Spotify Song Analysis

Enzo Moraes Mescall

2023-12-18

Introduction

This report presents a comprehensive historical analysis based on two years of meticulously collected personal workout data. The dataset employed in this report is sourced from my workouts on the Strong app, a workout tracker that enables users to record and monitor their training progress (CITATION). This workout data serves as a valuable repository of potential personal insights derived from a diverse range of workouts, reflecting the evolving nature of my personal fitness over the past two years. The dataset has also been augmented with sleep data collected from the Apple Health app, which may provide additional context to the workout performances (CITATION).

Dataset description and variables of interest

The exercises dataset contains 3,480 observations across 12 key variables, encompassing over 78 unique exercises stretching from January 26th, 2022 to December 12th, 2023. The sleep dataset contains 990 datapoints of hours slept stretching from as early as 2018 to December 18th, 2023. The variables of interest include per-day measures of the following figures:

- `Exercise Name`: The name of the exercise performed
- `Date`: The date of the workout
- `Weight`: The weight lifted in a given set
- `Reps`: The number of repetitions performed for a given exercise
- `Sleep`: The number of hours slept on a given day

And the dataset has been augmented with the following variables:

- `Volume`: The total weight lifted for a given exercise across all sets
- `Max Weight`: The maximum weight lifted for a given exercise across all sets
- `Volume per set`: The average weight lifted for a given exercise across all sets
- `Set number`: The index of the set for a given exercise relative to the whole workout, vaguely representative of how much fatigue has accumulated for a muscle group
- Percent metrics: The percent change in a given metric relative to the previous time an exercise was done (e.g. percent change in volume from the previous time an exercise was done)
- Running average percent metrics: The percent change in a given metric relative to the running average of previous times an exercise was done (e.g. percent change in volume from the running average of previous times an exercise was done)

Objective

Since the dataset surrounds personal data, the conclusions drawn from this report are not intended to be generalized to the broader population. Rather, the goal of this report is to provide a detailed analysis of my personal workout patterns, with the aim of uncovering potential correlations and insights that may be used to optimize future workout strategies.

The central focus of this report revolves around three research questions:

1. How many days of rest should Enzo Moraes Mescall take between workouts to optimize for 1 rep max increase (footnote)?
2. How effectiveley can Enzo Moraes Mescall predict his volume for a given exercise in the context of a workout?
3. What is the effect of sleep on Enzo Moraes Mescall's workout performance?

This report is not only an exploration of personal fitness journeys but also a testament to the potential of data-driven insights in enhancing individual training experiences.

Exploratory Data Analysis

```r
health = read_xml("exportar.xml")
```

```r
# Extract sleep data
sleep_data <- health %>%
  xml_find_all("//Record[@type='HKCategoryTypeIdentifierSleepAnalysis']") %>%
  map_df(~ {
    creationDate <- as.POSIXct(xml_attr(., "creationDate"), format = "%Y-%m-%d")
    startDate <- as.POSIXct(xml_attr(., "startDate"), format = "%Y-%m-%d %H:%M:%S %z")
    endDate <- as.POSIXct(xml_attr(., "endDate"), format = "%Y-%m-%d %H:%M:%S %z")

    sleep_duration <- as.numeric(difftime(endDate, startDate, units = "hours"))

    data.frame(
      Date = creationDate,
      sleep_duration = sleep_duration
    )
  }) %>%
  group_by(Date) %>%
  summarize(sleep = sum(sleep_duration)) %>%
  filter(sleep < 15)

weekly_avg_sleep = sleep_data %>%
  mutate(week_start = floor_date(Date, "week")) %>%
  group_by(week_start) %>%
  summarise(week_sleep = mean(sleep)) %>%
  ungroup()
```

```r
workouts = read_csv("strong.csv")
```

```
## Rows: 3480 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): Workout Name, Duration, Exercise Name, Notes, Workout Notes
## dbl  (5): Set Order, Weight, Reps, Distance, Seconds
## lgl  (1): RPE
## dttm (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
workout_count = workouts %>%
  filter(`Weight` > 0, !str_detect(`Exercise Name`, "Assisted")) %>%
  group_by(`Exercise Name`, `Date`) %>%
  summarise(count = n()) %>%
  group_by(`Exercise Name`) %>%
  summarise(count = n()) %>%
  arrange(desc(count))
```

