Ido and Noa’s Project:

This project is Ido Levy and Noa Levy’s final project in the course ‘Introduction to Statistics and Data Analysis with R’.

Introduction:

In this project we’re going to zoom in on a data set that records Ido’s resistance training sessions. Using the knowledge we acquired during the course ‘Introduction to Statistics and Data Analysis with R’ we’ll examine the data and try to extract useful insights.

About the data:

The data was recorded with and exported from FitNotes - a gym log app.

Each row in the data represents a set.

Background:

Why is tracking performance important?

An improvement in performance is probably one of the best and most easily accessible indicators of muscle growth.

Assuming muscle growth is the goal, this makes routinely recording performance crucial for

tracking progress.

Throughout the project we'll call an improvement in performance a PR – a personal record in a specific exercise's performance.

Our criteria for a personal record are performing more repetitions with a given weight, or given a number of repetitions using a heavier weight.

Data import and transformation:

Reticulate enables integration of Python in .rmd documents:

```
library(reticulate)
```

The Exercise class helps us determine when prs occur.

```
class Exercise:
    def __init__(self):
        # Key - value pairs of weight(kg): maximum number of repetitions performed with that weight
        self.weight_dict = {}
        # Key - value pairs of number of repetitions: maximum weight(kg) lifted with that number of repetitions
        self.rep_dict = {}
```

These functions create additional essential columns for our data table.

```
def is_pr(training_data_path):
    """
    Given a path to a training data file, returns a list of 1s and 0s indicating whether each set in the file
    represents a personal record (PR).

    :param training_data_path: A string representing the path to the training data file
```

```
    :return: A list of 1s and 0s indicating whether each set in the file represents a personal record (PR)
    """

    with open(training_data_path, 'r') as fo:
        line_list = fo.readlines()

        # Key - value pairs of exercise name: a corresponding Exercise object
        exercise_dict = {}
        is_pr_list = []
        for line in line_list[1:]:
            tokens_list = line.split(',')
            exercise = tokens_list[1]
            weight = float(tokens_list[3])
            rep = int(tokens_list[4])
            is_rep_pr = False
            is_weight_pr = False
            if exercise not in exercise_dict:
                exercise_dict[exercise] = Exercise()
                exercise_dict[exercise].weight_dict[weight] = rep
                exercise_dict[exercise].rep_dict[rep] = weight
            else:
                if weight not in exercise_dict[exercise].weight_dict:
                    exercise_dict[exercise].weight_dict[weight] = rep
                elif rep > exercise_dict[exercise].weight_dict[weight]:
                    exercise_dict[exercise].weight_dict[weight] = rep
                    is_rep_pr = True
                if rep not in exercise_dict[exercise].rep_dict:
                    exercise_dict[exercise].rep_dict[rep] = weight
                elif weight > exercise_dict[exercise].rep_dict[rep]:
                    exercise_dict[exercise].rep_dict[rep] = weight
                    is_weight_pr = True
            if is_rep_pr or is_weight_pr:
                is_pr_list.append(1)
            else:
                is_pr_list.append(0)
        return is_pr_list



def is_first(training_data_path):
```

```
    """
    Given a path to a training data file, returns a list of 1s and 0s indicating whether each set in the file
    represents the first set of a workout. The function assumes that the training data file is
    sorted by date in ascending order.

    :param training_data_path: A string representing the path to the training data file
    :return: A list of 1s and 0s indicating whether each set in the file represents the first set of a workout
    """

    with open(training_data_path, 'r') as fo:
        line_list = fo.readlines()
        tokens_list = line_list[1].split(',')[0].split('/')
        prev_date = datetime.date(int(tokens_list[2]), int(tokens_list[1]), int(tokens_list[0]))
        is_first_list = [1]
        for line in line_list[2:]:
            tokens_list = line.split(',')[0].split('/')
            curr_date = datetime.date(int(tokens_list[2]), int(tokens_list[1]), int(tokens_list[0]))
            if curr_date > prev_date:
                is_first_list.append(1)
            else:
                is_first_list.append(0)
            prev_date = curr_date
        return is_first_list
```

Using the above functions:

```
is_pr_list = is_pr(r"C:\Users\IDOLE\OneDrive\Desktop\Courses\Data Analysis\Project\training_data.csv")
is_first_list = is_first(r"C:\Users\IDOLE\OneDrive\Desktop\Courses\Data Analysis\Project\training_data.csv")
```

Reading the data and adding the lists we created as new columns to the data table:

```
training_data <- read_csv("training_data.csv", show_col_types = FALSE)
training_data$is_pr = py$is_pr_list
training_data$is_first = py$is_first_list
```

Is there a relationship between hitting a pr on a set and a set's position in the workout?

