

Case Study: effects of exercise on health

Sonia Mazzi

19 June, 2019

CASE STUDY. A study on the effects of exercise on health with data in four flat files

THE TRUE KEY TO WEIGHT LOSS

- ⇒ EXERCISE 20-30 MINUTES DAILY (THAT'S IT)
- ⇒ GIVE YOURSELF 8 HOURS OF SLEEP EVERY NIGHT
- ⇒ DRINK $\frac{1}{2}$ OUNCE OF WATER PER POUND OF YOUR BODYWEIGHT (DAILY)
- ⇒ SUNSHINE VITAMIN D (10 MINS EARLY MORNING, LATE AFTERNOON) ON HANDS OR FACE
- ⇒ DUMP THE JUNK IN YOUR DIET (NOTHING IN BOXES OR BAGS)
- ⇒ LIMIT GRAINS, REFINED SUGARS, AND COMMERCIAL DAIRY
- ⇒ MORE FRUITS AND VEGETABLES ON EVERY PLATE, EVERY DAY
- ⇒ QUIT SMOKING (YOU CAN DO IT)

Alternative-Doctor.com
Dr. Keith Scott-Mumby

- ▶ Assess the effectiveness of a new exercise programme which is believed to induce weight loss and improve self-rated health.
- ▶ 5000 people randomized into two groups: 2,500 people to the treatment and another 2,500 to the control group.
- ▶ Health outcomes, weight and self-rated health, were measured before and after the intervention.

The data was provided in 4 files:

- ▶ EXER_age_sex_race.csv : Subject demographics at baseline: age, sex, and race (we have seen this data when learning how to import comma separated values file)
- ▶ EXER_SRH.csv : Self-Rated Health (SRH) on an ordinal scale.
- ▶ EXER_weight_trt : Weight, in pounds, for Treatment Group.
- ▶ EXER_weight_con : Weight, in pounds, for Control Group.

Import these files using the `read_csv()` function and explore the variables they contain.

```
data1 <- read_csv("DataFiles/EXER_age_sex_race.csv")
```

```
glimpse(data1)
```

```
## Observations: 5,000  
## Variables: 2  
## $ subject_ID <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,...  
## $ SexAge_Race <chr> "MALE41.2_White", "FEMALE42.9_White", "FEMALE38.5_...
```

- ▶ There are 5000 subjects in the study.
- ▶ All the information about each subject is in one column.
Separate the information.
- ▶ There are missing values for race.

```
data2 <- read_csv("DataFiles/EXER_SRH.csv")
```

```
glimpse(data2)
```

```
## Observations: 10,035
## Variables: 4
## $ id   <int> 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10,...
## $ trt   <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ TIME  <chr> "POST", "PRE", "PRE", "POST", "PRE", "POST", "PRE", "POST...
## $ SRH   <chr> "Poor", "Good", "Poor", "Very Poor", "Satisfactory", "Goo...
```

```
unique(as.factor(data2$trt))
```

```
## [1] 1 0
```

```
## Levels: 0 1
```

- ▶ This file contains the patient id (note the different column name from the previous file),
- ▶ and each participant's self rated health before and after the study.
- ▶ trt takes on two values: 0 (individual didn't exercise, control), 1 (individual exercised, treatment).
- ▶ data2 has 10035 rows but there should only be 10000 rows

```
data3 <- read_csv("DataFiles/EXER_weight_trt.csv")
```

```
glimpse(data3)
```

```
## Observations: 5,017
## Variables: 3
## $ Id      <int> 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9,...
## $ PRE_WEIGHT <dbl> 135.2510, NA, 154.8713, NA, 128.1951, NA, 183.4600...
## $ POST_WEIGHT <dbl> NA, 125.6678, NA, 153.9882, NA, 115.5969, NA, 177....
```

- ▶ Information about the weight of patients who received the exercise plan.
- ▶ Patient id column has different name than in the previous two data sets,
- ▶ Pre- and post-intervention weight.
- ▶ Many NAs! if pre-intervention weight is recorded, post-intervention weight is an NA, and vice-versa.
- ▶ Expect 5000 observations, but there are 5017.

```
data4 <- read_csv("DataFiles/EXER_weight_con.csv")
```

```
glimpse(data4)
```

```
## Observations: 5,012
## Variables: 3
## $ obs_ID      <int> 2501, 2501, 2502, 2502, 2503, 2503, 2504, 2504, 25...
## $ PRE_WEIGHT  <dbl> 159.7587, NA, 176.1611, NA, 181.3907, NA, 175.6615...
## $ POST_WEIGHT <dbl> NA, 158.6920, NA, 174.8270, NA, 179.9042, NA, 175....
```

- ▶ Information about the weight of patients who are in the control group.
- ▶ Patient id column has different name than in the previous two data sets,
- ▶ Pre- and post-intervention weight.
- ▶ Many NAs! if pre-intervention weight is recorded, post-intervention weight is an NA, and vice-versa.
- ▶ Expect 5000 observations, but there are 5012.

In this study:

- ▶ an observational unit is a patient,
- ▶ fixed variables are age, sex and race,
- ▶ measured variables are self-rated health and weight at two time points, before and after the intervention.

This is messy data:

- ▶ a single observational unit is stored in multiple tables,
- ▶ multiple variables are stored in one column,
- ▶ there is possibly duplicated information,
- ▶ and more messy features to be discovered!

- ▶ The strategy is to deal with the data sets one by one.
- ▶ Manipulate each of them so that they can be joined.
- ▶ Have to join by subject id so let us name that column `id` in all four data frames

First data set (data1)

```
glimpse(data1)
```

```
## Observations: 5,000  
## Variables: 2  
## $ subject_ID <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,...  
## $ SexAge_Race <chr> "MALE41.2_White", "FEMALE42.9_White", "FEMALE38.5_...
```

```
# data1_1 will be the modified version of data1.  
data1_1 <- data1
```

```
#add a _ after MALE to separate sex and age  
data1_1 <- data1_1 %>%  
  mutate(SexAge_Race =  
    str_replace(SexAge_Race, "MALE", "MALE_"))
```

```
#separate values between _  
data1_1 <- data1_1 %>%  
  separate(SexAge_Race, c("Sex", "Age", "Race"), sep = "_")
```

```
glimpse(data1_1)
```

```
## Observations: 5,000
## Variables: 4
## $ subject_ID <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, ...
## $ Sex        <chr> "MALE", "FEMALE", "FEMALE", "FEMALE", "FEMALE", "FE...
## $ Age        <chr> "41.2", "42.9", "38.5", "35.6", "48.5", "36.9", "28...
## $ Race       <chr> "White", "White", "White", "Hispanic", "White", "NA..."
```

- Note Age is of type character. Should be numeric.

```
#make Age a numeric variable
```

```
data1_1 <- mutate(data1_1, Age = as.numeric(Age))
```

```
#rename the first column
```

```
data1_1 <- rename(data1_1, id = subject_ID)
```

```
glimpse(data1_1)
```

```
## Observations: 5,000
```

```
## Variables: 4
```

```
## $ id    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17...
```

```
## $ Sex   <chr> "MALE", "FEMALE", "FEMALE", "FEMALE", "FEMALE", "FEMALE", ...
```

```
## $ Age   <dbl> 41.2, 42.9, 38.5, 35.6, 48.5, 36.9, 28.9, 24.7, 47.1, 37...
```

```
## $ Race  <chr> "White", "White", "White", "Hispanic", "White", "NA", "Wh...
```

- ▶ Note that the Race column of data1_1 has the value “NA”, which R reads as a character string and not as a missing value.
- ▶ Need to transform the entries “NA” into NA.