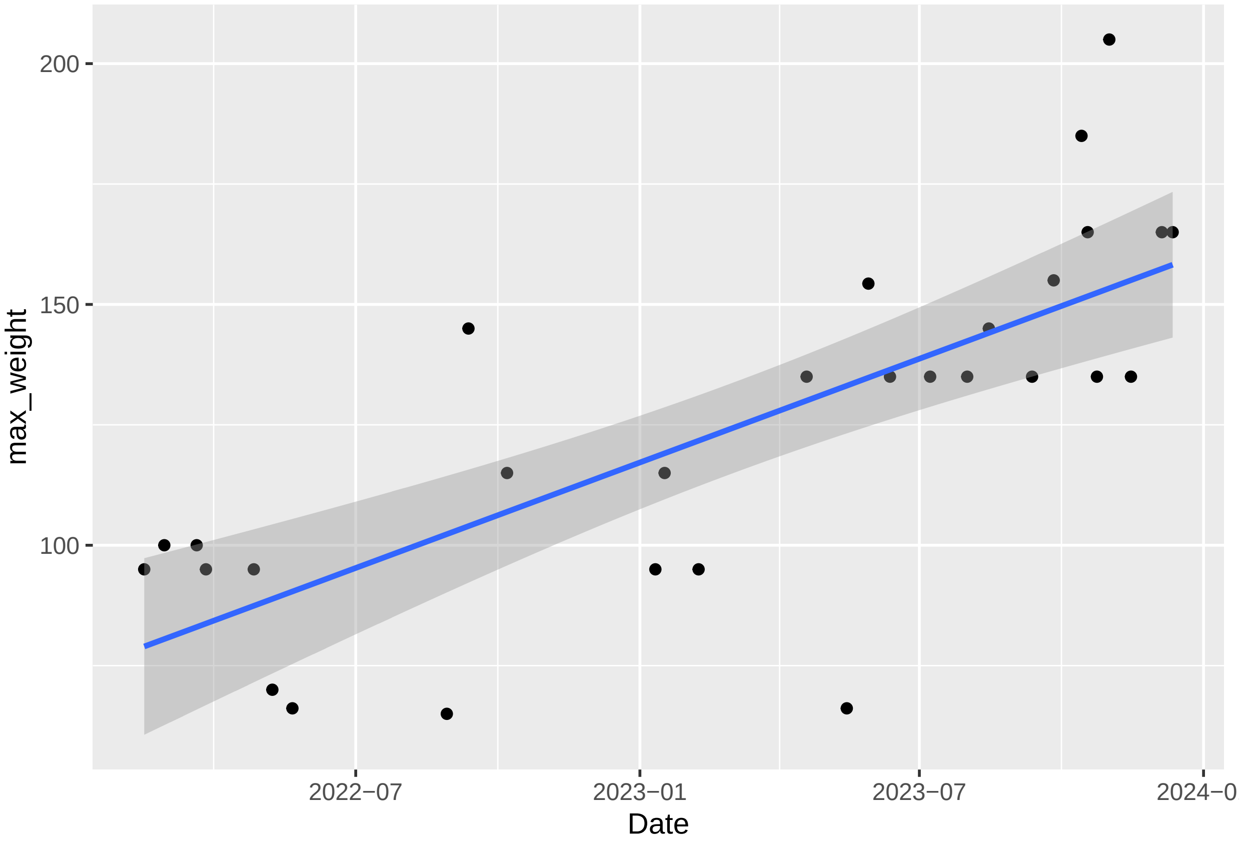
```
## `summarise()` has grouped output by 'Exercise Name'. You can override using the
## `.groups` argument.
```

```
# finding the time between this exercise and the last time it was done
workout_gaps = workouts %>%
  filter(`Weight` > 0) %>%
  group_by(`Date`) %>%
  mutate(set_number = row_number()) %>%
  ungroup() %>%
  group_by(`Exercise Name`, `Date`) %>%
  summarise(volume = sum(`Weight`*`Reps`), max_weight = max(`Weight`),
            volume_per_set = mean(`Weight`*`Reps`),
            set_number = min(`set_number`)) %>%
  arrange(`Exercise Name`, `Date`) %>%
  mutate(gap = `Date` - lag(`Date`),
         pct_volume = (`volume`-lag(`volume`))/lag(`volume`),
         pct_max_weight = (`max_weight`-lag(`max_weight`))/lag(`max_weight`),
         pct_volume_per_set = (`volume_per_set`-lag(`volume_per_set`))/lag(`volume_per_set`),
         pct_volume_runavg = (`volume`-cumsum(`volume`)/row_number())/lag(`volume`),
         pct_max_weight_runavg = (`max_weight`-cumsum(`max_weight`)/row_number())/lag(`max_weight`),
         pct_volume_per_set_runavg = (`volume_per_set`-cumsum(`volume_per_set`)/row_number())/lag(`volu
  ungroup() %>%
  filter(!is.na(gap)) %>%
  mutate(gap = as.numeric(gap, units = "days"),
         Date = as.Date(Date, format = "%Y-%m-%d")) %>%
  right_join(workout_count, by = "Exercise Name") %>%
  left_join(sleep_data, by = "Date") %>%
  mutate(week_start = floor_date(Date, "week")) %>%
  left_join(weekly_avg_sleep, by = "week_start")
```

```
## `summarise()` has grouped output by 'Exercise Name'. You can override using the
## `.groups` argument.
```

