Appendix

Tom Gause and Liam O'Brien

May 22, 2022

In the appendix is the R code we used for our data modifications and statistical tests as well as histograms of our sampled distributions.

A. Raw Data Collection

```
# Set seed so experiments can be replicated
set.seed(31)

############################################################
# CLASS
############################################################
# Extra row ommitted, 4 "prefer not to answer"
Class.Z <- c(2022, 2022.5, 2023, 2023.5, 2024, 2024.5, 2025, 2025.5, 0)
vals.Z <- c(180, 80, 190, 79, 242, 73, 236, 50, 4)
Class.M <- c(2022,2023,2024,2025)
# Rounded data based on proportions from fall 2021
vals.M <- c(600,690,727,638)
Class.Data.Z <- data.frame(Class = Class.Z,
                           Zeitgeist = vals.Z)
# We assume that the administration combined feb and reg classes
vals.Z2 <- c(180+80, 190+79, 242+73, 236+50)
Class.Data.Raw <- data.frame(Class = Class.M,
                             Zeitgeist = vals.Z2,
                             Middlebury = vals.M)


############################################################
# GEOGRAPHY
############################################################
Geography <- c("New England", "Mideast", "Southeast", "Great Lakes", "Plains",
               "Southwest", "Rocky Mountain", "Far West",
               "Inernational/Outside the U.S.", "I prefer not to answer")
vals.Z <- c(340, 272, 101, 63, 43, 33, 39, 136, 103, 4)
vals.M <- c(698, 604, 203, 164, 70, 78, 77, 301, 0, 143)
Geography.Data.Raw <- data.frame(Geographic.Location = Geography,
                                 Zeitgeist = vals.Z,
                                 Middlebury = vals.M)


############################################################
```

```r
# GENDER
############################################################
Gender <- c("Cisgender Female", "Cisgender Male", "Non-Binary",
            "I prefer not to answer", "Transgender Male", "Transgender Female",
            "These options don't define me")
vals.Z <- c(652, 401, 63, 10, 13, 5, 21)
# The data from Middlebury only has binary options for "Women" and "Men"
vals.M <- c(1416, 1240, 0, 0, 0, 0, 0)
Gender.Data.Raw <- data.frame(Gender = Gender,
                              Zeitgeist = vals.Z,
                              Middlebury = vals.M)


############################################################
# RACE
############################################################
# This data is tricky, because students had the option to check
# all boxes that applied
Race <- c("White", "Asian", "Hispanic or Latino Origin",
          "Black or African American", "Middle Eastern or North African",
          "I prefer not to answer", "American Indian or Alaskan Native",
          "Native Hawaiian or other Pacific Islander", "International",
          "Two or more races", "Race and/or ethnicity unknown")
vals.Z <- c(911, 173, 96, 57, 17, 13, 8, 5, 0, 0, 0)
# "Middle Eastern or North African" and "I prefer not to answer" are not
# categories in Middlebury's data
# "International", "Two or more races", and "Race and/or ethnicity unknown"
# are additional categories in Middlebury data
vals.M <- c(1556, 199, 287, 125, 0, 0, 3, 0, 308, 157, 19)
Race.Data.Raw <- data.frame(Race = Race,
                        Zeitgeist = vals.Z,
                        Middlebury = vals.M)


############################################################
# MAJORS
############################################################
Majors.Z <- c("Undeclared", "Environmental Studies", "Economics",
              "Political Science", "Neuroscience", "Computer Science",
              "English & American Literatures", "Psychology",
              "International & Global Studies",
              "Molecular Biology & Biochemistry", "Biology", "Mathematics",
              "International Politics & Economics", "History",
              "History of Art & Architecture", "Film & Media Culture",
              "Geography", "Anthropology", "Physics", "Spanish", "Chemistry",
              "Sociology", "Theatre", "Education Studies", "Music",
              "Philosophy", "Chinese", "Geology", "Biochemistry",
              "Independent Scholar", "Studio Art", "I prefer not to answer",
              "Arabic", "Gender, Sexuality, & Feminist Studies", "German",
              "Classics", "Japanese Studies", "Dance", "Russian", "French",
              "Religion", "American Studies", "Comparative Literature",
              "Literary Studies", "Black Studies",
              "History-Science, Medicine, and Technology", "Italian")
```

```
vals.Z <- c(162, 125, 113, 70, 68, 88, 60, 58, 50, 38, 35, 48, 44, 35, 33,
            30, 25, 24, 25, 23, 22, 24, 17, 17, 15, 14, 13, 11, 10, 10, 7, 12,
            12, 12, 12, 6, 5, 5, 5, 4, 4, 3, 3, 3, 2, 0, 0)
Major.Data.Z <- data.frame(Major = Majors.Z,
                           Zeitgeist = vals.Z)
Majors.M <- c("Dance","Film & Media Culture","Music",
              "Studio Art","Theatre","Classics","History",
              "History-Science, Medicine, and Technology",
              "Philosophy","Religion", "American Studies","Black Studies",
              "Environmental Studies","Gender, Sexuality, & Feminist Studies",
              "Independent Scholar",
              "International Politics & Economics",
              "International & Global Studies","Neuroscience","Arabic",
              "Chinese",
              "French","German","Italian",
              "Japanese Studies","Russian","Spanish","Comparative Literature",
              "English & American Literatures",
              "Literary Studies","Biochemistry","Biology",
              "Chemistry","Computer Science","Geology","Mathematics",
              "Molecular Biology & Biochemistry","Physics","Economics",
              "Education Studies","Geography",
              "Political Science","Psychology","Anthropology","Sociology",
              "Undeclared", "I prefer not to answer",
              "History of Art & Architecture")

vals.M <- c(5,43,19,15,21,9,55,0,24,11,8,3,166,13,6,80,75,152,9,
              20,7,8,3,14,7,21,5,93,7,20,86,29,158,29,68,80,46,298,13,34,
              99,118,41,30,923,0,61)

Major.Data.M <- data.frame(Major = Majors.M,
                           Middlebury = vals.M)
Major.Data.Raw <- merge(Major.Data.M, Major.Data.Z, all = TRUE)
```