#turn "NA" into missing values

```
data1_1 <- mutate(data1_1, Race = na_if(Race, "NA"))
```

```
glimpse(data1_1)
```

```
## Observations: 5,000
## Variables: 4
## $ id   <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17...
## $ Sex  <chr> "MALE", "FEMALE", "FEMALE", "FEMALE", "FEMALE", "FEMALE",...
## $ Age  <dbl> 41.2, 42.9, 38.5, 35.6, 48.5, 36.9, 28.9, 24.7, 47.1, 37...
## $ Race <chr> "White", "White", "White", "Hispanic", "White", NA, "Whit...
```


Summarise the wrangling

```
data1_tidy <- read_csv("DataFiles/EXER_age_sex_race.csv") %>%  
  rename(id = subject_ID) %>%  
  mutate(SexAge_Race =  
    str_replace(SexAge_Race, "MALE", "MALE_")) %>%  
  separate(SexAge_Race, c("Sex", "Age", "Race"), sep = "_") %>%  
  mutate(Age = as.numeric(Age)) %>%  
  mutate(Race = na_if(Race, "NA"))
```

```
summary(as.factor(data1_1$Race))
```

```
##      Asian      Black Hispanic      White      NA's  
##      450       424       846       2876       404
```

```
summary(as.factor(data1_1$Sex))
```

```
## FEMALE      MALE  
##   2556      2444
```

```
summary(data1_1$Age)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
##   18.10   33.40   38.80   38.89   44.10   69.30
```

Second data set (data2)

```
glimpse(data2)
```

```
## Observations: 10,035
## Variables: 4
## $ id   <int> 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10,...
## $ trt   <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ TIME  <chr> "POST", "PRE", "PRE", "POST", "PRE", "POST", "PRE", "POST...
## $ SRH   <chr> "Poor", "Good", "Poor", "Very Poor", "Satisfactory", "Goo...
```

- There are 10,035 rows. Remove exact duplicate rows.

```
data2_1 <- data2
```

```
#remove duplicate rows
data2_1 <- distinct(data2)
```

```
nrow(data2_1)
```

```
## [1] 10016
```

- After removing duplicate rows now there are 10,016 rows.

- Summaries of columns. Is there anything unusual?

```
summary(as.factor(data2_1$trt))
```

```
##      0      1  
## 5008 5008
```

```
summary(as.factor(data2_1$TIME))
```

```
## POST  PRE  
## 5003 5013
```

```
summary(as.factor(data2_1$SRH))
```

```
##      Excellent      Good      Poor Satisfactory      Very      Poor  
##          2947      1570      1711          1411          16  
##      Very Poor  
##          2361
```

- We spotted something unusual! The SRH level Very Poor appears also with a double space between Very and Poor.
- This happens for 16 rows.

```
data2_1 <- data2_1 %>%
  mutate(SRH = str_replace(SRH, "Very Poor", "Very Poor"))
```

```
summary(as.factor(data2_1$SRH))
```

```
##      Excellent      Good      Poor Satisfactory      Very Poor
##           2947           1570           1711           1411           2377
```

```
#remove duplicate rows
data2_1 <- distinct(data2_1)
```

```
glimpse(data2_1)
```

```
## Observations: 10,000
## Variables: 4
## $ id   <int> 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10,...
## $ trt  <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ TIME <chr> "POST", "PRE", "PRE", "POST", "PRE", "POST", "PRE", "POST...
## $ SRH  <chr> "Poor", "Good", "Poor", "Very Poor", "Satisfactory", "Goo...
```

- There are now two rows per patient.

- ▶ Merge all four data sets into one.
- ▶ The first data set has one row per observation. The second data set has two rows per observation.
- ▶ Spread the TIME column, to create two new columns with PRE-SRH and POST-SRH.

```
data2_1 <- spread(data2_1, TIME, SRH)
```

```
glimpse(data2_1)
```

```
## Observations: 5,000
## Variables: 4
## $ id    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17...
## $ trt    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ POST   <chr> "Poor", "Very Poor", "Good", "Good", "Poor", "Excellent",...
## $ PRE    <chr> "Good", "Poor", "Satisfactory", "Poor", "Poor", "Good", "...
```

- Change the names of the 3rd and 4th columns to POST_SRH and PRE_SRH and swap their order.

```
data2_1 <- rename(data2_1, POST_SRH = POST, PRE_SRH = PRE)
```

```
data2_1 <- select(data2_1, c(1,2,4,3))
```

```
glimpse(data2_1)
```

```
## Observations: 5,000
## Variables: 4
## $ id      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16...
## $ trt      <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ PRE_SRH  <chr> "Good", "Poor", "Satisfactory", "Poor", "Poor", "Good...
## $ POST_SRH <chr> "Poor", "Very Poor", "Good", "Good", "Poor", "Excele...
```


Summary of the wrangling

```
data2_tidy <- read_csv("DataFiles/EXER_SRH.csv") %>%  
  mutate(SRH =  
    str_replace(SRH, "Very Poor", "Very Poor")) %>%  
  distinct() %>%  
  spread(TIME, SRH) %>%  
  rename(POST_SRH = POST, PRE_SRH = PRE) %>%  
  select(c(1,2,4,3))
```

Third data set (data3)

- ▶ Pre- and post-weight of treatment patients.
- ▶ Change the name of the first column to `id`.

```
glimpse(data3)
```

```
## Observations: 5,017
## Variables: 3
## $ Id          <int> 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9,...
## $ PRE_WEIGHT  <dbl> 135.2510, NA, 154.8713, NA, 128.1951, NA, 183.4600...
## $ POST_WEIGHT <dbl> NA, 125.6678, NA, 153.9882, NA, 115.5969, NA, 177....
```

```
data3_1 <- data3
```

```
data3_1 <- rename(data3_1, id = Id)
```

- ▶ There should be 5,000 rows in `data3` but there are 5,017 rows.

```
#remove duplicate rows  
data3_1 <- distinct(data3_1)
```

```
nrow(data3_1)
```

```
## [1] 5017
```

- ▶ There are no exact duplicate rows in data3.
- ▶ Let us see if there are more than two records per patient.
- ▶ `count()`, in `dplyr`, used to count how many instances for each patient id there are.

Example

```
#count the instances of each patient id in the first 8 rows  
aux1 <- count(data3_1[1:8,],id)  
aux1
```

```
## # A tibble: 4 x 2  
##       id       n  
##   <int> <int>  
## 1     1     2  
## 2     2     2  
## 3     3     2  
## 4     4     2
```

- The result has two columns. One is id and the other one is the corresponding count, n.

```
#count the instances of each patient id  
aux1 <- count(data3_1,id)
```

```
aux2 <- which(aux1$n > 2)
```

```
aux3 <- aux1$id[aux2]  
aux3
```

```
## [1] 2394 2395 2396 2397 2398 2399 2400 2401 2402
```

- ▶ aux3 gives the patient id's with more than two entries. Let us explore the data for those patients.

```
ff <- filter(data3_1, id %in% aux3)
```

```
as.data.frame(ff)
```

##	id	PRE_WEIGHT	POST_WEIGHT
## 1	2394	137.0663	NA
## 2	2394	NA	128.2492
## 3	2394	NA	128.2000
## 4	2395	164.3000	NA
## 5	2395	164.3255	NA
## 6	2395	NA	164.3350
## 7	2395	NA	164.3500
## 8	2396	160.9983	NA
## 9	2396	160.9985	NA
## 10	2396	NA	165.7400
## 11	2396	NA	165.7270
## 12	2397	147.7983	NA
## 13	2397	147.7990	NA
## 14	2397	NA	137.8439
## 15	2397	NA	137.8500
## 16	2398	133.1364	NA
## 17	2398	133.1400	NA
## 18	2398	NA	118.9500
## 19	2398	NA	118.9398
## 20	2399	188.8684	NA
## 21	2399	188.9000	NA
## 22	2399	NA	179.1000
## 23	2399	NA	179.0564
## 24	2400	166.1632	NA
## 25	2400	166.2000	NA
## 26	2400	NA	150.2311
## 27	2400	NA	150.2000
## 28	2401	151.6768	NA
## 29	2401	151.7000	NA
## 30	2401	NA	133.3000
## 31	2401	NA	133.2508
## 32	2402	148.7000	NA