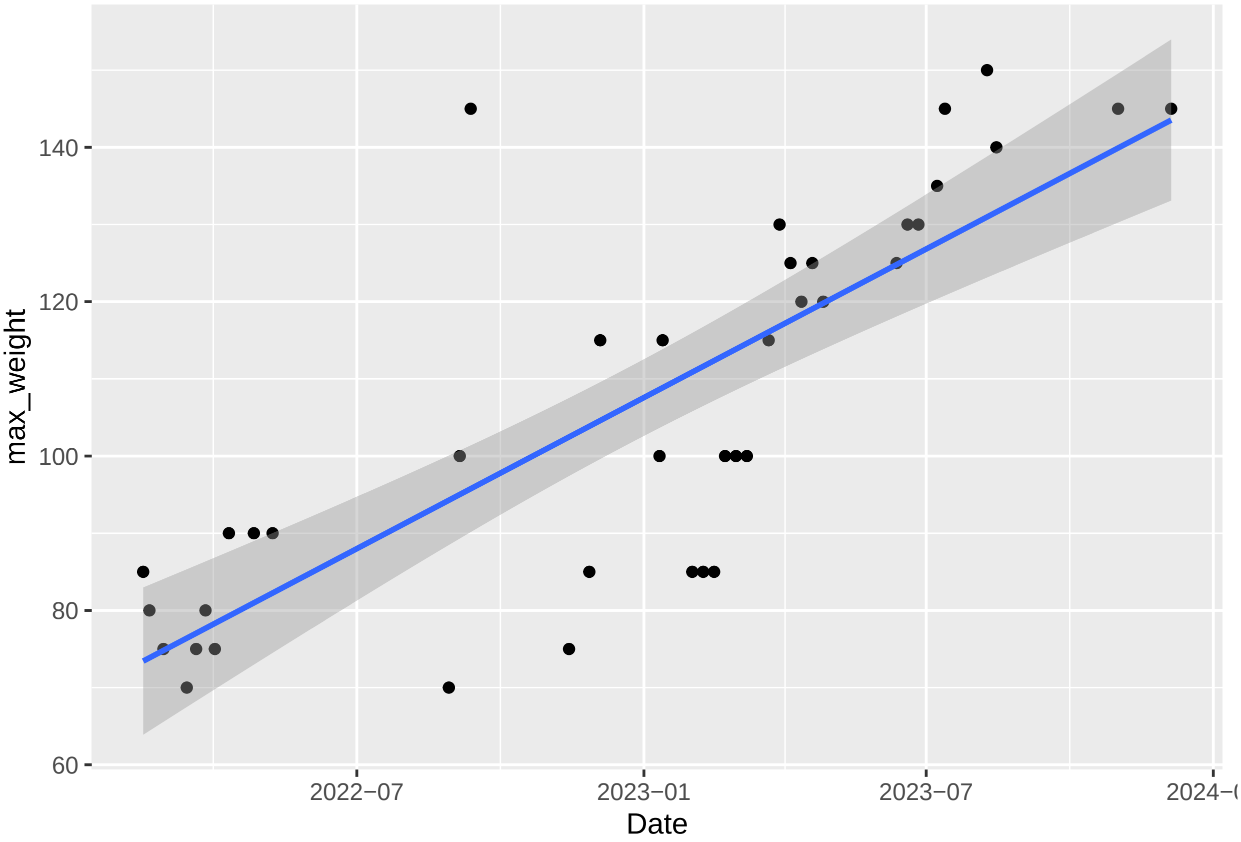
```
workout_gaps %>%
    filter(`Exercise Name` == "Bench Press (Barbell)") %>%
  ggplot(aes(x = Date, y = max_weight)) +
  geom_point() +
  geom_smooth(method = lm)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