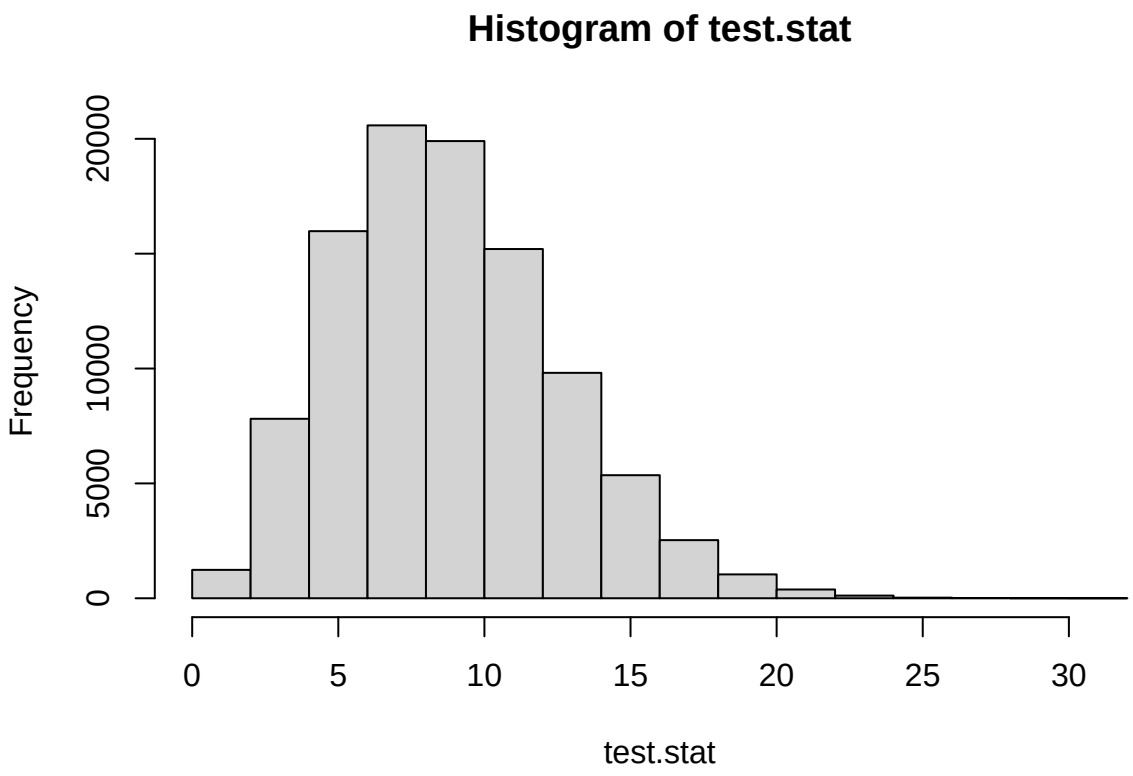
B. Phase 1 Data Preparation

```
library(tidyverse)
library(dplyr)

Class.Data.Phase1 <- Class.Data.Raw

Geography.Data.Phase1 <- Geography.Data.Raw

Gender.Data.Phase1 <- Gender.Data.Raw %>%
  filter(Gender == c("Cisgender Female", "Cisgender Male"))

# For some reason, the filter wasn't filtering the
# "American Indian or Alaskan Native" category, so I did it manually...
Race.Data.Phase1 <- Race.Data.Raw %>%
  filter(Race == c("White", "Asian", "Hispanic or Latino Origin",
                   "Black or African American"))
Race.Data.Phase1 <- rbind(Race.Data.Phase1, Race.Data.Raw[7,])
```

```r
Major.Data.Phase1 <- Major.Data.Raw

# Adjust p-value for multiple tests
alpha <- 0.05
k <- 5 # 5 tests
alpha.star <- 1 - (1-alpha)^(1/k)
```

C. Phase 1 Permutation Test for Class

```r
# Number of times that we will simulate samples for our sampling distribution
N <- 100000
# The population of graduation years provided by the administration
Population <- c(rep("2022", 600), rep("2023", 690),
                rep("2024", 727), rep("2025",638))
pop.size <- length(Population)
# Expected proportions
eprop22 <- 600/pop.size
eprop23 <- 690/pop.size
eprop24 <- 727/pop.size
eprop25 <- 638/pop.size
samp.size <- sum(Class.Data.Phase1$Zeitgeist)
# Expected values for sample of our size
epop22 <- eprop22*samp.size
epop23 <- eprop23*samp.size
epop24 <- eprop24*samp.size
epop25 <- eprop25*samp.size
expected.values <- c(epop22, epop23, epop24, epop25)
# test statistic is MAE
actual.test.stat <- (1/4)*sum(abs(Class.Data.Phase1$Zeitgeist - expected.values))
test.stat <- NULL
# Simulating our sampling distribution
for(i in 1:N){
  # Taking a sample from our population
  samp <- sample(Population, samp.size, replace = FALSE)

  samp22 <- sum(samp == "2022")
  samp23 <- sum(samp == "2023")
  samp24 <- sum(samp == "2024")
  samp25 <- sum(samp == "2025")

  actual.samp.values <- c(samp22,samp23,samp24,samp25)

  # Storing the test statistic for each simulated sample
  test.stat[i] <- (1/4)*sum(abs(actual.samp.values - expected.values))
}
hist(test.stat)
```




```
# p-value: probability of finding something more extreme than actual.test.stat
p.value.class <- mean(test.stat >= actual.test.stat)
print(paste0("Alpha-Star: ", alpha.star))
```

```
## [1] "Alpha-Star: 0.0102062183130115"
```

