Osteoporosis and Exercise

J Newell

17 October 2018

Osteoporosis is a fairly common condition in post-menopausal women. Exercise programmes have the potential to increase the well-being of osteoporosis sufferers.

A sample of 30 middle-aged osteoporotic women was randomised either to a twice-weekly physiotherapist-led exercise regime or to be a control. Physiological assessments, including a VO2 Max test, were conducted on all women both before and after two months of the ‘intervention’ (i.e. exercise or control).

Does such an exercise regime significantly improve, on average, the VO2Max of osteoporotic women over a two-month period of exercise over any ‘natural’ improvement of controls.

Update the code and include the interpretation from the notes

library(tolerance)  
library(beanplot)  
library(table1)

##   
## Attaching package: 'table1'

## The following objects are masked from 'package:base':  
##   
## units, units<-

library(viridis)

## Loading required package: viridisLite

library(ggridges)  
library(infer)  
library(tidyverse)

## ── Attaching packages  
## ───────────────────────────────────────  
## tidyverse 1.3.2 ──

## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ readr 2.1.2 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

# Read the data  
osteo.df <- read.csv("OSTEOPOROSIS.csv")  
glimpse(osteo.df)

## Rows: 60  
## Columns: 6  
## $ Regime <chr> "Exercise", "Exercise", "Exercise", "Exercise", "Exercise"…  
## $ Subject <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,…  
## $ Time <chr> "Before", "Before", "Before", "Before", "Before", "Before"…  
## $ VO2max <dbl> 28.20, 33.74, 18.98, 22.90, 9.70, 30.62, 25.20, 28.88, 21.…  
## $ Balance <dbl> 34.290, 26.670, 21.590, 33.020, 19.050, 26.670, 12.700, 27…  
## $ Flexibility <int> 36, 27, 10, 21, 25, 23, 25, 24, 13, 26, 34, 33, 31, 31, 19…

# Design

osteo.df %>% select(Subject, Time, Regime) %>% table()

## , , Regime = Control  
##   
## Time  
## Subject After Before  
## 1 0 0  
## 2 0 0  
## 3 0 0  
## 4 0 0  
## 5 0 0  
## 6 0 0  
## 7 0 0  
## 8 0 0  
## 9 0 0  
## 10 0 0  
## 11 0 0  
## 12 0 0  
## 13 0 0  
## 14 0 0  
## 15 0 0  
## 16 0 0  
## 17 1 1  
## 18 1 1  
## 19 1 1  
## 20 1 1  
## 21 1 1  
## 22 1 1  
## 23 1 1  
## 24 1 1  
## 25 1 1  
## 26 1 1  
## 27 1 1  
## 28 1 1  
## 29 1 1  
## 30 1 1  
##   
## , , Regime = Exercise  
##   
## Time  
## Subject After Before  
## 1 1 1  
## 2 1 1  
## 3 1 1  
## 4 1 1  
## 5 1 1  
## 6 1 1  
## 7 1 1  
## 8 1 1  
## 9 1 1  
## 10 1 1  
## 11 1 1  
## 12 1 1  
## 13 1 1  
## 14 1 1  
## 15 1 1  
## 16 1 1  
## 17 0 0  
## 18 0 0  
## 19 0 0  
## 20 0 0  
## 21 0 0  
## 22 0 0  
## 23 0 0  
## 24 0 0  
## 25 0 0  
## 26 0 0  
## 27 0 0  
## 28 0 0  
## 29 0 0  
## 30 0 0

There were 16 subjects assigned to Exercise Regime and 14 assigned to Control regime (i.e.n=30 in total) and each had their VO2Max, VO2max and FLexibility measured at 2 time points (i.e. Before and two months After).

For this analyis we will start by comparing the improvement in VO2max.

To start we will need to create an ‘improvement’ score in VO2max where positive values represent an improvement (i.e. After - Before).

The data are in the wrong shape as the before vlaues are stacked on top of the after values where it would be easier to calculate the difference if the variables were side by side.