- ▶ Truncate the values of weight (pre- and post-). This will hardly affect the results of the study
- ▶ This will create exact row duplicates and then eliminate duplicates.
- ▶ The function `mutate_all()` in the `dplyr` package is useful to apply a function to each column.
- ▶ In this case the function we want to apply is `round()`.

```
#this call truncates all the columns  
data3_1 <- mutate_all(data3_1, trunc)
```

```
glimpse(data3_1)
```

```
## Observations: 5,017  
## Variables: 3  
## $ id      <dbl> 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, ...  
## $ PRE_WEIGHT <dbl> 135, NA, 154, NA, 128, NA, 183, NA, 166, NA, 120, ...  
## $ POST_WEIGHT <dbl> NA, 125, NA, 153, NA, 115, NA, 177, NA, 163, NA, 1...
```

► Now, remove duplicate rows

```
data3_1 <- distinct(data3_1)
```

```
glimpse(data3_1)
```

```
## Observations: 5,000  
## Variables: 3  
## $ id      <dbl> 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, ...  
## $ PRE_WEIGHT <dbl> 135, NA, 154, NA, 128, NA, 183, NA, 166, NA, 120, ...  
## $ POST_WEIGHT <dbl> NA, 125, NA, 153, NA, 115, NA, 177, NA, 163, NA, 1...
```

► Now we have 5000 rows.

- ▶ Need one row per patient with `id`, `pre-` and `post-weight`.
- ▶ Strategy: select `id` and `pre-weight`, eliminate the rows where `pre-weight` is `NA`.
- ▶ Similarly with `id` and `post-weight`.
- ▶ Next merge both data frames into one by patient `id`.

```
#select all columns except POST_WEIGHT
```

```
temp1 <- select(data3_1, -POST_WEIGHT)
```

```
#filter out the NAs in PRE_WEIGHT
```

```
temp1 <- filter(temp1, is.na(PRE_WEIGHT) == "FALSE")
```

```
glimpse(temp1)
```

```
## Observations: 2,500
```

```
## Variables: 2
```

```
## $ id      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, ...
```

```
## $ PRE_WEIGHT <dbl> 135, 154, 128, 183, 166, 120, 158, 160, 127, 187, 1...
```

Alternatively,

```
temp1 <- data3_1 %>%  
  select(-POST_WEIGHT) %>%  
  filter(is.na(PRE_WEIGHT) == "FALSE")
```

```
#select all columns except PRE_WEIGHT
```

```
temp2 <- select(data3_1, -PRE_WEIGHT)
```

```
#filter out the NAs in POST_WEIGHT
```

```
temp2 <- filter(temp2, is.na(POST_WEIGHT) == "FALSE")
```

```
glimpse(temp2)
```

```
## Observations: 2,500
```

```
## Variables: 2
```

```
## $ id      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,...
```

```
## $ POST_WEIGHT <dbl> 125, 153, 115, 177, 163, 105, 151, 143, 119, 159, ...
```

Alternatively,

```
temp2 <- data3_1 %>%  
  select(-PRE_WEIGHT) %>%  
  filter(is.na(POST_WEIGHT) == "FALSE")
```

```
#join temp1 and temp2  
data3_1 <- inner_join(temp1,temp2)
```

```
## Joining, by = "id"
```

```
glimpse(data3_1)
```

```
## Observations: 2,500  
## Variables: 3  
## $ id          <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,...  
## $ PRE_WEIGHT  <dbl> 135, 154, 128, 183, 166, 120, 158, 160, 127, 187, ...  
## $ POST_WEIGHT <dbl> 125, 153, 115, 177, 163, 105, 151, 143, 119, 159, ...
```

Summary, using %>%

```
data3_aux <- read_csv("DataFiles/EXER_weight_trt.csv") %>%  
  rename(id = Id) %>%  
  mutate_all(trunc) %>%  
  distinct()
```

```
temp1 <- data3_aux %>%  
  select(-POST_WEIGHT) %>%  
  filter(is.na(PRE_WEIGHT) == "FALSE")
```

```
temp2 <- data3_aux %>%  
  select(-PRE_WEIGHT) %>%  
  filter(is.na(POST_WEIGHT) == "FALSE")
```

```
data3_tidy <- inner_join(temp1,temp2)
```

```
## Joining, by = "id"
```

Fourth data set (data4)

```
glimpse(data4)
```

```
## Observations: 5,012  
## Variables: 3  
## $ obs_ID      <int> 2501, 2501, 2502, 2502, 2503, 2503, 2504, 2504, 25...  
## $ PRE_WEIGHT  <dbl> 159.7587, NA, 176.1611, NA, 181.3907, NA, 175.6615...  
## $ POST_WEIGHT <dbl> NA, 158.6920, NA, 174.8270, NA, 179.9042, NA, 175...
```

```
data4_1 <- data4
```

```
data4_1 <- rename(data4_1, id = obs_ID)
```

- ▶ There are 5,012 rows in this data set when there should be just 5,000.
- ▶ Similar inefficient way of recording information as in previous data set.

- ▶ Let us first investigate if there are patient with exact duplicate records.

```
data4_1 <- distinct(data4_1)
```

```
glimpse(data4_1)
```

```
## Observations: 5,012
## Variables: 3
## $ id      <int> 2501, 2501, 2502, 2502, 2503, 2503, 2504, 2504, 25...
## $ PRE_WEIGHT <dbl> 159.7587, NA, 176.1611, NA, 181.3907, NA, 175.6615...
## $ POST_WEIGHT <dbl> NA, 158.6920, NA, 174.8270, NA, 179.9042, NA, 175...
```

- ▶ There aren't any exact duplicate rows.

Let us investigate if any patients have duplicate records.

```
#counts instances of each patient id in the whole data set  
aux1 <- count(data4_1,id)  
aux2 <- which(aux1$n > 2)  
aux3 <- aux1$id[aux2]  
aux3
```

```
## [1] 4980 4981 4982 4983 4984 4985
```

Patients 4980, 4981, 4982, 4983, 4984 and 4985 have more than two records each.

```
ff <- filter(data4_1, id %in% aux3)
```

```
as.data.frame(ff)
```

##	id	PRE_WEIGHT	POST_WEIGHT
## 1	4980	151.6736	NA
## 2	4980	151.7000	NA
## 3	4980	NA	150.5249
## 4	4980	NA	150.5300
## 5	4981	171.0954	NA
## 6	4981	171.1000	NA
## 7	4981	NA	168.0745
## 8	4981	NA	168.0800
## 9	4982	154.6500	NA
## 10	4982	154.6518	NA
## 11	4982	NA	153.5068
## 12	4982	NA	153.5100
## 13	4983	141.0200	NA
## 14	4983	141.0217	NA
## 15	4983	NA	140.1000
## 16	4983	NA	140.1410
## 17	4984	137.0652	NA
## 18	4984	137.1000	NA
## 19	4984	NA	134.1708
## 20	4984	NA	134.2000
## 21	4985	195.0055	NA
## 22	4985	195.0100	NA
## 23	4985	NA	193.3600
## 24	4985	NA	193.3532

As before, truncate the data and eliminate any duplicate rows.

```
#this call rounds all the columns to the nearest integer  
data4_1 <- mutate_all(data4_1, trunc)
```

```
#eliminate duplicate rows  
data4_1 <- distinct(data4_1)
```

```
glimpse(data4_1)
```

```
## Observations: 5,000
## Variables: 3
## $ id          <dbl> 2501, 2501, 2502, 2502, 2503, 2503, 2504, 2504, 25...
## $ PRE_WEIGHT  <dbl> 159, NA, 176, NA, 181, NA, 175, NA, 136, NA, 160, ...
## $ POST_WEIGHT <dbl> NA, 158, NA, 174, NA, 179, NA, 175, NA, 137, NA, 1...
```

► We have 5,000 rows now.