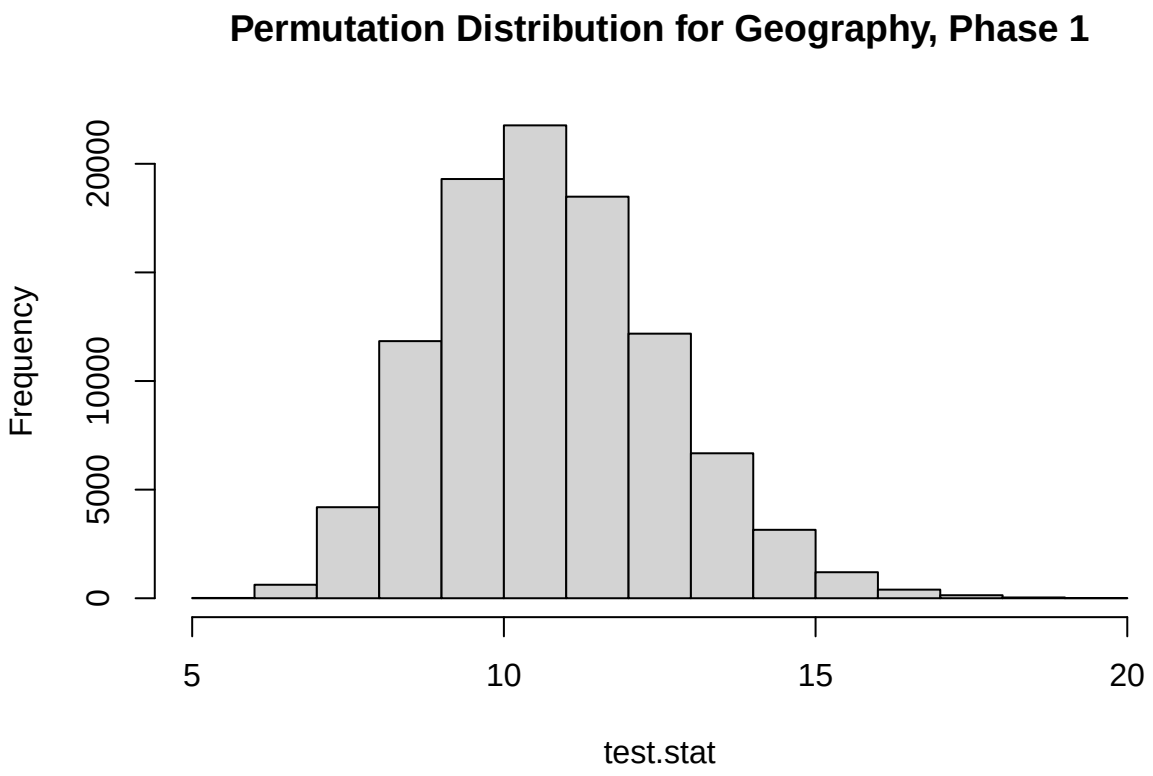
```
print(paste0("P-Value: ", p.value.class))
```

```
## [1] "P-Value: 0.17437"
```

D. Phase 1 Permutation Test for Geography

```
# The population of regions as given by the administration data
Population <- c(rep("New England", 698), rep("Mideast", 604),
                rep("Southeast", 203), rep("Great Lakes",164),
                rep("Plains", 70), rep("Southwest", 78),
                rep("Rocky Mountain",77), rep("Far West", 301))
pop.size <- length(Population)
# Expected proportions
epropNE <- 698/pop.size
epropME <- 604/pop.size
epropSE <- 101/pop.size
epropGL <- 164/pop.size
epropP <- 70/pop.size
epropSW <- 78/pop.size
epropRM <- 77/pop.size
```

```r
epropFW <- 301/pop.size
samp.size <- sum(Geography.Data.Phase1$Zeitgeist[1:8])
# Expected values in a sample our size
epopNE <- epropNE*samp.size
epopME <- epropME*samp.size
epopSE <- epropSE*samp.size
epopGL <- epropGL*samp.size
epopP <- epropP*samp.size
epopSW <- epropSW*samp.size
epopRM <- epropRM*samp.size
epopFW <- epropFW*samp.size
expected.values <- c(epopNE, epopME, epopSE, epopGL,
                     epopP, epopSW, epopRM, epopFW)
#Don't want last two responses ("International" & "I prefer not to answer")
actual.test.stat <- (1/8)*sum(abs(Geography.Data.Raw$Zeitgeist[1:8] - expected.values))
test.stat <- NULL
# Simulating our sampling distribution
for(i in 1:N){
  samp <- sample(Population, samp.size, replace = FALSE)

  sampNE <- sum(samp == "New England")
  sampME <- sum(samp == "Mideast")
  sampSE <- sum(samp == "Southeast")
  sampGL <- sum(samp == "Great Lakes")
  sampP <- sum(samp == "Plains")
  sampSW <- sum(samp == "Southwest")
  sampRM <- sum(samp == "Rocky Mountain")
  sampFW <- sum(samp == "Far West")

  actual.samp.values <- c(sampNE, sampME, sampSE, sampGL,
                          sampP, sampSW, sampRM, sampFW)

  # Computing the test stat for our sample
  test.stat[i] <- (1/8)*sum(abs(actual.samp.values - expected.values))
}
hist(test.stat, main = "Permutation Distribution for Geography, Phase 1")
```




```
# p-value: the probability of observing something rarer than actual.test.stat
p.value.geography <- mean(test.stat >= actual.test.stat)
print(paste0("Alpha-Star: ", alpha.star))
```

```
## [1] "Alpha-Star: 0.0102062183130115"
```

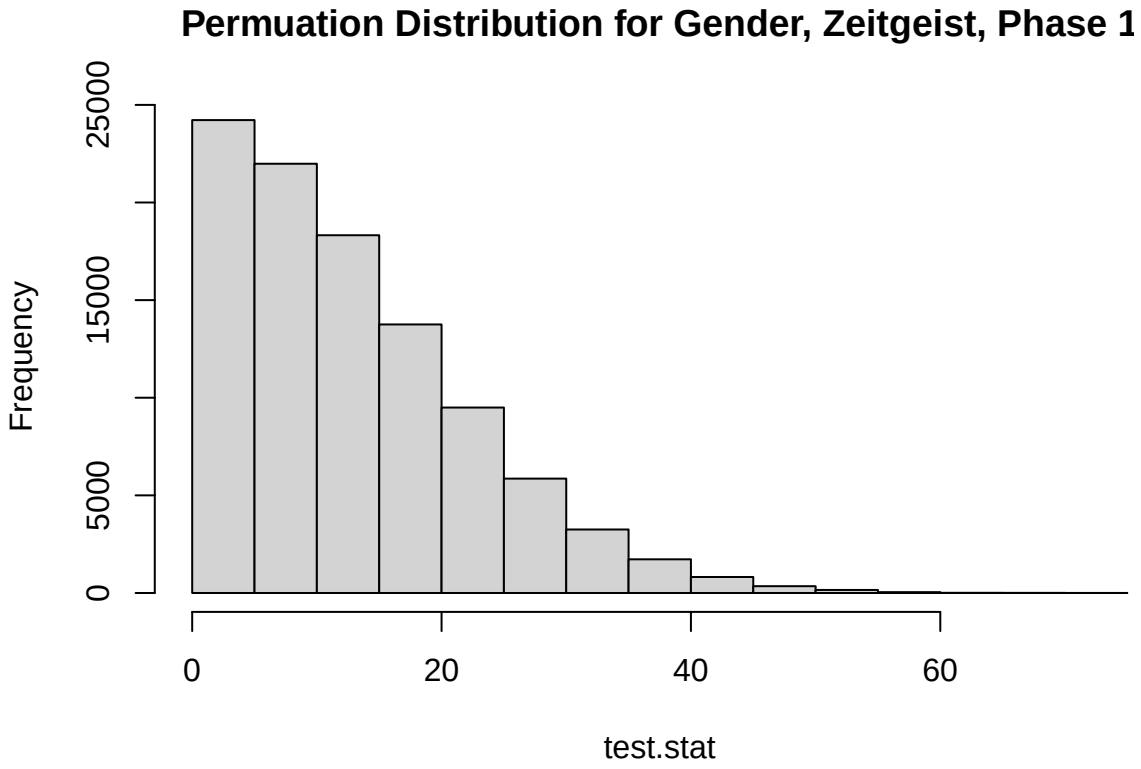
```
print(paste0("P-Value: ", p.value.geography))
```

```
## [1] "P-Value: 0.04387"
```