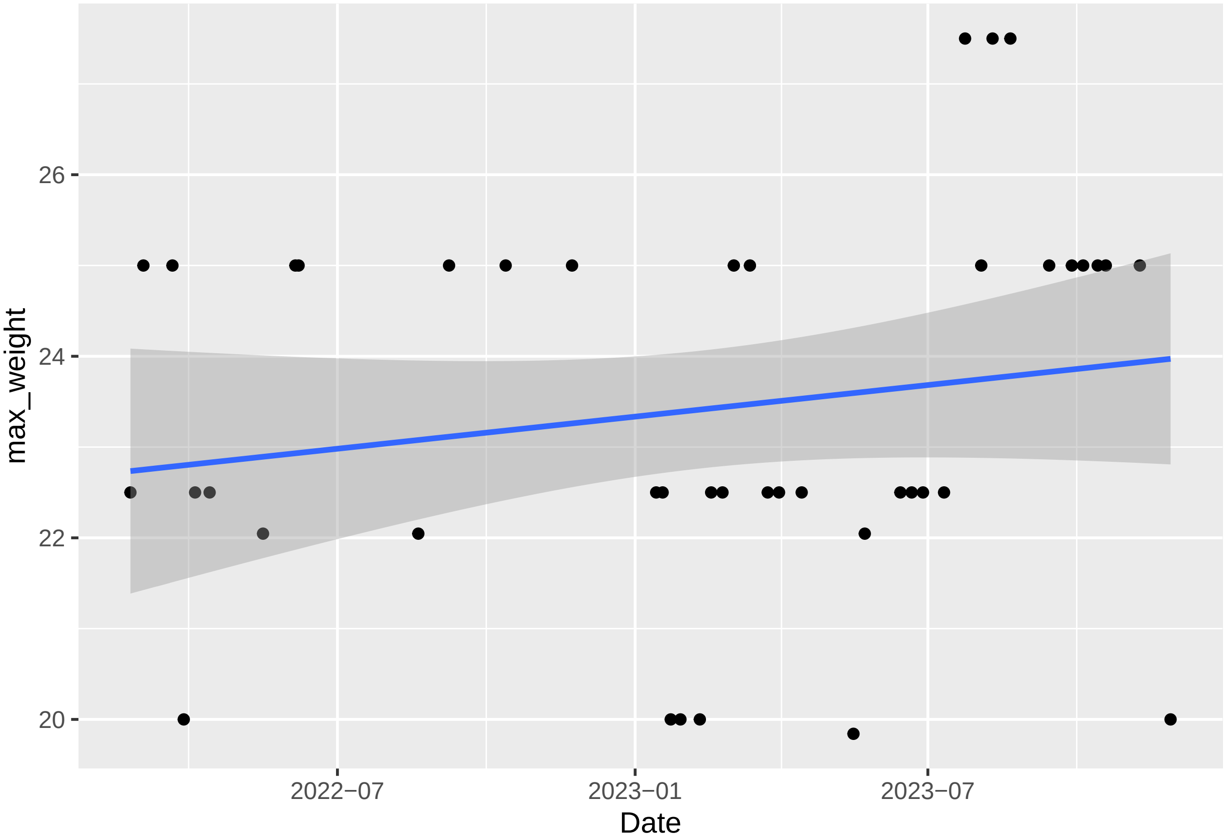
```
workout_gaps %>%
    filter(`Exercise Name` == "Chest Fly") %>%
  ggplot(aes(x = Date, y = max_weight)) +
  geom_point() +
  geom_smooth(method = lm)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



