LEARN Center
EPFL

Analysis of Exam Scores

Data for Only "New" Students (Year 1 & Year 2)

Himanshu Verma, Cécile Hardebolle

Flipped Classroom Project

13 novembre 2019

Table des matières

1 Background and Rationale **2**

2 R Package Imports **2**

3 Data Import and Pre-Processing **3**
3.0.1 Data Import 2
3.0.2 Data Summary 2
3.0.3 Data Filtering 3
3.0.4 Data For Visualization 5

4 Gender Differences **5**
4.1 Visualizing Scores 5
4.1.1 Year1 6
4.1.2 Year2 7
4.2 Gender Differences Across Course Parts 8
4.2.1 Year1 8
4.2.2 Year2 9

5 Background Differences (Category) **10**
5.1 Visualizing Scores 11
5.1.1 Year1 11
5.1.2 Year2 11
5.2 Background Differences Across Course Parts 12
5.2.1 Year1 12
5.2.2 Year2 13

6 Grouping Flipped and Post-Flipped Part **15**
6.1 Preparing the Data 15
6.2 Visualizing Gender Gap (Pre-Flipped vs. Post-Flipped) 16
6.2.1 Year1 16
6.2.2 Year2 16
6.3 Computing the Learning Gain 17
6.3.1 Year1 17
6.3.2 Year2 18
6.4 Learning Gain across Condition 18
6.4.1 Year1 18
6.4.2 Year2 19

7 Gender Differences on Aggregated Data **19**
7.1 Visualizing Scores 19
7.1.1 Year1 19
7.1.2 Year2 21

8 Learning Gain and Gender Differences **22**
8.1 Year1 22
8.1.1 Visualization 22
8.1.2 ANOVA – Gender as IV 23
8.1.3 ANOVA – Condition as IV 24
8.2 Year2 24
8.2.1 Visualization 24
8.2.2 ANOVA – Gender as IV 25
8.2.3 ANOVA – Condition as IV 26

9 Category Differences on Aggregated Data **26**
9.1 Visualizing Scores 26
9.1.1 Year1 26
9.1.2 Year2 29

10 Learning Gain and Category Differences **31**
10.1 Year1 . . . 31
10.1.1 Visualization . . . 31
10.1.2 ANOVA – Category as IV . . . 32
10.1.3 ANOVA – Condition as IV . . . 33
10.2 Year2 . . . 34
10.2.1 Visualization . . . 34
10.2.2 ANOVA – Category as IV . . . 34
10.2.3 ANOVA – Condition as IV . . . 35

11 Influence of Gender, Condition, and Category (Chi-Square and Linear Regression) **36**
11.1 Year1 . . . 36
11.1.1 Chi-Square Test of Independence . . . 36
11.1.2 Linear Regression . . . 38
11.2 Year2 . . . 40
11.2.1 Chi Square Test of Independence . . . 40
11.2.2 Linear Regression . . . 42

12 Analysis of Students' Trajectory (Drop-outs / MAN / Passed) **44**
12.0.1 Read IS-Academia Data . . . 44
12.1 Year1 . . . 46
12.1.1 Students who Passed . . . 46
12.1.2 Students who Failed First Semester + Drop-Outs . . . 49
12.1.3 Students who went to MAN . . . 50
12.1.4 Students who Dropped-Out of MAN . . . 52
12.2 Alluvial Diagram – Year1 . . . 52
12.2.1 Preparing the Data . . . 52
12.2.2 Combining Data . . . 53
12.2.3 Visualization : Alluvial Plot . . . 53
12.3 Year2 . . . 55
12.3.1 Students who Passed . . . 56
12.3.2 Students who Dropped their studies after 1st Semester . . . 58
12.3.3 Students who went to MAN . . . 58
12.4 Alluvial Diagram – Year2 . . . 59
12.4.1 Preparing the Data . . . 59
12.4.2 Combining Data . . . 59
12.4.3 Visualization : Alluvial Plot . . . 60

13 MAN Students – Rate of Failure across Condition **61**
13.1 Year1 . . . 61
13.1.1 Filtering out Students who went to MAN . . . 61
13.1.2 Filtering out Students who Dropped Out from MAN . . . 61
13.1.3 Filtering out Students who Failed from MAN . . . 62
13.1.4 Analyzing MAN Dropouts . . . 63
13.1.5 Analyzing MAN Failures . . . 63

14 Students at the Margin – Rate of Failures/Drop-outs **64**
14.1 Year1 . . . 64
14.1.1 Filtering BA1 Students Who Took Exams . . . 64
14.1.2 Students Who Passed/Failed BA1 . . . 65
14.1.3 Computing Mean Grade of BA1 studens . . . 66
14.1.4 Filtering the BA1 Students at the Margin . . . 66
14.1.5 Filtering the Students who Passed/Failed BA2 . . . 67
14.1.6 Identifying Risky Students who Passed/Failed in BA2 . . . 67

15 Disconnect **69**

1 Background and Rationale

Analysis in a previous project "`Score-Analysis.Rnw`" revealed that different populations at school behave differently in response to the teaching in the Linear Algebra course (both `Control` and `Flipped` conditions). One possible explanation for these observed differences in behavior and performance can be attributed to the different backgrounds of the different population Categories – French, Swiss-PAM, etc. The background can be considered as a confounding variable.

Also, the repeating students – who have been previously exposed to the Linear Algebra course – could bring more variability and noise to the dataset. Therefore, in this project, we will conduct the analysis only on the set of **New** students who have freshly enrolled in the semester.

2 R Package Imports

In this section, we will import all the required packages for importing, cleaning, and pre-processing the data.

```
library(readxl)
library(dplyr)
library(tidyr)
library(ggplot2)
library(scales)
library(gridExtra)
library(gplots)
library(RColorBrewer)
library(FactoMineR)
library(factoextra)
library(nlme)
library(rcompanion)
library(here)
library(RMariaDB)
library(keyringr)
library(ggalluvial)
library(wesanderson)
library(ggpubr)
```

3 Data Import and Pre-Processing

3.0.1 Data Import

First, we will load the data which we have previously cleand and normalized in "`Score-Analysis.Rnw`" project :

```
# Setting the path.
path = paste(here(), "/Data/Scores/Normalized-Volunteer-Data/", sep = "")

# Year1
dt.y1 = read.csv(paste(path, "Year1-Normalized-Score-DI-Filtered.csv", sep = ""), header = TRUE)
dt.y1$X = NULL

# Year2
dt.y2 = read.csv(paste(path, "Year2-Normalized-Score-DI-Filtered.csv", sep = ""), header = TRUE)
dt.y2$X = NULL

# Convert ID.Anon to Character
dt.y1$ID.Anon = as.character(dt.y1$ID.Anon)
dt.y2$ID.Anon = as.character(dt.y2$ID.Anon)

# Clean-up the path variable.
rm(path)
```

3.0.2 Data Summary

Secondly, we will produce a small summary of data, i.e. how many students are new and how many are repeating.

```r
# Summarize.
# Year1
dt.y1 %>% group_by(Code.BA, Condition) %>%
  summarise(N = n()) %>%
  spread(Condition, N)
```

```
## # A tibble: 4 x 3
## # Groups:   Code.BA [4]
##   Code.BA   Control Flipped
##   <fct>       <int>   <int>
## 1 Ex-CMS         10      NA
## 2 Ex-MAN         36       8
## 3 New           302      75
## 4 Repeating      34       6
```

... also for Year2

```r
# Summarize.
# Year2
dt.y2 %>% group_by(Code.BA, Condition) %>%
  summarise(N = n()) %>%
  spread(Condition, N)
```

```
## # A tibble: 4 x 3
## # Groups:   Code.BA [4]
##   Code.BA   Control Flipped
##   <fct>       <int>   <int>
## 1 Ex-CMS          3       4
## 2 Ex-MAN         15      18
## 3 New           100     102
## 4 Repeating       8      10
```

3.0.3 Data Filtering

Thirdly, we will filter out only the **New** students.

```r
# Year1
dt.y1 = dt.y1 %>% filter(Code.BA == "New")

# Year2
dt.y2 = dt.y2 %>% filter(Code.BA == "New")
```

... We will also filter out the Category of **Etranger.Autres** because this is one category with very small nunmber of students and could add noise to our analysis.

```r
# Filter out Etranger.Autres
# Year1
dt.y1 = dt.y1 %>% filter(!(Category == "Etranger.Autres"))

# Year2
dt.y2 = dt.y2 %>% filter(!(Category == "Etranger.Autres"))
```

With all this filtering, we have in total 351 rows for Year1 and 196 rows for Year2.
... some summary.

```r
# Year 1
t.stat = dt.y1 %>% group_by(Condition) %>%
  summarise(N = n(),
            m = mean(Nor.Score),
            sd = sd(Nor.Score))
t.stat
```

```
## # A tibble: 2 x 4
##   Condition     N       m    sd
##   <fct>     <int>   <dbl> <dbl>
## 1 Control     281 -0.118  0.982
## 2 Flipped      70 -0.0701 0.939
```

```
t.stat = dt.y1 %>% group_by(Condition) %>%
  summarise(N = n(),
            m = mean(Total.Score),
            sd = sd(Total.Score))
t.stat
```

```
## # A tibble: 2 x 4
##   Condition     N     m    sd
##   <fct>     <int> <dbl> <dbl>
## 1 Control     281  31.0  15.6
## 2 Flipped      70  31.7  15.0
```

```
# T-Test
t.test(dt.y1$Total.Score~dt.y1$Condition)
```

```
##
##  Welch Two Sample t-test
##
## data:  dt.y1$Total.Score by dt.y1$Condition
## t = -0.37557, df = 109.67, p-value = 0.708
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.751855  3.237772
## sample estimates:
## mean in group Control mean in group Flipped
##              30.97153              31.72857
```

```
# Year 2
t.stat = dt.y2 %>% group_by(Condition) %>%
  summarise(N = n(),
            m = mean(Nor.Score),
            sd = sd(Nor.Score))
t.stat
```

```
## # A tibble: 2 x 4
##   Condition     N      m    sd
##   <fct>     <int>  <dbl> <dbl>
## 1 Control      97 -0.191 1.01
## 2 Flipped      99 -0.269 0.951
```

```
t.stat = dt.y2 %>% group_by(Condition) %>%
  summarise(N = n(),
            m = mean(Total.Score),
            sd = sd(Total.Score))
t.stat
```

```
## # A tibble: 2 x 4
##   Condition     N     m    sd
##   <fct>     <int> <dbl> <dbl>
## 1 Control      97  33.2  17.5
## 2 Flipped      99  31.9  16.5
```

```
# T-Test
t.test(dt.y2$Total.Score~dt.y2$Condition)
```

```
##
##  Welch Two Sample t-test
##
## data:  dt.y2$Total.Score by dt.y2$Condition
```

```
## t = 0.55408, df = 192.77, p-value = 0.5802
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.450507  6.146539
## sample estimates:
## mean in group Control mean in group Flipped
##              33.22680              31.87879
```

3.0.4 Data For Visualization

YEAR1 :

```
# Gathering the score variables
temp.y1 = gather(dt.y1,
              "Nor.Score.A", "Nor.Score.B", "Nor.Score.C",
              key = "Course.Parts",
              value = "Score.Parts")
```

YEAR2 :

```
# Gathering the score variables
temp.y2 = gather(dt.y2,
              "Nor.Score.A", "Nor.Score.B", "Nor.Score.C",
              key = "Course.Parts",
              value = "Score.Parts")
```

Now, that the data is ready, we can start analyzing it.

4 Gender Differences

Let us summarize the data first, for YEAR1 :

```
# Year1
dt.y1 %>% group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  spread(Gender, N)

## # A tibble: 2 x 3
## # Groups:   Condition [2]
##   Condition     F     M
##   <fct>     <int> <int>
## 1 Control      92   189
## 2 Flipped      23    47
```

... also for YEAR2 :

```
# Year2
dt.y2 %>% group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  spread(Gender, N)

## # A tibble: 2 x 3
## # Groups:   Condition [2]
##   Condition     F     M
##   <fct>     <int> <int>
## 1 Control      35    62
## 2 Flipped      35    64
```

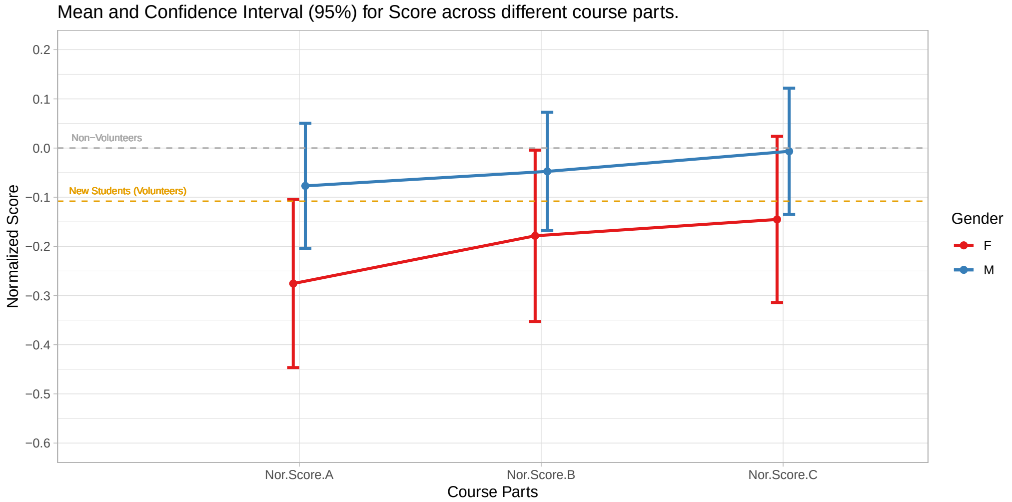
4.1 Visualizing Scores

Before we continue with the visualizations, we have to compute the **mean** value of the scores :

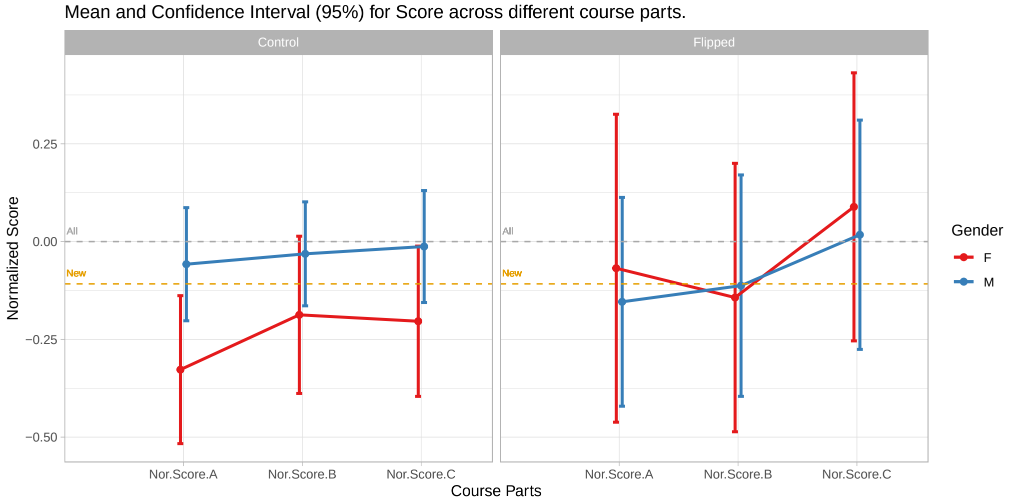
```r
# Year1
y1.mean = mean(dt.y1$Nor.Score)

# Year2
y2.mean = mean(dt.y2$Nor.Score)
```

4.1.1 Year1

Gender differences for all the volunteers :



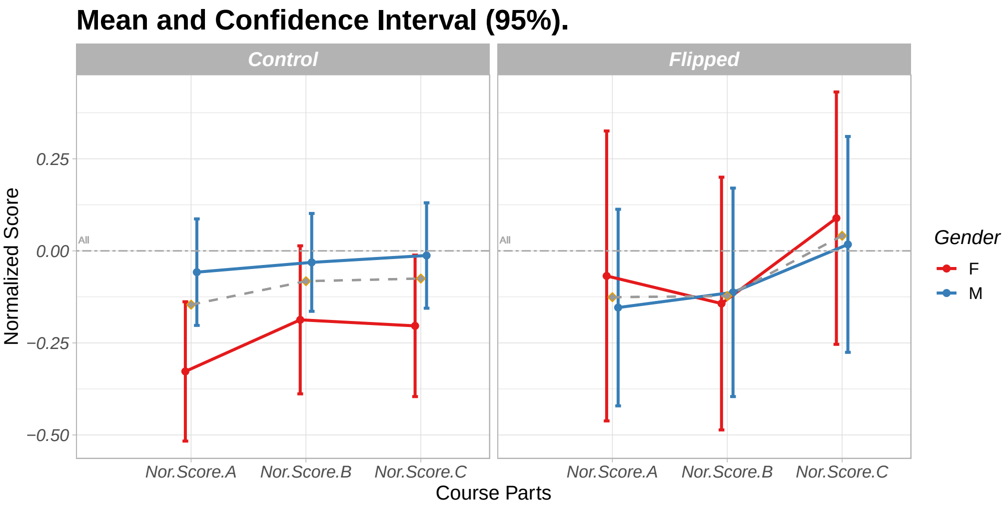
Gender differences across the Condition :



Summary :

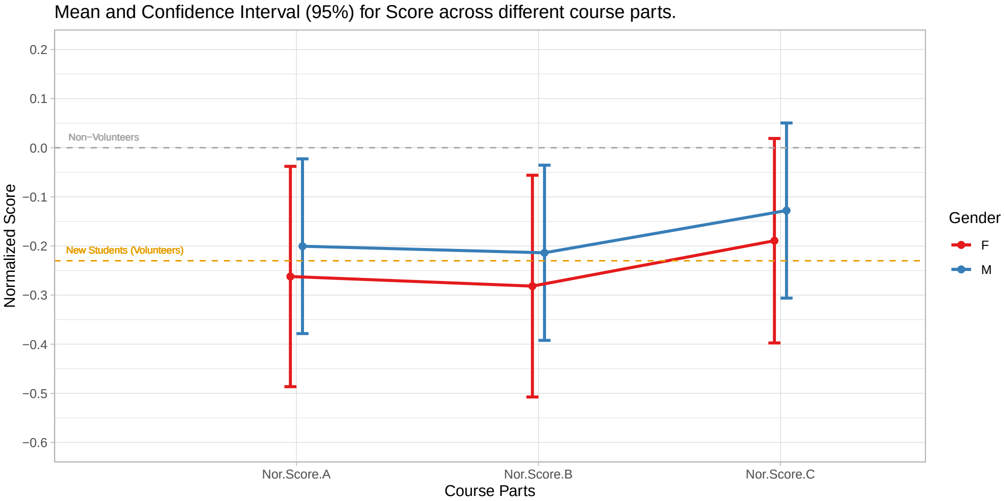
```r
# Table
dt.y1 %>% group_by(Condition, Gender) %>%
  summarise(N = n())

## # A tibble: 4 x 3
## # Groups:   Condition [2]
##   Condition Gender     N
##   <fct>     <fct>  <int>
## 1 Control   F         92
## 2 Control   M        189
## 3 Flipped   F         23
## 4 Flipped   M         47
```

Gender differences across the Condition (with weighted mean) :

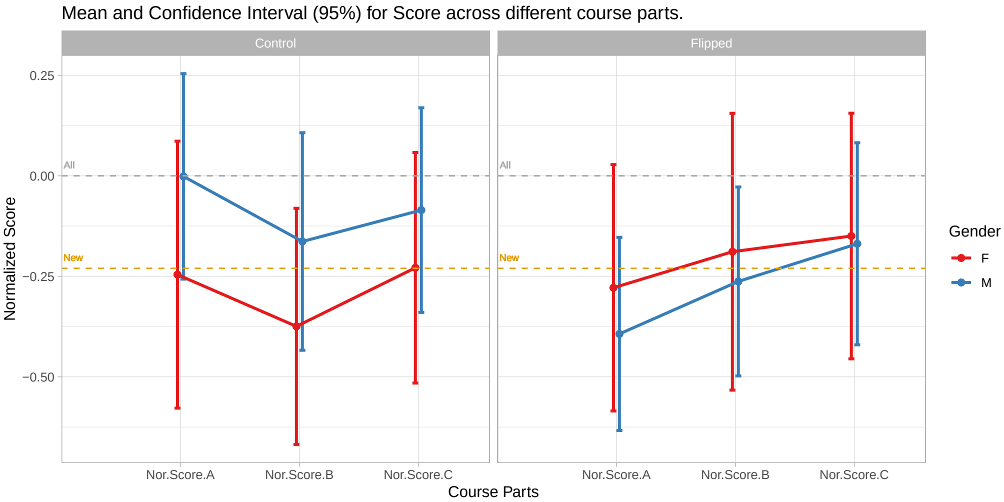


4.1.2 Year2

Gender differences for all the volunteers :



Gender differences across the Condition :

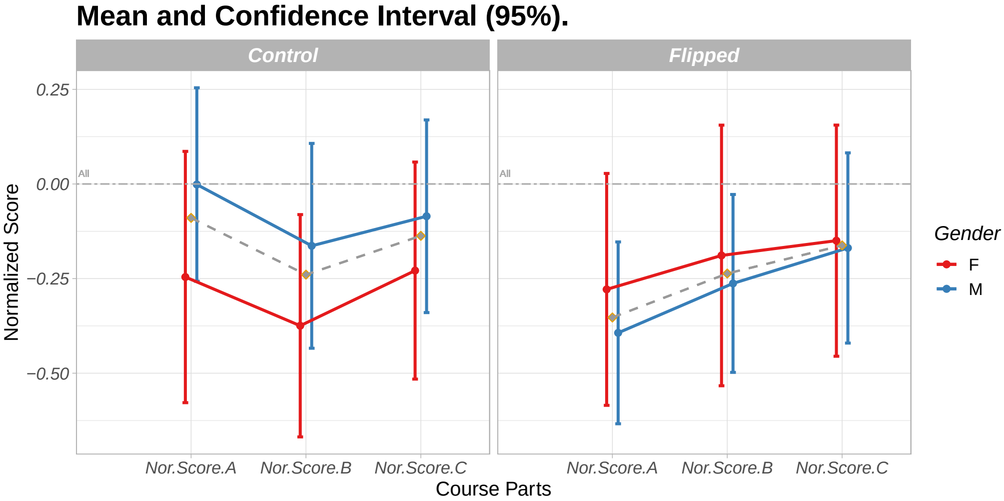


Summary :

```
# Table
dt.y2 %>% group_by(Condition, Gender) %>%
  summarise(N = n())

## # A tibble: 4 x 3
## # Groups:   Condition [2]
##   Condition Gender     N
##   <fct>     <fct>  <int>
```

```
## 1 Control   F          35
## 2 Control   M          62
## 3 Flipped   F          35
## 4 Flipped   M          64
```

Gender differences across the Condition (with weighted mean) :



4.2 Gender Differences Across Course Parts

Now, we will examine if there is a statistical difference between `Gender` across the different `Course.Parts`

4.2.1 Year1

Control Condition :

```
# Data for analysis
t.stat = dt.y1 %>% filter(Condition == "Control")

# Part A
oneway.test(t.stat$Nor.Score.A~t.stat$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.A and t.stat$Gender
## F = 4.9238, num df = 1.00, denom df = 196.07, p-value = 0.02764

# Part B
oneway.test(t.stat$Nor.Score.B~t.stat$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.B and t.stat$Gender
## F = 1.6098, num df = 1.00, denom df = 172.05, p-value = 0.2062

# Part C
oneway.test(t.stat$Nor.Score.C~t.stat$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.C and t.stat$Gender
## F = 2.4428, num df = 1.00, denom df = 191.55, p-value = 0.1197
```

Flipped Condition :

```r
# Data for analysis
t.stat = dt.y1 %>% filter(Condition == "Flipped")

# Part A
oneway.test(t.stat$Nor.Score.A~t.stat$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.A and t.stat$Gender
## F = 0.12521, num df = 1.000, denom df = 42.573, p-value = 0.7252

# Part B
oneway.test(t.stat$Nor.Score.B~t.stat$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.B and t.stat$Gender
## F = 0.017979, num df = 1.000, denom df = 50.865, p-value = 0.8939

# Part C
oneway.test(t.stat$Nor.Score.C~t.stat$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.C and t.stat$Gender
## F = 0.096138, num df = 1.000, denom df = 52.523, p-value = 0.7577
```

4.2.2 Year2

Control Condition :

```r
# Data for analysis
t.stat = dt.y2 %>% filter(Condition == "Control")

# Part A
oneway.test(t.stat$Nor.Score.A~t.stat$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.A and t.stat$Gender
## F = 1.3076, num df = 1.000, denom df = 72.122, p-value = 0.2566

# Part B
oneway.test(t.stat$Nor.Score.B~t.stat$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.B and t.stat$Gender
## F = 1.074, num df = 1.000, denom df = 82.927, p-value = 0.3031

# Part C
oneway.test(t.stat$Nor.Score.C~t.stat$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.C and t.stat$Gender
## F = 0.5381, num df = 1.000, denom df = 80.752, p-value = 0.4653
```