To answer this question we'll perform an independence test between is_pr and is_first.

Let’s construct a contingency table:

```
contingency_table <- training_data %>% count(is_pr, is_first) %>% pivot_wider(names_from = is_pr, values_from = n, names_prefix = 'is_pr: ')
contingency_table
```

```
## # A tibble: 2 x 3
##   is_first `is_pr: 0` `is_pr: 1`
##      <int>      <int>      <int>
## 1       0        185         37
## 2       1         10         14
```

Null hypothesis: is_pr and is_first are independent variables.

Alternative hypothesis: is_pr and is_first are dependent variables.

```
chisq.test(contingency_table %>% select(2:3) %>% as.matrix(), correct = FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table %>% select(2:3) %>% as.matrix()
## X-squared = 22.881, df = 1, p-value = 1.724e-06
```

Since the p - value is less than 0.05 we reject the null hypothesis. Meaning there is a statistically significant relationship between pring and a set’s position in the workout.

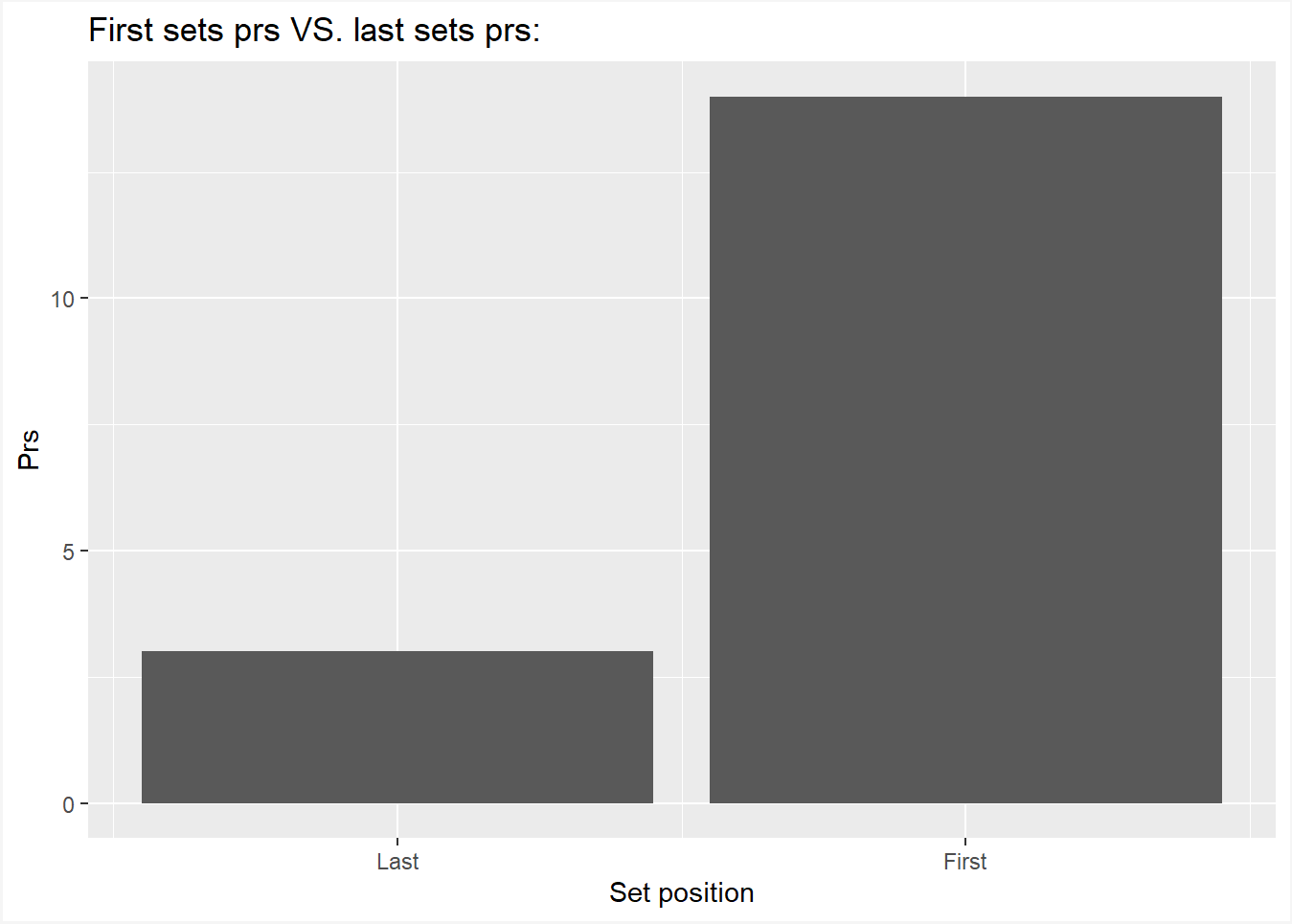
Let’s further explore this relationship.