One way to do this is to select the variables we want (i.e. Subject, Time, Regime, VO2max) and unstack them by the time variable:

# Unstack the data

The pivot\_wider() function is very handy and one you will usea lot to reshape data into the shape you need.

The function takes at least four arguments:

1. the data which is passed to it via the pipe %>%
2. a specification for which columns to gather – here it is specified that we want to gather the values from Subject, Time, Regime and VO2max.
3. a names\_from argument which specifies the name of the column which will contains the column names of the original data frame that we want to spread the data across (i.e one column contaning the Before VO2max and one column the After VO2max)
4. a values\_from argument which specifies the name of the column which contains the values in the original data frame that you want to spread (VO2max in this case).

osteo.df.Balance.wide <- osteo.df %>%   
 select(Subject, Time, Regime, Balance) %>%  
 spread(key=Time, Balance)

Another way to do this is to use the pivot\_wider function as follows:

osteo.df.VO2max.wide <- osteo.df %>%   
 select(Subject, Time, Regime, VO2max) %>%  
 pivot\_wider(names\_from = Time,  
 values\_from = VO2max)

This can be further extended to unstack all the response variables, namely VO2Max, Balance and Flexibility.

osteo.df.VO2max.wide <- osteo.df %>%   
# select(Subject, Time, Regime, VO2max) %>%  
 pivot\_wider(names\_from = Time,  
 values\_from = VO2max:Flexibility)

Let’s unpack the code .. it can be ‘read’ as follows : set up a new data frame called osteo.df.VO2max.wide by taking the osteo.df then selecting the variables needed and spreading the data over the time time points for VO2max variable. To see what the result is have a look at the first few rows using the head function.

head(osteo.df.VO2max.wide)

## # A tibble: 6 × 8  
## Regime Subject VO2max\_Before VO2max\_After Balance\_…¹ Balan…² Flexi…³ Flexi…⁴  
## <chr> <int> <dbl> <dbl> <dbl> <dbl> <int> <int>  
## 1 Exercise 1 28.2 34.7 34.3 35.6 36 40  
## 2 Exercise 2 33.7 40.9 26.7 31.8 27 37  
## 3 Exercise 3 19.0 22.2 21.6 22.9 10 10  
## 4 Exercise 4 22.9 29.1 33.0 35.6 21 37  
## 5 Exercise 5 9.70 24 19.0 30.5 25 34  
## 6 Exercise 6 30.6 31.9 26.7 33.0 23 27  
## # … with abbreviated variable names ¹​Balance\_Before, ²​Balance\_After,  
## # ³​Flexibility\_Before, ⁴​Flexibility\_After

Getting familiar with pivot\_wider() takes some time plus trial and error. So don’t be discouraged if you don’t get what you want straight away. Once you’ve mastered this function, it will make it much easier to arrange your data frames as required.

After having done some transformations like this, it’s worth checking that nothing went wrong. I often compare a few values in the transformed and original data frame to make sure everything went ok.

All looks good! You are now ready to do the analysis.

Start by creating the Improvement variable for VO2\_Max using the mutate function.

osteo.df.VO2max.wide <- osteo.df.VO2max.wide %>%   
 mutate(Improvement = VO2max\_After - VO2max\_Before)

# Summary Statistics

The first step is to generate the necessary summary statistics to comapre the Improvment in VO2Max between the regimes.