Flipped Condition :

```
# Data for analysis
t.stat = dt.y2 %>% filter(Condition == "Flipped")

# Part A
oneway.test(t.stat$Nor.Score.A~t.stat$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.A and t.stat$Gender
## F = 0.33415, num df = 1.000, denom df = 73.635, p-value = 0.565

# Part B
oneway.test(t.stat$Nor.Score.B~t.stat$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.B and t.stat$Gender
## F = 0.12076, num df = 1.000, denom df = 65.402, p-value = 0.7293

# Part C
oneway.test(t.stat$Nor.Score.C~t.stat$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.C and t.stat$Gender
## F = 0.0091348, num df = 1.000, denom df = 76.657, p-value = 0.9241
```

5 Background Differences (Category)

Let us summarize the data first, for Year1 :

```
# Year1
dt.y1 %>% group_by(Category, Condition) %>%
  summarise(N = n()) %>%
  spread(Condition, N)

## # A tibble: 3 x 3
## # Groups:   Category [4]
##   Category      Control Flipped
##   <fct>           <int>   <int>
## 1 France            168      39
## 2 Suisse.Autres      54      16
## 3 Suisse.PAM         59      15
```

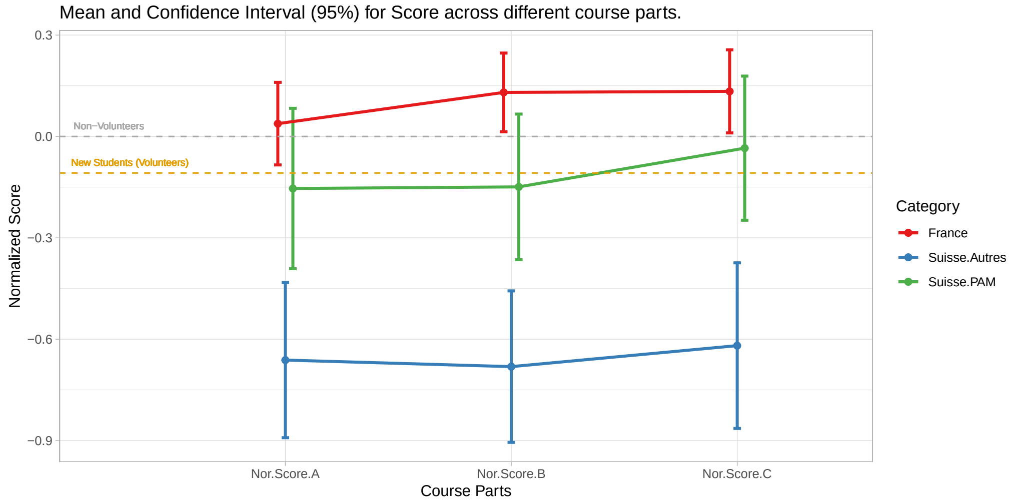
... also for Year2 :

```
# Year2
dt.y2 %>% group_by(Category, Condition) %>%
  summarise(N = n()) %>%
  spread(Condition, N)

## # A tibble: 3 x 3
## # Groups:   Category [4]
##   Category      Control Flipped
##   <fct>           <int>   <int>
## 1 France             53      50
## 2 Suisse.Autres      20      25
## 3 Suisse.PAM         24      24
```

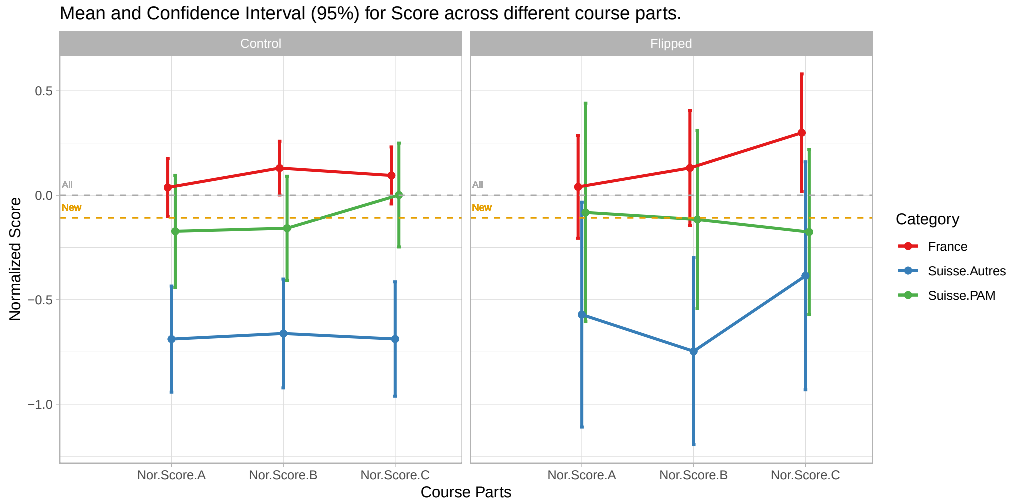
5.1 Visualizing Scores

5.1.1 Year1

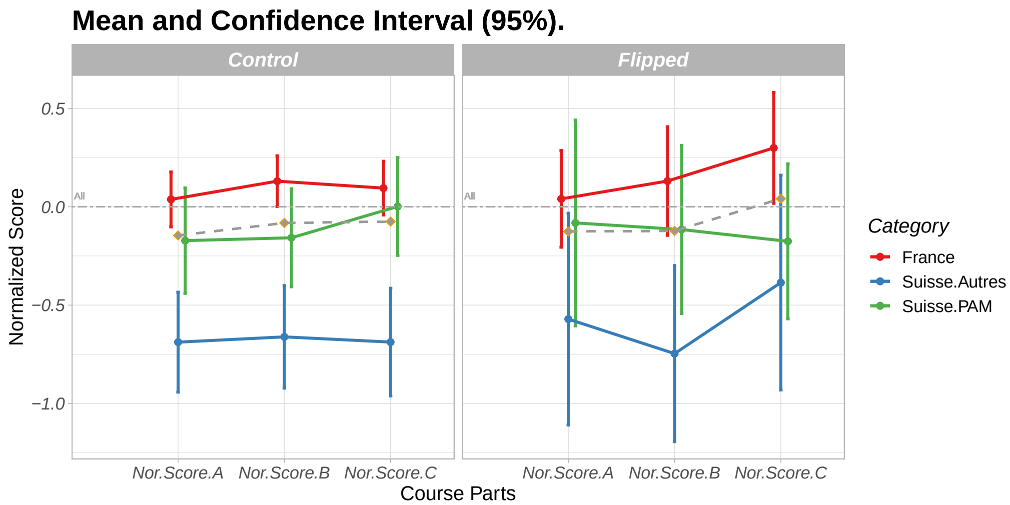
Category differences for all the volunteers :



Category differences across the Condition :

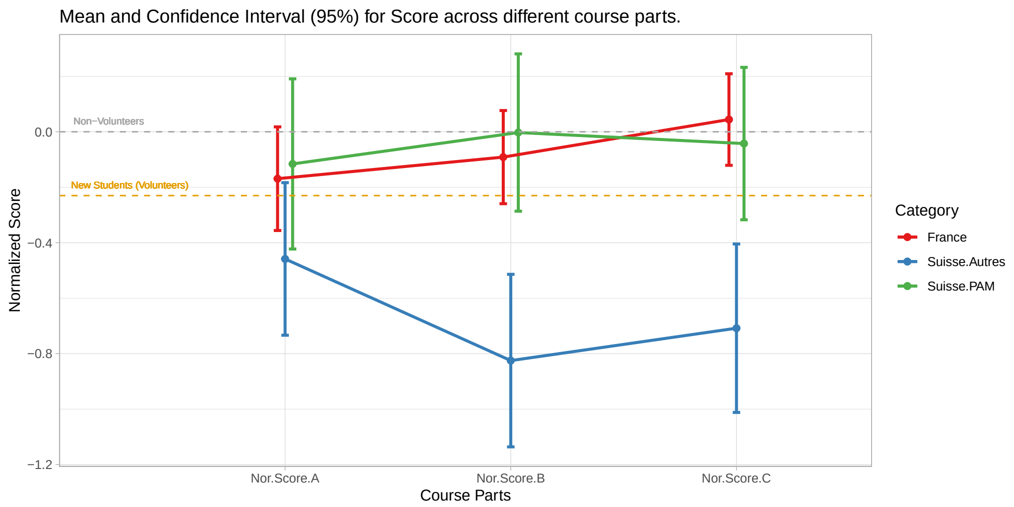


Category differences across the Condition (with weighted mean) :

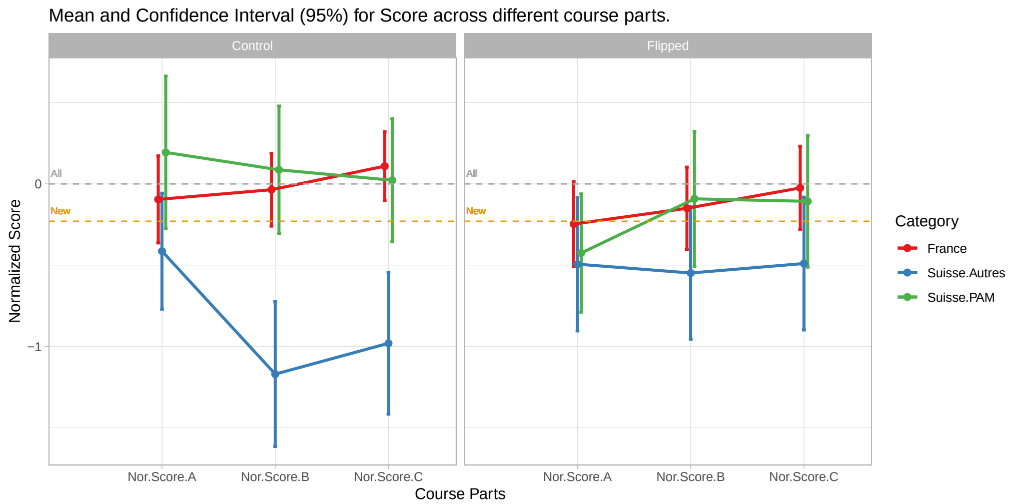


5.1.2 Year2

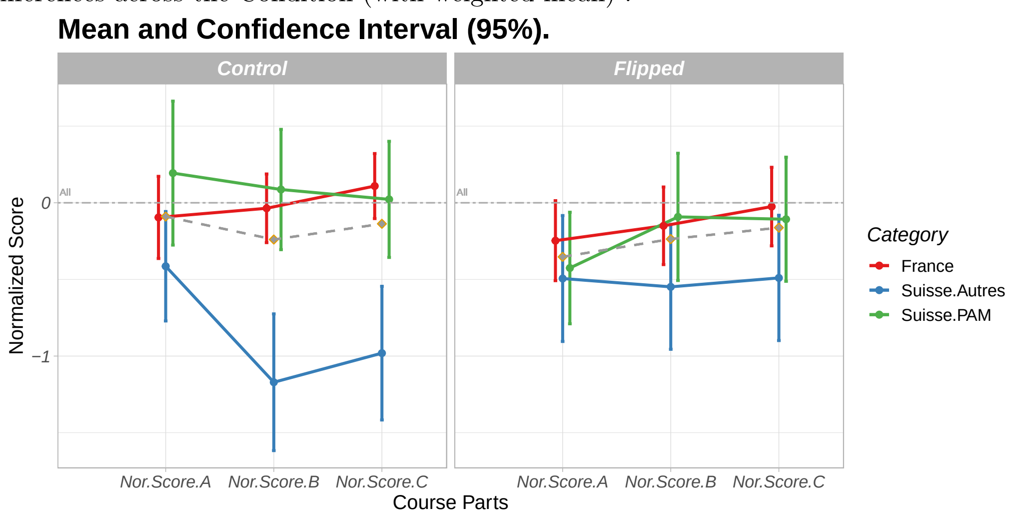
Category differences for all the volunteers :



Category differences across the Condition :



Category differences across the Condition (with weighted mean) :



5.2 Background Differences Across Course Parts

Now, we will examine if there is statistical differences between different `Categories` across the `Course.Parts` :

5.2.1 Year1

Control Condition :

```
# Data for analysis
t.stat = dt.y1 %>% filter(Condition == "Control")

# Part A
```

```r
oneway.test(t.stat$Nor.Score.A~t.stat$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.A and t.stat$Category
## F = 12.014, num df = 2, denom df = 107, p-value = 1.964e-05

# Part B
oneway.test(t.stat$Nor.Score.B~t.stat$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.B and t.stat$Category
## F = 14.611, num df = 2.00, denom df = 104.03, p-value = 2.557e-06

# Part C
oneway.test(t.stat$Nor.Score.C~t.stat$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.C and t.stat$Category
## F = 12.615, num df = 2.00, denom df = 105.63, p-value = 1.222e-05
```

Flipped Condition :

```r
# Data for analysis
t.stat = dt.y1 %>% filter(Condition == "Flipped")

# Part A
oneway.test(t.stat$Nor.Score.A~t.stat$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.A and t.stat$Category
## F = 2.0011, num df = 2.000, denom df = 26.115, p-value = 0.1554

# Part B
oneway.test(t.stat$Nor.Score.B~t.stat$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.B and t.stat$Category
## F = 5.2339, num df = 2.000, denom df = 30.054, p-value = 0.01121

# Part C
oneway.test(t.stat$Nor.Score.C~t.stat$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.C and t.stat$Category
## F = 3.2824, num df = 2.000, denom df = 29.807, p-value = 0.05148
```

5.2.2 Year2

Control Condition :

```r
# Data for analysis
t.stat = dt.y2 %>% filter(Condition == "Control")

# Part A
oneway.test(t.stat$Nor.Score.A~t.stat$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.A and t.stat$Category
## F = 2.1246, num df = 2.000, denom df = 44.476, p-value = 0.1314

# Part B
oneway.test(t.stat$Nor.Score.B~t.stat$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.B and t.stat$Category
## F = 10.974, num df = 2.000, denom df = 39.726, p-value = 0.0001605

# Part C
oneway.test(t.stat$Nor.Score.C~t.stat$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.C and t.stat$Category
## F = 9.7029, num df = 2.000, denom df = 39.281, p-value = 0.0003763
```

Flipped Condition :

```r
# Data for analysis
t.stat = dt.y2 %>% filter(Condition == "Flipped")

# Part A
oneway.test(t.stat$Nor.Score.A~t.stat$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.A and t.stat$Category
## F = 0.60892, num df = 2.000, denom df = 49.432, p-value = 0.548

# Part B
oneway.test(t.stat$Nor.Score.B~t.stat$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.B and t.stat$Category
## F = 1.5552, num df = 2.000, denom df = 47.454, p-value = 0.2217

# Part C
oneway.test(t.stat$Nor.Score.C~t.stat$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Nor.Score.C and t.stat$Category
## F = 1.7859, num df = 2.000, denom df = 47.939, p-value = 0.1786
```

6 Grouping Flipped and Post-Flipped Part

The experimental design for both YEAR1 and YEAR2 has parts which are *pre-flipped*, *flipped*, and *post-flipped* (only for YEAR1). Owing to this organization, maybe, it is not so wise to examine the scores of the `Course.Part` which has not been flipped.

Therefore, we will combine the score of only the flipped and post-flipped parts to examine Gender and Background differences.

6.1 Preparing the Data

YEAR1 :

```
# Adding the scores of B and C parts.
#dt.y1$Nor.Score.BC = dt.y1$Nor.Score.B + dt.y1$Nor.Score.C

# ... and computing the mean.
y1.mean = mean(dt.y1$Nor.Score.BC)
y1.mean.ABC = mean(dt.y1$Nor.Score)
```

Also, gathering to prepare new data :

```
# Gathering the score variables
temp.y1.pre.post = gather(dt.y1, "Nor.Score.A", "Nor.Score.BC",
                          key = "Pre.Post",
                          value = "Score.Parts")
```

... and summarizing :

```
dt.y1 %>% group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  spread(Gender, N)

## # A tibble: 2 x 3
## # Groups:   Condition [2]
##   Condition     F     M
##   <fct>     <int> <int>
## 1 Control      92   189
## 2 Flipped      23    47
```

YEAR2 :

```
# Adding the scores of B and C parts.
#dt.y2$Nor.Score.BC = dt.y2$Nor.Score.B + dt.y2$Nor.Score.C

# ... and computing the mean.
y2.mean = mean(dt.y2$Nor.Score.BC)
y2.mean.ABC = mean(dt.y2$Nor.Score)
```

Also, gathering to prepare new data :

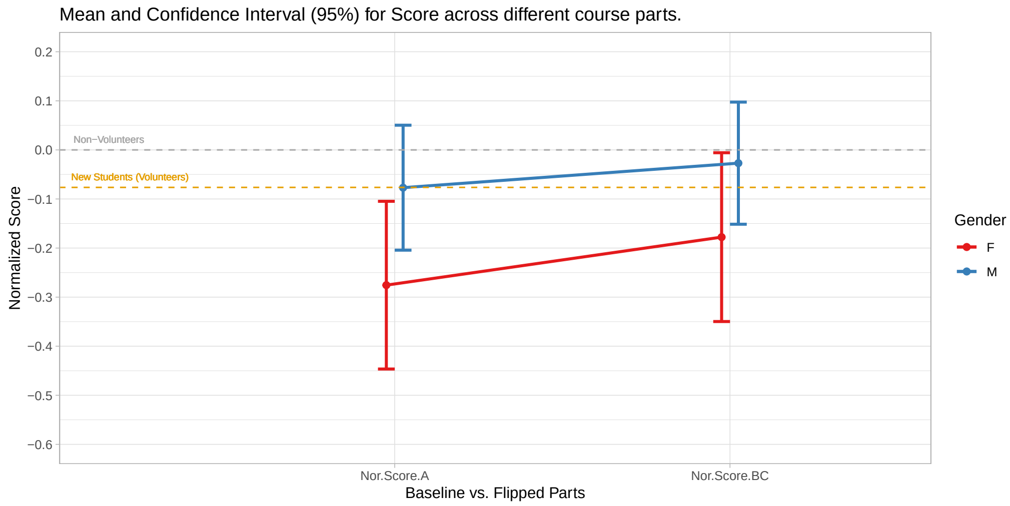
```
# Gathering the score variables
temp.y2.pre.post = gather(dt.y2, "Nor.Score.A", "Nor.Score.BC",
                          key = "Pre.Post",
                          value = "Score.Parts")
```

... and summarizing :

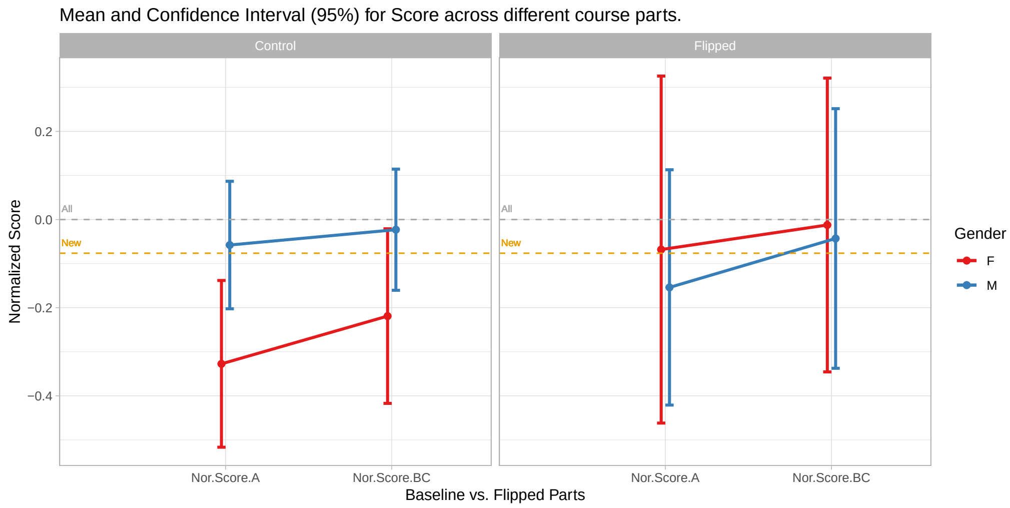
```
dt.y2 %>% group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  spread(Gender, N)

## # A tibble: 2 x 3
## # Groups:   Condition [2]
##   Condition     F     M
##   <fct>     <int> <int>
## 1 Control      35    62
## 2 Flipped      35    64
```

6.2 Visualizing Gender Gap (Pre-Flipped vs. Post-Flipped)

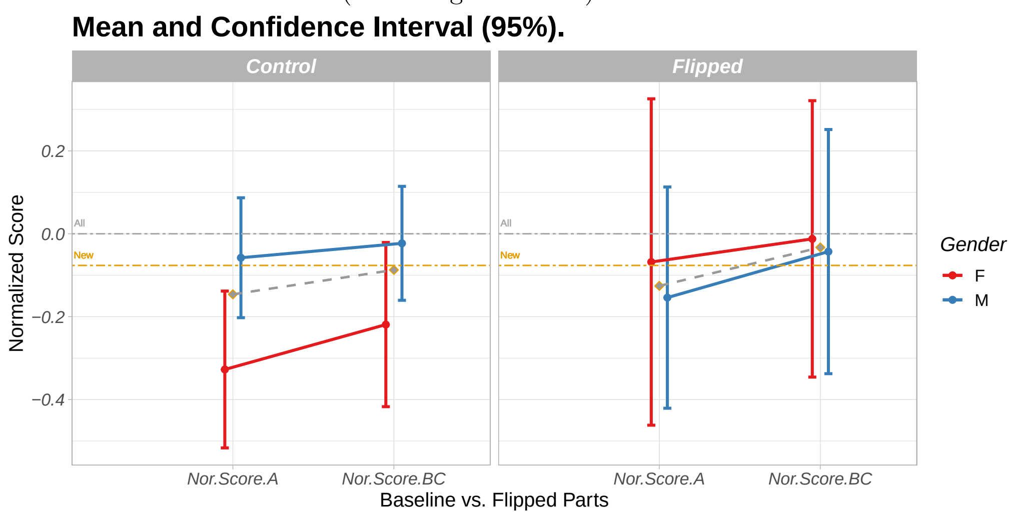
6.2.1 Year1



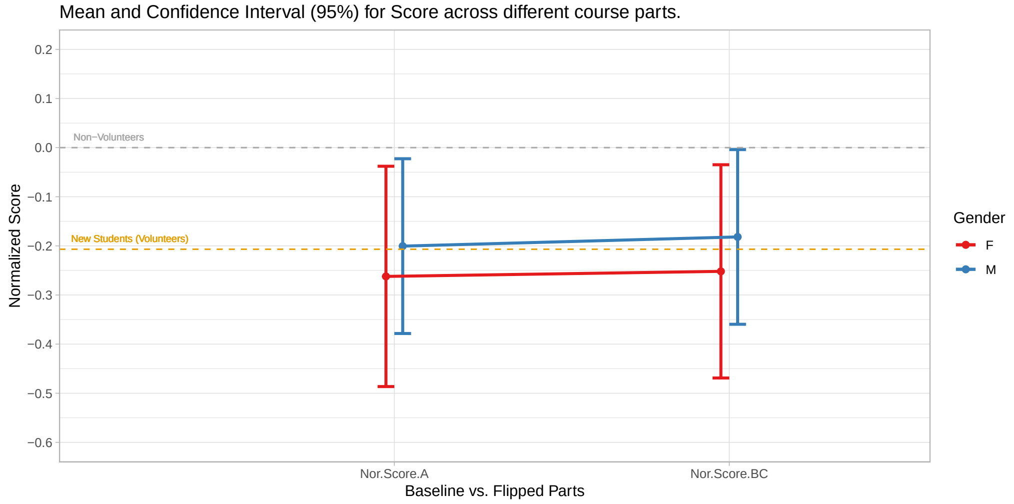
Gender differences across the Condition :



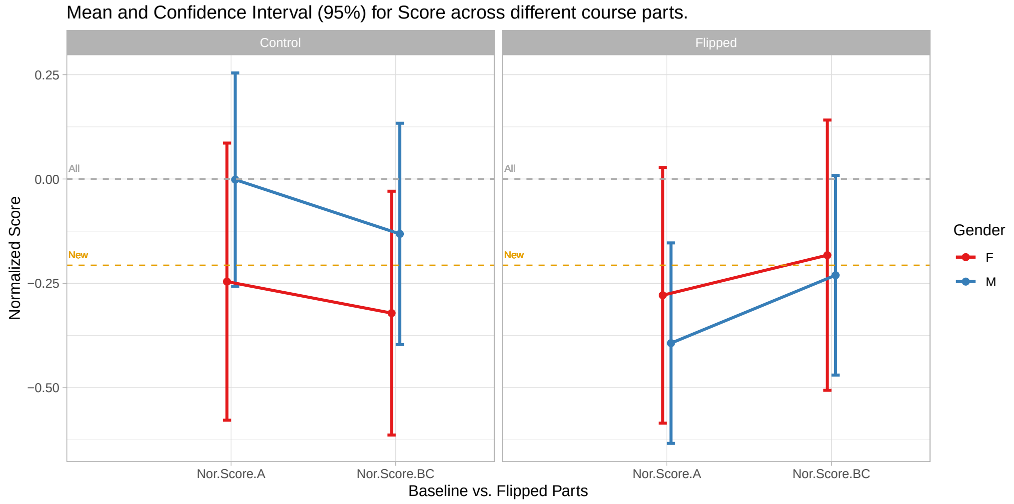
Gender differences across the Condition (with weighted mean) :



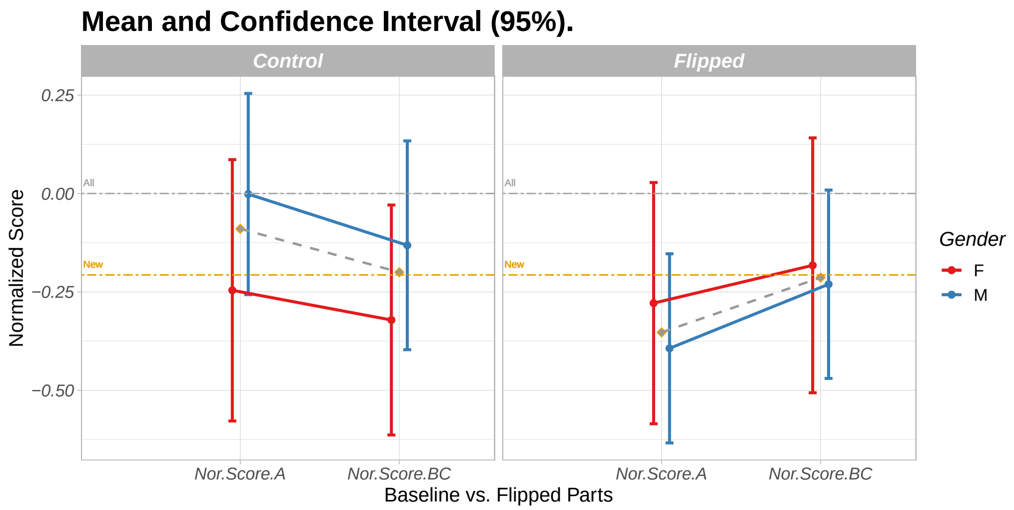
6.2.2 Year2



Gender differences across the Condition :



Gender differences across the Condition (with weighted mean) :



6.3 Computing the Learning Gain

In this section, we will compute the learning gain based on the difference between the *mean* score attained in Part A and the mean score in part BC.