- We have to create a data frame with one row per patient, as before.

```
#select all columns except POST_WEIGHT  
temp1 <- select(data4_1, -POST_WEIGHT)
```

```
#filter out the NAs in PRE_WEIGHT  
temp1 <- filter(temp1, is.na(PRE_WEIGHT) == "FALSE")
```

```
glimpse(temp1)
```

```
## Observations: 2,500  
## Variables: 2  
## $ id      <dbl> 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 250...  
## $ PRE_WEIGHT <dbl> 159, 176, 181, 175, 136, 160, 153, 163, 161, 178, 1...
```

```
#select all columns except PRE_WEIGHT
```

```
temp2 <- select(data4_1, -PRE_WEIGHT)
```

```
#filter out the NAs in POST_WEIGHT
```

```
temp2 <- filter(temp2, is.na(POST_WEIGHT) == "FALSE")
```

```
glimpse(temp2)
```

```
## Observations: 2,500
```

```
## Variables: 2
```

```
## $ id      <dbl> 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 25...
```

```
## $ POST_WEIGHT <dbl> 158, 174, 179, 175, 137, 159, 155, 164, 160, 177, ...
```

```
data4_1 <- inner_join(temp1,temp2)
```

```
## Joining, by = "id"
```

```
glimpse(data4_1)
```

```
## Observations: 2,500  
## Variables: 3  
## $ id      <dbl> 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 25...  
## $ PRE_WEIGHT <dbl> 159, 176, 181, 175, 136, 160, 153, 163, 161, 178, ...  
## $ POST_WEIGHT <dbl> 158, 174, 179, 175, 137, 159, 155, 164, 160, 177, ...
```


- ▶ Next, join the rows of data3_1 and data4_1 to create data3_2 so that all patients are in one data frame.
- ▶ First, add a column in data3_1 called trt with all values equal to 1, and a column in data4_1 called trt as well but with all values equal to zero.

```
data3_1 <- mutate(data3_1, trt = 1)
```

```
data4_1 <- mutate(data4_1, trt = 0)
```

```
glimpse(data3_1)
```

```
## Observations: 2,500
## Variables: 4
## $ id          <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,...
## $ PRE_WEIGHT  <dbl> 135, 154, 128, 183, 166, 120, 158, 160, 127, 187, ...
## $ POST_WEIGHT <dbl> 125, 153, 115, 177, 163, 105, 151, 143, 119, 159, ...
## $ trt         <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
```

```
glimpse(data4_1)
```

```
## Observations: 2,500
## Variables: 4
## $ id          <dbl> 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 25...
## $ PRE_WEIGHT  <dbl> 159, 176, 181, 175, 136, 160, 153, 163, 161, 178, ...
## $ POST_WEIGHT <dbl> 158, 174, 179, 175, 137, 159, 155, 164, 160, 177, ...
## $ trt         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
```

```
data3_2 <- bind_rows(data3_1,data4_1)
```

```
glimpse(data3_2)
```

```
## Observations: 5,000
## Variables: 4
## $ id      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,...
## $ PRE_WEIGHT <dbl> 135, 154, 128, 183, 166, 120, 158, 160, 127, 187, ...
## $ POST_WEIGHT <dbl> 125, 153, 115, 177, 163, 105, 151, 143, 119, 159, ...
## $ trt      <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
```

A unified data set

- Merge data1_1, data2_1 and data3_2.

```
data_exer1 <- inner_join(data1_1, data2_1)
```

```
## Joining, by = "id"
```

```
data_exer1 <- inner_join(data_exer1, data3_2)
```

```
## Joining, by = c("id", "trt")
```

```
glimpse(data_exer1)
```

```
## Observations: 5,000
## Variables: 9
## $ id      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,...
## $ Sex     <chr> "MALE", "FEMALE", "FEMALE", "FEMALE", "FEMALE", "F...
## $ Age     <dbl> 41.2, 42.9, 38.5, 35.6, 48.5, 36.9, 28.9, 24.7, 47...
## $ Race    <chr> "White", "White", "White", "Hispanic", "White", NA...
## $ trt     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ PRE_SRH <chr> "Good", "Poor", "Satisfactory", "Poor", "Poor", "G...
## $ POST_SRH <chr> "Poor", "Very Poor", "Good", "Good", "Poor", "Exce...
## $ PRE_WEIGHT <dbl> 135, 154, 128, 183, 166, 120, 158, 160, 127, 187, ...
## $ POST_WEIGHT <dbl> 125, 153, 115, 177, 163, 105, 151, 143, 119, 159, ...
```

- ▶ Further narrow the number of variables introducing a variable Time, with values PRE and POST, and gather SRH and Weight.
- ▶ To do that, first create one data frame with all the fixed variables and pre- and post-SRH and another data frame with all the fixed variables and pre- and post-WEIGHT.
- ▶ Gather the pre- and post- column into Time and SRH or WEIGHT and then we will join both data sets.

```
temp1 <- select(data_exer1, 1:7)
```

```
temp1 <- gather(temp1, "Time", "SRH", 6:7)
```

```
temp1 <- mutate(temp1,  
                 Time = str_replace(Time, "PRE_SRH", "PRE"))
```

```
temp1 <- mutate(temp1,  
                 Time = str_replace(Time, "POST_SRH", "POST"))
```

```
glimpse(temp1)
```

```
## Observations: 10,000  
## Variables: 7  
## $ id   <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17...  
## $ Sex  <chr> "MALE", "FEMALE", "FEMALE", "FEMALE", "FEMALE", "FEMALE",...  
## $ Age  <dbl> 41.2, 42.9, 38.5, 35.6, 48.5, 36.9, 28.9, 24.7, 47.1, 37...  
## $ Race <chr> "White", "White", "White", "Hispanic", "White", NA, "Whit...  
## $ trt  <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...  
## $ Time <chr> "PRE", "PRE", "PRE", "PRE", "PRE", "PRE", "PRE", "PRE", "...  
## $ SRH  <chr> "Good", "Poor", "Satisfactory", "Poor", "Poor", "Good", "...
```

Using "%>%"

```
temp1 <- select(data_exer1, 1:7) %>%  
  gather(temp1, "Time", "SRH", 6:7) %>%  
  mutate(  
    Time = str_replace(Time, "PRE_SRH", "PRE")) %>%  
  mutate(  
    Time = str_replace(Time, "POST_SRH", "POST"))
```

```
temp2 <- select(data_exer1, c(1:5,8,9))
```

```
temp2 <- gather(temp2, "Time", "WEIGHT", 6:7)
```

```
temp2 <- mutate(  
  temp2, Time = str_replace(Time, "PRE_WEIGHT", "PRE"))
```

```
temp2 <- mutate(  
  temp2, Time = str_replace(Time, "POST_WEIGHT", "POST"))
```

```
glimpse(temp2)
```

```
## Observations: 10,000  
## Variables: 7  
## $ id      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ...  
## $ Sex     <chr> "MALE", "FEMALE", "FEMALE", "FEMALE", "FEMALE", "FEMALE...  
## $ Age     <dbl> 41.2, 42.9, 38.5, 35.6, 48.5, 36.9, 28.9, 24.7, 47.1, 3...  
## $ Race    <chr> "White", "White", "White", "Hispanic", "White", NA, "Wh...  
## $ trt     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...  
## $ Time    <chr> "PRE", "PRE", "PRE", "PRE", "PRE", "PRE", "PRE", "PRE",...  
## $ WEIGHT  <dbl> 135, 154, 128, 183, 166, 120, 158, 160, 127, 187, 149, ...
```