Note that we first use the group\_by function to group the data by the two regimes (Control and Exercise) and then use summarize() to calculate the mean, median and standard deviation of Improvement for each regime as follows:

osteo.df.VO2max.wide %>% group\_by(Regime) %>%   
 summarize(Mean=mean(Improvement),  
 Median=median(Improvement),  
 SD= sd(Improvement),  
 sample.size = n())

## # A tibble: 2 × 5  
## Regime Mean Median SD sample.size  
## <chr> <dbl> <dbl> <dbl> <int>  
## 1 Control 0.197 0.630 2.76 14  
## 2 Exercise 6.14 6.00 4.02 16

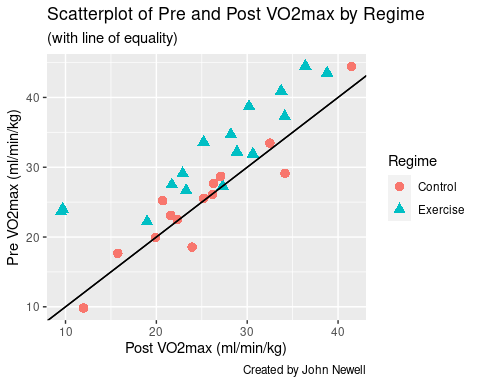
## Nicer table of summary statistics

##   Control Exercise Overall  
## 1 (N=14) (N=16) (N=30)  
## 2 VO2max\_Before   
## 3   Mean (SD) 24.9 (7.54) 26.2 (8.46) 25.6 (7.93)  
## 4   Median [Min, Max] 24.6 [12.0, 41.5] 27.8 [9.51, 38.8] 25.7 [9.51, 41.5]  
## 5 VO2max\_After   
## 6   Mean (SD) 25.1 (8.09) 32.4 (7.11) 29.0 (8.30)  
## 7   Median [Min, Max] 25.4 [9.82, 44.5] 32.1 [22.2, 44.5] 27.6 [9.82, 44.5]  
## 8 Improvement   
## 9   Mean (SD) 0.197 (2.76) 6.14 (4.02) 3.37 (4.57)  
## 10   Median [Min, Max] 0.630 [-5.38, 4.51] 6.00 [0, 14.3] 3.06 [-5.38, 14.3]

# Graphical Summaries

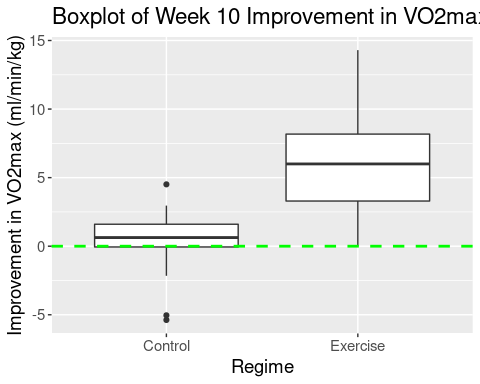
Scatterplot

osteo.df.VO2max.wide %>%  
 ggplot(aes(x = VO2max\_Before,   
 y = VO2max\_After,   
 color=Regime, shape = Regime)) +  
 geom\_point(size = 3) +   
 geom\_abline(slope=1, intercept=0) +  
# ylab("Post VO2max (ml/min/kg)") +  
# xlab("Pre VO2max (ml/min/kg)")   
 geom\_abline(slope=1, intercept=0) +  
 labs(title = "Scatterplot of Pre and Post VO2max by Regime ",  
 subtitle = "(with line of equality)",  
 caption = "Created by John Newell",   
 x = "Post VO2max (ml/min/kg)", y = "Pre VO2max (ml/min/kg)")

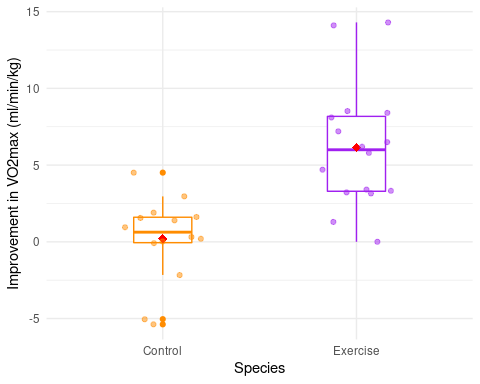


Boxplot

ggplot(osteo.df.VO2max.wide, aes(x = Regime, y = Improvement)) +  
 geom\_boxplot() +   
 ggtitle("Boxplot of Week 10 Improvement in VO2max by Regime") +  
 ylab("Improvement in VO2max (ml/min/kg)") +  
 geom\_hline(yintercept=0, linetype="dashed",color = "green", size=1) +  
 theme(text = element\_text(size=14))