6.3.1 Year1

The following list shows the number of questions in each part :

- **Part.A** 6 questions
- **Part.B** 6 questions
- **Part.C** 9 questions

We will use this information to compute the mean scores :

```
# Compute the mean pre-score
dt.y1$Mean.Pre.Score = dt.y1$Nor.Score.A

# Compute the mean post-score
dt.y1$Mean.Post.Score = dt.y1$Nor.Score.BC

# Compute the Learning Gain
dt.y1$Learning.Gain = dt.y1$Mean.Post.Score - dt.y1$Mean.Pre.Score
```

6.3.2 Year2

The following list shows the number of questions in each part :

– **Part.A** 4 questions
– **Part.B** 8 questions
– **Part.C** 10 questions

We will use this information to compute the mean scores :

```
# Compute the mean pre-score
dt.y2$Mean.Pre.Score = dt.y2$Nor.Score.A

# Compute the mean post-score
dt.y2$Mean.Post.Score = dt.y2$Nor.Score.BC

# Compute the Learning Gain
dt.y2$Learning.Gain = dt.y2$Mean.Post.Score - dt.y2$Mean.Pre.Score
```

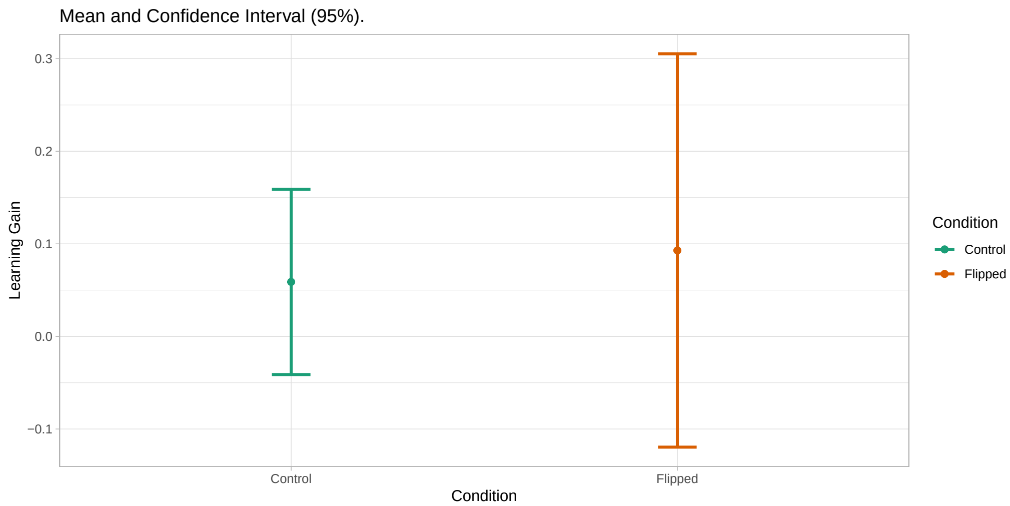
6.4 Learning Gain across Condition

In this section, we will analyse whether the learning gain differs significantly across the FLIPPED or the CONTROL condition :

6.4.1 Year1

```
# ANOVA
oneway.test(dt.y1$Learning.Gain ~ dt.y1$Condition)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  dt.y1$Learning.Gain and dt.y1$Condition
## F = 0.080491, num df = 1.00, denom df = 101.77, p-value = 0.7772
```

The results show **NO Statistically Signifiant** difference in `Learning.Gain`.

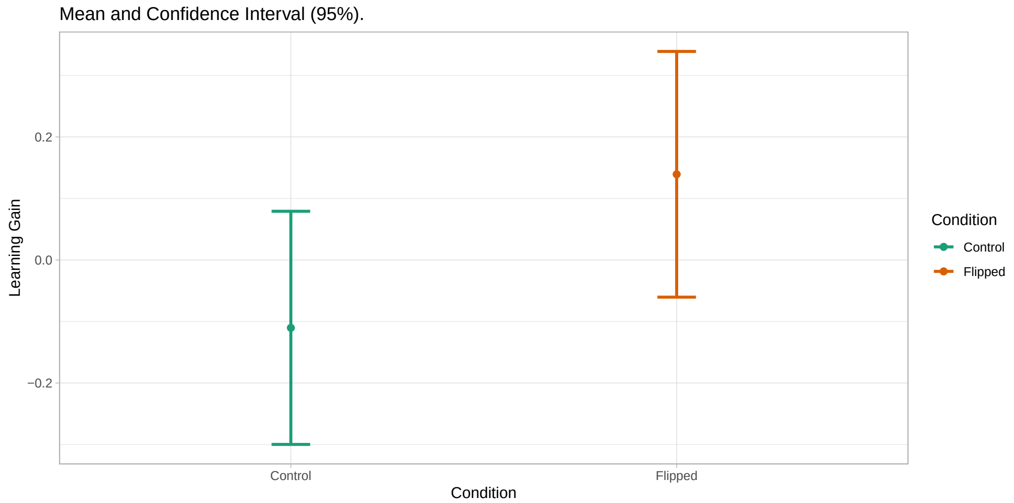


From the plot above, we see as well that the mean of `Learning.Gain` is almost the same across `Conditions`. In other words, students in both the conditions learn about same.

6.4.2 Year2

```
# ANOVA
oneway.test(dt.y2$Learning.Gain ~ dt.y2$Condition)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  dt.y2$Learning.Gain and dt.y2$Condition
## F = 3.1576, num df = 1.00, denom df = 193.66, p-value = 0.07714
```

The results show **Marginal Significant** difference in `Learning.Gain`.



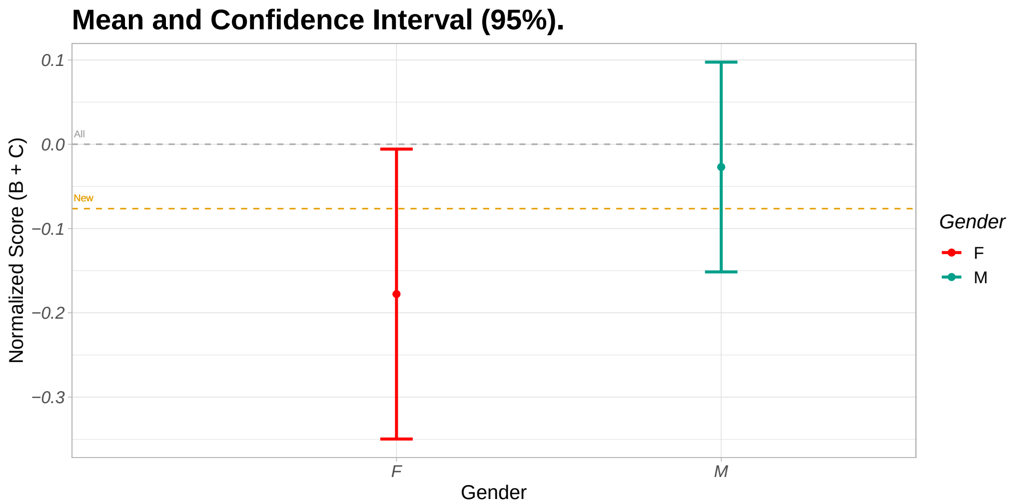
From the plot above, that the learning gain is **higher** in the FLIPPED condition as compared to the CONTROL condition.

7 Gender Differences on Aggregated Data

7.1 Visualizing Scores

7.1.1 Year1

Gender differences for all the volunteers :



... and the ANOVA :

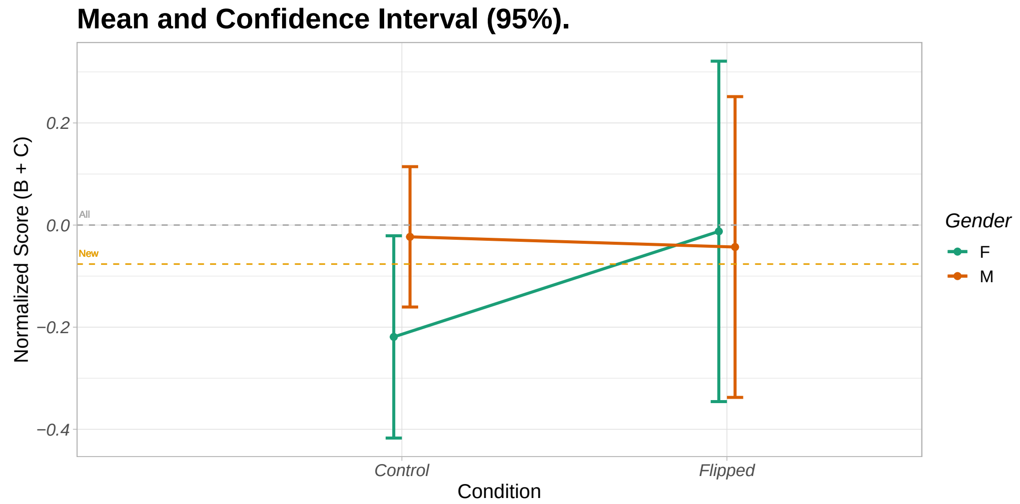
```
oneway.test(dt.y1$Nor.Score.BC~dt.y1$Gender)

##
```

```
##  One-way analysis of means (not assuming equal variances)
##
## data:  dt.y1$Nor.Score.BC and dt.y1$Gender
## F = 1.9365, num df = 1.00, denom df = 233.57, p-value = 0.1654
```

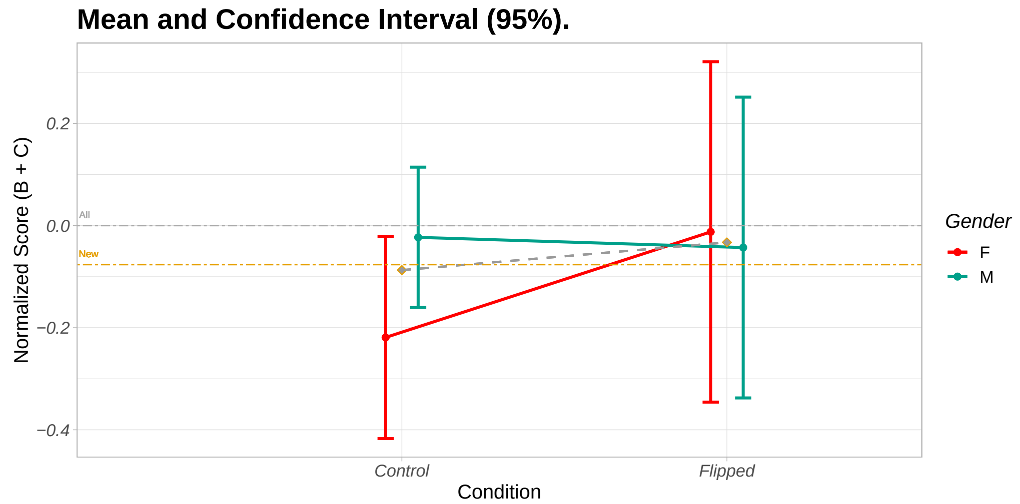
... and the Kruskal-Wallis :

```
kruskal.test(dt.y1$Nor.Score.BC~dt.y1$Gender)

##
##  Kruskal-Wallis rank sum test
##
## data:  dt.y1$Nor.Score.BC by dt.y1$Gender
## Kruskal-Wallis chi-squared = 2.133, df = 1, p-value = 0.1442

epsilonSquared(x = dt.y1$Nor.Score.BC,
               g = dt.y1$Gender)

## epsilon.squared
##         0.00609
```

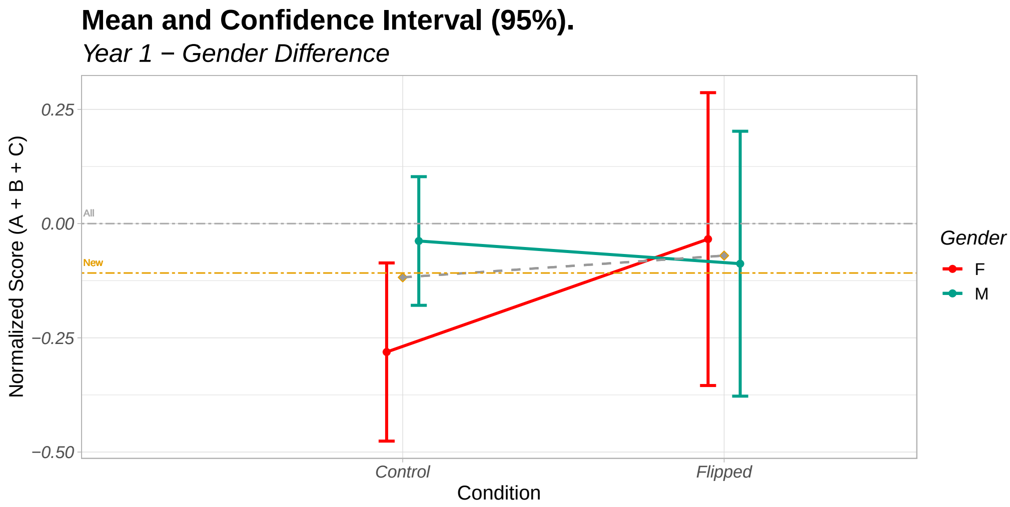
Gender differences across the Condition :



Gender differences across condition (with weighted mean) :

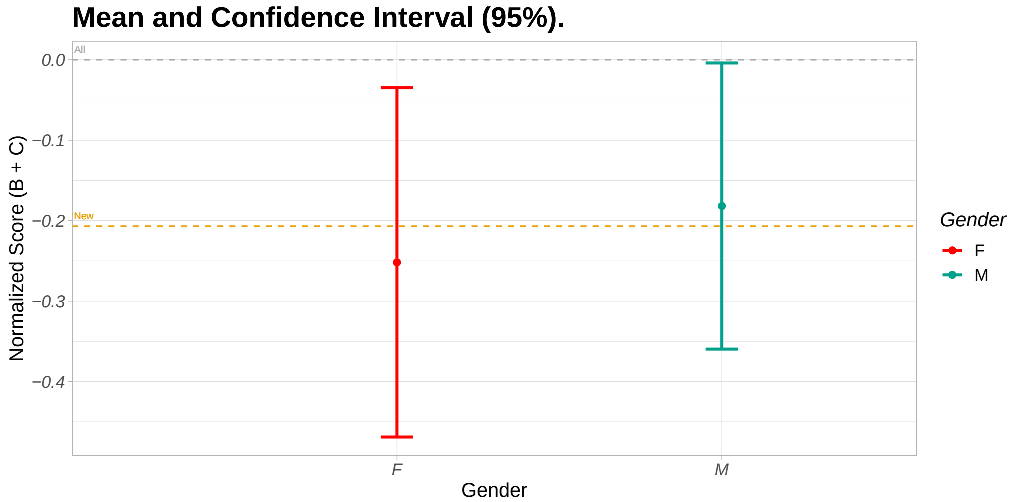


Gender differences across condition for **A+B+C** (with weighted mean) :



7.1.2 Year2

Gender differences for all the volunteers :



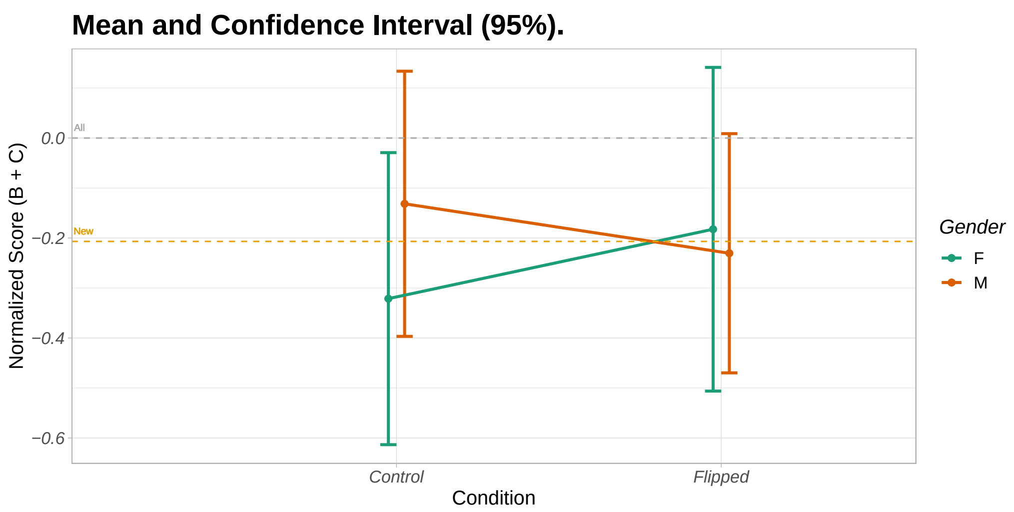
... and the ANOVA :

```
oneway.test(dt.y2$Nor.Score.BC~dt.y2$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  dt.y2$Nor.Score.BC and dt.y2$Gender
## F = 0.23961, num df = 1.00, denom df = 154.32, p-value = 0.6252
```

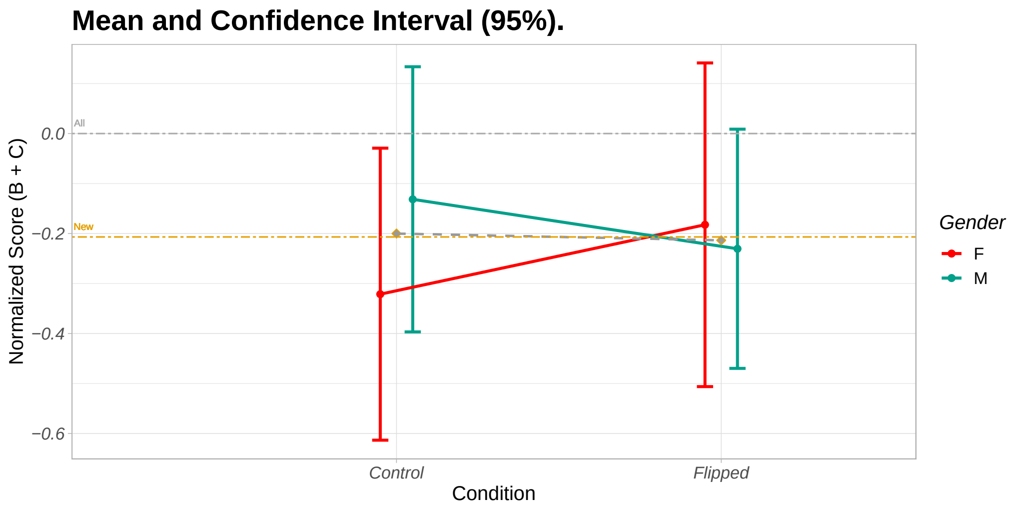
... and the Kruskal-Wallis :

```
kruskal.test(dt.y2$Nor.Score.BC~dt.y2$Gender)

##
##  Kruskal-Wallis rank sum test
##
## data:  dt.y2$Nor.Score.BC by dt.y2$Gender
## Kruskal-Wallis chi-squared = 0.46197, df = 1, p-value = 0.4967

epsilonSquared(x = dt.y2$Nor.Score.BC,
               g = dt.y2$Gender)

## epsilon.squared
##         0.00237
```

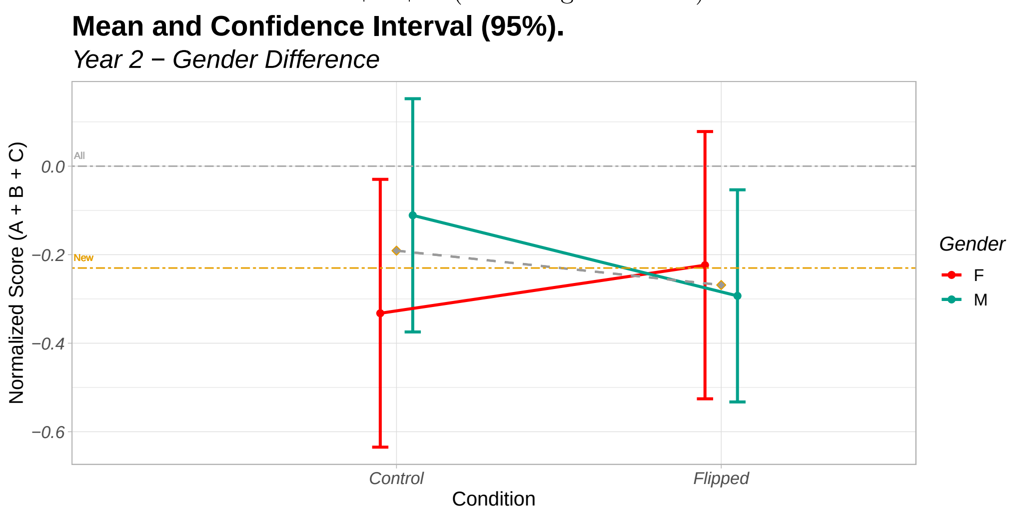
Gender differences across the Condition :



Gender differences across condition (with weighted mean) :



Gender differences across condition for **A+B+C** (with weighted mean) :



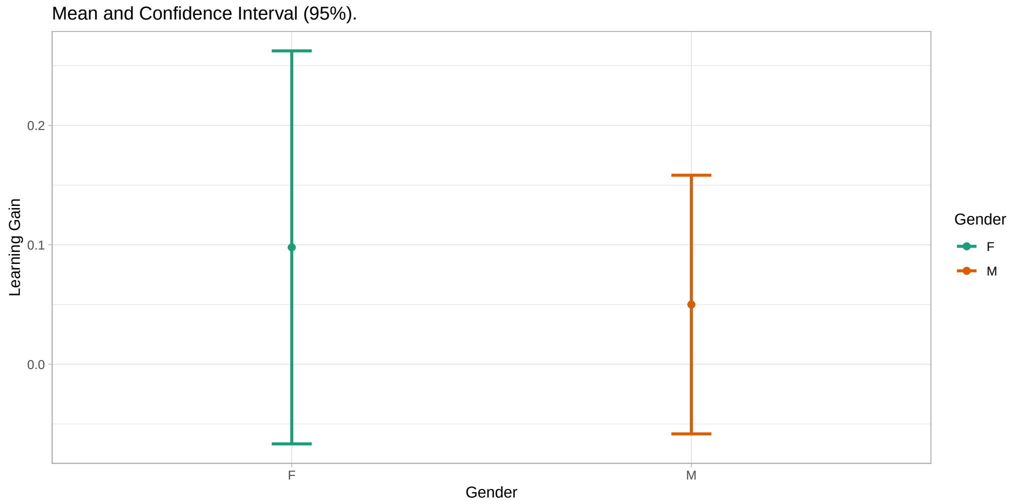
8 Learning Gain and Gender Differences

In this section, we will examine the differences in `Learning.Gain` across the `Gender`.