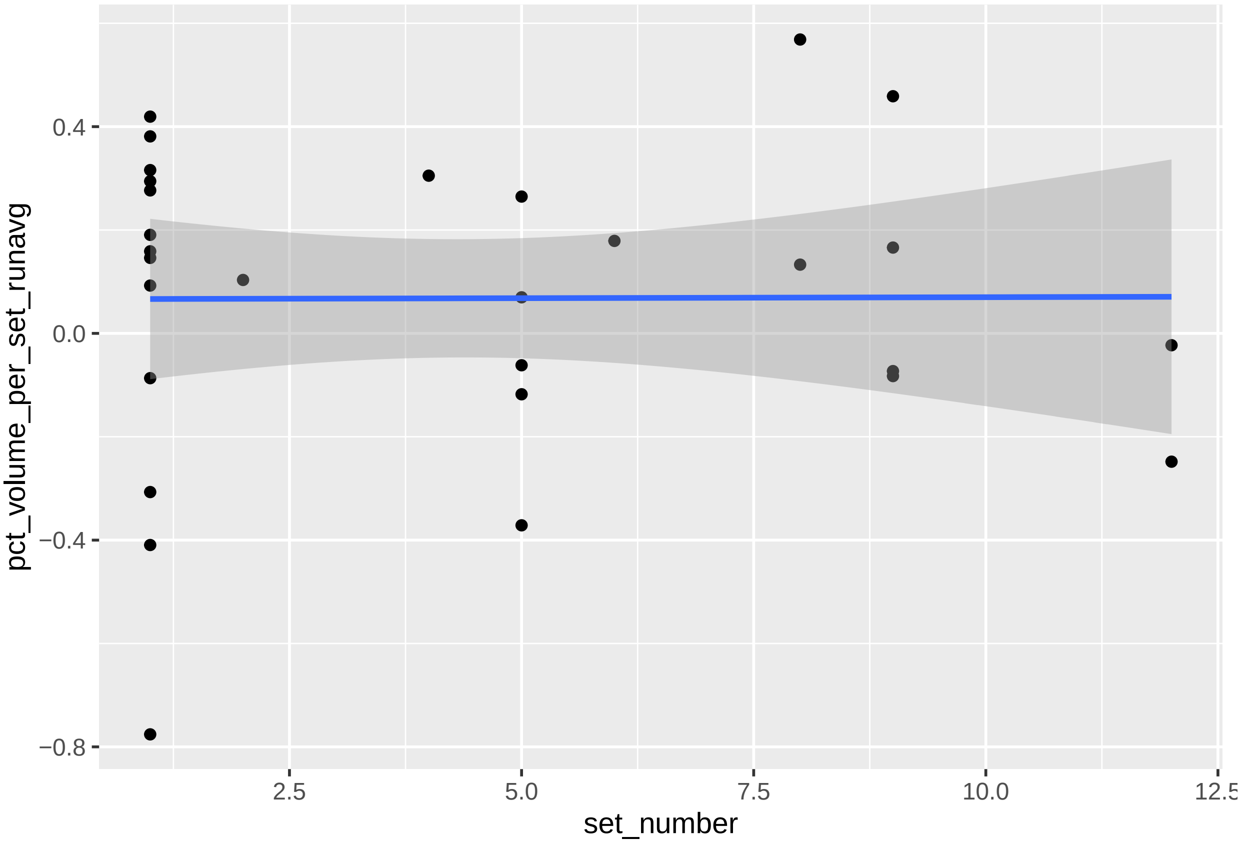
```
workout_gaps %>%
    filter(`Exercise Name` == "Hammer Curl (Dumbbell)") %>%
    filter(max_weight < 40) %>%
  ggplot(aes(x = Date, y = max_weight)) +
  geom_point() +
  geom_smooth(method = lm)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



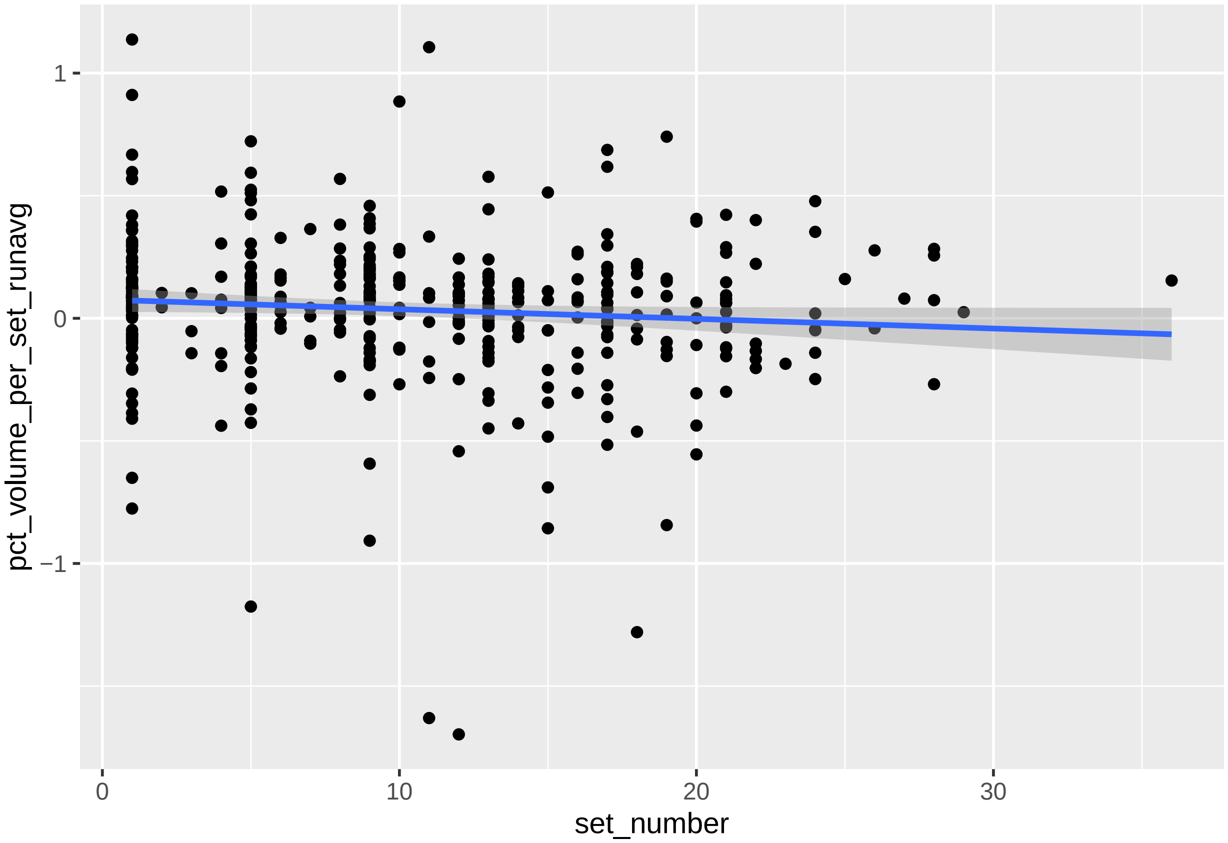
```
workout_gaps %>%
  filter(count > 20, `Exercise Name` == "Bench Press (Barbell)") %>%
  ggplot(aes(x = set_number, y = pct_volume_per_set_runavg)) +
  geom_point() +
  geom_smooth(method = lm)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
workout_gaps %>%
    filter(count > 20) %>%
  ggplot(aes(x = set_number, y = pct_volume_per_set_runavg)) +
  geom_point() +
  geom_smooth(method = lm)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