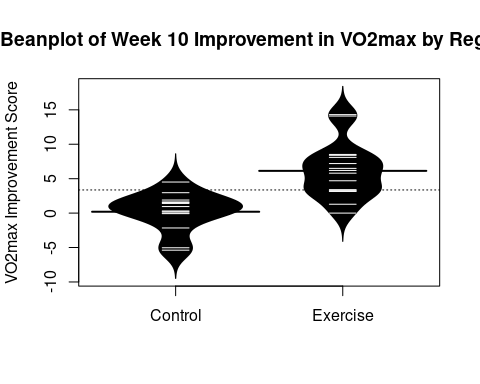


## Fancier Boxplot



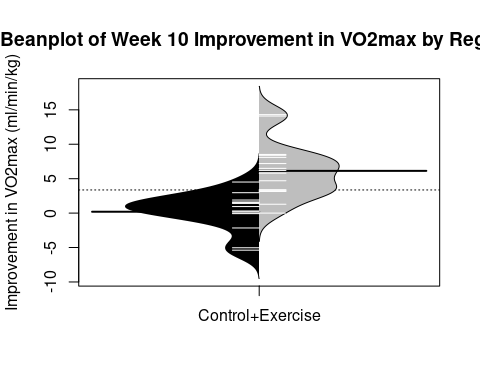
## Bean plot

beanplot(Improvement ~ Regime, data = osteo.df.VO2max.wide, main = "Beanplot of Week 10 Improvement in VO2max by Regime", ylab="VO2max Improvement Score")



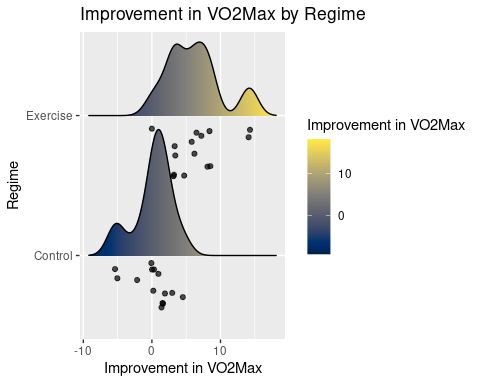
## Bean plot (side by side)

beanplot(Improvement ~ Regime, data = osteo.df.VO2max.wide,   
 side = "both", main = "Beanplot of Week 10 Improvement in VO2max by Regime", ylab="Improvement in VO2max (ml/min/kg)",  
 col = list("black", c("grey", "white")))



## Rain Cloud Plot

ggplot(osteo.df.VO2max.wide, aes(x = Regime, y = Improvement)) +



95% Confidence Interval for difference in the population mean

t.test(Improvement ~ Regime, data= osteo.df.VO2max.wide)

##   
## Welch Two Sample t-test  
##   
## data: Improvement by Regime  
## t = -4.7651, df = 26.621, p-value = 5.916e-05  
## alternative hypothesis: true difference in means between group Control and group Exercise is not equal to 0  
## 95 percent confidence interval:  
## -8.499946 -3.380766  
## sample estimates:  
## mean in group Control mean in group Exercise   
## 0.1971433 6.1374996

# better version of t test

osteo.df.VO2max.wide %>%  
t\_test(Improvement ~ Regime, order = c("Exercise", "Control"),   
 alternative = "two\_sided",  
 mu=0,  
 conf\_level = 0.95,  
 paired=FALSE,  
 var.equal=FALSE)

## # A tibble: 1 × 7  
## statistic t\_df p\_value alternative estimate lower\_ci upper\_ci  
## <dbl> <dbl> <dbl> <chr> <dbl> <dbl> <dbl>  
## 1 4.77 26.6 0.0000592 two.sided 5.94 3.38 8.50