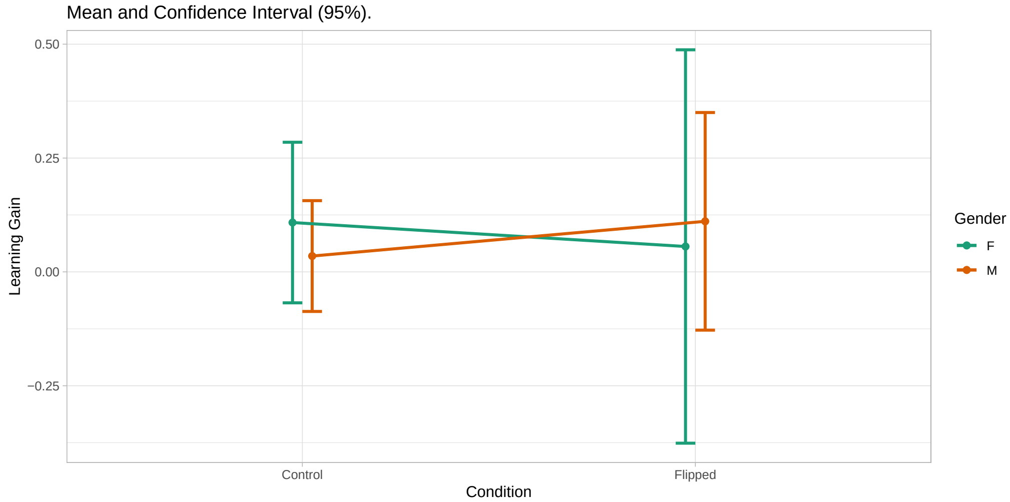
8.1 Year1

8.1.1 Visualization

Gender differences for all the volunteers :



Gender differences across the Condition :



8.1.2 ANOVA – Gender as IV

```
# Difference in Learning.Gain across Gender
oneway.test(dt.y1$Learning.Gain~dt.y1$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  dt.y1$Learning.Gain and dt.y1$Gender
## F = 0.22699, num df = 1.00, denom df = 214.62, p-value = 0.6342
```

We observe **NO Statistical difference** in `Learning.Gain` across `Gender`.
Next, we will examine the differences for the FLIPPED and CONTROL condition separately.

```
# Flipped Condition: Difference in Learning Gain across Gender
t.stat = dt.y1 %>% filter(Condition == "Flipped")

# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Gender
## F = 0.048132, num df = 1.000, denom df = 35.916, p-value = 0.8276
```

We observe **No Statistical Significance.**

```r
# Flipped Condition: Difference in Learning Gain across Gender
t.stat = dt.y1 %>% filter(Condition == "Control")

# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Gender
## F = 0.45316, num df = 1.00, denom df = 178.69, p-value = 0.5017

# Clean-up
rm(t.stat)
```

We observe **No Statistical Significance.**

8.1.3 ANOVA – Condition as IV

In this section, we will examine differences in `Learning.Gain` across `Condition` for different `Gender` :

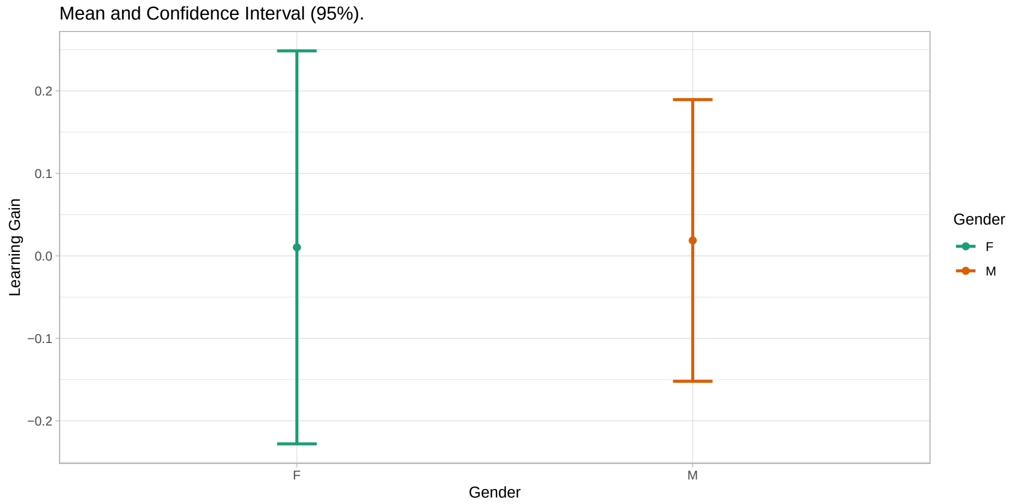
```r
# Subsetting Males
t.stat = dt.y1 %>% filter(Gender == "M")

# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Condition)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Condition
## F = 0.31056, num df = 1.000, denom df = 71.772, p-value = 0.5791
```

```r
# Subsetting Females
t.stat = dt.y1 %>% filter(Gender == "F")

# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Condition)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Condition
## F = 0.048825, num df = 1.000, denom df = 29.744, p-value = 0.8266
```

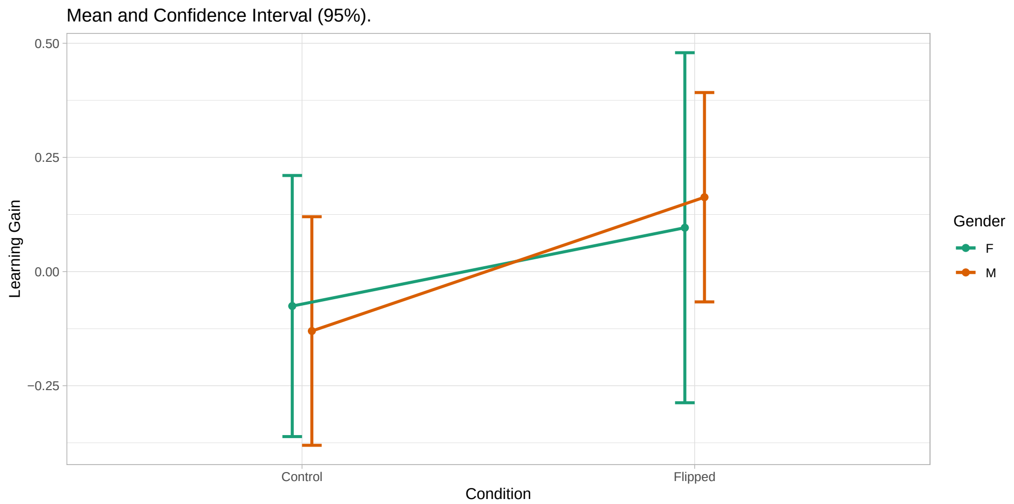
8.2 Year2

8.2.1 Visualization

Gender differences for all the volunteers :



Gender differences across the Condition :



8.2.2 ANOVA – Gender as IV

```
# Difference in Learning.Gain across Gender
oneway.test(dt.y2$Learning.Gain~dt.y2$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  dt.y2$Learning.Gain and dt.y2$Gender
## F = 0.0031435, num df = 1.00, denom df = 138.01, p-value = 0.9554
```

We observe **NO Statistical difference** in `Learning.Gain` across `Gender`.
Next, we will examine the differences for the FLIPPED and CONTROL condition separately.

```
# Flipped Condition: Difference in Learning Gain across Gender
t.stat = dt.y2 %>% filter(Condition == "Flipped")

# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Gender
## F = 0.085912, num df = 1.000, denom df = 58.623, p-value = 0.7705
```

We observe **No Statistical Significance.**

```r
# Flipped Condition: Difference in Learning Gain across Gender
t.stat = dt.y2 %>% filter(Condition == "Control")

# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Gender)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Gender
## F = 0.079522, num df = 1.000, denom df = 79.921, p-value = 0.7787

# Clean-up
rm(t.stat)
```

We observe **No Statistical Significance.**

8.2.3 ANOVA – Condition as IV

In this section, we will examine differences in `Learning.Gain` across `Condition` for different `Gender` :

```r
# Subsetting Males
t.stat = dt.y2 %>% filter(Gender == "M")

# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Condition)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Condition
## F = 2.8624, num df = 1.00, denom df = 122.68, p-value = 0.09321
```

```r
# Subsetting Females
t.stat = dt.y2 %>% filter(Gender == "F")

# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Condition)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Condition
## F = 0.4942, num df = 1.000, denom df = 62.891, p-value = 0.4847
```

9 Category Differences on Aggregated Data

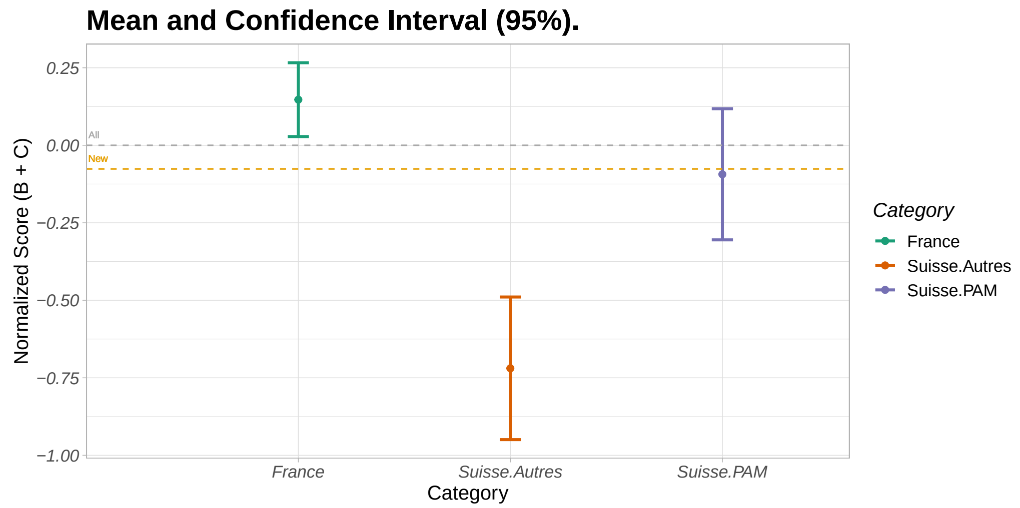
9.1 Visualizing Scores

9.1.1 Year1

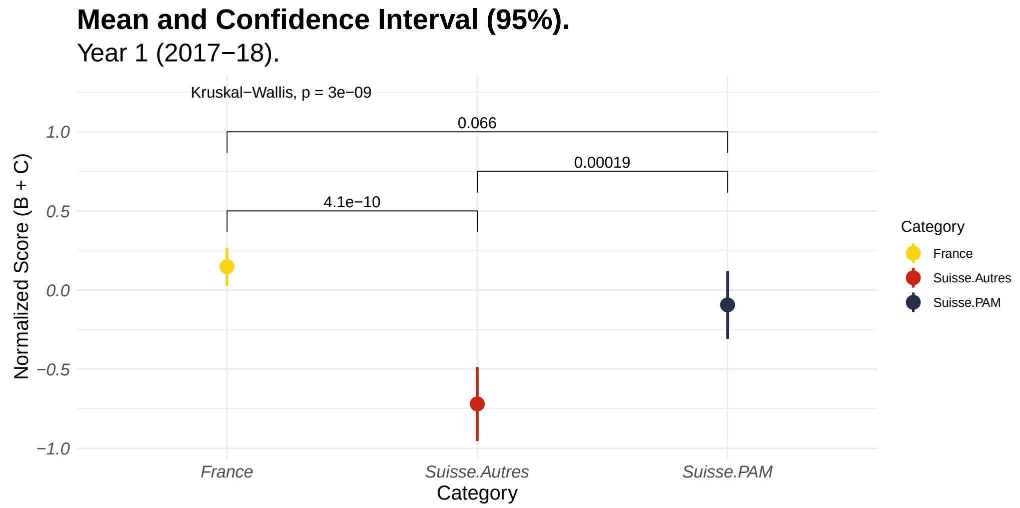
Summarizing the data first :

```r
dt.y1 %>% group_by(Category, Condition) %>%
  summarise(N = n()) %>%
  spread(Condition, N)

## # A tibble: 3 x 3
## # Groups:   Category [4]
##   Category      Control Flipped
##   <fct>           <int>   <int>
## 1 France            168      39
## 2 Suisse.Autres      54      16
## 3 Suisse.PAM         59      15
```

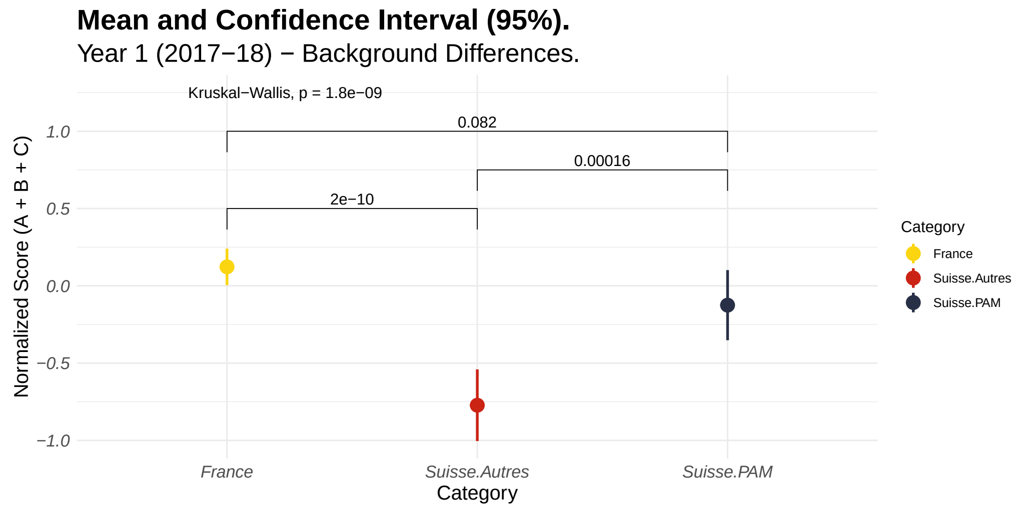
Category differences for all the volunteers :



Plot with pairwise cmparisions :



Plot with pairwise cmparisions (A + B + C) :



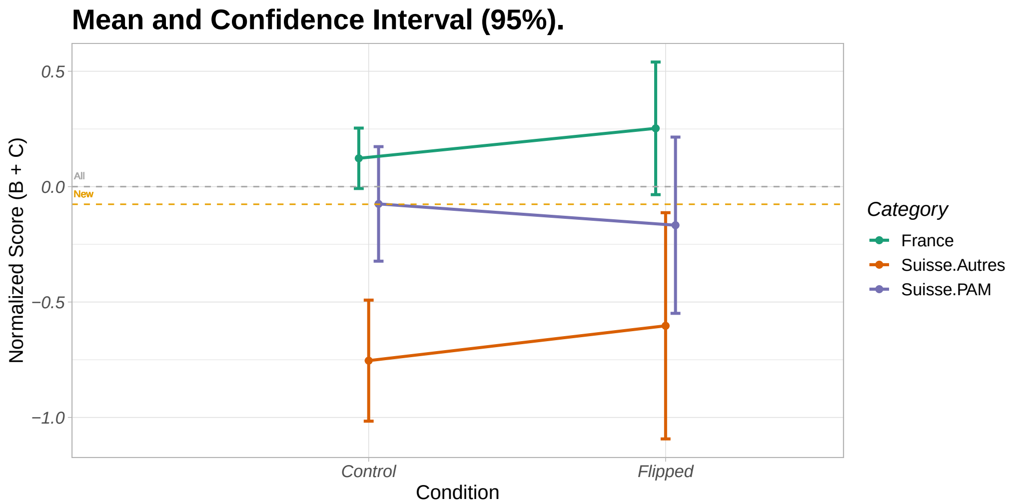
... and the ANOVA :

```
oneway.test(dt.y1$Nor.Score.BC~dt.y1$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  dt.y1$Nor.Score.BC and dt.y1$Category
## F = 21.505, num df = 2.00, denom df = 137.23, p-value = 7.516e-09
```

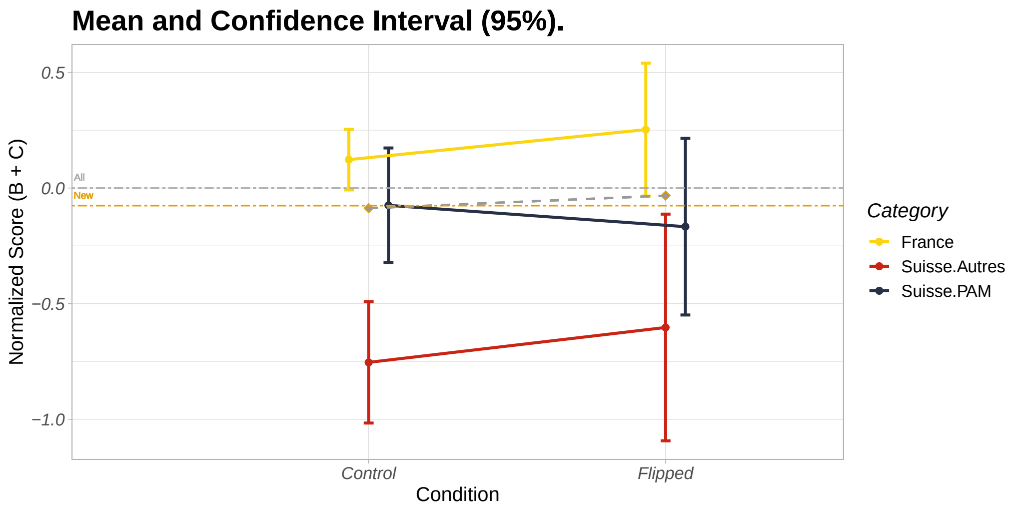
... and the Kruskal-Wallis :

```
kruskal.test(dt.y1$Nor.Score.BC~dt.y1$Category)

##
##  Kruskal-Wallis rank sum test
##
## data:  dt.y1$Nor.Score.BC by dt.y1$Category
## Kruskal-Wallis chi-squared = 39.282, df = 2, p-value = 2.952e-09

epsilonSquared(x = dt.y1$Nor.Score.BC,
               g = dt.y1$Category)

## epsilon.squared
##           0.112
```

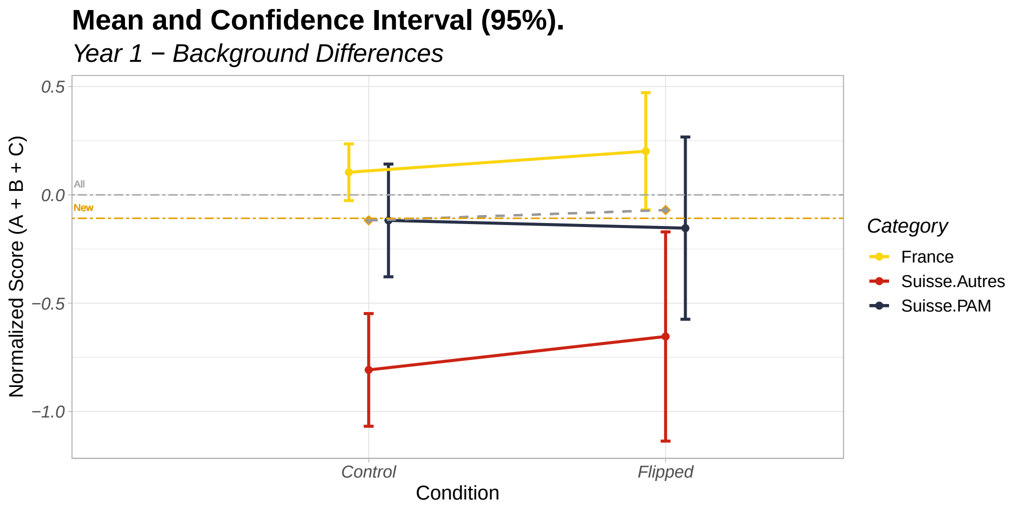
Category differences across the Condition :



Category differences across condition (with weighted mean) :



Category differences across condition and $\mathbf{A} + \mathbf{B} + \mathbf{C}$ (with weighted mean) :



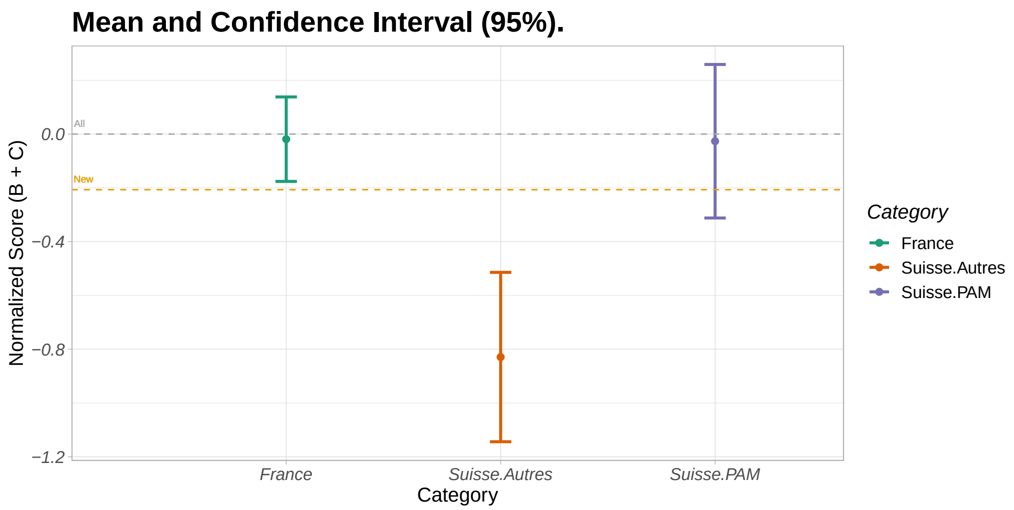
9.1.2 Year2

Summarizing the data first :

```
dt.y2 %>% group_by(Category, Condition) %>%
  summarise(N = n()) %>%
  spread(Condition, N)
```

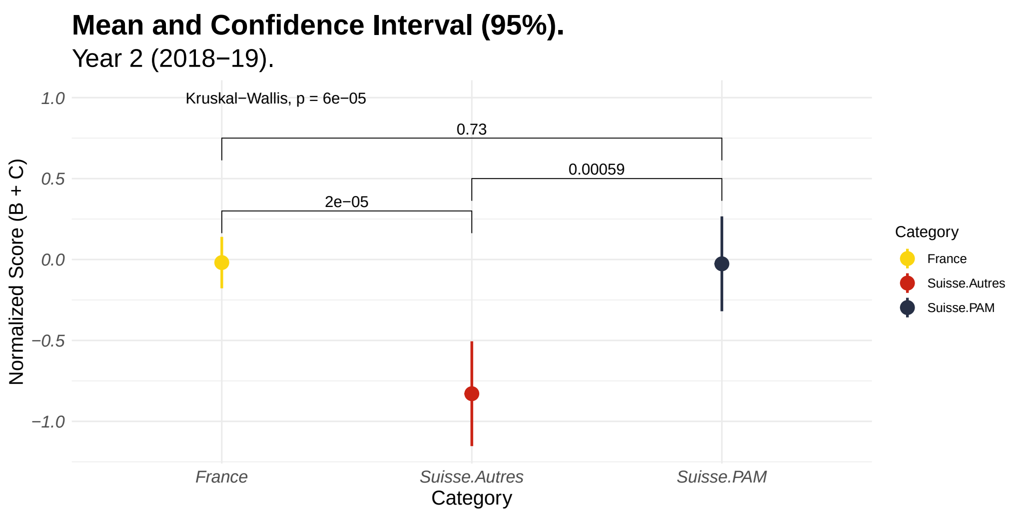
```
## # A tibble: 3 x 3
## # Groups:   Category [4]
##   Category      Control Flipped
##   <fct>           <int>   <int>
## 1 France             53      50
## 2 Suisse.Autres      20      25
## 3 Suisse.PAM         24      24
```

Category differences for all the volunteers :



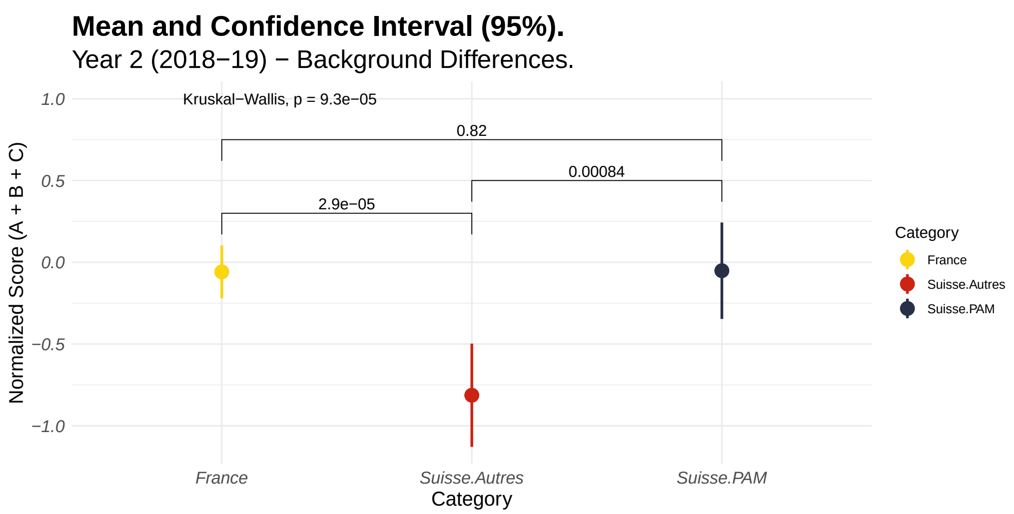
Plot with pairwise cmparisions :

```
## Warning in wilcox.test.default(c(-0.0123356589575437, -0.833688951103404, : cannot compute exact
p-value with ties
```