E. Phase 1 Permutation Test for Gender

```
n <- sum(Gender.Data.Phase1$Zeitgeist)

# Expected number of cisgender females in Zeitgeist sample
expected.prob.f <- (Gender.Data.Phase1$Middlebury[1] /
                    sum(Gender.Data.Phase1$Middlebury))
expected.count.f <- expected.prob.f * n
# Expected number of cisgender males in Zeitgeist sample
expected.prob.m <- (Gender.Data.Phase1$Middlebury[2] /
                    sum(Gender.Data.Phase1$Middlebury))
expected.count.m <- expected.prob.m * n

# Test stat
real.test.stat <- mean(
  c(abs(Gender.Data.Phase1$Zeitgeist[1] - expected.count.f),
```

```
    abs(Gender.Data.Phase1$Zeitgeist[2] - expected.count.m)))

test.stat <- NULL
k <- 100000
for (i in c(1:k)) {
  random.data <- sample(c("Cisgender Female", "Cisgender Male"),
                        size = n,
                        replace = TRUE,
                        prob = c(expected.prob.f,expected.prob.m))
  counts <- table(random.data)
  test.stat[i] <- mean(c(abs(counts[["Cisgender Female"]] - expected.count.f),
                         abs(counts[["Cisgender Male"]] - expected.count.m)))
}
hist(test.stat, main = "Permuation Distribution for Gender, Zeitgeist, Phase 1")
```




```
p.value.gender <- sum(real.test.stat < test.stat) / k
print(paste0("Alpha-Star: ", alpha.star))
```

```
## [1] "Alpha-Star: 0.0102062183130115"
```

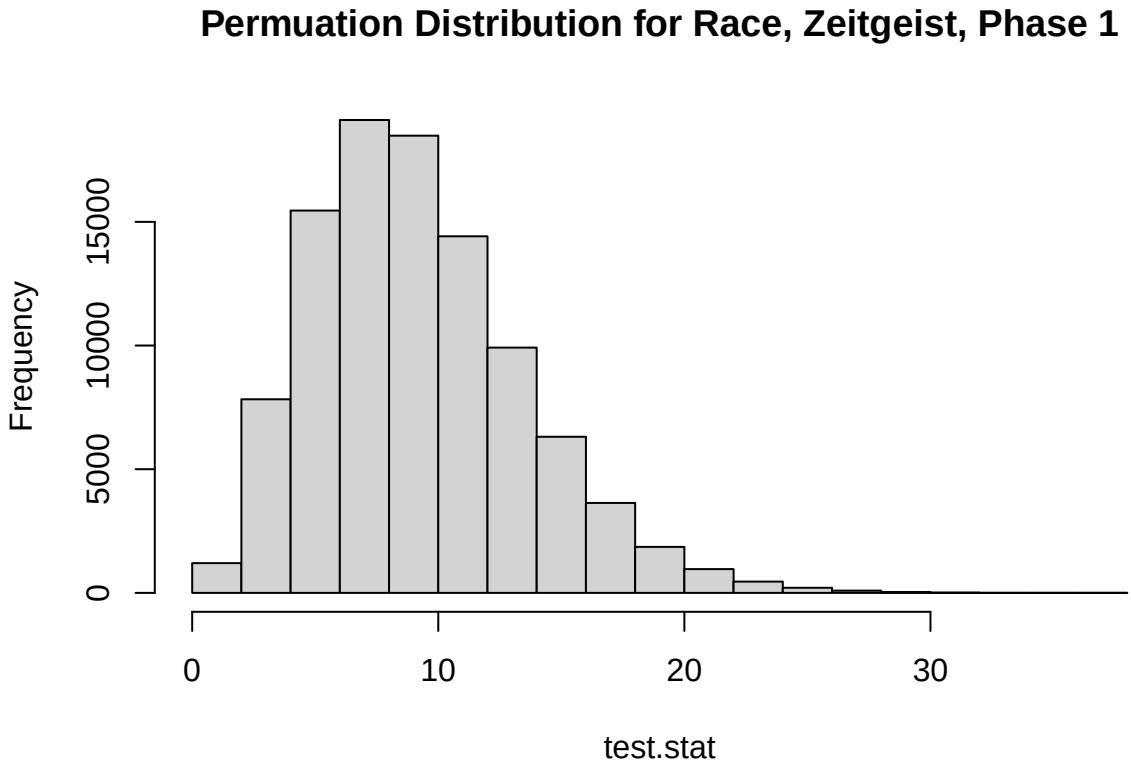
```
print(paste0("P-Value: ", p.value.gender))
```

```
## [1] "P-Value: 0"
```