## Tolerance Interval

## Controls

Controls.Improvement <- osteo.df.VO2max.wide %>%   
 filter(Regime %in% 'Control') %>% select(Improvement) %>% as.data.frame()  
  
normtol.int(Controls.Improvement$Improvement, alpha = 0.05, P = 0.95, side = 2)

## alpha P x.bar 2-sided.lower 2-sided.upper  
## 1 0.05 0.95 0.1971433 -8.174975 8.569262

## Exercisers

Exercisers.Improvement <- osteo.df.VO2max.wide %>%   
 filter(Regime %in% 'Exercise') %>% select(Improvement) %>% as.data.frame()  
  
normtol.int(Exercisers.Improvement$Improvement, alpha = 0.05, P = 0.95, side = 2)

## alpha P x.bar 2-sided.lower 2-sided.upper  
## 1 0.05 0.95 6.1375 -5.587382 17.86238

# 95% Bootstrap CI for difference in medians

osteo.boot <- osteo.df.VO2max.wide %>%  
 specify(response = Improvement, explanatory = Regime) %>%  
 generate(reps = 1000, type = "bootstrap") %>%  
 calculate(stat = "diff in medians", order = c("Exercise", "Control"))  
  
percentile\_ci <- get\_ci(osteo.boot)  
  
percentile\_ci

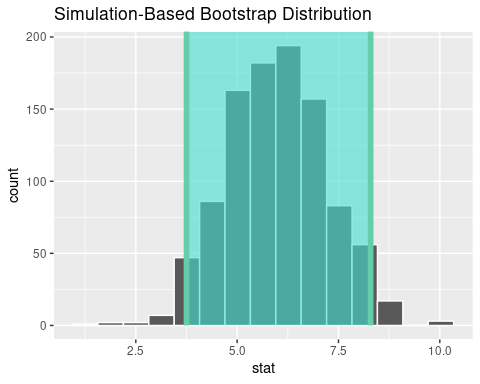
## # A tibble: 1 × 2  
## lower\_ci upper\_ci  
## <dbl> <dbl>  
## 1 2.00 7.90

# 95% Bootstrap CI for difference in means

osteo.boot <- osteo.df.VO2max.wide %>%  
 specify(response = Improvement, explanatory = Regime) %>%  
 generate(reps = 1000, type = "bootstrap") %>%  
 calculate(stat = "diff in means", order = c("Exercise", "Control"))  
  
percentile\_ci <- get\_ci(osteo.boot)  
  
percentile\_ci

## # A tibble: 1 × 2  
## lower\_ci upper\_ci  
## <dbl> <dbl>  
## 1 3.75 8.29

osteo.boot %>%  
 visualize() +  
 shade\_confidence\_interval(percentile\_ci,  
 color = "mediumaquamarine",   
 fill = "turquoise")



## Permutations test

samp.diff <- osteo.df.VO2max.wide %>%  
specify(response = Improvement, explanatory = Regime) %>%  
 calculate(stat = "diff in means", order = c("Exercise", "Control"))  
  
# Simulate the diffs under the null hypothesis  
  
Improvement.null.dist <- osteo.df.VO2max.wide %>%  
specify(response = Improvement, explanatory = Regime) %>%  
 hypothesize(null = "independence") %>%  
 generate(reps = 1000, Group = "permute") %>%  
 calculate(stat = "diff in means", order = c("Exercise", "Control"))

## Setting `type = "permute"` in `generate()`.

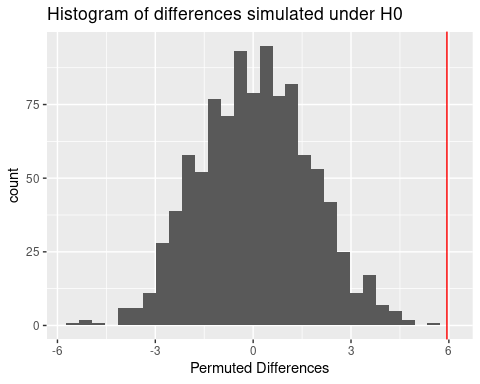
Improvement.null.dist %>%   
 get\_pvalue(obs\_stat = samp.diff, direction = "two\_sided")

## Warning: Please be cautious in reporting a p-value of 0. This result is an  
## approximation based on the number of `reps` chosen in the `generate()` step. See  
## `?get\_p\_value()` for more information.