Plot with pairwise cmparisions (A + B + C) :

```
## Warning in wilcox.test.default(c(0.0264149421745262, -1.01031095344588, : cannot compute exact p-value
with ties
```



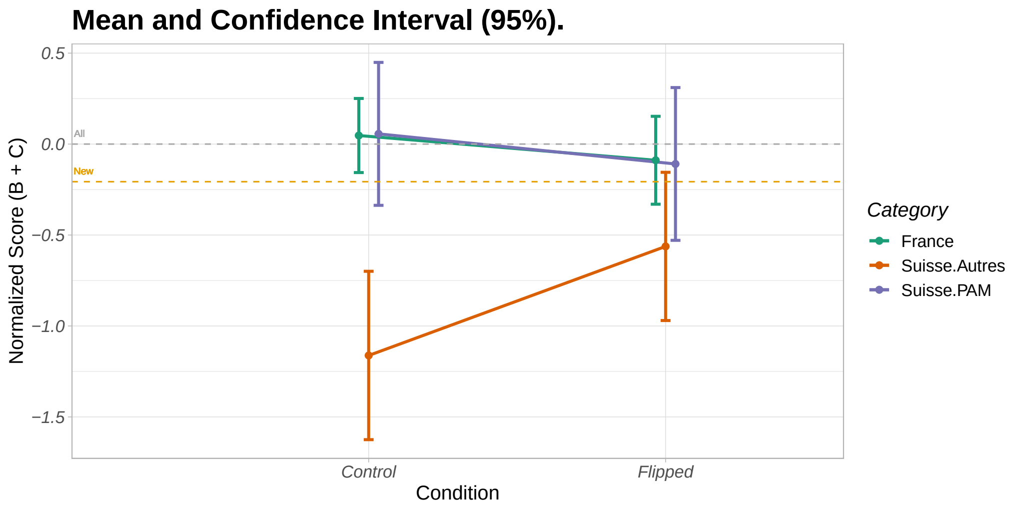
... and the ANOVA :

```
oneway.test(dt.y2$Nor.Score.BC~dt.y2$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  dt.y2$Nor.Score.BC and dt.y2$Category
## F = 10.546, num df = 2.000, denom df = 85.898, p-value = 8.024e-05
```

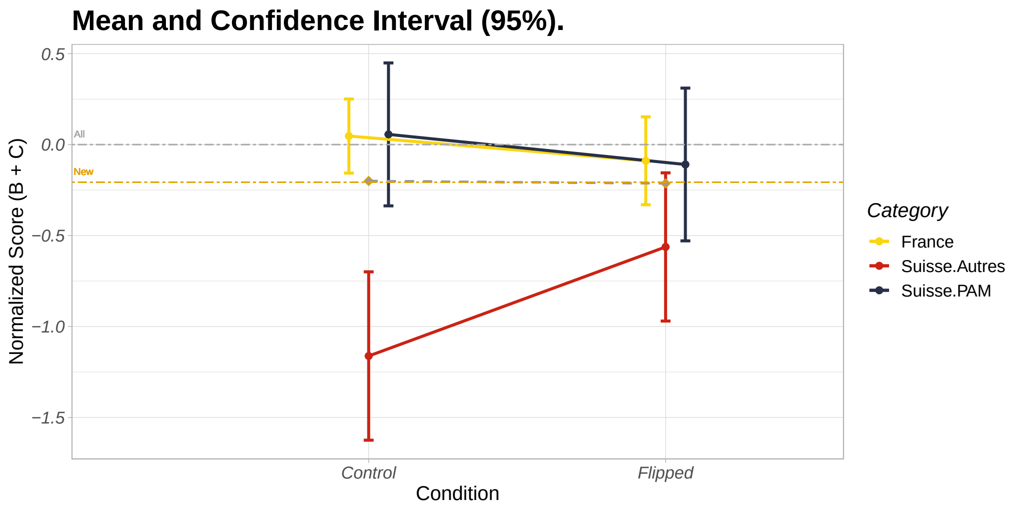
... and the Kruskal-Wallis :

```
kruskal.test(dt.y2$Nor.Score.BC~dt.y2$Category)

##
##  Kruskal-Wallis rank sum test
##
## data:  dt.y2$Nor.Score.BC by dt.y2$Category
## Kruskal-Wallis chi-squared = 19.432, df = 2, p-value = 6.03e-05

epsilonSquared(x = dt.y2$Nor.Score.BC,
               g = dt.y2$Category)

## epsilon.squared
##          0.0997
```

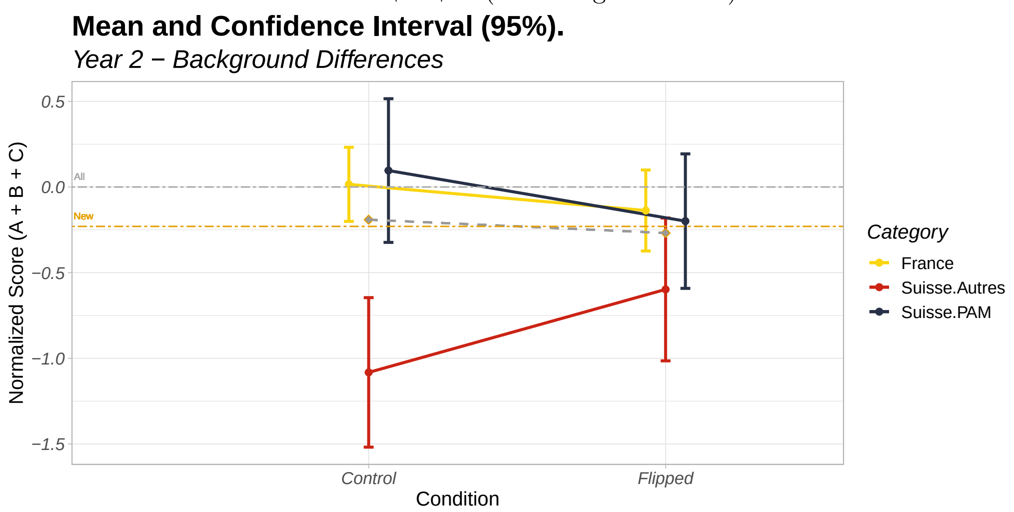
Category differences across the Condition :



Category differences across condition (with weighted mean) :



Category differences across condition and **A+B+C** (with weighted mean) :



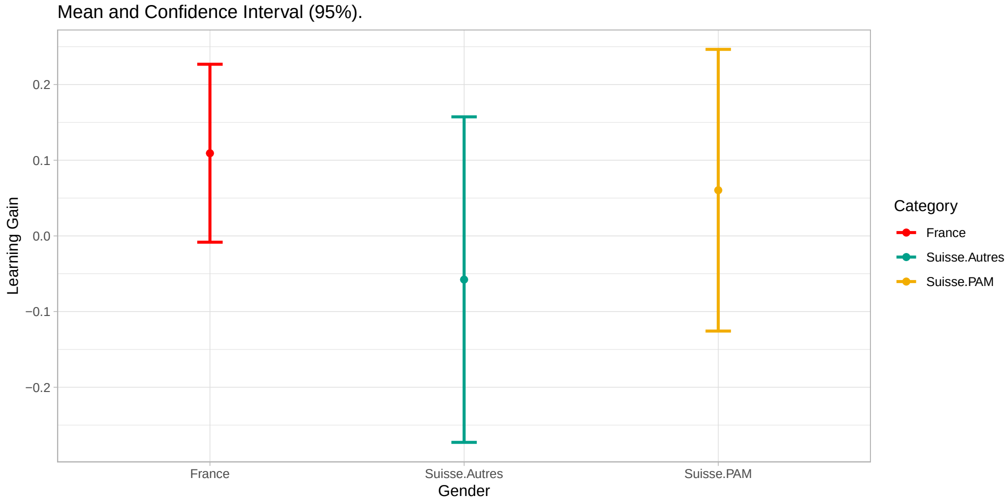
10 Learning Gain and Category Differences

In this section, we will examine the differences in `Learning.Gain` across the different `Categories`.

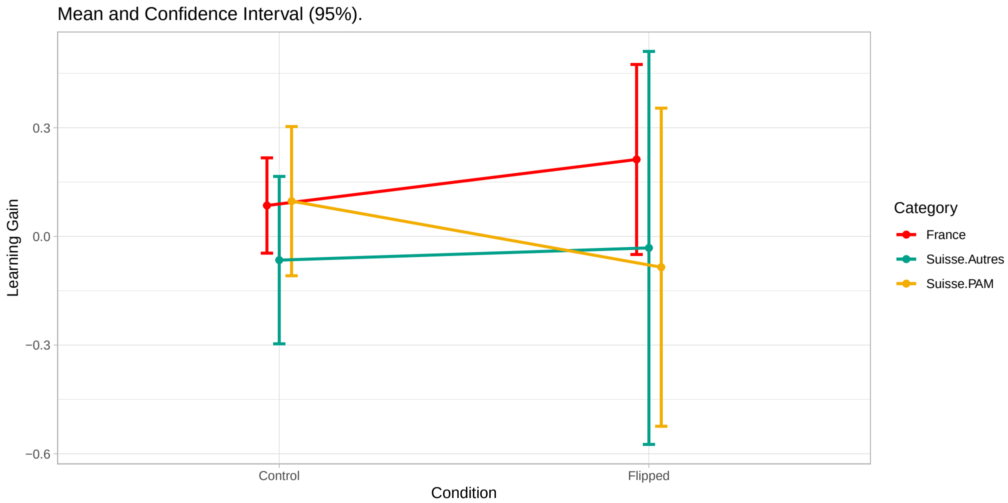
10.1 Year1

10.1.1 Visualization

Category differences for all the volunteers :



Category differences across the Condition :



10.1.2 ANOVA – Category as IV

```
# Difference in Learning.Gain across Category
oneway.test(dt.y1$Learning.Gain~dt.y1$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  dt.y1$Learning.Gain and dt.y1$Category
## F = 0.89184, num df = 2.00, denom df = 142.78, p-value = 0.4122
```

We observe **MArginally Significant Difference** in `Learning.Gain` across `Category`.
Next, we will examine the differences for the Flipped and Control condition separately.

```
# Flipped Condition: Difference in Learning Gain across Category
t.stat = dt.y1 %>% filter(Condition == "Flipped")

# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Category
## F = 0.78403, num df = 2.000, denom df = 28.037, p-value = 0.4663
```

We observe **No Statistical Significance.**

```r
# Flipped Condition: Difference in Learning Gain across Category
t.stat = dt.y1 %>% filter(Condition == "Control")

# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Category
## F = 0.69626, num df = 2.00, denom df = 112.94, p-value = 0.5006

# Clean-up
rm(t.stat)
```

We observe **No Statistical Significance.**

10.1.3 ANOVA – Condition as IV

We will examine the specific cohorts (France, Suisse.PAM, Suisse.Autres), and examine if the `Learning.Gain` differs across `Condition` :

France :

```r
# Subsetting the data
t.stat = dt.y1 %>% filter(Category == "France")

# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Condition)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Condition
## F = 0.72044, num df = 1.000, denom df = 58.683, p-value = 0.3994
```

We observe **No Statistical Significance.**

Suisse.PAM :

```r
# Subsetting the data
t.stat = dt.y1 %>% filter(Category == "Suisse.PAM")

# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Condition)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Condition
## F = 0.54302, num df = 1.000, denom df = 20.595, p-value = 0.4695
```

We observe **No Statistical Significance.**

Suisse.Autres :

```r
# Subsetting the data
t.stat = dt.y1 %>% filter(Category == "Suisse.Autres")

# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Condition)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Condition
## F = 0.012543, num df = 1.000, denom df = 20.744, p-value = 0.9119
```

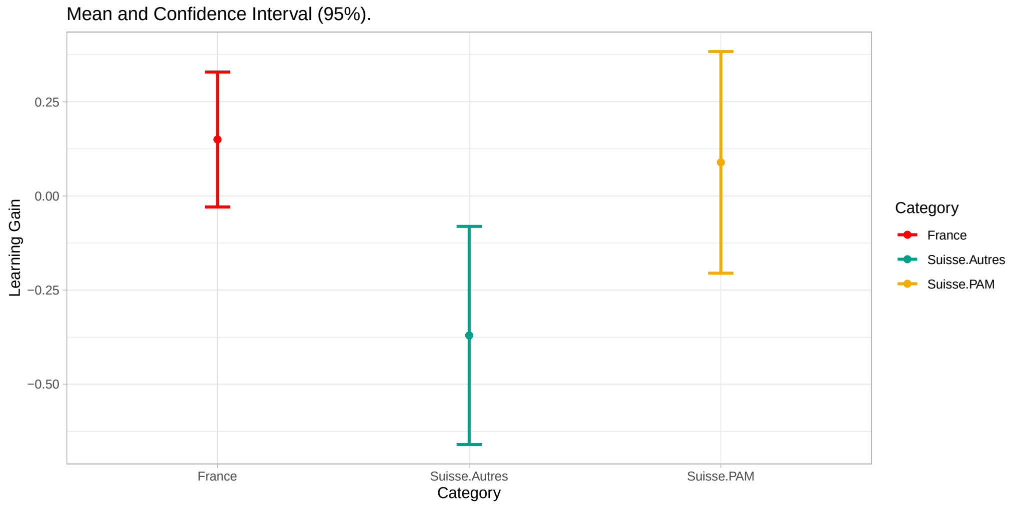
```r
# Clean-up
rm(t.stat)
```

We observe **NO Statistical Significance.**

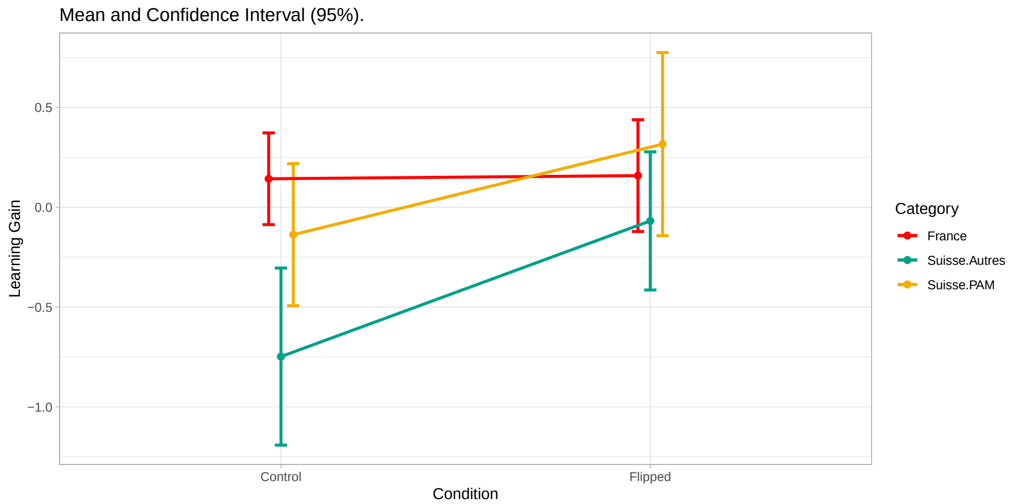
10.2 Year2

10.2.1 Visualization

Category differences for all the volunteers :



Category differences across the Condition :



10.2.2 ANOVA – Category as IV

```r
# Difference in Learning.Gain across Category
oneway.test(dt.y2$Learning.Gain~dt.y2$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  dt.y2$Learning.Gain and dt.y2$Category
## F = 4.5751, num df = 2.00, denom df = 92.17, p-value = 0.01275
```

We observe **Statistical difference** in `Learning.Gain` across `Category`.
Next, we will examine the differences for the FLIPPED and CONTROL condition separately.

```r
# Flipped Condition: Difference in Learning Gain across Category
t.stat = dt.y2 %>% filter(Condition == "Flipped")

# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Category)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Category
## F = 0.94364, num df = 2.000, denom df = 50.031, p-value = 0.396
```

We observe **No Statistical Significance.**

```r
# Flipped Condition: Difference in Learning Gain across Category
t.stat = dt.y2 %>% filter(Condition == "Control")

# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Category)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Category
## F = 6.1061, num df = 2.000, denom df = 41.009, p-value = 0.004773

# Clean-up
rm(t.stat)
```

We observe **Statistical Significance.**

10.2.3 ANOVA – Condition as IV

We will examine the specific cohorts (France, Suisse.PAM, Suisse.Autres), and examine if the `Learning.Gain` differs across `Condition` :

France :

```r
# Subsetting the data
t.stat = dt.y2 %>% filter(Category == "France")

# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Condition)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Condition
## F = 0.0071302, num df = 1.00, denom df = 96.12, p-value = 0.9329
```

We observe **No Statistical Significance.**

Suisse.PAM :

```r
# Subsetting the data
t.stat = dt.y2 %>% filter(Category == "Suisse.PAM")

# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Condition)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Condition
## F = 2.3466, num df = 1.000, denom df = 43.335, p-value = 0.1328
```

We observe **No Statistical Significance.**

Suisse.Autres :

```r
# Subsetting the data
t.stat = dt.y2 %>% filter(Category == "Suisse.Autres")
```

```r
# ANOVA
oneway.test(t.stat$Learning.Gain~t.stat$Condition)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  t.stat$Learning.Gain and t.stat$Condition
## F = 5.6088, num df = 1.00, denom df = 38.01, p-value = 0.02306
```

We observe **Statistical Significance.**

11 Influence of Gender, Condition, and Category (Chi-Square and Linear Regression)

In this section, we will perform linear regression to understand the influence of `Gender`, `Condition`, and `Category` on the post-flipped scores (not the Grade).

11.1 Year1

11.1.1 Chi-Square Test of Independence

First, let's try the **Chi-Square Test of Independence** to examine the relationship between these variables :

```r
# Chi-Square Test of Independence.
t = chisq.test(table(dt.y1$Gender, dt.y1$Category, exclude = "Etranger.Autres"))
t
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(dt.y1$Gender, dt.y1$Category, exclude = "Etranger.Autres")
## X-squared = 15.75, df = 2, p-value = 0.0003801
```

```r
# ... we also show the residuals for different classes.
t$residuals
```

```
##
##         France Suisse.Autres Suisse.PAM
##   F -0.5853461     2.7282399 -1.6744809
##   M  0.4086068    -1.9044759  1.1688886
```

```r
# ... the Observed values.
t$observed
```

```
##
##     France Suisse.Autres Suisse.PAM
##   F     63            36         16
##   M    144            34         58
```

```r
# ... and the expected frequency corresponding to each class.
t$expected
```

```
##
##        France Suisse.Autres Suisse.PAM
##   F  67.82051      22.93447   24.24501
##   M 139.17949      47.06553   49.75499
```

We see that there is **Significant Relationship** between `Gender` and `Category`.

Next, we will perform the same test between `Condition` and `Category` :

```r
# Chi-Square Test of Independence.
t = chisq.test(table(dt.y1$Condition, dt.y1$Category, exclude = "Etranger.Autres"))
t
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(dt.y1$Condition, dt.y1$Category, exclude = "Etranger.Autres")
## X-squared = 0.53487, df = 2, p-value = 0.7653

# ... we also show the residuals for different classes.
t$residuals

##
##                 France Suisse.Autres   Suisse.PAM
##   Control  0.17727220   -0.27249421 -0.03146273
##   Flipped -0.35517694    0.54596075  0.06303772

# ... the Observed values.
t$observed

##
##           France Suisse.Autres Suisse.PAM
##   Control    168            54         59
##   Flipped     39            16         15

# ... and the expected frequency corresponding to each class.
t$expected

##
##               France Suisse.Autres Suisse.PAM
##   Control 165.71795      56.03989   59.24217
##   Flipped  41.28205      13.96011   14.75783
```

There is **NO Significant Relationship** between `Condition` and `Category`.
Finally, let's do the same for `Gender` and `Condition` :

```
# Chi-Square Test of Independence.
t = chisq.test(table(dt.y1$Gender, dt.y1$Condition))
t

##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(dt.y1$Gender, dt.y1$Condition)
## X-squared = 1.0689e-30, df = 1, p-value = 1

# ... we also show the residuals for different classes.
t$residuals

##
##          Control      Flipped
##   F -0.006829237  0.013682843
##   M  0.004767219 -0.009551449

# ... the Observed values.
t$observed

##
##     Control Flipped
##   F      92      23
##   M     189      47

# ... and the expected frequency corresponding to each class.
t$expected

##
##       Control  Flipped
##   F  92.06553 22.93447
##   M 188.93447 47.06553
```

There is **NO Significant Relationship** between `Gender` and `Condition`.

11.1.2 Linear Regression

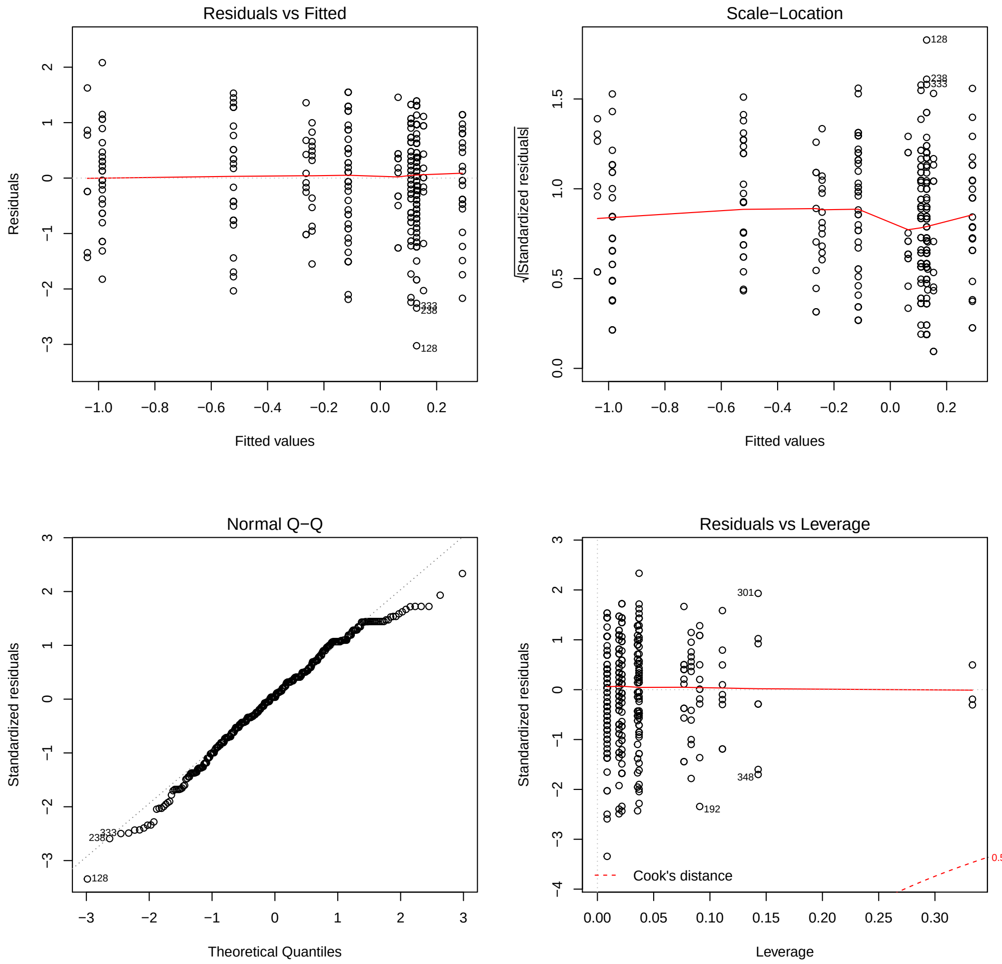
Linear regression :

```
# Model fitting
m = lm(Nor.Score.BC ~ Gender * Category * Condition,
       data = dt.y1)

# Printing the model coefficients
print(m)

##
## Call:
## lm(formula = Nor.Score.BC ~ Gender * Category * Condition, data = dt.y1)
##
## Coefficients:
##                                    (Intercept)
##                                        0.10889
##                                        GenderM
##                                        0.02004
##                          CategorySuisse.Autres
##                                       -1.09555
##                             CategorySuisse.PAM
##                                       -0.04572
##                               ConditionFlipped
##                                        0.04453
##                  GenderM:CategorySuisse.Autres
##                                        0.44536
##                     GenderM:CategorySuisse.PAM
##                                       -0.19710
##                       GenderM:ConditionFlipped
##                                        0.11807
##         CategorySuisse.Autres:ConditionFlipped
##                                        0.67874
##            CategorySuisse.PAM:ConditionFlipped
##                                        0.02514
## GenderM:CategorySuisse.Autres:ConditionFlipped
##                                       -1.35975
##    GenderM:CategorySuisse.PAM:ConditionFlipped
##                                       -0.31601

# Next, we print the model summary
summary(m)

##
## Call:
## lm(formula = Nor.Score.BC ~ Gender * Category * Condition, data = dt.y1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.02440 -0.56213  0.03221  0.64659  2.08176
##
## Coefficients:
##                                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                                     0.10889    0.12602   0.864   0.3881
## GenderM                                         0.02004    0.15166   0.132   0.8949
## CategorySuisse.Autres                          -1.09555    0.21556  -5.082 6.17e-07 ***
## CategorySuisse.PAM                             -0.04572    0.28179  -0.162   0.8712
## ConditionFlipped                                0.04453    0.30158   0.148   0.8827
## GenderM:CategorySuisse.Autres                   0.44536    0.29012   1.535   0.1257
## GenderM:CategorySuisse.PAM                     -0.19710    0.32323  -0.610   0.5424
## GenderM:ConditionFlipped                        0.11807    0.35716   0.331   0.7412
```

```
## CategorySuisse.Autres:ConditionFlipped          0.67874    0.46184   1.470   0.1426
## CategorySuisse.PAM:ConditionFlipped             0.02514    0.65555   0.038   0.9694
## GenderM:CategorySuisse.Autres:ConditionFlipped -1.35975    0.63124  -2.154   0.0319 *
## GenderM:CategorySuisse.PAM:ConditionFlipped    -0.31601    0.74372  -0.425   0.6712
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9087 on 339 degrees of freedom
## Multiple R-squared:  0.1421,Adjusted R-squared:  0.1142
## F-statistic: 5.103 on 11 and 339 DF,  p-value: 2.107e-07

# We also show the ANOVA table
anova(m)

## Analysis of Variance Table
##
## Response: Nor.Score.BC
##                          Df  Sum Sq Mean Sq F value    Pr(>F)
## Gender                    1   1.757  1.7566  2.1271   0.14564
## Category                  2  37.633 18.8164 22.7857 5.191e-10 ***
## Condition                 1   0.422  0.4219  0.5109   0.47525
## Gender:Category           2   1.183  0.5913  0.7161   0.48941
## Gender:Condition          1   1.184  1.1838  1.4335   0.23203
## Category:Condition        2   0.318  0.1591  0.1927   0.82483
## Gender:Category:Condition   2   3.855  1.9274  2.3340   0.09847 .
## Residuals               339 279.946  0.8258
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We will also plot the diagnostic plots for the regression analysis :