Using "%>%"

```
temp2 <- select(data_exer1, c(1:5, 8, 9)) %>%  
  gather(temp1, "Time", "WEIGHT", 6:7) %>%  
  mutate(  
    Time = str_replace(Time, "PRE_WEIGHT", "PRE")) %>%  
  mutate(  
    Time = str_replace(Time, "POST_WEIGHT", "POST"))
```


► Merge temp1 and temp2

```
data_exer2 <- inner_join(temp1, temp2)
```

```
## Joining, by = c("id", "Sex", "Age", "Race", "trt", "Time")
```

```
glimpse(data_exer2)
```

```
## Observations: 10,000
```

```
## Variables: 8
```

```
## $ id      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ...
## $ Sex      <chr> "MALE", "FEMALE", "FEMALE", "FEMALE", "FEMALE", "FEMALE", "FEMALE..."
## $ Age      <dbl> 41.2, 42.9, 38.5, 35.6, 48.5, 36.9, 28.9, 24.7, 47.1, 3...
## $ Race     <chr> "White", "White", "White", "Hispanic", "White", NA, "Wh...
## $ trt      <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ Time     <chr> "PRE", "PRE", "PRE", "PRE", "PRE", "PRE", "PRE", "PRE", "PRE",...
## $ SRH      <chr> "Good", "Poor", "Satisfactory", "Poor", "Poor", "Good", "Good",...
## $ WEIGHT   <dbl> 135, 154, 128, 183, 166, 120, 158, 160, 127, 187, 149, ...
```

- ▶ Both `data_exer1` and `data_exer2` are tidy, and it is a matter of preference and how the data will be used to choose a format to work with.
- ▶ We will work with both data frames for the purpose of visualising the data.

- ▶ The variable SRH is an ordinal categorical variable. Let us tell R about the ordinal features of SRH.
- ▶ Also, Time is categorical ordinal. Sex and Race are simply categorical.
- ▶ Treatment, trt, is also a factor and we will let R know and re-label the levels from 0 to Control and 1 to Treatment.
- ▶ We will do this for both data frames data_exer1 and data_exer2.

We introduce now a new Tidyverse package called `forcats` which is designed to deal with factors.



```
library(forcats)
```

```
#ordered levels of SRH
```

```
srhlev <- c("Very Poor", "Poor", "Satisfactory", "Good", "Excellent")
```

```
#declare trt, srh, sex and race as factors, relabel trt and order srh
```

```
data_exer1 <- data_exer1 %>%
```

```
  mutate(trt = factor(trt), PRE_SRH = factor(PRE_SRH),
```

```
         POST_SRH = factor(POST_SRH),
```

```
         Sex = factor(Sex), Race = factor(Race)) %>%
```

```
  mutate(trt = fct_recode(trt, Control = "0", Treatment = "1")) %>%
```

```
  mutate(PRE_SRH = fct_relevel(PRE_SRH, srhlev)) %>%
```

```
  mutate(POST_SRH = fct_relevel(POST_SRH, srhlev))
```

```
glimpse(data_exer1)
```

```
## Observations: 5,000
## Variables: 9
## $ id          <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,...
## $ Sex         <fct> MALE, FEMALE, FEMALE, FEMALE, FEMALE, FEMALE, FEMA...
## $ Age         <dbl> 41.2, 42.9, 38.5, 35.6, 48.5, 36.9, 28.9, 24.7, 47...
## $ Race        <fct> White, White, White, Hispanic, White, NA, White, H...
## $ trt         <fct> Treatment, Treatment, Treatment, Treatment, Treatm...
## $ PRE_SRH     <fct> Good, Poor, Satisfactory, Poor, Poor, Good, Excell...
## $ POST_SRH    <fct> Poor, Very Poor, Good, Good, Poor, Excellent, Exce...
## $ PRE_WEIGHT  <dbl> 135, 154, 128, 183, 166, 120, 158, 160, 127, 187, ...
## $ POST_WEIGHT <dbl> 125, 153, 115, 177, 163, 105, 151, 143, 119, 159, ...
```

```
summary(data_exer1)
```

```
##           id           Sex           Age           Race
## Min.      : 1    FEMALE:2556   Min.      :18.10   Asian    : 450
## 1st Qu.:1251   MALE  :2444   1st Qu.:33.40   Black    : 424
## Median :2500                                Median :38.80   Hispanic: 846
## Mean    :2500                                Mean    :38.89   White    :2876
## 3rd Qu.:3750                                3rd Qu.:44.10   NA's     : 404
## Max.     :5000                                Max.     :69.30
##           trt           PRE_SRH           POST_SRH           PRE_WEIGHT
## Control   :2500   Very Poor   :1321   Very Poor   :1040   Min.      : 82.0
## Treatment:2500   Poor        : 959   Poor        : 752   1st Qu.:145.0
##                                     Satisfactory: 611   Satisfactory: 800   Median :159.0
##                                     Good         : 736   Good         : 834   Mean    :159.4
##                                     Excellent    :1373   Excellent    :1574   3rd Qu.:174.0
##                                     Max.         :242.0
## POST_WEIGHT
## Min.      : 78.0
## 1st Qu.:140.0
## Median :154.0
## Mean    :154.5
## 3rd Qu.:169.0
## Max.     :229.0
```

For `data_exer2`, we also set the order of the levels of `Time` to be PRE and POST.

```
data_exer2 <- data_exer2 %>%  
  mutate(trt = factor(trt), SRH = factor(SRH), Sex = factor(Sex),  
         Race = factor(Race), Time = factor(Time)) %>%  
  mutate(trt = fct_recode(trt, Control = "0", Treatment = "1")) %>%  
  mutate(SRH = fct_relevel(SRH, srhlev)) %>%  
  mutate(Time = fct_relevel(Time, "PRE", "POST"))
```



```
glimpse(data_exer2)
```

```
## Observations: 10,000
## Variables: 8
## $ id      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ...
## $ Sex     <fct> MALE, FEMALE, FEMALE, FEMALE, FEMALE, FEMALE, FEMALE, M...
## $ Age     <dbl> 41.2, 42.9, 38.5, 35.6, 48.5, 36.9, 28.9, 24.7, 47.1, 3...
## $ Race    <fct> White, White, White, Hispanic, White, NA, White, Hispan...
## $ trt     <fct> Treatment, Treatment, Treatment, Treatment, Treatment, ...
## $ Time    <fct> PRE, PRE, PRE, PRE, PRE, PRE, PRE, PRE, PRE, PRE, PRE, ...
## $ SRH     <fct> Good, Poor, Satisfactory, Poor, Poor, Good, Excellent, ...
## $ WEIGHT  <dbl> 135, 154, 128, 183, 166, 120, 158, 160, 127, 187, 149, ...
```

- ▶ Age is continuous in nature.
- ▶ However, it is not really expected that at each small change in age we would observe changes in treatment effects.
- ▶ Also, in case effects change according to age, a change of, say 5 years, at young age will not possibly see the same effect as a change of 5 years at middle or old age.
- ▶ Construct age groups.
- ▶ Note that we will use Age from data_exer1 because it contains just one row per observational unit.

```
data_exer1 %>% select(Age) %>% summary()
```

```
##           Age
##  Min.      :18.10
##  1st Qu.:33.40
##  Median :38.80
##  Mean     :38.89
##  3rd Qu.:44.10
##  Max.     :69.30
```

```
aux <- mutate(  
  data_exer1, Age = cut(Age, c(18, 20, 30, 40, 50, 60,70)))
```

```
aux %>% select(Age) %>% summary()
```

```
##      Age  
## (18,20]: 33  
## (20,30]: 649  
## (30,40]: 2133  
## (40,50]: 1779  
## (50,60]: 374  
## (60,70]: 32
```