F. Phase 1 Permutation Test for Race

```r
# Total Zeitgeist Responses
n <- sum(Race.Data.Phase1$Zeitgeist)

# Expected number of White in Zeitgeist sample
expected.prob.w <- Race.Data.Phase1$Middlebury[1] /
                   sum(Race.Data.Phase1$Middlebury)
expected.count.w <- expected.prob.w * sum(Race.Data.Phase1$Zeitgeist)

# Expected number of Asian in Zeitgeist sample
expected.prob.a <- Race.Data.Phase1$Middlebury[2] /
                   sum(Race.Data.Phase1$Middlebury)
expected.count.a <- expected.prob.a * sum(Race.Data.Phase1$Zeitgeist)

# Expected number of Hispanic or Latino Origin
expected.prob.h <- Race.Data.Phase1$Middlebury[3] /
                   sum(Race.Data.Phase1$Middlebury)
expected.count.h <- expected.prob.h * sum(Race.Data.Phase1$Zeitgeist)

# Expected number of Black or African American
expected.prob.b <- Race.Data.Phase1$Middlebury[4] /
                   sum(Race.Data.Phase1$Middlebury)
expected.count.b <- expected.prob.b * sum(Race.Data.Phase1$Zeitgeist)

# Expected number of American Indian or Alaskan Native
expected.prob.i <- Race.Data.Phase1$Middlebury[5] /
                   sum(Race.Data.Phase1$Middlebury)
expected.count.i <- expected.prob.i * sum(Race.Data.Phase1$Zeitgeist)

# Test stat
real.test.stat = mean(
  c(abs(Race.Data.Phase1$Zeitgeist[1] - expected.count.w),
    abs(Race.Data.Phase1$Zeitgeist[2] - expected.count.a),
    abs(Race.Data.Phase1$Zeitgeist[3] - expected.count.h),
    abs(Race.Data.Phase1$Zeitgeist[4] - expected.count.b),
    abs(Race.Data.Phase1$Zeitgeist[5] - expected.count.i)))

test.stat = NULL
k <- 100000
for (i in c(1:k)) {
  random.data <- sample(c("White", "Asian", "Hispanic or Latino Origin",
                          "Black or African American",
                          "American Indian or Alaskan Native"),
                        size = n,
                        replace = TRUE,
                        prob = c(expected.prob.w,expected.prob.a,
                                 expected.prob.h,expected.prob.b,
                                 expected.prob.i))
  counts <- table(random.data)
  sum <- c(abs(counts[["White"]] - expected.count.w),
           abs(counts[["Asian"]] - expected.count.a),
           abs(counts[["Hispanic or Latino Origin"]] - expected.count.h),
           abs(counts[["Black or African American"]] - expected.count.b))
  if (length(counts) == 5) {
```

```
    rbind(sum, abs(counts[["American Indian or Alaskan Native"]]
                - expected.count.i))
  }

  test.stat[i] <- mean(sum)

}
hist(test.stat, main = "Permuation Distribution for Race, Zeitgeist, Phase 1")
```




```
p.value.race = sum(real.test.stat < test.stat) / k
print(paste0("Alpha-Star: ", alpha.star))
```

```
## [1] "Alpha-Star: 0.0102062183130115"
```

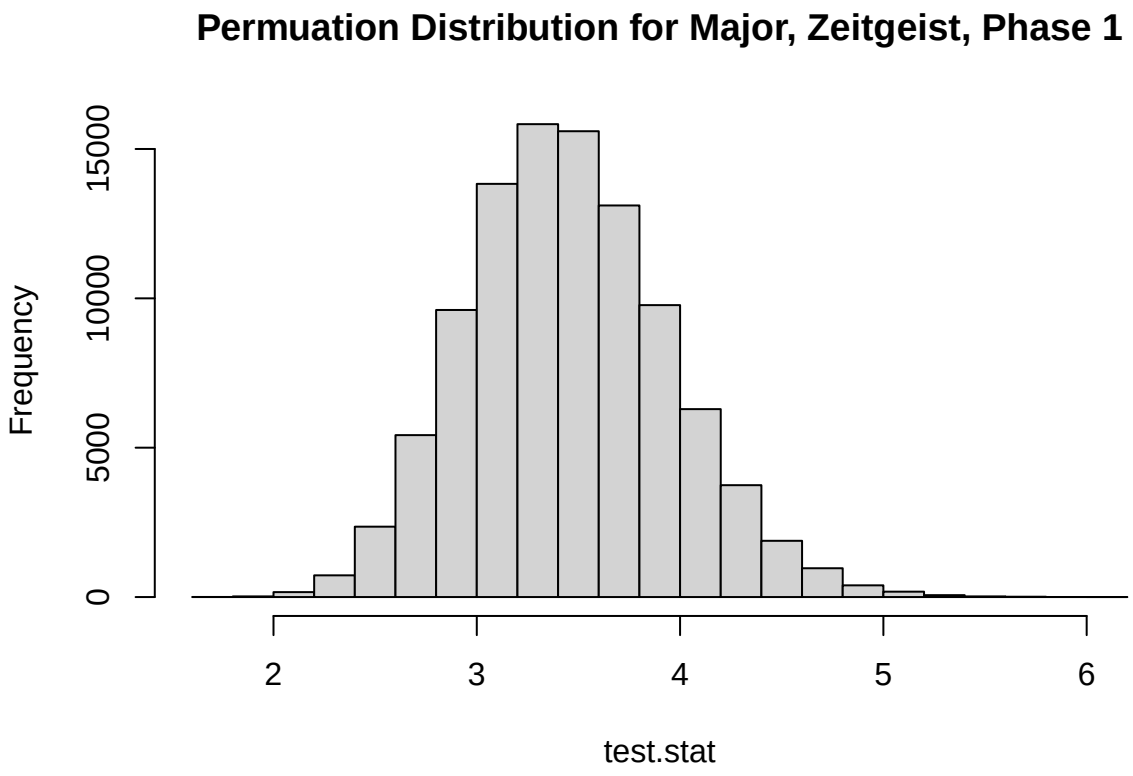
```
print(paste0("P-Value: ", p.value.race))
```

```
## [1] "P-Value: 4e-05"
```