```
# Diagnostic plots
layout(matrix(c(1,2,3,4), 2, 2))
plot(m)
```



The results from YEAR1 show that SUISSE.AUTRES perform significantly badly overall. In addition, there is an *interaction effect* between `Gender` – particularly MALE, `Category` – SUISSE.AUTRES, and `Condition` – FLIPPED. This signifies that MALE students belonging to SUISSE.AUTRES and in the FLIPPED condition performed significantly worse (a drop of 2.7 points in `Nor.Score.BC`.

Furthermore, the ANOVA of the model shows that `Category` significantly influences the `Nor.Score.BC`, and there is an interaction effect between `Gender`, `Category`, and `Condition`.

11.2 Year2

11.2.1 Chi Square Test of Independence

First, let's try the **Chi-Square Test of Independence** to examine the relationship between these variables :

```
# Chi-Square Test of Independence.
t = chisq.test(table(dt.y2$Gender, dt.y2$Category, exclude = "Etranger.Autres"))
t

##
##  Pearson's Chi-squared test
##
## data:  table(dt.y2$Gender, dt.y2$Category, exclude = "Etranger.Autres")
## X-squared = 3.9562, df = 2, p-value = 0.1383

# ... we also show the residuals for different classes.
t$residuals

##
##          France Suisse.Autres Suisse.PAM
##    F  0.2002079     0.9799579 -1.2421180
##    M -0.1492262    -0.7304175  0.9258201

# ... the Observed values.
t$observed
```

```
##
##     France Suisse.Autres Suisse.PAM
##   F     38            20         12
##   M     65            25         36

# ... and the expected frequency corresponding to each class.
t$expected

##
##       France Suisse.Autres Suisse.PAM
##   F 36.78571      16.07143   17.14286
##   M 66.21429      28.92857   30.85714
```

We see that there is **NO Significant Relationship** between `Gender` and `Category`.
Next, we will perform the same test between `Condition` and `Category` :

```
# Chi-Square Test of Independence.
t = chisq.test(table(dt.y2$Condition, dt.y2$Category, exclude = "Etranger.Autres"))
t

##
##  Pearson's Chi-squared test
##
## data:  table(dt.y2$Condition, dt.y2$Category, exclude = "Etranger.Autres")
## X-squared = 0.62259, df = 2, p-value = 0.7325

# ... we also show the residuals for different classes.
t$residuals

##
##                 France Suisse.Autres  Suisse.PAM
##   Control  0.28369912   -0.48110498  0.05024660
##   Flipped -0.28081885    0.47622054 -0.04973647

# ... the Observed values.
t$observed

##
##            France Suisse.Autres Suisse.PAM
##   Control     53            20         24
##   Flipped     50            25         24

# ... and the expected frequency corresponding to each class.
t$expected

##
##              France Suisse.Autres Suisse.PAM
##   Control 50.97449      22.27041    23.7551
##   Flipped 52.02551      22.72959    24.2449
```

There is **NO Significant Relationship** between `Condition` and `Category`.
Finally, let's do the same for `Gender` and `Condition` :

```
# Chi-Square Test of Independence.
t = chisq.test(table(dt.y2$Gender, dt.y2$Condition))
t

##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(dt.y2$Gender, dt.y2$Condition)
## X-squared = 0, df = 1, p-value = 1

# ... we also show the residuals for different classes.
t$residuals
```

```
##
##         Control     Flipped
##   F  0.06067854 -0.06006250
##   M -0.04522711  0.04476794

# ... the Observed values.
t$observed

##
##     Control Flipped
##   F      35      35
##   M      62      64

# ... and the expected frequency corresponding to each class.
t$expected

##
##      Control  Flipped
##   F 34.64286 35.35714
##   M 62.35714 63.64286
```

There is **NO Significant Relationship** between `Gender` and `Condition`.

11.2.2 Linear Regression

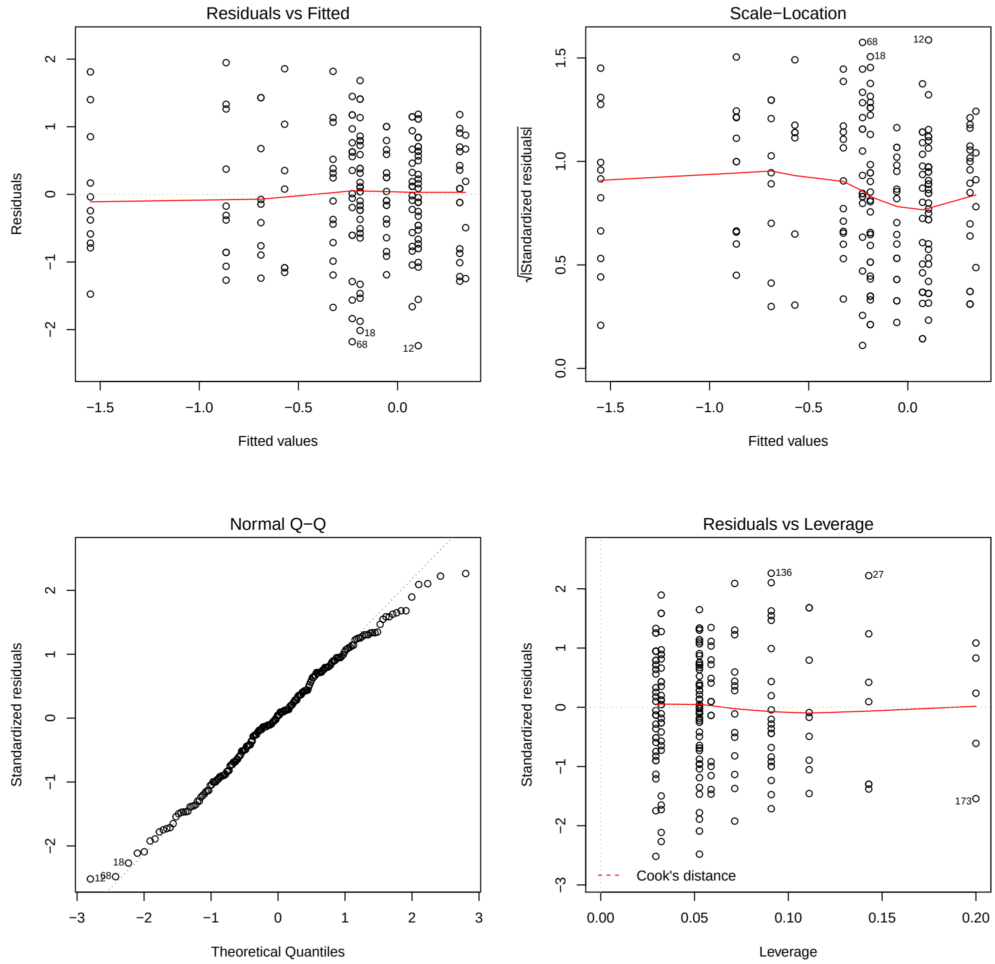
Linear regression :

```
# Model fitting
m = lm(Nor.Score.BC ~ Gender * Category * Condition,
       data = dt.y2)

# Printing the model coefficients
print(m)

##
## Call:
## lm(formula = Nor.Score.BC ~ Gender * Category * Condition, data = dt.y2)
##
## Coefficients:
##                                    (Intercept)
##                                       -0.05556
##                                        GenderM
##                                        0.15999
##                          CategorySuisse.Autres
##                                       -0.63363
##                             CategorySuisse.PAM
##                                       -0.51412
##                               ConditionFlipped
##                                        0.12969
##                  GenderM:CategorySuisse.Autres
##                                       -1.02006
##                     GenderM:CategorySuisse.PAM
##                                        0.72348
##                       GenderM:ConditionFlipped
##                                       -0.42308
##         CategorySuisse.Autres:ConditionFlipped
##                                       -0.30530
##            CategorySuisse.PAM:ConditionFlipped
##                                        0.78358
## GenderM:CategorySuisse.Autres:ConditionFlipped
##                                        1.82272
##    GenderM:CategorySuisse.PAM:ConditionFlipped
##                                       -1.03245

# Next, we print the model summary
```

```r
summary(m)

##
## Call:
## lm(formula = Nor.Score.BC ~ Gender * Category * Condition, data = dt.y2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.23859 -0.61421  0.03974  0.66105  1.94760
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              -0.05556    0.20708  -0.268  0.78875
## GenderM                                   0.15999    0.25854   0.619  0.53681
## CategorySuisse.Autres                    -0.63363    0.36525  -1.735  0.08446 .
## CategorySuisse.PAM                       -0.51412    0.39909  -1.288  0.19929
## ConditionFlipped                          0.12969    0.29285   0.443  0.65840
## GenderM:CategorySuisse.Autres            -1.02006    0.48108  -2.120  0.03532 *
## GenderM:CategorySuisse.PAM                0.72348    0.48080   1.505  0.13410
## GenderM:ConditionFlipped                 -0.42308    0.36879  -1.147  0.25279
## CategorySuisse.Autres:ConditionFlipped   -0.30530    0.50036  -0.610  0.54251
## CategorySuisse.PAM:ConditionFlipped       0.78358    0.60424   1.297  0.19632
## GenderM:CategorySuisse.Autres:ConditionFlipped  1.82272    0.65793   2.770  0.00617 **
## GenderM:CategorySuisse.PAM:ConditionFlipped    -1.03245    0.71145  -1.451  0.14843
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9026 on 184 degrees of freedom
## Multiple R-squared:  0.2071,Adjusted R-squared:  0.1597
## F-statistic:  4.37 on 11 and 184 DF,  p-value: 8.11e-06

# We also show the ANOVA table
anova(m)

## Analysis of Variance Table
##
## Response: Nor.Score.BC
##                           Df  Sum Sq Mean Sq F value   Pr(>F)
## Gender                     1   0.221  0.2210  0.2712 0.603140
## Category                   2  22.399 11.1996 13.7460 2.73e-06 ***
## Condition                  1   0.028  0.0283  0.0348 0.852266
## Gender:Category            2   0.528  0.2642  0.3243 0.723429
## Gender:Condition           1   0.648  0.6475  0.7948 0.373828
## Category:Condition         2   4.560  2.2798  2.7981 0.063518 .
## Gender:Category:Condition  2  10.783  5.3916  6.6174 0.001678 **
## Residuals                184 149.915  0.8148
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We will also plot the diagnostic plots for the regression analysis :

```r
# Diagnostic plots
layout(matrix(c(1,2,3,4), 2, 2))
plot(m)
```



The results from YEAR2 show an *inverse* effect as compared to YEAR1. *Firstly*, MALES in `Category` SUISSE.AUTRES and FLIPPED `Condition` had a significant positive score (which was not the case in YEAR1). In addition, the analysis of ANOVA shows significant influence of `Category`, as well as significant interaction efffects between `Category` and `Condition`, and between `Gender`, `Category`, and `Condition`.

12 Analysis of Students' Trajectory (Drop-outs / MAN / Passed)

In this section, we will look at the students who failed after the end of their "Bachelor Semester I". We will examine the population who simply *dropped-out*, *repeated*, or went to *MAN*.

In order to do so, we will look at the following sources :

- The data from the Spring semester to see if some students were registered in MAN.
- Data from the Autumn semester in the following year, to identify the students who were repeating the semester/year.

12.0.1 Read IS-Academia Data

Since, the next sections will need some information from the IS-Academia, we will connect to the database and fetch the relevant data :

Step 1 : Establishing the connection

```
# Establish a connection to the database.
connection = dbConnect(
  RMariaDB::MariaDB(),
  user = 'hverma',
  password = decrypt_kc_pw("CEDE_DB_EPFL"),
  dbname = 'project_himanshu',
  host = 'cedegemac8.epfl.ch',
  port = 3306
)
```

Step 2 : Course data

```r
# Firstly, getting all the course codes.
t.course = dbReadTable(connection, "isa_course_codes")

# This list is not organized, so we will just sort it by CourseCode.
t.course = arrange(t.course, CourseCode)

# Since one CourseCode is repeated several times.
# We pick only distinct values.
t.course = t.course %>% distinct()
```

Step 3 : Grades data

```r
# Secondly, we fetch the grades.
t.grades = dbReadTable(connection, "isa_grades")

# Sorting the data by SCIPERs
t.grades = t.grades %>% arrange(SCIPER)

# Few columns need to be unlisted.
# Student details
t.grades$IsStudent = as.character(unlist(t.grades$IsStudent))
t.grades$IsStudent = ifelse(t.grades$IsStudent == "00", "NO", "YES")

t.grades$IsEnrolled = as.character(unlist(t.grades$IsEnrolled))
t.grades$IsEnrolled = ifelse(t.grades$IsEnrolled == "00", "NO", "YES")

t.grades$IsOutOfPlan = as.character(unlist(t.grades$IsOutOfPlan))
t.grades$IsOutOfPlan = ifelse(t.grades$IsOutOfPlan == "00", "NO", "YES")

t.grades$IsInactive = as.character(unlist(t.grades$IsInactive))
t.grades$IsInactive = ifelse(t.grades$IsInactive == "00", "NO", "YES")

# Course details
t.grades$IsSubject = as.character(unlist(t.grades$IsSubject))
t.grades$IsSubject = ifelse(t.grades$IsSubject == "00", "NO", "YES")

t.grades$IsTaught = as.character(unlist(t.grades$IsTaught))
t.grades$IsTaught = ifelse(t.grades$IsTaught == "00", "NO", "YES")

t.grades$IsExamined = as.character(unlist(t.grades$IsExamined))
t.grades$IsExamined = ifelse(t.grades$IsExamined == "00", "NO", "YES")

# Since there are many irrelevant columns, we only select the relevant ones:
t.grades = t.grades %>% select(SCIPER, IsStudent, IsEnrolled, IsOutOfPlan,
                               IsLocked, IsInactive, Session, Grade,
                               GradeDate, Credit, Status, YearName,
                               SemesterName, SemesterType, LevelName, SubjectID,
                               SubjectName, IsSubject, IsTaught, IsExamined,
                               TeachingLanguage, StudyDomain, UnitAcronym, UnitCode,
                               PlanID, SchoolName, SectionName, SectionAcronym,
                               SectionCode, AcademicName, AcademicCode, PedagogicalName,
                               PedagogicalCode, BirthDate)
```

Step 4 : Joining the two.

```r
# Creating a table of Course Codes and Course Names
t.course.names = merge(x = t.grades %>% select(SubjectID, SubjectName, PlanID),
                       y = t.course,
                       by = c("PlanID", "SubjectID"),
                       all.x = TRUE) %>%
  distinct() %>% arrange(SubjectName)

# Cleaning-up.
```

```
# Remove the initial table t.course
rm(t.course)
```

12.1 Year1

In Year1 there are in total 351 New students (excluding the Etranger.Autres). Now, in order to follow their progress, we will first extract their SCIPERS.

Following is the summary of the new students :

```
# Summary of distribution
dt.y1 %>% group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  spread(Gender, N)

## # A tibble: 2 x 3
## # Groups:   Condition [2]
##   Condition     F     M
##   <fct>     <int> <int>
## 1 Control      92   189
## 2 Flipped      23    47
```

... and the proportions :

```
# Computing the proportion of males, females.
t.stat = dt.y1 %>% group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  #mutate(Total = sum(N)) %>%
  #mutate(Prop.Female = (N / Total) * 100) %>%
  #spread(Gender, N)
  spread(Gender, N)
t.stat$Total = t.stat$F + t.stat$M
t.stat$Prop.Female = (t.stat$F / t.stat$Total) * 100
t.stat$Prop.Male = (t.stat$M / t.stat$Total) * 100
t.stat

## # A tibble: 2 x 6
## # Groups:   Condition [2]
##   Condition     F     M Total Prop.Female Prop.Male
##   <fct>     <int> <int> <int>       <dbl>     <dbl>
## 1 Control      92   189   281        32.7      67.3
## 2 Flipped      23    47    70        32.9      67.1
```

```
# Convert ID.Anon to SCIPER
dt.y1$SCIPER = lapply(dt.y1$ID.Anon, GenerateSCIPER)
dt.y1$SCIPER = unlist(dt.y1$SCIPER)
```

12.1.1 Students who Passed

In order to compute the number of students who passed the Autumn Semester (2017-18), we will do a join operation with the dataset which shows the **total number of registrations in the Spring semester**.

```
# Registrations in the Spring Semester
t.stat = t.grades %>% filter(
  YearName == "2017-2018" & PedagogicalCode == "BA2")
```

There are a few students, for whom the `Session` value is **'NA'**. There are in total 27 students with 'NA' value for `Session`. Also, there are 4 students for whom the `Session` value is **RAT** (what does this mean ?)

```r
# Checking the students, who have a missing Session value.
na.val = t.stat %>% filter(is.na(Session)) %>%
  arrange(SCIPER)

# Checking the students, who have a Session value of RAT
rat.val = t.stat %>% filter(Session == "RAT") %>%
  arrange(SCIPER)
```

If we look at the subjects for `na.val` dataset, we see that there are mainly the following subjects :

```r
# Overview of the name of Subjects
levels(as.factor(na.val$SubjectName))

## [1] "Biologie I et II"
## [2] "Introduction à la science des matériaux + technologie"
## [3] "Matériaux: de la chimie aux propriétés + chimie"
```

In addition, the value for `Grade` is available for the `na.val` dataset, however the `GradeDate` is different.

Another way to verify if the students who are part of `na.val` dataset are repeaters is to perform a join operation with the `AcademicYear` of 2018-2019 :

```r
# First, filter the students from the year 2018-2019
temp = t.grades %>% filter(
  YearName == "2018-2019" & PedagogicalCode == "BA1")

# Second, we perform the join operation with na.val to
# see how many people are in the second year.
temp.merge = merge(x = temp,
                   y = na.val %>% select(SCIPER),
                   by = "SCIPER")
```

What we notice is that, people for whom the `Session` value is <NA> –in the `AcademicYear` of **2017-2018** and `PedagogicalCode` of **BA2**– are also appearing in the next year's *first* semester students. We see that there are in total **26** students who started their *first semester* in the next academic year, and 1 person dropped out.

```r
# Examining the 1 student who had dropped-out
t.drop = subset(na.val, !(na.val$SCIPER %in% unlist(temp.merge %>% distinct(SCIPER))))
#t.drop
```

Next, we also look at the subjects for `rat.val` dataset, and see that there are the following subjects :

```r
# Overview of the name of Subjects
levels(as.factor(rat.val$SubjectName))

## [1] "Chimie générale"           "Chimie générale (anglais)" "Geometry"
```

However, the `Grade` values for this dataset are strange :

```r
# Overview of the Grade values.
levels(as.factor(rat.val$Grade))

## [1] "d"             "STATUT_NOTE_D"
```

... and the `GradeDate` has different values.

To proceed, we can filter out the rows for `Session` column which contain RAT or <NA> values :

```r
# Filtering out based on Session
t.stat = t.stat %>% filter(Session == "ETE")
```

Now, the actual number of students who passed the *first* semester are the ones who remain. There are in total 311 students who passed from 351 students. This number includes, New, Ex-MAN, Ex-CMS, etc.

```r
# Now, we only extract the distinct SCIPERs of students who passed.
t.sciper = t.stat %>% select(SCIPER) %>%
  distinct()
```

Following dataset contains the complete details of these student :

```r
# Filtering the SCIPERS which are contained in t.sciper dataset
t.sciper$SCIPER = as.character(t.sciper$SCIPER)
#t1 = subset(dt.y1, dt.y1$SCIPER %in% unlist(t.sciper))
t.passed = merge(x = dt.y1,
                 y = t.sciper,
                 by = "SCIPER")

# Cleaning-up
rm(t.sciper, t.stat)
```

Now, the total number of students who passed amongst the **"New"** students are **202**.
Amongst these passed students, following is the distribution in different conditions :

```r
# Distribution of Passed Students (Condition, Gender):
t.passed %>% group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 4 x 5
## # Groups:   Condition [2]
##   Condition Gender     N Total  Prop
##   <fct>     <fct>  <int> <int> <dbl>
## 1 Control   F         47   161  29.2
## 2 Control   M        114   161  70.8
## 3 Flipped   F         11    41  26.8
## 4 Flipped   M         30    41  73.2

# Distribution of Passed Students (Condition, Category):
t.passed %>% group_by(Condition, Category) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 6 x 5
## # Groups:   Condition [2]
##   Condition Category          N Total  Prop
##   <fct>     <fct>         <int> <int> <dbl>
## 1 Control   France          114   161 70.8
## 2 Control   Suisse.Autres    14   161  8.70
## 3 Control   Suisse.PAM       33   161 20.5
## 4 Flipped   France           29    41 70.7
## 5 Flipped   Suisse.Autres     2    41  4.88
## 6 Flipped   Suisse.PAM       10    41 24.4

# Distribution of Passed Students (Condition, Category, Gender):
t.passed %>% group_by(Condition, Category, Gender) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 11 x 6
## # Groups:   Condition, Category [6]
##    Condition Category      Gender     N Total  Prop
##    <fct>     <fct>         <fct>  <int> <int> <dbl>
##  1 Control   France        F         35   114  30.7
##  2 Control   France        M         79   114  69.3
##  3 Control   Suisse.Autres F          4    14  28.6
```

```
##   4 Control   Suisse.Autres M          10     14  71.4
##   5 Control   Suisse.PAM    F           8     33  24.2
##   6 Control   Suisse.PAM    M          25     33  75.8
##   7 Flipped   France        F           8     29  27.6
##   8 Flipped   France        M          21     29  72.4
##   9 Flipped   Suisse.Autres F           2      2 100
##  10 Flipped   Suisse.PAM    F           1     10  10
##  11 Flipped   Suisse.PAM    M           9     10  90
```

12.1.2 Students who Failed First Semester + Drop-Outs

We found in the previous section that some students had **<NA>** values for `Session`. We also found out that these 27 students include 26 failed students and 1 dropout student. Including we have 4 students who have the "*rattrapage*" status, and we include them in the same category. So, we will create a separate dataset called `t.failed` :

```
# Data of failed students (including drop-out)
t.failed = merge(x = dt.y1,
                 y = rbind(na.val, rat.val) %>% select(SCIPER) %>% distinct(),
                 by = "SCIPER")
```

We see that there are **26** of **New** students who failed in their first semester.