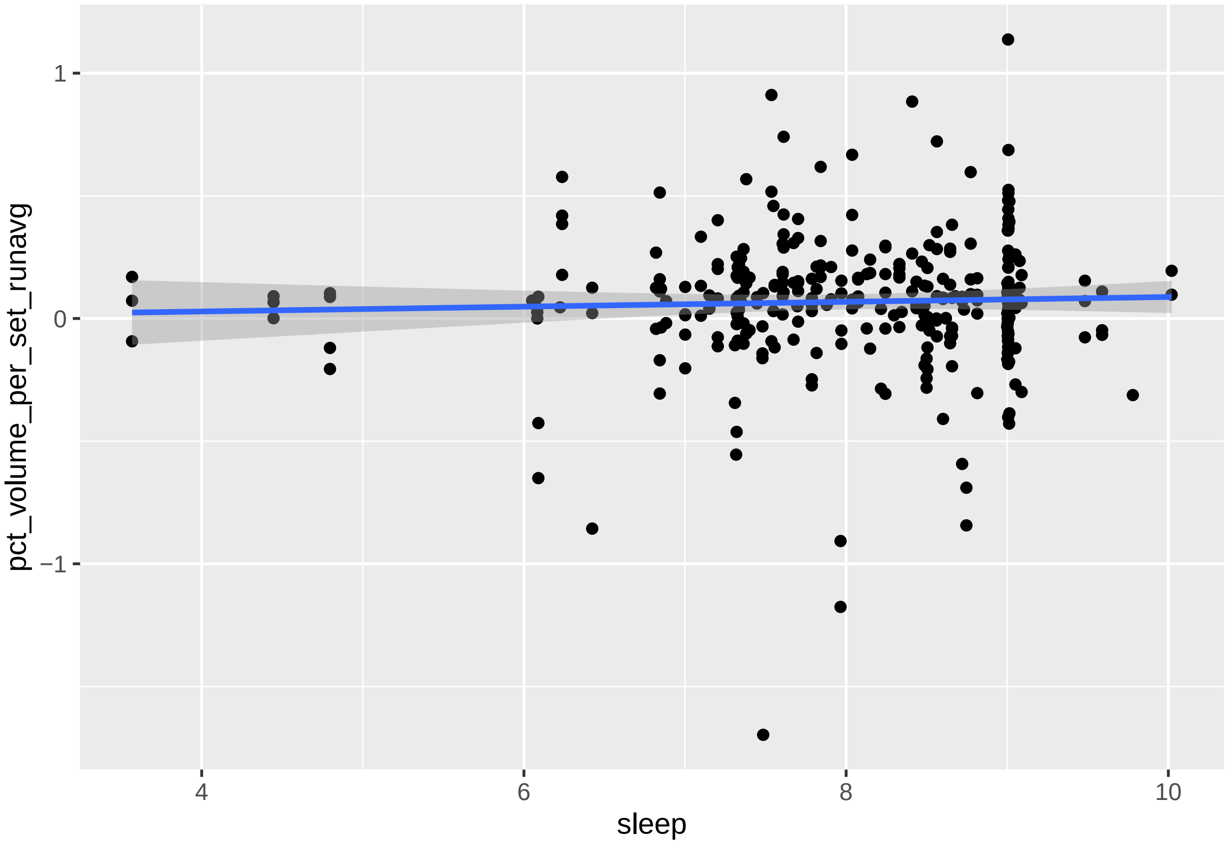


```
workout_gaps %>%
    filter(count > 20) %>%
  ggplot(aes(x = sleep, y = pct_volume_per_set_runavg)) +
  geom_point() +
  geom_smooth(method = lm)
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 98 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 98 rows containing missing values (`geom_point()`).
```