G. Phase 1 Permutation Test for Major

```
n <- sum(Major.Data.Phase1$Zeitgeist)
expected.prob.majors <- NULL
expected.count.majors <- NULL

# Calculate probability using loop bc no way I'm doing it by hand
```

```r
for (i in c(1:length(Majors.M))) {
  expected.prob.majors[i] <- Major.Data.Phase1$Middlebury[i] / sum(Major.Data.Phase1$Middlebury)
  expected.count.majors[i] <- expected.prob.majors[i] * n
}

# Test stat
real.test.stat <- mean(abs(Major.Data.Phase1$Zeitgeist - expected.count.majors))

# For merging with the random sample, remove I prefer not to answer
merge.frame <- data.frame(expected.count.majors,
                          random.data = Major.Data.Phase1$Major)[-26, ]

test.stat <- NULL
k <- 100000
for (i in c(1:k)) {
  random.data <- sample(Major.Data.Phase1$Major,
                        size = n,
                        replace = TRUE,
                        prob = expected.prob.majors)
  counts <- table(random.data)
  df <- merge(data.frame(counts), merge.frame, all = FALSE)
  test.stat[i] <- mean(c(abs(df$Freq - df$expected.count.majors)))

}

hist(test.stat, main = "Permuation Distribution for Major, Zeitgeist, Phase 1")
```



```r
p.value.race <- sum(real.test.stat < test.stat) / k
print(paste0("Alpha-Star: ", alpha.star))
```

```
## [1] "Alpha-Star: 0.0102062183130115"
```

```r
print(paste0("P-Value: ", p.value.race))
```

```
## [1] "P-Value: 0"
```

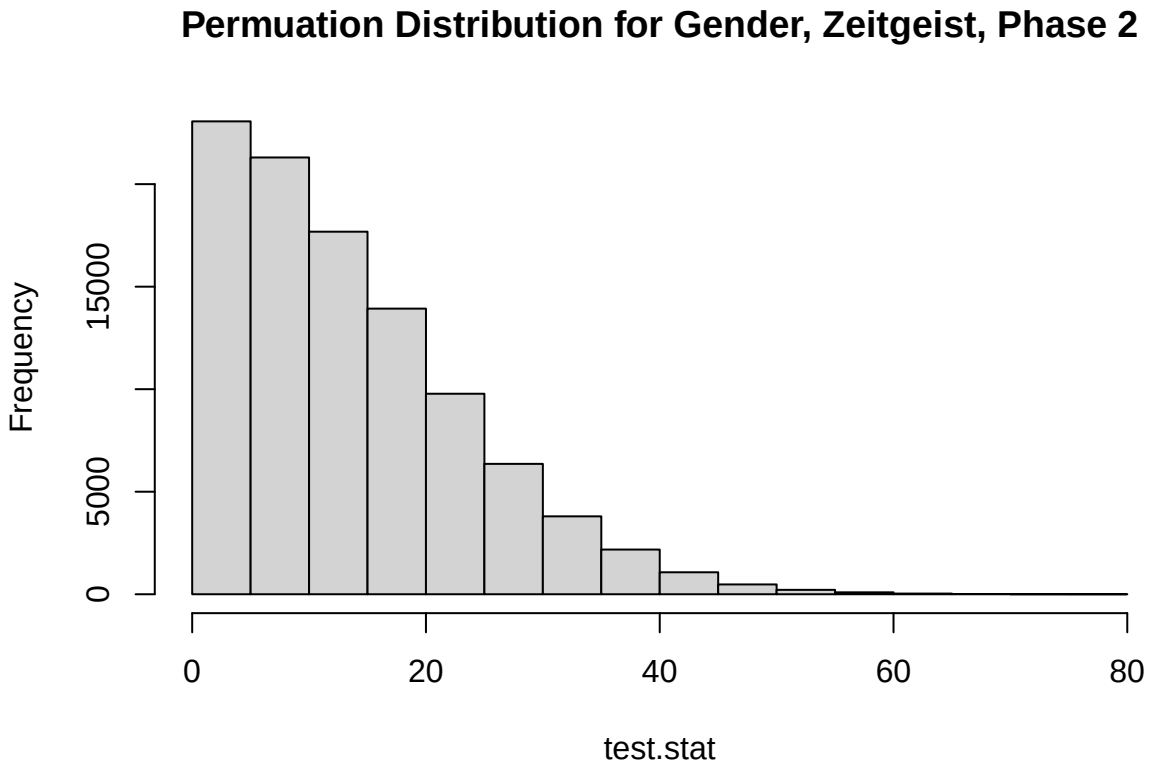
H. Phase 2 Data Preparation

```r
library(tidyverse)
library(dplyr)

# Phase 2 testing, we randomly distribute all Zeitgeist respondents who
# selected gender-identities other than CGF and CGM between these options
Gender.Data.Phase2 <- Gender.Data.Raw %>%
  filter(Gender == c("Cisgender Female", "Cisgender Male"))
non.CG.count <- sum((Gender.Data.Raw %>%
  filter(Gender != c("Cisgender Female", "Cisgender Male")))$Zeitgeist)
CGF.artificial.sample <- sum(sample(c(0,1), non.CG.count, replace = TRUE))
CGM.artificial.sample <- non.CG.count - CGF.artificial.sample
Gender.Data.Phase2$Zeitgeist[1] <- Gender.Data.Phase2$Zeitgeist[1] +
  CGF.artificial.sample
Gender.Data.Phase2$Zeitgeist[2] <- Gender.Data.Phase2$Zeitgeist[2] +
  CGM.artificial.sample

# We aren't adjusting the Zeitgeist data for phase 2 of race testing,
# we're adjusting the Middlebury Admin distribution as described in Methods
Race.Data.Phase2 <- Race.Data.Phase1
Admin.Distribution.Phase2 <- Race.Data.Raw %>% filter(
  Race == c("White", "Asian", "Hispanic or Latino Origin",
            "Black or African American"))
two.or.more <- Race.Data.Raw$Middlebury[10]
two.race.samples <- NULL
for (i in c(1:two.or.more)) {
  two.race.samples <- append(two.race.samples,
                             sample(c("White", "Asian", "Hispanic or Latino Origin",
                                      "Black or African American",
                                      "American Indian or Alaskan Native"),
                                    size = 2,
                                    replace = FALSE,
                                    prob = c(expected.prob.w,expected.prob.a,
                                             expected.prob.h,expected.prob.b,
                                             expected.prob.i)))
}
counts <- data.frame(table(two.race.samples))
colnames(counts) [1] <- "Race"
Admin.Distribution.Phase2 <- merge(counts, Admin.Distribution.Phase2)
Admin.Distribution.Phase2 <- Admin.Distribution.Phase2 %>%
```

```
                            mutate(Middlebury = Freq + Middlebury) %>%
                            select(-Freq)

# Just voiding undeclared
Major.Data.Phase2 <- Major.Data.Phase1[-47,]
```