## # A tibble: 1 × 1  
## p\_value  
## <dbl>  
## 1 0

Improvement.null.dist %>%   
 ggplot(aes(stat)) +   
 geom\_histogram() +  
 geom\_vline(data=samp.diff, aes(xintercept = stat), colour="red") +  
 ggtitle("Histogram of differences simulated under H0") +  
 xlab("Permuted Differences")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



# How often is the sample difference in means bigger than those generated by the 1000 permutations ?

sum(samp.diff$stat > Improvement.null.dist$stat )

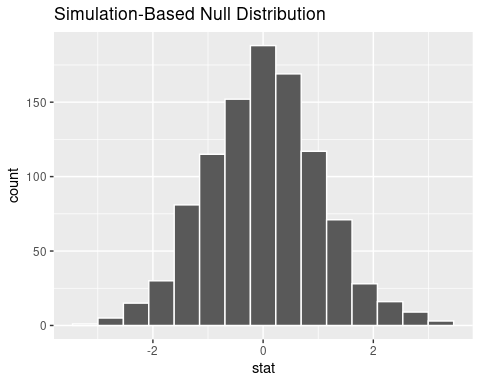
## [1] 1000

1 - (sum(samp.diff$stat > Improvement.null.dist$stat )/1000)

## [1] 0

osteo.df.VO2max.wide %>%  
specify(response = Improvement, explanatory = Regime) %>%  
 hypothesize(null = "independence") %>%  
 generate(reps = 1000, Group = "permute") %>%  
 calculate(stat = "t", order = c("Exercise", "Control")) %>%  
 visualize(method = "simulation")

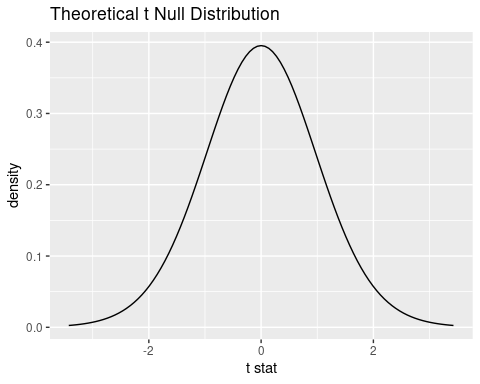
## Setting `type = "permute"` in `generate()`.



# Theoretical t distribution  
# using t statistic  
  
osteo.df.VO2max.wide %>%  
specify(response = Improvement, explanatory = Regime) %>%  
 hypothesize(null = "independence") %>%  
# generate(reps = 1000, Group = "permute") %>%  
 calculate(stat = "t", order = c("Exercise", "Control")) %>%  
 visualize(method = "theoretical")

## Rather than setting `method = "theoretical"` with a simulation-based null distribution, the preferred method for visualizing theory-based distributions with infer is now to pass the output of `assume()` as the first argument to `visualize()`.

## Warning: Check to make sure the conditions have been met for the theoretical  
## method. {infer} currently does not check these for you.



# Overlay theoretical distribution on top of randomised t-statistics  
osteo.df.VO2max.wide %>%  
specify(response = Improvement, explanatory = Regime) %>%  
 hypothesize(null = "independence") %>%  
 generate(reps = 1000, Group = "permute") %>%  
 calculate(stat = "t", order = c("Exercise", "Control")) %>%  
 visualize(method = "both")

## Setting `type = "permute"` in `generate()`.

## Warning: Check to make sure the conditions have been met for the theoretical  
## method. {infer} currently does not check these for you.

