Randomization exercise

1. Assigning Treatment

Control or treatment (same setup with the excel)

```
number_of_people <- 30 # number of participants

# make a dataframe for each people
df <- data.frame(Participant = 1:number_of_people)
# randomly assign the treatment status for each people
set.seed(2) # fix random seed
df <- df %>%
  # generate random number for each person
  dplyr::mutate(random_number = runif(number_of_people)) %>%
  # assign treatment/control depending on the above random number
  dplyr::mutate(treatment = ifelse(random_number > 0.5, 1, 0))
df
```

```
##    Participant random_number treatment
## 1            1    0.18488226         0
## 2            2    0.70237404         1
## 3            3    0.57332633         1
## 4            4    0.16805192         0
## 5            5    0.94383934         1
## 6            6    0.94347496         1
## 7            7    0.12915898         0
## 8            8    0.83344882         1
## 9            9    0.46801852         0
## 10          10    0.54998374         1
## 11          11    0.55267407         1
## 12          12    0.23889476         0
## 13          13    0.76051331         1
## 14          14    0.18082010         0
## 15          15    0.40528218         0
## 16          16    0.85354845         1
## 17          17    0.97639849         1
## 18          18    0.22582546         0
## 19          19    0.44480923         0
## 20          20    0.07497942         0
## 21          21    0.66189876         1
## 22          22    0.38754954         0
## 23          23    0.83688918         1
## 24          24    0.15050144         0
## 25          25    0.34727225         0
## 26          26    0.48877323         0
## 27          27    0.14924686         0
## 28          28    0.35706259         0
```

```
## 29          29    0.96264405         1
## 30          30    0.13237200         0
```

```
sum(df$treatment)
```

```
## [1] 13
```

To be more precise, this is a bit different from what you did in excel. In excel, you first sorted and force to be a treatment for half of the people. Essentially, there is no much difference. Here, you just assign people who happen to get a higher value to the treatment.

Just for completeness, you can also do the exact same way in your excel as follows.

```
df <- data.frame(Participant = 1:number_of_people)
# randomly assign the treatment status for each people
set.seed(2) # fix random seed
df <- df %>%
  # generate random number for each person
  dplyr::mutate(random_number = runif(number_of_people)) %>%
  # sort by the value
  dplyr::arrange(random_number) %>%
  # assign treatment/control depending on the above random number
  dplyr::mutate(row_num = 1:number_of_people) %>%
  dplyr::mutate(treatment = ifelse(row_num > number_of_people/2, 1, 0))
df
```

```
##    Participant random_number row_num treatment
## 1           20    0.07497942       1         0
## 2            7    0.12915898       2         0
## 3           30    0.13237200       3         0
## 4           27    0.14924686       4         0
## 5           24    0.15050144       5         0
## 6            4    0.16805192       6         0
## 7           14    0.18082010       7         0
## 8            1    0.18488226       8         0
## 9           18    0.22582546       9         0
## 10          12    0.23889476      10         0
## 11          25    0.34727225      11         0
## 12          28    0.35706259      12         0
## 13          22    0.38754954      13         0
## 14          15    0.40528218      14         0
## 15          19    0.44480923      15         0
## 16           9    0.46801852      16         1
## 17          26    0.48877323      17         1
## 18          10    0.54998374      18         1
## 19          11    0.55267407      19         1
## 20           3    0.57332633      20         1
## 21          21    0.66189876      21         1
## 22           2    0.70237404      22         1
## 23          13    0.76051331      23         1
## 24           8    0.83344882      24         1
## 25          23    0.83688918      25         1
## 26          16    0.85354845      26         1
## 27           6    0.94347496      27         1
## 28           5    0.94383934      28         1
## 29          29    0.96264405      29         1
## 30          17    0.97639849      30         1
```

```
sum(df$treatment)
```

```
## [1] 15
```

Note that this should not work in the case when odd `number_of_people`. You just need to modify a bit to take into account the tie (not a big deal!). I leave it for your exercise.