I. Phase 2 Permutation Test for Gender

```
# 1 is Cisgender Female, 0 is Cisgender Male
n <- sum(Gender.Data.Phase2$Zeitgeist)

# Expected number of cisgender females in Zeitgeist sample
expected.prob.f <- (Gender.Data.Phase2$Middlebury[1] /
                    sum(Gender.Data.Phase2$Middlebury))
expected.count.f <- expected.prob.f * sum(Gender.Data.Phase2$Zeitgeist)
# Expected number of cisgender males in Zeitgeist sample
expected.prob.m <- (Gender.Data.Phase2$Middlebury[2] /
                    sum(Gender.Data.Phase2$Middlebury))
expected.count.m <- expected.prob.m * sum(Gender.Data.Phase2$Zeitgeist)

# Test stat
real.test.stat <- mean(
  c(abs(Gender.Data.Phase2$Zeitgeist[1] - expected.count.f),
    abs(Gender.Data.Phase2$Zeitgeist[2] - expected.count.m)))

test.stat <- NULL
k <- 100000
for (i in c(1:k)) {
  random.data <- sample(c("Cisgender Female", "Cisgender Male"),
                        size = n,
                        replace = TRUE,
                        prob = c(expected.prob.f,expected.prob.m))
  counts <- table(random.data)
  test.stat[i] <- mean(c(abs(counts[["Cisgender Female"]] - expected.count.f),
                         abs(counts[["Cisgender Male"]] - expected.count.m)))
}
hist(test.stat, main = "Permuation Distribution for Gender, Zeitgeist, Phase 2")
```




```
p.value.gender <- sum(real.test.stat < test.stat) / k
print(paste0("Alpha-Star: ", alpha.star))
```

```
## [1] "Alpha-Star: 0.0102062183130115"
```

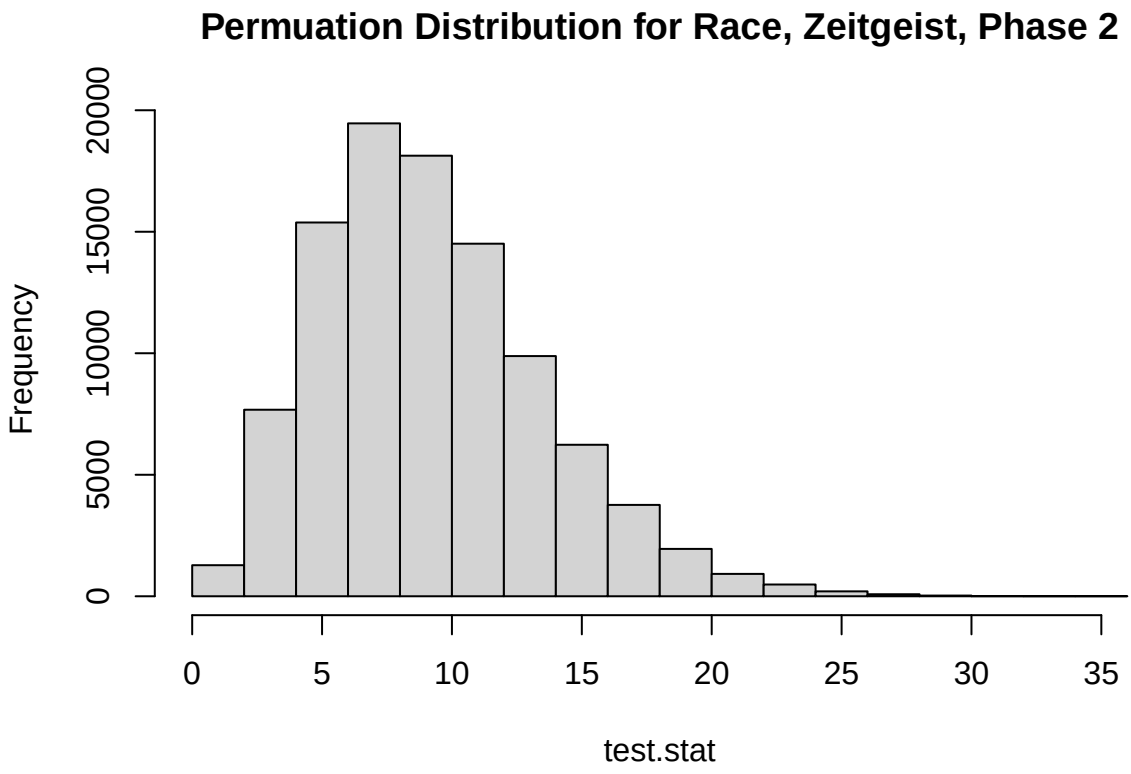
```
print(paste0("P-Value: ", p.value.gender))
```

```
## [1] "P-Value: 0"
```