- ▶ Most participants are 30 - 50 years of age.
- ▶ No participants are younger than 18 years or older than 70 years.
- ▶ Split age into 18-34yrs, 35-40yrs, 41-45yrs, 46-70yrs, so that we don't have an age category over-represented with number of patients.
- ▶ Create a new column in both data_exer1 and data_exer2 called Age_cat which indicates the age group of the patient.

```
data_exer1 <- data_exer1 %>%  
  mutate(Age_cat = cut(Age, c(18, 34, 39, 44, 70)))
```

```
data_exer2 <- data_exer2 %>%  
  mutate(Age_cat = cut(Age, c(18, 34, 39, 44, 70)))
```

```
data_exer1 %>% select(Age_cat) %>% summary()
```

```
##      Age_cat  
## (18,34]:1345  
## (34,39]:1223  
## (39,44]:1162  
## (44,70]:1270
```

or in base R

```
summary(data_exer1$Age_cat)
```

```
## (18,34] (34,39] (39,44] (44,70]  
##      1345      1223      1162      1270
```

Let us see how the data looks like now

```
glimpse(data_exer1)
```

```
## Observations: 5,000
## Variables: 10
## $ id          <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,...
## $ Sex         <fct> MALE, FEMALE, FEMALE, FEMALE, FEMALE, FEMALE, FEMALE...
## $ Age         <dbl> 41.2, 42.9, 38.5, 35.6, 48.5, 36.9, 28.9, 24.7, 47...
## $ Race        <fct> White, White, White, Hispanic, White, NA, White, H...
## $ trt         <fct> Treatment, Treatment, Treatment, Treatment, Treatm...
## $ PRE_SRH     <fct> Good, Poor, Satisfactory, Poor, Poor, Good, Excell...
## $ POST_SRH    <fct> Poor, Very Poor, Good, Good, Poor, Excellent, Exce...
## $ PRE_WEIGHT  <dbl> 135, 154, 128, 183, 166, 120, 158, 160, 127, 187, ...
## $ POST_WEIGHT <dbl> 125, 153, 115, 177, 163, 105, 151, 143, 119, 159, ...
## $ Age_cat     <fct> (39,44], (39,44], (34,39], (34,39], (44,70], (34,3...
```

```
glimpse(data_exer2)
```

```
## Observations: 10,000
## Variables: 9
## $ id      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ...
## $ Sex     <fct> MALE, FEMALE, FEMALE, FEMALE, FEMALE, FEMALE, FEMALE, ...
## $ Age     <dbl> 41.2, 42.9, 38.5, 35.6, 48.5, 36.9, 28.9, 24.7, 47.1, ...
## $ Race    <fct> White, White, White, Hispanic, White, NA, White, Hispa...
## $ trt     <fct> Treatment, Treatment, Treatment, Treatment, Treatment, ...
## $ Time    <fct> PRE, PRE, PRE, PRE, PRE, PRE, PRE, PRE, PRE, PRE, PRE, ...
## $ SRH     <fct> Good, Poor, Satisfactory, Poor, Poor, Good, Excellent, ...
## $ WEIGHT  <dbl> 135, 154, 128, 183, 166, 120, 158, 160, 127, 187, 149, ...
## $ Age_cat <fct> (39,44], (39,44], (34,39], (34,39], (44,70], (34,39], ...
```

Let us shuffle the columns so that Age and Age_cat are contiguous.

```
data_exer1 <- data_exer1 %>% select(c(1:3,10,4:9))
```

```
data_exer2 <- data_exer2 %>% select(c(1:3,9,4:8))
```

```
glimpse(data_exer1)
```

```
## Observations: 5,000
## Variables: 10
## $ id          <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,...
## $ Sex         <fct> MALE, FEMALE, FEMALE, FEMALE, FEMALE, FEMALE, FEMA...
## $ Age         <dbl> 41.2, 42.9, 38.5, 35.6, 48.5, 36.9, 28.9, 24.7, 47...
## $ Age_cat     <fct> (39,44], (39,44], (34,39], (34,39], (44,70], (34,3...
## $ Race        <fct> White, White, White, Hispanic, White, NA, White, H...
## $ trt         <fct> Treatment, Treatment, Treatment, Treatment, Treatm...
## $ PRE_SRH     <fct> Good, Poor, Satisfactory, Poor, Poor, Good, Excell...
## $ POST_SRH    <fct> Poor, Very Poor, Good, Good, Poor, Excellent, Exce...
## $ PRE_WEIGHT  <dbl> 135, 154, 128, 183, 166, 120, 158, 160, 127, 187, ...
## $ POST_WEIGHT <dbl> 125, 153, 115, 177, 163, 105, 151, 143, 119, 159, ...
```



```
glimpse(data_exer2)
```

```
## Observations: 10,000
## Variables: 9
## $ id      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,...
## $ Sex      <fct> MALE, FEMALE, FEMALE, FEMALE, FEMALE, FEMALE, FEMALE, ...
## $ Age      <dbl> 41.2, 42.9, 38.5, 35.6, 48.5, 36.9, 28.9, 24.7, 47.1, ...
## $ Age_cat  <fct> (39,44], (39,44], (34,39], (34,39], (44,70], (34,39], ...
## $ Race     <fct> White, White, White, Hispanic, White, NA, White, Hispa...
## $ trt      <fct> Treatment, Treatment, Treatment, Treatment, Treatment,...
## $ Time     <fct> PRE, PRE, PRE, PRE, PRE, PRE, PRE, PRE, PRE, PRE, PRE,...
## $ SRH      <fct> Good, Poor, Satisfactory, Poor, Poor, Good, Excellent,...
## $ WEIGHT   <dbl> 135, 154, 128, 183, 166, 120, 158, 160, 127, 187, 149,...
```

```
summary(data_exer2$trt)
```

```
## Control Treatment
##      5000      5000
```

- ▶ Let us see a few ways in which we can tabulate some aspects of the data.
- ▶ We can use `table()`, `ftable()` (more than two variables) or `xtabs()`.

```
with(data_exer2, table(SRH,Time))
```

```
##              Time
## SRH          PRE POST
## Very Poor    1321 1040
## Poor         959  752
## Satisfactory 611  800
## Good         736  834
## Excellent    1373 1574
```

- ▶ Without considering any other factor, there are less patients in the Poor and Very Poor categories after the intervention.
- ▶ there are more patients in the Satisfactory, Good and Excellent categories after the intervention than before the intervention.