Comparing these two ways, here you are kind of enforcing the number of people in the treatment/control. Notice that we fix the random generator, so they are the same in both ways. Although, the number of people allocated into the treatment in the first way is `13` and it is `15` in the second way. Statistically, it won't matter much whichever way you go (try larger `number_of_people`). But in practice, how much control/treatment you want depends on the question, so this `0.5` may depend on the case.

What if you have multiple treatments?

This is not a long journey from the original way. You just need to make a kind of placeholder to assign every treatment and control. There are many ways to do the same thing but here is my preffered way.

```
number_of_people <- 30 # number of participants
number_of_treatment <- 3 # control, treatment A, ...

# make a dataframe for each people
df <- data.frame(Participant = 1:number_of_people)
# randomly assign the treatment status for each people
treatment_placeholder <- rep(1:number_of_treatment, number_of_people / number_of_treatment)
treatment_placeholder
```

```
##  [1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
```

```
# randomly assign the treatment status for each people
set.seed(2) # fix random seed
df <- df %>%
  # generate random number for each person
  dplyr::mutate(random_number = runif(number_of_people)) %>%
  # assign treatment/control depending on the above random number
  dplyr::mutate(treatment = treatment_placeholder[order(random_number)]) # just randomize treatment_pla
df
```

```
##    Participant random_number treatment
## 1            1    0.18488226         2
## 2            2    0.70237404         1
## 3            3    0.57332633         3
## 4            4    0.16805192         3
## 5            5    0.94383934         3
## 6            6    0.94347496         1
## 7            7    0.12915898         2
## 8            8    0.83344882         1
## 9            9    0.46801852         3
## 10          10    0.54998374         3
## 11          11    0.55267407         1
## 12          12    0.23889476         1
## 13          13    0.76051331         1
## 14          14    0.18082010         3
## 15          15    0.40528218         1
## 16          16    0.85354845         3
## 17          17    0.97639849         2
## 18          18    0.22582546         1
```

```
## 19          19    0.44480923         2
## 20          20    0.07497942         3
## 21          21    0.66189876         3
## 22          22    0.38754954         2
## 23          23    0.83688918         1
## 24          24    0.15050144         2
## 25          25    0.34727225         2
## 26          26    0.48877323         1
## 27          27    0.14924686         3
## 28          28    0.35706259         2
## 29          29    0.96264405         2
## 30          30    0.13237200         2
```

Clearly, by using `treatment_placeholder`, we are imposing the number of treatment A, B, and control.

2. Blocking/Stratifying

Let's think about the data that includes some contexts (gender).

```
number_of_people <- 30
number_of_treatment <- 2

set.seed(2)
df <- data.frame(Participant = 1:number_of_people)
df <- df %>%
  dplyr::mutate(group = sample(c("male","female"), number_of_people, replace = TRUE))
df
```

```
##    Participant  group
## 1            1   male
## 2            2   male
## 3            3 female
## 4            4 female
## 5            5 female
## 6            6 female
## 7            7   male
## 8            8   male
## 9            9   male
## 10          10 female
## 11          11   male
## 12          12   male
## 13          13 female
## 14          14   male
## 15          15   male
## 16          16   male
## 17          17 female
## 18          18 female
## 19          19 female
## 20          20 female
## 21          21   male
## 22          22   male
## 23          23   male
## 24          24 female
## 25          25   male
## 26          26   male
```

```
## 27          27 female
## 28          28   male
## 29          29 female
## 30          30 female
```

```
# randomly assign the treatment status for each people by block
set.seed(2) # fix random seed
df_assigned <- df %>%
  # generate random number for each person
  dplyr::mutate(random_number = runif(number_of_people)) %>%
  # sort random_number by group
  dplyr::arrange(group, random_number) %>%
  # assign treatment/control depending on the above random number
  dplyr::group_by(group) %>%
  dplyr::mutate(row_number = dplyr::row_number(),
                median_rownumber = median(row_number)) %>%
  dplyr::mutate(treatment = ifelse(row_number >= floor(median_rownumber), 1, 0)) %>%
  dplyr::select(Participant, group, random_number, treatment)
as.data.frame(df_assigned)
```

```
##    Participant  group random_number treatment
## 1           20 female    0.07497942         0
## 2           30 female    0.13237200         0
## 3           27 female    0.14924686         0
## 4           24 female    0.15050144         0
## 5            4 female    0.16805192         0
## 6           18 female    0.22582546         0
## 7           19 female    0.44480923         1
## 8           10 female    0.54998374         1
## 9            3 female    0.57332633         1
## 10          13 female    0.76051331         1
## 11           6 female    0.94347496         1
## 12           5 female    0.94383934         1
## 13          29 female    0.96264405         1
## 14          17 female    0.97639849         1
## 15           7   male    0.12915898         0
## 16          14   male    0.18082010         0
## 17           1   male    0.18488226         0
## 18          12   male    0.23889476         0
## 19          25   male    0.34727225         0
## 20          28   male    0.35706259         0
## 21          22   male    0.38754954         0
## 22          15   male    0.40528218         1
## 23           9   male    0.46801852         1
## 24          26   male    0.48877323         1
## 25          11   male    0.55267407         1
## 26          21   male    0.66189876         1
## 27           2   male    0.70237404         1
## 28           8   male    0.83344882         1
## 29          23   male    0.83688918         1
## 30          16   male    0.85354845         1
```