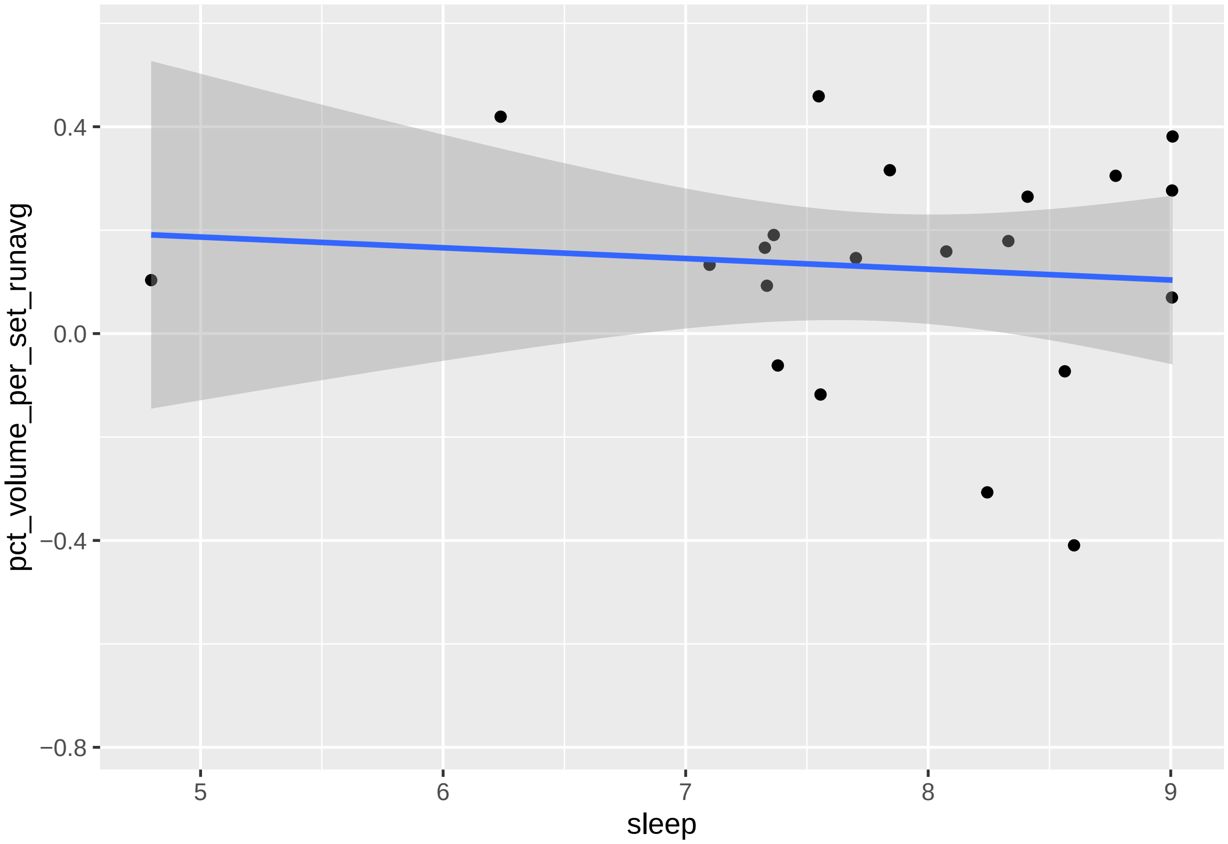


```
workout_gaps %>%
  filter(count > 20, `Exercise Name` == "Bench Press (Barbell)") %>%
  ggplot(aes(x = sleep, y = pct_volume_per_set_runavg)) +
  geom_point() +
  geom_smooth(method = lm)
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 8 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 8 rows containing missing values (`geom_point()`).
```



Idea: predict volume / max for next workout 2078