Amongst these failed students, following is the distribution in different conditions :

```
# Distribution of Passed Students (Condition, Gender):
t.failed %>% group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 4 x 5
## # Groups:   Condition [2]
##   Condition Gender     N Total  Prop
##   <fct>     <fct>  <int> <int> <dbl>
## 1 Control   F          8    22  36.4
## 2 Control   M         14    22  63.6
## 3 Flipped   F          2     4  50
## 4 Flipped   M          2     4  50

# Distribution of Passed Students (Condition, Category):
t.failed %>% group_by(Condition, Category) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 5 x 5
## # Groups:   Condition [2]
##   Condition Category          N Total  Prop
##   <fct>     <fct>         <int> <int> <dbl>
## 1 Control   France           15    22  68.2
## 2 Control   Suisse.Autres     3    22  13.6
## 3 Control   Suisse.PAM        4    22  18.2
## 4 Flipped   France            2     4  50
## 5 Flipped   Suisse.PAM        2     4  50

# Distribution of Passed Students (Condition, Category, Gender):
t.failed %>% group_by(Condition, Category, Gender) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 10 x 6
## # Groups:   Condition, Category [5]
##    Condition Category      Gender     N Total  Prop
```

```
##    <fct>     <fct>         <fct>  <int> <int> <dbl>
##  1 Control   France        F          5    15  33.3
##  2 Control   France        M         10    15  66.7
##  3 Control   Suisse.Autres F          1     3  33.3
##  4 Control   Suisse.Autres M          2     3  66.7
##  5 Control   Suisse.PAM    F          2     4  50
##  6 Control   Suisse.PAM    M          2     4  50
##  7 Flipped   France        F          1     2  50
##  8 Flipped   France        M          1     2  50
##  9 Flipped   Suisse.PAM    F          1     2  50
## 10 Flipped   Suisse.PAM    M          1     2  50
```

12.1.3 Students who went to MAN

Looking at the `t.grades` dataset, we observe that there are some students who have the `PedagogicalCode` as **MAN**. So, let's start by filtering them out. We will only filter out the students who belong to the `AcademicYear` of 2017-2018.

```
# Filtering out the MAN students.
t.stat = t.grades %>% filter(
  YearName == "2017-2018" & PedagogicalCode == "MAN"
)
```

We see that the students who appeared in the MAN semester take the follwoing courses :

```
# Courses taken by MAN Students
levels(as.factor(t.stat$SubjectName))

## [1] "Mathématiques 1A (pour MAN)" "Mathématiques 1B (pour MAN)"
## [3] "Mathématiques 2 (pour MAN)"  "Physique (pour MAN)"
```

Amongst these students, some of them did not appear for exam (**STATUT_NOTE_NA**) or reported sickness (**STATUT_NOTE_M**). There are in total **22**.

Let's make new datasets about these :

```
# Students who were absent in exam.
na.stu = t.stat %>% filter(Grade =="STATUT_NOTE_NA" | is.na(Grade)) %>%
  arrange(SCIPER)
na.stu.sciper = na.stu %>% distinct(SCIPER)

# Students who were sick during exam.
mal.stu = t.stat %>% filter(Grade == "STATUT_NOTE_M") %>%
  arrange(SCIPER)
mal.stu.sciper = mal.stu %>% distinct(SCIPER)
```

We see that **2** students reported sick and **20** were absent. Performing an inner join with the volunteer data will tell us if these students were from the sample of NEW students.

```
# Join operation with the volunteer data.
t.man.drop = merge(x = dt.y1,
              y = rbind(na.stu.sciper, mal.stu.sciper),
              by = "SCIPER")

# Cleaning up
rm(na.stu, na.stu.sciper, mal.stu, mal.stu.sciper)
```

We see that there are **15** students amongst the "New" students who dropped from MAN (so to speak).

Now, we identify the students who actually took exams for MAN :

```
# Students who got grade for MAN.
graded.stu = t.stat %>% filter(
  !(Grade == "STATUT_NOTE_M" | Grade == "STATUT_NOTE_NA" | is.na(Grade)))
```

```r
# Unique students who had grades.
graded.stu.sciper = graded.stu %>% select(SCIPER) %>% distinct(SCIPER)

# Identify who took exam in MAN
t.man.exam = merge(x = dt.y1,
                   y = graded.stu.sciper,
                   by = "SCIPER")
```

Amongst these MAN students, following is the distribution in different conditions :

```r
# Distribution of Passed Students (Condition, Gender):
t.man.exam %>% group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 4 x 5
## # Groups:   Condition [2]
##   Condition Gender     N Total  Prop
##   <fct>     <fct>  <int> <int> <dbl>
## 1 Control   F         41   105  39.0
## 2 Control   M         64   105  61.0
## 3 Flipped   F         12    27  44.4
## 4 Flipped   M         15    27  55.6

# Distribution of Passed Students (Condition, Category):
t.man.exam %>% group_by(Condition, Category) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 6 x 5
## # Groups:   Condition [2]
##   Condition Category          N Total  Prop
##   <fct>     <fct>         <int> <int> <dbl>
## 1 Control   France           51   105  48.6
## 2 Control   Suisse.Autres    32   105  30.5
## 3 Control   Suisse.PAM       22   105  21.0
## 4 Flipped   France            9    27  33.3
## 5 Flipped   Suisse.Autres    13    27  48.1
## 6 Flipped   Suisse.PAM        5    27  18.5

# Distribution of Passed Students (Condition, Category, Gender):
t.man.exam %>% group_by(Condition, Category, Gender) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 12 x 6
## # Groups:   Condition, Category [6]
##    Condition Category      Gender     N Total  Prop
##    <fct>     <fct>         <fct>  <int> <int> <dbl>
##  1 Control   France        F         16    51  31.4
##  2 Control   France        M         35    51  68.6
##  3 Control   Suisse.Autres F         20    32  62.5
##  4 Control   Suisse.Autres M         12    32  37.5
##  5 Control   Suisse.PAM    F          5    22  22.7
##  6 Control   Suisse.PAM    M         17    22  77.3
##  7 Flipped   France        F          3     9  33.3
##  8 Flipped   France        M          6     9  66.7
##  9 Flipped   Suisse.Autres F          7    13  53.8
## 10 Flipped   Suisse.Autres M          6    13  46.2
## 11 Flipped   Suisse.PAM    F          2     5  40
## 12 Flipped   Suisse.PAM    M          3     5  60
```

12.1.4 Students who Dropped-Out of MAN

Amongst these MAN drop-outs students, following is the distribution in different conditions :

```
# Distribution of Passed Students (Condition, Gender):
t.man.drop %>% group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 3 x 5
## # Groups:   Condition [2]
##   Condition Gender     N Total  Prop
##   <fct>     <fct>  <int> <int> <dbl>
## 1 Control   F          2    13  15.4
## 2 Control   M         11    13  84.6
## 3 Flipped   M          2     2 100

# Distribution of Passed Students (Condition, Category):
t.man.drop %>% group_by(Condition, Category) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 5 x 5
## # Groups:   Condition [2]
##   Condition Category          N Total  Prop
##   <fct>     <fct>         <int> <int> <dbl>
## 1 Control   France            1    13  7.69
## 2 Control   Suisse.Autres     7    13 53.8
## 3 Control   Suisse.PAM        5    13 38.5
## 4 Flipped   France            1     2 50
## 5 Flipped   Suisse.Autres     1     2 50

# Distribution of Passed Students (Condition, Category, Gender):
t.man.drop %>% group_by(Condition, Category, Gender) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 6 x 6
## # Groups:   Condition, Category [5]
##   Condition Category      Gender     N Total  Prop
##   <fct>     <fct>         <fct>  <int> <int> <dbl>
## 1 Control   France        M          1     1 100
## 2 Control   Suisse.Autres F          2     7  28.6
## 3 Control   Suisse.Autres M          5     7  71.4
## 4 Control   Suisse.PAM    M          5     5 100
## 5 Flipped   France        M          1     1 100
## 6 Flipped   Suisse.Autres M          1     1 100
```

12.2 Alluvial Diagram – Year1

12.2.1 Preparing the Data

Students who passed the first semester amongst "New" volunteers :

```
# Selecting the relevant columns.
t.passed = t.passed %>% select(Condition, Category, Gender)

# Add a new variable: Result.BA1
t.passed$Result.BA1 = "Passed.BA1"

# Add an empty variable for MAN
```

```r
t.passed$Result.MAN = "NA"
```

Students who failed the first semester amongst "New" volunteers :

```r
# Selecting the relevant columns.
t.failed = t.failed %>% select(Condition, Category, Gender)

# Add a new variable: Result.BA1
t.failed$Result.BA1 = "Dropped.BA1"

# Add an empty variable for MAN
t.failed$Result.MAN = "NA"
```

Students who went to MAN amongst "New" volunteers :

```r
# Selecting the relevant columns.
t.man.exam = t.man.exam %>% select(Condition, Category, Gender)

# Add a new variable: Result.BA1
t.man.exam$Result.BA1 = "Failed.BA1"

# Add a new variable for MAN: Result.MAN
t.man.exam$Result.MAN = "MAN"
```

Students who dropped-out in MAN amongst "New" volunteers :

```r
# Selecting the relevant columns.
t.man.drop = t.man.drop %>% select(Condition, Category, Gender)

# Add a new variable: Result.BA1
t.man.drop$Result.BA1 = "Failed.BA1"

# Add a new variable for MAN: Result.MAN
t.man.drop$Result.MAN = "Dropped.MAN"
```

12.2.2 Combining Data

```r
# Combining all the datasets.
t.alluvial = rbind(t.passed, t.failed,
                   t.man.drop, t.man.exam)

# Cleaning-up.
rm(t.passed, t.failed, t.man.drop, t.man.exam)
```

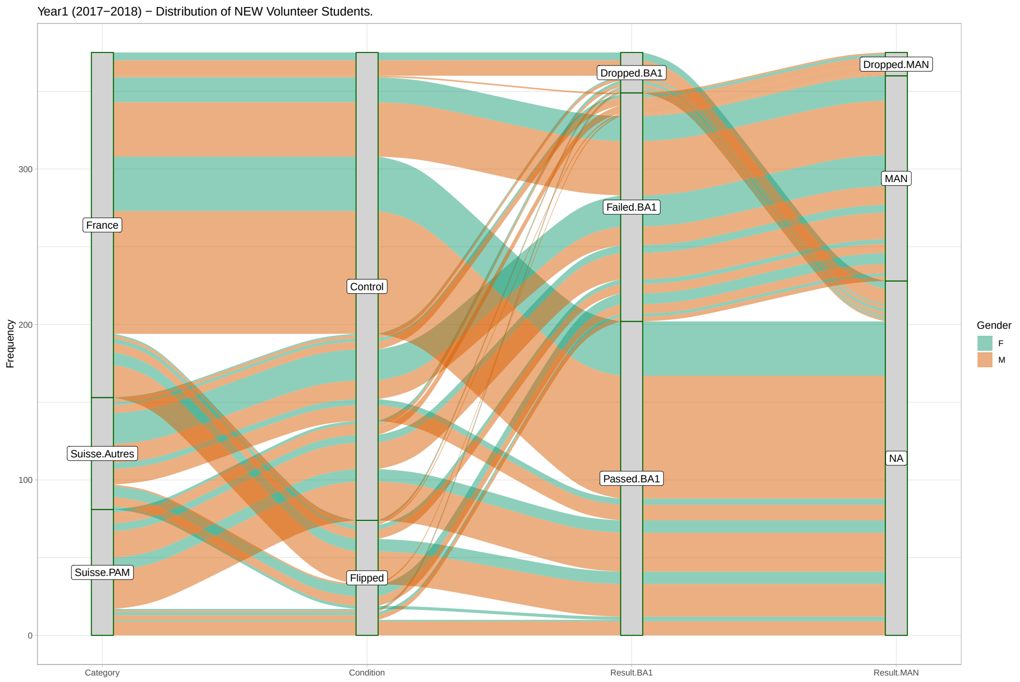
Computing the frequency of each category :

```r
# Summarizing the data to compute frequency.
t.plot = t.alluvial %>% group_by(Condition, Category, Gender, Result.BA1, Result.MAN) %>%
  summarise(Freq = n())
```

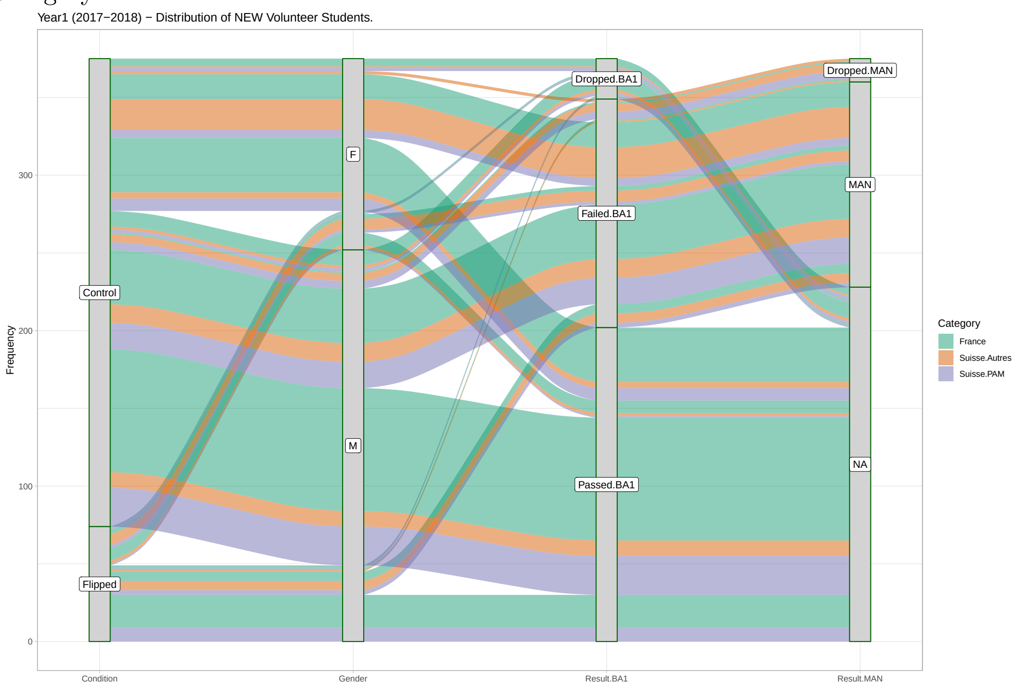
12.2.3 Visualization : Alluvial Plot

```r
# Checking if data is in order for visualization.
is_alluvia_form(t.plot, axes = 1:5, silent = TRUE)

## [1] TRUE
```

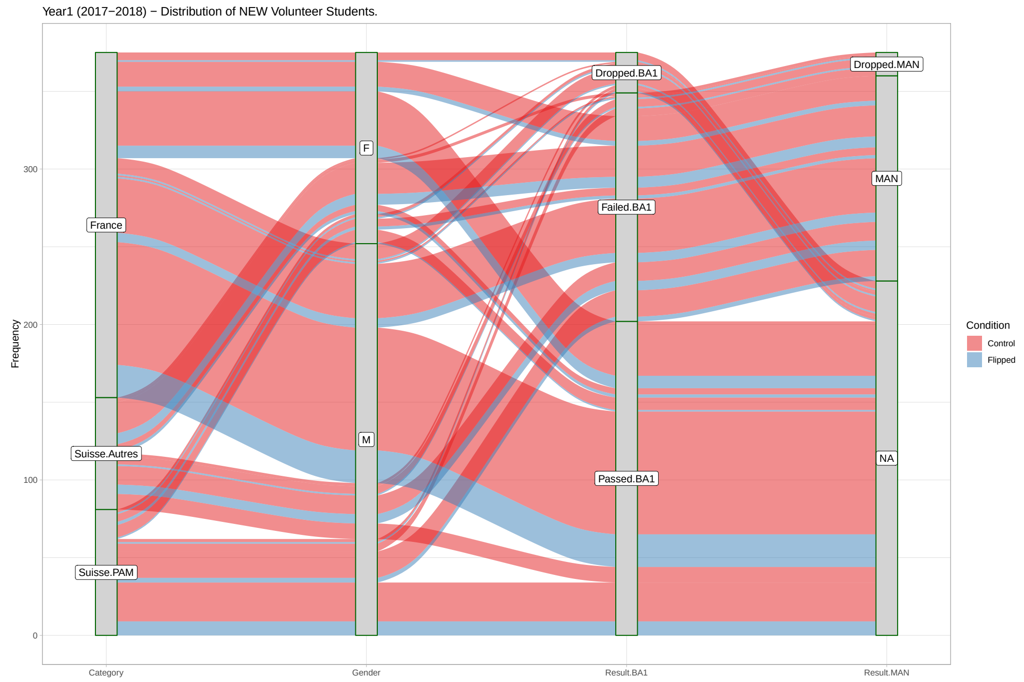
Plot with Gender as color :



Plot with Category as color :



Plot with Condition as color :



12.3 Year2

In Year2 there are in total 196 New students (excluding the Etranger.Autres). Now, in order to follow their progress, we will first extract their SCIPERS.

Following is the summary of the new students :

```
# Summary of distribution
dt.y2 %>% group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  spread(Gender, N)

## # A tibble: 2 x 3
## # Groups:   Condition [2]
##   Condition     F     M
##   <fct>     <int> <int>
## 1 Control      35    62
## 2 Flipped      35    64
```

... and the proportions :

```
# Computing the proportion of males, females.
t.stat = dt.y2 %>% group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  #mutate(Total = sum(N)) %>%
  #mutate(Prop.Female = (N / Total) * 100) %>%
  #spread(Gender, N)
  spread(Gender, N)
t.stat$Total = t.stat$F + t.stat$M
t.stat$Prop.Female = (t.stat$F / t.stat$Total) * 100
t.stat$Prop.Male = (t.stat$M / t.stat$Total) * 100
t.stat

## # A tibble: 2 x 6
## # Groups:   Condition [2]
##   Condition     F     M Total Prop.Female Prop.Male
##   <fct>     <int> <int> <int>       <dbl>     <dbl>
## 1 Control      35    62    97        36.1      63.9
## 2 Flipped      35    64    99        35.4      64.6
```

```r
# Convert ID.Anon to SCIPER
dt.y2$SCIPER = lapply(dt.y2$ID.Anon, GenerateSCIPER)
dt.y2$SCIPER = unlist(dt.y2$SCIPER)
```

12.3.1 Students who Passed

In order to compute the number of students who passed the Autumn Semester (2018-19), we will do a join operation with the dataset which shows the **total number of registrations in the Spring semester**.

```r
# Registrations in the Spring Semester
t.stat = t.grades %>% filter(
  YearName == "2018-2019" & PedagogicalCode == "BA2")
```

There are a few students, for whom the `Session` value is **'NA'**. There are in total 52 students with 'NA' value for `Session`.

```r
# Checking the students, who have a missing Session value.
na.val = t.stat %>% filter(is.na(Session)) %>%
  arrange(SCIPER)
```

If we look at the subjects for `na.val` dataset, we see that there are mainly the following subjects :

```r
# Overview of the name of Subjects
levels(as.factor(na.val$SubjectName))

## [1] "Biologie + Biochimie"
## [2] "Chimie générale"
## [3] "Enjeux mondiaux: climat A"
## [4] "Enjeux mondiaux: santé A"
## [5] "Fonctions et réactions organiques I"
## [6] "Introduction à la mécanique des structures"
## [7] "Physique générale : thermodynamique"
## [8] "Programmation II"
## [9] "Statique et dynamique"

# Overview of the SemesterName
levels(as.factor(na.val$SemesterName))

## [1] "Bachelor semestre 1" "Bachelor semestre 2"
```

Some students have **"Bachelor Semester 1"** for the `SemesterName` variable. Propbably these are the students who did not start their semester in September'2018. Let's have a detailed look at them :

```r
# SemesterName == "Bachelor semestre 1"
temp = na.val %>% filter(SemesterName == "Bachelor semestre 1")

# These students can be removed from na.val
na.val = na.val %>% filter(SemesterName == "Bachelor semestre 2")
```

Some of these students have `Grades` and others are drop-outs. Let's examine the ones who got grades :

```r
# Grade == Double value
temp = na.val %>% filter(Grade != "STATUT_NOTE_D")
```

It could be possible that despite their grades, these **4** students dropped-out after 1st semester.

To proceed, we can filter out the rows for `Session` column which contain <NA> values :

```r
# Filtering out based on Session
t.stat = t.stat %>% filter(Session == "ETE")
```

Now, the actual number of students who passed the *first* semester are the ones who remain. There are in total 294 students who passed. This number includes, New, Ex-MAN, Ex-CMS, etc.

```
# Now, we only extract the distinct SCIPERs of students who passed.
t.sciper = t.stat %>% select(SCIPER) %>%
  distinct()
```

Following dataset contains the complete details of these student :

```
# Filtering the SCIPERS which are contained in t.sciper dataset
t.sciper$SCIPER = as.character(t.sciper$SCIPER)
#t1 = subset(dt.y1, dt.y1$SCIPER %in% unlist(t.sciper))
t.passed = merge(x = dt.y2,
                 y = t.sciper,
                 by = "SCIPER")

# Cleaning-up
rm(t.sciper, t.stat)
```

Now, the total number of students who passed amongst the **"New"** students are **102**.
Amongst these passed students, following is the distribution in different conditions :

```
# Distribution of Passed Students (Condition, Gender):
t.passed %>% group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 4 x 5
## # Groups:   Condition [2]
##   Condition Gender     N Total  Prop
##   <fct>     <fct>  <int> <int> <dbl>
## 1 Control   F         15    49  30.6
## 2 Control   M         34    49  69.4
## 3 Flipped   F         19    53  35.8
## 4 Flipped   M         34    53  64.2

# Distribution of Passed Students (Condition, Category):
t.passed %>% group_by(Condition, Category) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 6 x 5
## # Groups:   Condition [2]
##   Condition Category          N Total  Prop
##   <fct>     <fct>         <int> <int> <dbl>
## 1 Control   France           26    49  53.1
## 2 Control   Suisse.Autres     6    49  12.2
## 3 Control   Suisse.PAM       17    49  34.7
## 4 Flipped   France           32    53  60.4
## 5 Flipped   Suisse.Autres     7    53  13.2
## 6 Flipped   Suisse.PAM       14    53  26.4

# Distribution of Passed Students (Condition, Category, Gender):
t.passed %>% group_by(Condition, Category, Gender) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 12 x 6
## # Groups:   Condition, Category [6]
##    Condition Category      Gender     N Total  Prop
##    <fct>     <fct>         <fct>  <int> <int> <dbl>
##  1 Control   France        F          7    26  26.9
##  2 Control   France        M         19    26  73.1
##  3 Control   Suisse.Autres F          4     6  66.7
```

```
##  4 Control   Suisse.Autres M          2     6  33.3
##  5 Control   Suisse.PAM    F          4    17  23.5
##  6 Control   Suisse.PAM    M         13    17  76.5
##  7 Flipped   France        F         13    32  40.6
##  8 Flipped   France        M         19    32  59.4
##  9 Flipped   Suisse.Autres F          3     7  42.9
## 10 Flipped   Suisse.Autres M          4     7  57.1
## 11 Flipped   Suisse.PAM    F          3    14  21.4
## 12 Flipped   Suisse.PAM    M         11    14  78.6
```

12.3.2 Students who Dropped their studies after 1st Semester

So, we will create a separate dataset called `t.dropped` :

```
# Data of failed students (including drop-out)
t.dropped = merge(x = dt.y2,
                  y = na.val %>% select(SCIPER) %>% distinct(),
                  by = "SCIPER")
```

We see that there are **0 New** students who dropped out after the first semester.

12.3.3 Students who went to MAN

Looking at the `t.grades` dataset, we observe that there are some students who have the `PedagogicalCode` as **MAN**. So, let's start by filtering them out. We will only filter out the students who belong to the `AcademicYear` of 2018-2019.

```
# Filtering out the MAN students.
t.stat = t.grades %>% filter(
  YearName == "2018-2019" & PedagogicalCode == "MAN"
)
```

We see that the students who appeared in the MAN semester take the follwoing courses :

```
# Courses taken by MAN Students
levels(as.factor(t.stat$SubjectName))

## [1] "Mathématiques 1A (pour MAN)" "Mathématiques 1B (pour MAN)"
## [3] "Mathématiques 2 (pour MAN)"  "Physique (pour MAN)"
```

There are in total **125** students in MAN right now.