Let us see if the trend stays when we consider also the patient's gender.

```
aux2 <- with(data_exer2, table(SRH,Time,Sex))
aux2
```

```
## , , Sex = FEMALE
##
##           Time
## SRH         PRE POST
##  Very Poor    660  576
##   Poor       443  414
## Satisfactory 337  367
##   Good       391  382
##  Excellent   725  817
##
## , , Sex = MALE
##
##           Time
## SRH         PRE POST
##  Very Poor    661  464
##   Poor       516  338
## Satisfactory 274  433
##   Good       345  452
##  Excellent   648  757
```

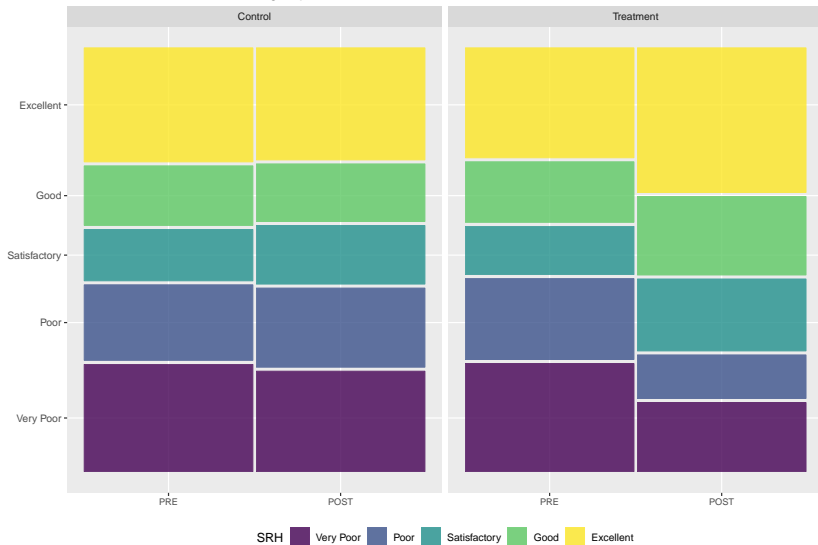
- ▶ The trend of positive effect of the exercise plan is still present when we consider gender.
- ▶ It seems to be stronger for men than for women.

Visualising SRH using mosaic plots in ggplot2

First, we simply consider SRH by gender for the different groups, before and after the treatment.

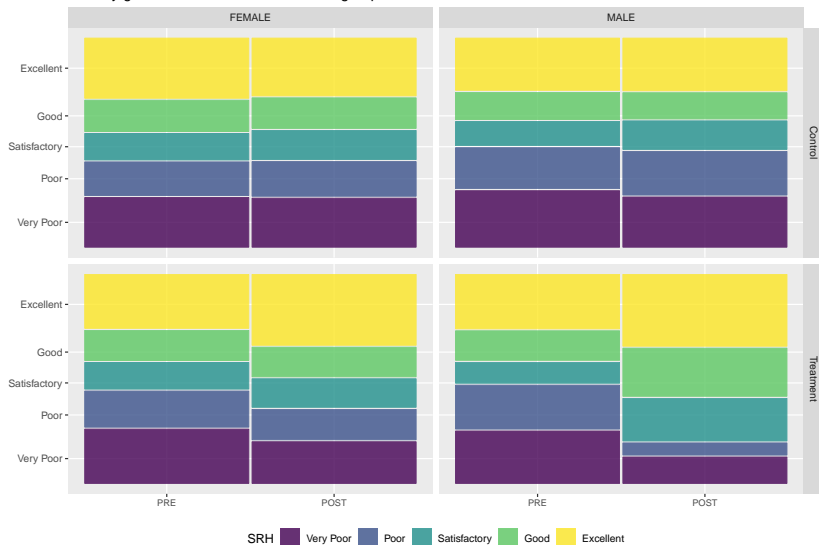
```
ggplot(data = data_exer2) +  
  geom_mosaic(aes( x = product(SRH, Time), fill = SRH), na.rm = TRUE) +  
  theme(axis.text.x=element_text(angle = -90, hjust = .1, size = 4)) +  
  facet_grid(.~trt, drop = TRUE) +  
  theme(axis.text.x = element_text(angle = 0, hjust = .5, size = 8)) +  
  theme(axis.title = element_blank()) +  
  theme(legend.position = "bottom") +  
  ggtitle ('SRH for treatment and control groups')
```

SRH for treatment and control groups




```
ggplot(data = data_exer2) +  
  geom_mosaic(aes( x = product(SRH, Time), fill = SRH), na.rm = TRUE) +  
  theme(axis.text.x=element_text(angle = -90, hjust = .1, size = 4)) +  
  facet_grid(trt~Sex, drop = TRUE) +  
  theme(axis.text.x = element_text(angle = 0, hjust = .5, size = 8)) +  
  theme(axis.title = element_blank()) +  
  theme(legend.position = "bottom") +  
  ggtitle ('SRH by gender for treatment and control groups')
```

SRH by gender for treatment and control groups

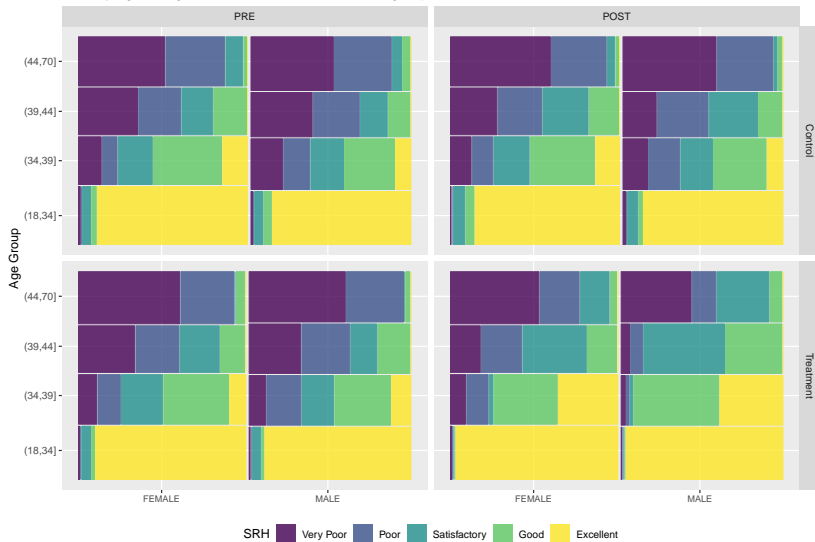


```

ggplot(data = data_exer2) +
  geom_mosaic(aes(x = product(SRH, Age_cat, Sex), fill=SRH), na.rm=TRUE) +
  theme(axis.text.x=element_text(angle = -90, hjust = .1, size = 4)) +
  facet_grid(trt~Time, drop = TRUE) +
  scale_x_productlist(breaks = c(0.25, 0.75), labels = c("FEMALE", "MALE")) +
  labs(y = "Age Group") +
  theme(axis.text.x = element_text(angle = 0, hjust = .5, size = 8)) +
  theme(axis.title.x = element_blank()) +
  theme(legend.position = "bottom") +
  ggtitle('SRH by age and gender for treatment and control groups')

```

SRH by age and gender for treatment and control groups

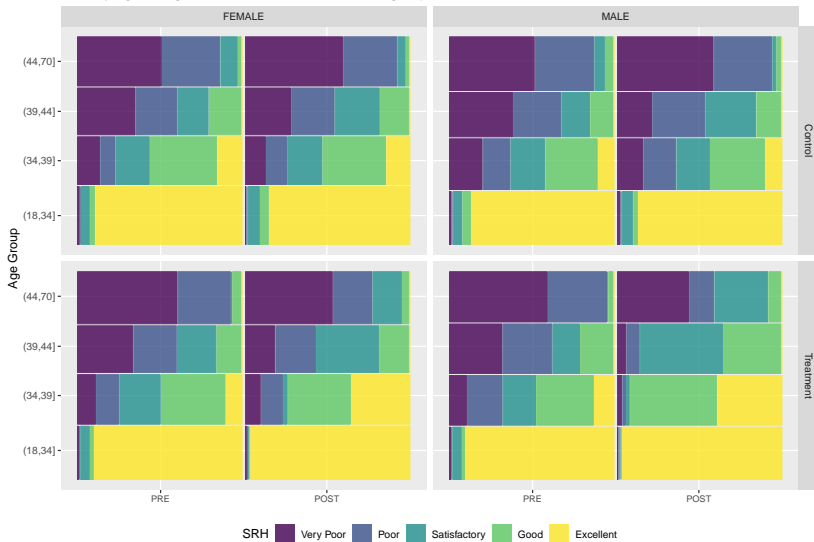


```

ggplot(data = data_exer2) +
  geom_mosaic(aes( x = product(SRH, Age_cat, Time), fill=SRH), na.rm=TRUE) +
  theme(axis.text.x=element_text(angle = -90, hjust = .1, size = 4)) +
  facet_grid(trt~Sex, drop = TRUE) +
  scale_x_productlist(breaks = c(0.25, 0.75), labels = c("PRE", "POST")) +
  labs(y = "Age Group") +
  theme(axis.text.x = element_text(angle = 0, hjust = .5, size = 8)) +
  theme(axis.title.x = element_blank()) +
  theme(legend.position = "bottom") +
  ggtitle ('SRH by age and gender for treatment and control groups')