How many prs were hit on first sets of workouts compared to last sets of workouts?

```
# Similar concept to is_first_list. A list containing an element for each set recorded in the data.
# If the set was the last set of the workout it's corresponding element in the list is 1. Otherwise it's corresponding element in the list is 0.
```

```
 is_last_list = is_first_list[1:] + [1]
```

```
training_data$is_last = py$is_last_list
ggplot(training_data %>% filter(is_first == 1 | is_last == 1), aes(x = is_first, y = is_pr)) + geom_col() + scale_x_continuous(name = 'Set position', breaks = c(0, 1), labels = c('Last', 'First')) + ylab('Prs') + ggtitle('First sets prs VS. last sets prs:')
```



This figure leads us to the following question - *is the probability to pr given it's the first set of the workout greater than the probability to pr given it's the last set of the workout?*

Let's perform a two populations proportions test.

Null hypothesis: P(pr | first set of the workout) = P(pr | last set of the workout).

Alternative hypothesis: P(pr | first set of the workout) > P(pr | last set of the workout).

```
# Again, similar concept. A list containing an element for each set recorded in the data.
# The workouts recorded in the data get a number based on their order of execution.
# Each set's corresponding element in the list is the number of the workout it was performed in.

workout_number_list = [1]
for i, is_first in enumerate(is_first_list[1:]):
    workout_number_list.append(workout_number_list[i - 1] + is_first)

first_set_prs <- Training_data %>% filter(is_pr == 1, is_first == 1) %>% nrow()
last_set_prs <- Training_data %>% filter(is_pr == 1, is_last == 1) %>% nrow()
```

```
training_data$workout_number = py$workout_number_list
num_of_workouts <- training_data %>% distinct(workout_number) %>% nrow()
prop.test(x = c(first_set_prs, last_set_prs), n = c(num_of_workouts, num_of_workouts), alternative =
"greater", correct = FALSE)

##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(first_set_prs, last_set_prs) out of c(num_of_workouts, num_of_workouts)
## X-squared = 11.021, df = 1, p-value = 0.0004505
## alternative hypothesis: greater
## 95 percent confidence interval:
##  0.2590099 1.0000000
## sample estimates:
##    prop 1    prop 2
## 0.5833333 0.1250000
```

Since the p - value is less than 0.05 we reject the null hypothesis, and we can almost certainly say that the probability to pr given it's the first set of the workout is greater than the probability to pr given it's the last set of the workout.

One way to explain these findings is that in the first set of the workout you're fresh. You haven't accumulated performance damaging fatigue yet. Hence, you can perform your best, and this leads to a higher chance to pr.

We'll argue that performing better not only leads to more prs. More importantly, it gives your target muscles a better stimulus, which ultimately leads to more muscle growth.

So, if you want to prioritize a specific muscle group, then maybe it's a good idea to routinely start your workouts with this muscle group. But, if you want to take a balanced approach to your training, then maybe it's best to do your workouts in a cyclical manner.

To wrap up what we did:

We found a relationship between a workout's first set and performance.

We looked deeper into this relationship and found that the chance to pr given it's the first set of the workout is greater than the chance to pr given it's the last set of the workout.