Since, these students have not yet taken their MAN exams, we cannot say if they passed their MAN or not. So, we will make just one dataset called `t.man` :

```
# New students who are in MAN
t.man = merge(x = dt.y2,
              y = t.stat %>% select(SCIPER) %>% distinct(),
              by = "SCIPER")
```

Amongst these MAN students, following is the distribution in different conditions :

```
# Distribution of Passed Students (Condition, Gender):
t.man %>% group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 4 x 5
## # Groups:   Condition [2]
##   Condition Gender     N Total  Prop
##   <fct>     <fct>  <int> <int> <dbl>
## 1 Control   F         18    44  40.9
## 2 Control   M         26    44  59.1
## 3 Flipped   F         14    43  32.6
## 4 Flipped   M         29    43  67.4
```

```
# Distribution of Passed Students (Condition, Category):
t.man %>% group_by(Condition, Category) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 6 x 5
## # Groups:   Condition [2]
##   Condition Category          N Total  Prop
##   <fct>     <fct>         <int> <int> <dbl>
## 1 Control   France           26    44  59.1
## 2 Control   Suisse.Autres    13    44  29.5
## 3 Control   Suisse.PAM        5    44  11.4
## 4 Flipped   France           17    43  39.5
## 5 Flipped   Suisse.Autres    17    43  39.5
## 6 Flipped   Suisse.PAM        9    43  20.9

# Distribution of Passed Students (Condition, Category, Gender):
t.man %>% group_by(Condition, Category, Gender) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 12 x 6
## # Groups:   Condition, Category [6]
##    Condition Category      Gender     N Total  Prop
##    <fct>     <fct>         <fct>  <int> <int> <dbl>
##  1 Control   France        F         12    26  46.2
##  2 Control   France        M         14    26  53.8
##  3 Control   Suisse.Autres F          4    13  30.8
##  4 Control   Suisse.Autres M          9    13  69.2
##  5 Control   Suisse.PAM    F          2     5  40
##  6 Control   Suisse.PAM    M          3     5  60
##  7 Flipped   France        F          5    17  29.4
##  8 Flipped   France        M         12    17  70.6
##  9 Flipped   Suisse.Autres F          7    17  41.2
## 10 Flipped   Suisse.Autres M         10    17  58.8
## 11 Flipped   Suisse.PAM    F          2     9  22.2
## 12 Flipped   Suisse.PAM    M          7     9  77.8
```

12.4 Alluvial Diagram – Year2

12.4.1 Preparing the Data

Students who passed the first semester amongst "New" volunteers :

```
# Selecting the relevant columns.
t.passed = t.passed %>% select(Condition, Category, Gender)

# Add a new variable: Result.BA1
t.passed$Result.BA1 = "Passed"
```

Students who went to MAN amongst "New" volunteers :

```
# Selecting the relevant columns.
t.man = t.man %>% select(Condition, Category, Gender)

# Add a new variable: Result.BA1
t.man$Result.BA1 = "MAN"
```

12.4.2 Combining Data

```r
# Combining all the datasets.
t.alluvial = rbind(t.passed, t.man)
```

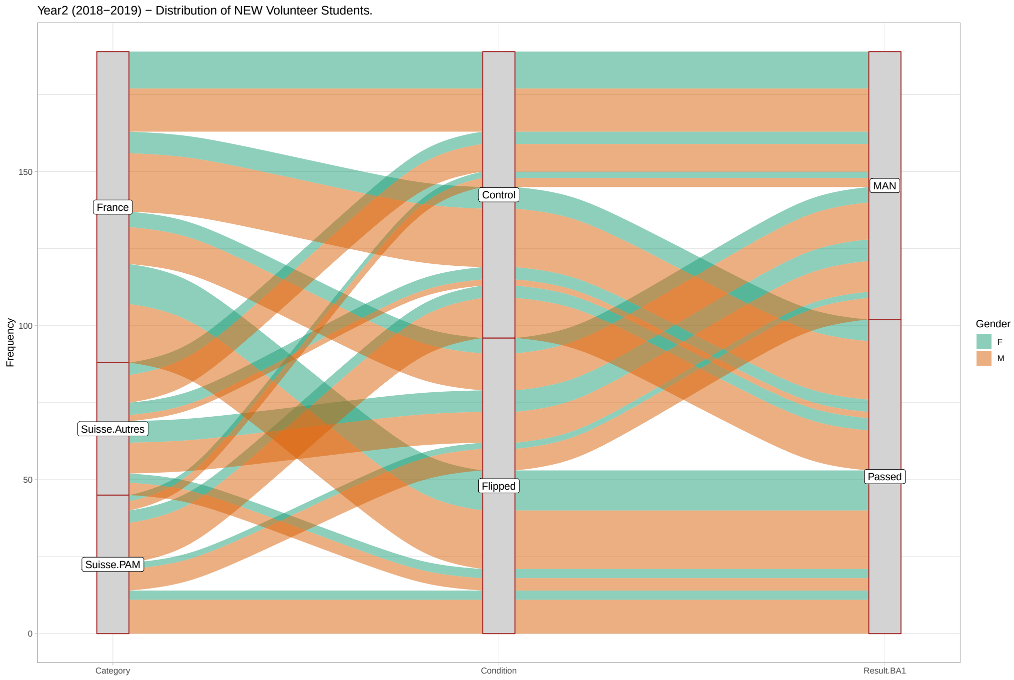
Computing the frequency of each category :

```r
# Summarizing the data to compute frequency.
t.plot = t.alluvial %>% group_by(Condition, Category, Gender, Result.BA1) %>%
  summarise(Freq = n())
```

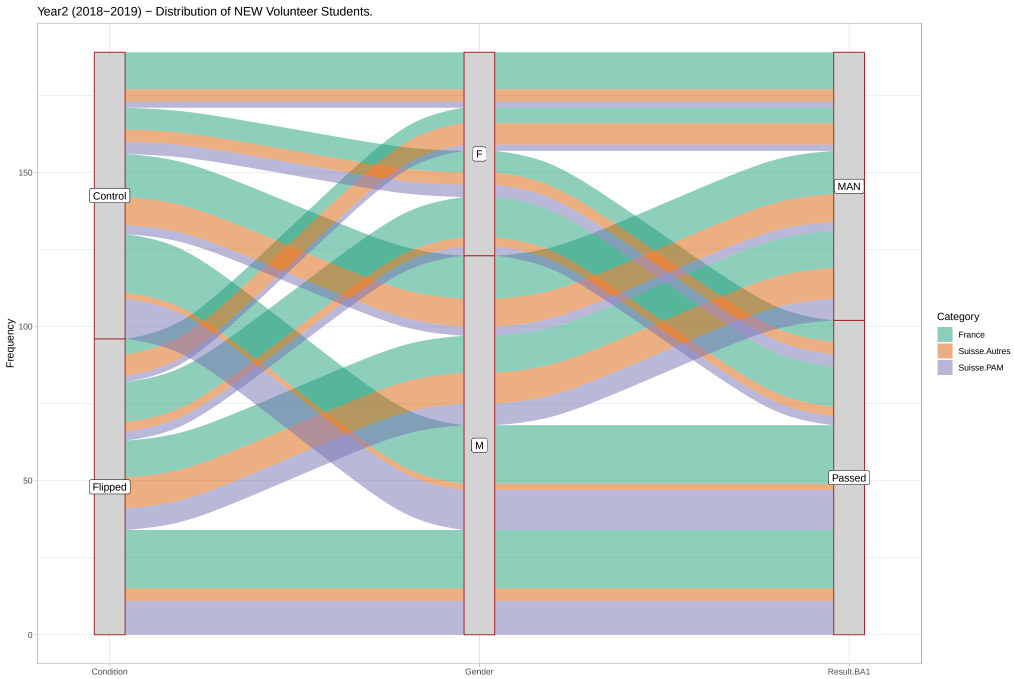
12.4.3 Visualization : Alluvial Plot

```r
# Checking if data is in order for visualization.
is_alluvia_form(t.plot, axes = 1:4, silent = TRUE)

## [1] TRUE
```

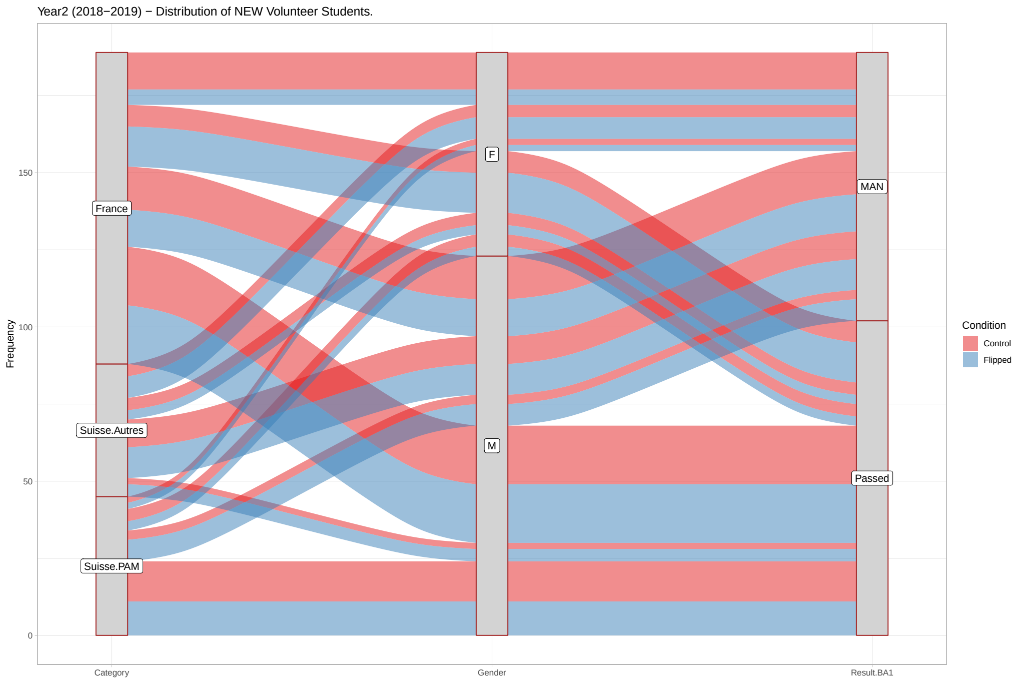
Plot with Gender as color :



Plot with Category as color :



Plot with Condition as color :



13 MAN Students – Rate of Failure across Condition

In this section, we will analyze *whether there is a difference in failure/droup-out rates in the "flipped" and "control" condition*.

NOTE : Currently this analysis is performed only for YEAR1 because we do not have the grades for YEAR2.

13.1 Year1

13.1.1 Filtering out Students who went to MAN

```
# Creating a data frame out of students who went to MAN.
t.man = t.grades %>% filter(
  YearName == "2017-2018" & PedagogicalCode == "MAN"
)
```

```
# Summarizing the distribution of the MAN students.
temp = merge(x = dt.y1,
             y = t.man %>% distinct(SCIPER),
             by = "SCIPER")
temp %>% group_by(Condition) %>% summarise(N = n())

## # A tibble: 2 x 2
##   Condition     N
##   <fct>     <int>
## 1 Control     117
## 2 Flipped      29

# Cleaning up
rm(temp)
```

13.1.2 Filtering out Students who Dropped Out from MAN

First, we will examine if students took exams of all the subjects or if they did not appear for exams for a few subjects.

```
# Computing the number of exams taken by each student.
t.stat = t.man %>%
  filter(!(Grade =="STATUT_NOTE_NA" | is.na(Grade) | Grade == "STATUT_NOTE_M")) %>%
```

```
  group_by(SCIPER) %>%
  summarise(N = n())
```

From a quick look, we see that there are 175 students who appeared in all the *four* exams. Also, there are 2 students who did not take exams of all the subjects. We will consider these latter set of students as failed and remove them from passed students.

```
# Student who did not appear for all exams: Drop-Outs
t.less.exams = t.stat %>% filter(N < 4) %>%
  select(SCIPER)
```

We will also make list of students who did not appear for their exams or reported sickness.

```
# Students who did not appear for exams or reported sick.
t.miss.exams = t.man %>%
  filter(Grade =="STATUT_NOTE_NA" | is.na(Grade) | Grade == "STATUT_NOTE_M") %>%
  group_by(SCIPER) %>%
  summarise(N = n()) %>%
  select(SCIPER)
```

Now, the total list of students who dropped-out of MAN is :

```
# Compiling the list of students who dropped-out:
t.man.dropout = rbind(
  t.less.exams,
  t.miss.exams
)

# Cleaning up.
rm(t.less.exams, t.miss.exams, t.stat)
```

There are in total 24 students who dropped out of MAN.

13.1.3 Filtering out Students who Failed from MAN

To compute the students who failed in MAN, we will see the list of students who took all 4 exams, and then their mean grade is less than 4.0.

```
# Identifying students who appeared for all exams.
t.all.exams = t.man %>%
  filter(!(Grade =="STATUT_NOTE_NA" | is.na(Grade) | Grade == "STATUT_NOTE_M")) %>%
  group_by(SCIPER) %>%
  summarise(N.Exams = n()) %>%
  filter(N.Exams == 4) %>%
  select(SCIPER)
```

Now, filter out students whose SCIPERs are in the above list :

```
# Filter students who took all exams.
t.stat = merge(x = t.man,
               y = t.all.exams,
               by = "SCIPER")
```

Computing the mean grade.

```
# Convert Grade from Character to Double
t.stat$Grade = as.double(t.stat$Grade)

# Compute the mean grade
t.stat = t.stat %>%
  group_by(SCIPER) %>%
  summarise(N.Exams = n(),
            Mean.MAN.Grade = mean(Grade))
```

Students who have failed, have a mean grade less than 4.0 :

```
t.man.failed = t.stat %>%
  filter(Mean.MAN.Grade < 4.0)
```

We see that there are in total 61 students who failed in their MAN semester.

13.1.4 Analyzing MAN Dropouts

In this section, we will perform an Inner Join with the volunteer dataset, and then analyze the distribution of dropouts.

```
# Compiling dataset by joining with volunteer data.
t.stat = merge(
  x = dt.y1,
  y = t.man.dropout,
  by = "SCIPER"
)
```

Computing the proportion of students in different `Conditions` :

```
# Prop. of students who dropped-out.
t.stat %>% group_by(Condition) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 2 x 4
##   Condition     N Total  Prop
##   <fct>     <int> <int> <dbl>
## 1 Control      14    16  87.5
## 2 Flipped       2    16  12.5
```

... also exploring the distribution across `Categories` :

```
# Prop. of students (in different Categories) who dropped-out.
t.stat %>% group_by(Condition, Category) %>%
  summarise(N = n()) %>%
  spread(Category, N)

## # A tibble: 2 x 4
## # Groups:   Condition [2]
##   Condition France Suisse.Autres Suisse.PAM
##   <fct>      <int>         <int>      <int>
## 1 Control        1             7          6
## 2 Flipped        1             1         NA
```

... also exploring the distribution across `Genders` :

```
# Prop. of students (in different Gender) who dropped-out.
t.stat %>% group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  spread(Gender, N)

## # A tibble: 2 x 3
## # Groups:   Condition [2]
##   Condition     F     M
##   <fct>     <int> <int>
## 1 Control       2    12
## 2 Flipped      NA     2
```

13.1.5 Analyzing MAN Failures

In this section, we will perform an Inner Join with the volunteer dataset, and then analyze the distribution of failures.

```r
# Compiling dataset by joining with volunteer data.
t.stat = merge(
  x = dt.y1,
  y = t.man.failed,
  by = "SCIPER"
)
```

Computing the proportion of students in different `Conditions` :

```r
t.stat %>% group_by(Condition) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 2 x 4
##   Condition     N Total  Prop
##   <fct>     <int> <int> <dbl>
## 1 Control      45    56  80.4
## 2 Flipped      11    56  19.6
```

... also exploring the distribution across `Categories` :

```r
# Prop. of students (in different Categories) who dropped-out.
t.stat %>% group_by(Condition, Category) %>%
  summarise(N = n()) %>%
  spread(Category, N)

## # A tibble: 2 x 4
## # Groups:   Condition [2]
##   Condition France Suisse.Autres Suisse.PAM
##   <fct>      <int>         <int>      <int>
## 1 Control       14            18         13
## 2 Flipped        3             6          2
```

... also exploring the distribution across `Genders` :

```r
# Prop. of students (in different Gender) who dropped-out.
t.stat %>% group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  spread(Gender, N)

## # A tibble: 2 x 3
## # Groups:   Condition [2]
##   Condition     F     M
##   <fct>     <int> <int>
## 1 Control      21    24
## 2 Flipped       5     6
```

14 Students at the Margin – Rate of Failures/Drop-outs

In this section, we will examine the students who are at *risk of failing as they are just passing in the first semester.* We would like to study, *whether these students fail/dropout more in one `Condition` than another.*

NOTE : At this stage, this analysis can only be conducted for YEAR1 students.

14.1 Year1

14.1.1 Filtering BA1 Students Who Took Exams

Students who passed BA1 are the ones who registered in BA2. In addition, the passed students scored a mean `Grade` of 4.0 or above.

```r
# Number of students registered in BA1
t.BA1 = t.grades %>%
  filter(YearName == "2017-2018" & PedagogicalCode == "BA1")
```

... now, removing the dropouts.

```r
# Dataset without the dropouts.
t.stat = t.BA1 %>%
  filter(!(Session == "RAT" | is.na(Session)))
```

... also removing people who failed to appear in their exams.

```r
# Identifying students who appeared for their exams.
t.BA1.exams = t.stat %>%
  filter(!(Grade %in% c("STATUT_NOTE_D", "STATUT_NOTE_M", "STATUT_NOTE_NA"))) %>%
  distinct(SCIPER)
```

14.1.2 Students Who Passed/Failed BA1

... are the ones who were registered for BA2.

```r
# Students who registered for BA2
t.BA2 = t.grades %>%
  filter(YearName == "2017-2018" & PedagogicalCode == "BA2" & Session == "ETE") %>%
  distinct(SCIPER)
```

... now performing the join operation with **BA1** dataset to compute their mean grades.

```r
# Filtering students who passed BA1 definitely.
t.BA1.passed = merge(x = t.grades,
                     y = t.BA2,
                     by = "SCIPER")

# Filtering based on Academic Year.
t.BA1.passed = t.BA1.passed %>%
  filter(YearName == "2017-2018" & PedagogicalCode == "BA1" & Session == "HIV")
```

... now identifying the students who failed their BA1. One way to achieve this is to identify the students who were registered for BA1 but are not in the BA2 list.

```r
# Students who registered for BA1
t.BA1 = t.grades %>%
  filter(YearName == "2017-2018" & PedagogicalCode == "BA1" & Session == "HIV") %>%
  distinct(SCIPER)
```

... looking at the students who failed their BA1 :

```r
# Students who failed/dropout BA1
t.BA1.failed = anti_join(t.BA1,
                         t.BA2,
                         by = "SCIPER")
```

Joining the above datasets with the volunteer data to have all the details.

```r
# Joining with the Volunteer data.
# Passed BA1
y1.BA1.passed = merge(x = dt.y1,
                      y = t.BA2,
                      by = "SCIPER")

# Failed/Dropouts
y1.BA1.failed = merge(x = dt.y1,
                      y = t.BA1.failed,
                      by = "SCIPER")
```

14.1.3 Computing Mean Grade of BA1 studens

Printing the number of exams taken by each student.

```
# Summarizing the number of exams taken by each student.
t.stat = t.BA1.passed %>%
  group_by(SCIPER) %>%
  summarise(N = n())
```

We see that there is no uniformity in the number of exams taken by the students. Some students took 8 exams and others only 1. Is there a threshold that we could use to filter out the true students to compute the mean grade ?

Let's compute the mean grade anyways, and we will figure out how to deal with this problem later.

```
# Convert Grade from Character to Double
t.BA1.passed$Grade = as.double(t.BA1.passed$Grade)

# Computing the mean Grade.
t.stat = t.BA1.passed %>%
  group_by(SCIPER) %>%
  summarise(N.Exams = n(),
            Mean.Grade = mean(Grade))
```

14.1.4 Filtering the BA1 Students at the Margin

```
# Filtering out students at the margin.
t.BA1.Margin = t.stat %>% filter(Mean.Grade >= 3.5 & Mean.Grade <= 4.5)
```

Joining this dataset with the volunteer dataset.

```
# Volunteers who are at the margin.
y1.BA1.margin = merge(x = dt.y1,
                      y = t.BA1.Margin,
                      by = "SCIPER")
```

Summarizing the students at the margin :

```
# Distribution across Condition
y1.BA1.margin %>%
  group_by(Condition) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 2 x 4
##   Condition     N Total  Prop
##   <fct>     <int> <int> <dbl>
## 1 Control      84   108  77.8
## 2 Flipped      24   108  22.2

# Distribution across Condition and Gender
y1.BA1.margin %>%
  group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 4 x 5
## # Groups:   Condition [2]
##   Condition Gender     N Total  Prop
##   <fct>     <fct>  <int> <int> <dbl>
## 1 Control   F         26    84  31.0
## 2 Control   M         58    84  69.0
## 3 Flipped   F          7    24  29.2
## 4 Flipped   M         17    24  70.8
```

```
# Distribution across Condition and Category
y1.BA1.margin %>%
  group_by(Condition, Category) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 6 x 5
## # Groups:   Condition [2]
##   Condition Category          N Total  Prop
##   <fct>     <fct>         <int> <int> <dbl>
## 1 Control   France           60    84 71.4
## 2 Control   Suisse.Autres    10    84 11.9
## 3 Control   Suisse.PAM       14    84 16.7
## 4 Flipped   France           16    24 66.7
## 5 Flipped   Suisse.Autres     1    24  4.17
## 6 Flipped   Suisse.PAM        7    24 29.2
```

14.1.5 Filtering the Students who Passed/Failed BA2

Students who did pass BA2 are the ones who registered for BA3.

```
# All Students who registered for BA2
t.BA2 = t.grades %>%
  filter(YearName == "2017-2018" & PedagogicalCode == "BA2" & Session == "ETE") %>%
  distinct(SCIPER)
```

```
# Filtering students who passed BA2
t.BA3 = t.grades %>%
  filter(YearName == "2018-2019" & PedagogicalCode == "BA3" & Session == "HIV") %>%
  distinct(SCIPER)

# Choosing the ones who were registered for BA2 (and not repeaters)
t.BA2.passed = merge(x = t.BA2,
                     y = t.BA3,
                     by = "SCIPER")
```

Now, the students who failed/dropped-out BA2 :

```
# Filtering students who passed BA2
t.BA2.failed = anti_join(x = t.BA2,
                         y = t.BA2.passed,
                         by = "SCIPER")
```

14.1.6 Identifying Risky Students who Passed/Failed in BA2

Risky students who passed BA2 :

```
# Marginal Students who passed
margin.passed.BA2 = merge(x = y1.BA1.margin,
                          y = t.BA2.passed,
                          by = "SCIPER")

# Summarising the passed marginal students
margin.passed.BA2 %>% group_by(Condition) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 2 x 4
##   Condition     N Total  Prop
```

```
##   <fct>     <int> <int> <dbl>
## 1 Control      43    54  79.6
## 2 Flipped      11    54  20.4

# Distribution across Condition and Gender
margin.passed.BA2 %>%
  group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 4 x 5
## # Groups:   Condition [2]
##   Condition Gender     N Total  Prop
##   <fct>     <fct>  <int> <int> <dbl>
## 1 Control   F         13    43  30.2
## 2 Control   M         30    43  69.8
## 3 Flipped   F          3    11  27.3
## 4 Flipped   M          8    11  72.7

# Distribution across Condition and Category
margin.passed.BA2 %>%
  group_by(Condition, Category) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 6 x 5
## # Groups:   Condition [2]
##   Condition Category          N Total  Prop
##   <fct>     <fct>         <int> <int> <dbl>
## 1 Control   France           31    43 72.1
## 2 Control   Suisse.Autres     6    43 14.0
## 3 Control   Suisse.PAM        6    43 14.0
## 4 Flipped   France            7    11 63.6
## 5 Flipped   Suisse.Autres     1    11  9.09
## 6 Flipped   Suisse.PAM        3    11 27.3
```

Risky students who failed BA2 :

```
# Marginal Students who failed
margin.failed.BA2 = merge(x = y1.BA1.margin,
                          y = t.BA2.failed,
                          by = "SCIPER")

# Summarising the passed marginal students
margin.failed.BA2 %>% group_by(Condition) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 2 x 4
##   Condition     N Total  Prop
##   <fct>     <int> <int> <dbl>
## 1 Control      41    54  75.9
## 2 Flipped      13    54  24.1

# Distribution across Condition and Gender
margin.failed.BA2 %>%
  group_by(Condition, Gender) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)

## # A tibble: 4 x 5
```

```
## # Groups:   Condition [2]
##   Condition Gender     N Total  Prop
##   <fct>     <fct>  <int> <int> <dbl>
## 1 Control   F         13    41  31.7
## 2 Control   M         28    41  68.3
## 3 Flipped   F          4    13  30.8
## 4 Flipped   M          9    13  69.2
```

```
# Distribution across Condition and Category
margin.failed.BA2 %>%
  group_by(Condition, Category) %>%
  summarise(N = n()) %>%
  mutate(Total = sum(N)) %>%
  mutate(Prop = (N / Total) * 100)
```

```
## # A tibble: 5 x 5
## # Groups:   Condition [2]
##   Condition Category          N Total  Prop
##   <fct>     <fct>         <int> <int> <dbl>
## 1 Control   France           29    41 70.7
## 2 Control   Suisse.Autres     4    41  9.76
## 3 Control   Suisse.PAM        8    41 19.5
## 4 Flipped   France            9    13 69.2
## 5 Flipped   Suisse.PAM        4    13 30.8
```

15 Disconnect

This is the final step, to close the connection with the database.

```
dbDisconnect(connection)
```