```

SRH by age and gender for treatment and control groups

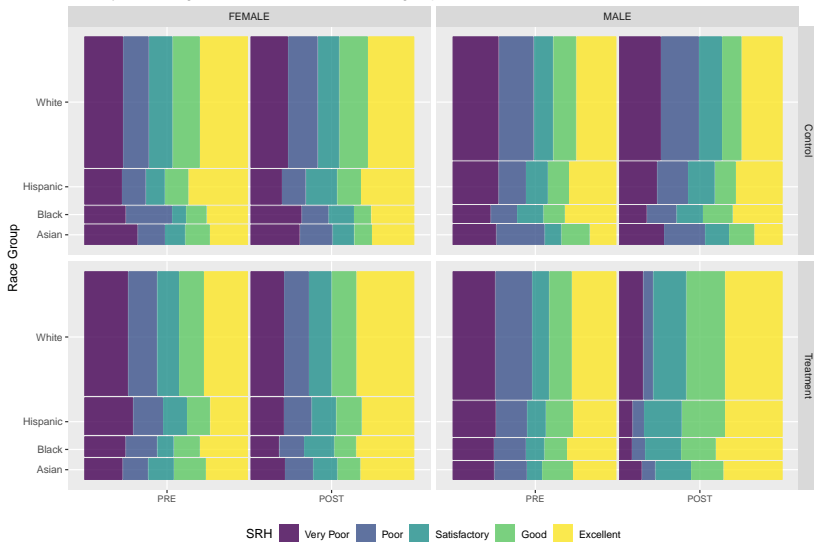


```

ggplot(data = subset(data_exer2, is.na(Race) == "FALSE")) +
  geom_mosaic(aes( x = product(SRH, Race, Time), fill = SRH), na.rm=TRUE) +
  theme(axis.text.x = element_text(angle = -90, hjust = .1, size = 4)) +
  facet_grid(trt ~ Sex) +
  scale_x_productlist(breaks = c(0.25, 0.75), labels = c("PRE", "POST")) +
  labs(y = "Race Group") +
  theme(axis.text.x = element_text(angle = 0, hjust = .5, size = 8)) +
  theme(axis.title.x = element_blank()) +
  theme(legend.position = "bottom") +
  ggtitle('SRH by race and gender for treatment and control groups') +
  theme(legend.position = "bottom")

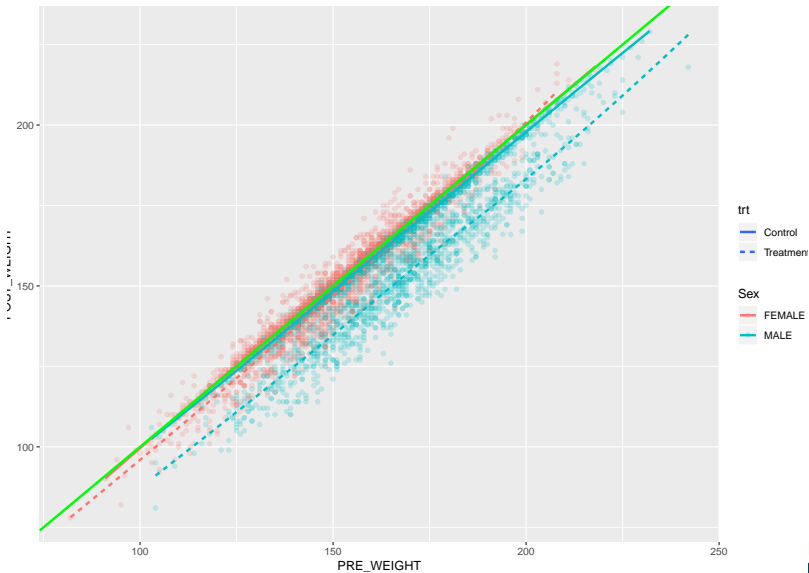
```

SRH by race and gender for treatment and control groups



- ▶ Let us visualise now the measured variables that are continuous, namely Weight.
- ▶ Plot POST_WEIGHT vs PRE_WEIGHT for males (blue) and females (red), before (solid line) and after (dotted line) the intervention.
- ▶ Add a 45 degree line through zero (symbolising no treatment effect on weight, i.e. weight before is equal to weight after).

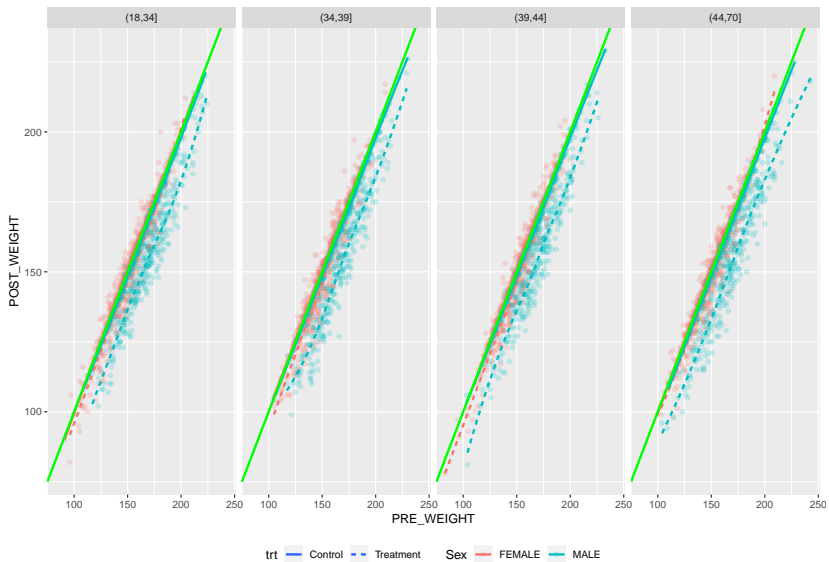
```
ggplot(data_exer1, aes(x=PRE_WEIGHT, y=POST_WEIGHT, col=Sex, linetype=trt))+  
  geom_point(alpha=0.2) +  
  stat_smooth(method = "loess", se = FALSE, lwd=1) +  
  geom_abline(intercept = 0, slope = 1, color = "green", lwd = 1)
```



- ▶ For both sexes the curves of the control patients are nearly overlapping the no-effect green line, as expected.
- ▶ The treatment curve for women is very near the no-effect green line. There seems to be a very mild positive effect for women whose weights before the intervention were below 150 pounds.
- ▶ The intervention seems to be effective for weight loss for men who underwent the exercise programme, as the blue dotted curve is clearly below the no-effect green line.

- Let us split the data into age categories.

```
ggplot(data_exer1, aes(x = PRE_WEIGHT, y = POST_WEIGHT, col = Sex,  
                      linetype = trt )) +  
  geom_point(alpha=0.2) +  
  facet_grid(.~Age_cat) +  
  stat_smooth(method = "loess", se = FALSE, lwd=1) +  
  geom_abline(intercept = 0, slope = 1, color = "green", lwd = 1) +  
  theme(legend.position = "bottom")
```



The observations we made before still hold when we split the group by age category. The effect of the exercise programme seems to be more beneficial for men aged 45 or older who weigh over 200 pounds.

<https://www.smartsurvey.co.uk/s/AL6P8/>