisations using the grammar of graphics*. <https://ggplot2.tidyverse.org>
- Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *Dplyr: A grammar of*

data manipulation. <https://dplyr.tidyverse.org>

Wickham, H., & Henry, L. (2023). *Purrr: Functional programming tools.* <https://purrr.tidyverse.org/>

Wickham, H., Hester, J., & Bryan, J. (2024). *Readr: Read rectangular text data.* <https://readr.tidyverse.org>

Wickham, H., Vaughan, D., & Girlich, M. (2024). *Tidyr: Tidy messy data.* <https://tidyr.tidyverse.org>