I am a little loose here for the tie person. You can toss a coin to decide their destiny. I leave it for your exercise.

Let's just sort them by participants.

```
df_assigned %>% dplyr::arrange(Participant, group) %>% as.data.frame()
```

```
##    Participant  group random_number treatment
## 1            1   male    0.18488226         0
## 2            2   male    0.70237404         1
## 3            3 female    0.57332633         1
## 4            4 female    0.16805192         0
## 5            5 female    0.94383934         1
## 6            6 female    0.94347496         1
## 7            7   male    0.12915898         0
## 8            8   male    0.83344882         1
## 9            9   male    0.46801852         1
## 10          10 female    0.54998374         1
## 11          11   male    0.55267407         1
## 12          12   male    0.23889476         0
## 13          13 female    0.76051331         1
## 14          14   male    0.18082010         0
## 15          15   male    0.40528218         1
## 16          16   male    0.85354845         1
## 17          17 female    0.97639849         1
## 18          18 female    0.22582546         0
## 19          19 female    0.44480923         1
## 20          20 female    0.07497942         0
## 21          21   male    0.66189876         1
## 22          22   male    0.38754954         0
## 23          23   male    0.83688918         1
## 24          24 female    0.15050144         0
## 25          25   male    0.34727225         0
## 26          26   male    0.48877323         1
## 27          27 female    0.14924686         0
## 28          28   male    0.35706259         0
## 29          29 female    0.96264405         1
## 30          30 female    0.13237200         0
```

These procedure may look complicated. Here is one-linear version by Ariel.

```
df_assigned <- df[order(df$group, runif(number_of_people)),]
df_assigned$treatment <- rep(1:number_of_treatment, number_of_people / number_of_treatment)
df_assigned %>% dplyr::arrange(Participant, group)
```

```
##    Participant  group treatment
## 1            1   male         2
## 2            2   male         1
## 3            3 female         2
## 4            4 female         2
## 5            5 female         1
## 6            6 female         1
## 7            7   male         1
## 8            8   male         1
## 9            9   male         1
## 10          10 female         2
## 11          11   male         1
## 12          12   male         2
## 13          13 female         1
## 14          14   male         2
```

```
## 15          15   male         2
## 16          16   male         1
## 17          17 female         2
## 18          18 female         1
## 19          19 female         2
## 20          20 female         1
## 21          21   male         1
## 22          22   male         1
## 23          23   male         2
## 24          24 female         1
## 25          25   male         2
## 26          26   male         2
## 27          27 female         1
## 28          28   male         2
## 29          29 female         2
## 30          30 female         2
```

If you are enthusiastic, try blocking exercise in the case of the multiple treatments.