J. Phase 2 Permutation Test for Race

```
# Total Zeitgeist Responses
n <- sum(Race.Data.Phase2$Zeitgeist)

# Expected number of White in Zeitgeist sample
expected.prob.w <- Race.Data.Phase2$Middlebury[1] /
                   sum(Race.Data.Phase2$Middlebury)
expected.count.w <- expected.prob.w * sum(Race.Data.Phase2$Zeitgeist)

# Expected number of Asian in Zeitgeist sample
expected.prob.a <- Race.Data.Phase2$Middlebury[2] /
                   sum(Race.Data.Phase2$Middlebury)
expected.count.a <- expected.prob.a * sum(Race.Data.Phase2$Zeitgeist)

# Expected number of Hispanic or Latino Origin
expected.prob.h <- Race.Data.Phase2$Middlebury[3] /
```

```
                    sum(Race.Data.Phase2$Middlebury)
expected.count.h <- expected.prob.h * sum(Race.Data.Phase2$Zeitgeist)

# Expected number of Black or African American
expected.prob.b <- Race.Data.Phase2$Middlebury[4] /
                    sum(Race.Data.Phase2$Middlebury)
expected.count.b <- expected.prob.b * sum(Race.Data.Phase2$Zeitgeist)

# Expected number of American Indian or Alaskan Native
expected.prob.i <- Race.Data.Phase2$Middlebury[5] /
                    sum(Race.Data.Phase2$Middlebury)
expected.count.i <- expected.prob.i * sum(Race.Data.Phase2$Zeitgeist)

# Test stat
real.test.stat = mean(
  c(abs(Race.Data.Phase2$Zeitgeist[1] - expected.count.w),
    abs(Race.Data.Phase2$Zeitgeist[2] - expected.count.a),
    abs(Race.Data.Phase2$Zeitgeist[3] - expected.count.h),
    abs(Race.Data.Phase2$Zeitgeist[4] - expected.count.b),
    abs(Race.Data.Phase2$Zeitgeist[5] - expected.count.i)))

test.stat = NULL
k <- 100000
for (i in c(1:k)) {
  random.data <- sample(c("White", "Asian", "Hispanic or Latino Origin",
                          "Black or African American",
                          "American Indian or Alaskan Native"),
                        size = n,
                        replace = TRUE,
                        prob = c(expected.prob.w,expected.prob.a,
                                 expected.prob.h,expected.prob.b,
                                 expected.prob.i))
  counts <- table(random.data)
  sum <- c(abs(counts[["White"]] - expected.count.w),
           abs(counts[["Asian"]] - expected.count.a),
           abs(counts[["Hispanic or Latino Origin"]] - expected.count.h),
           abs(counts[["Black or African American"]] - expected.count.b))
  if (length(counts) == 5) {
    rbind(sum, abs(counts[["American Indian or Alaskan Native"]]
                - expected.count.i))
  }

  test.stat[i] <- mean(sum)

}
hist(test.stat, main = "Permuation Distribution for Race, Zeitgeist, Phase 2")
```




```
p.value.race = sum(real.test.stat < test.stat) / k
print(paste0("Alpha-Star: ", alpha.star))
```

```
## [1] "Alpha-Star: 0.0102062183130115"
```

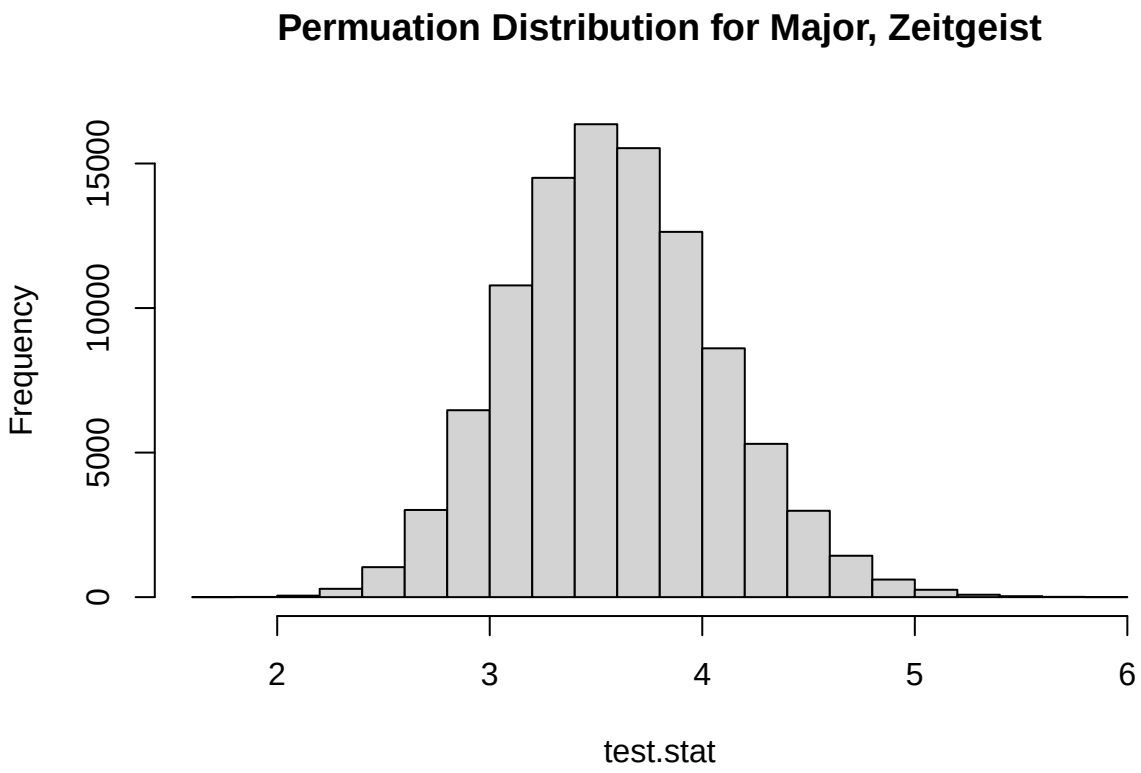
```
print(paste0("P-Value: ", p.value.race))
```

```
## [1] "P-Value: 5e-05"
```

K. Phase 2 Permutation Test for Major

```
n <- sum(Major.Data.Phase2$Zeitgeist)
expected.prob.majors <- NULL
expected.count.majors <- NULL

# Calculate probability using loop bc no way I'm doing it by hand
for (i in c(1:length(Major.Data.Phase2$Major))) {
  expected.prob.majors[i] <- Major.Data.Phase2$Middlebury[i] /
    sum(Major.Data.Phase2$Middlebury)
  expected.count.majors[i] <- expected.prob.majors[i] * n
}

# Test stat
real.test.stat <- mean(abs(Major.Data.Phase2$Zeitgeist - expected.count.majors))

# For merging with the random sample, remove I prefer not to answer
```

```
merge.frame <- data.frame(expected.count.majors,
                          random.data = Major.Data.Phase2$Major)[-26, ]

test.stat <- NULL
k <- 100000
for (i in c(1:k)) {
  random.data <- sample(Major.Data.Phase2$Major,
                        size = n,
                        replace = TRUE,
                        prob = expected.prob.majors)
  counts <- table(random.data)
  df <- merge(data.frame(counts), merge.frame, all = FALSE)
  test.stat[i] <- mean(c(abs(df$Freq - df$expected.count.majors)))

}

hist(test.stat, main = "Permuation Distribution for Major, Zeitgeist")
```




```
p.value.race <- sum(real.test.stat < test.stat) / k
print(paste0("Alpha-Star: ", alpha.star))
```

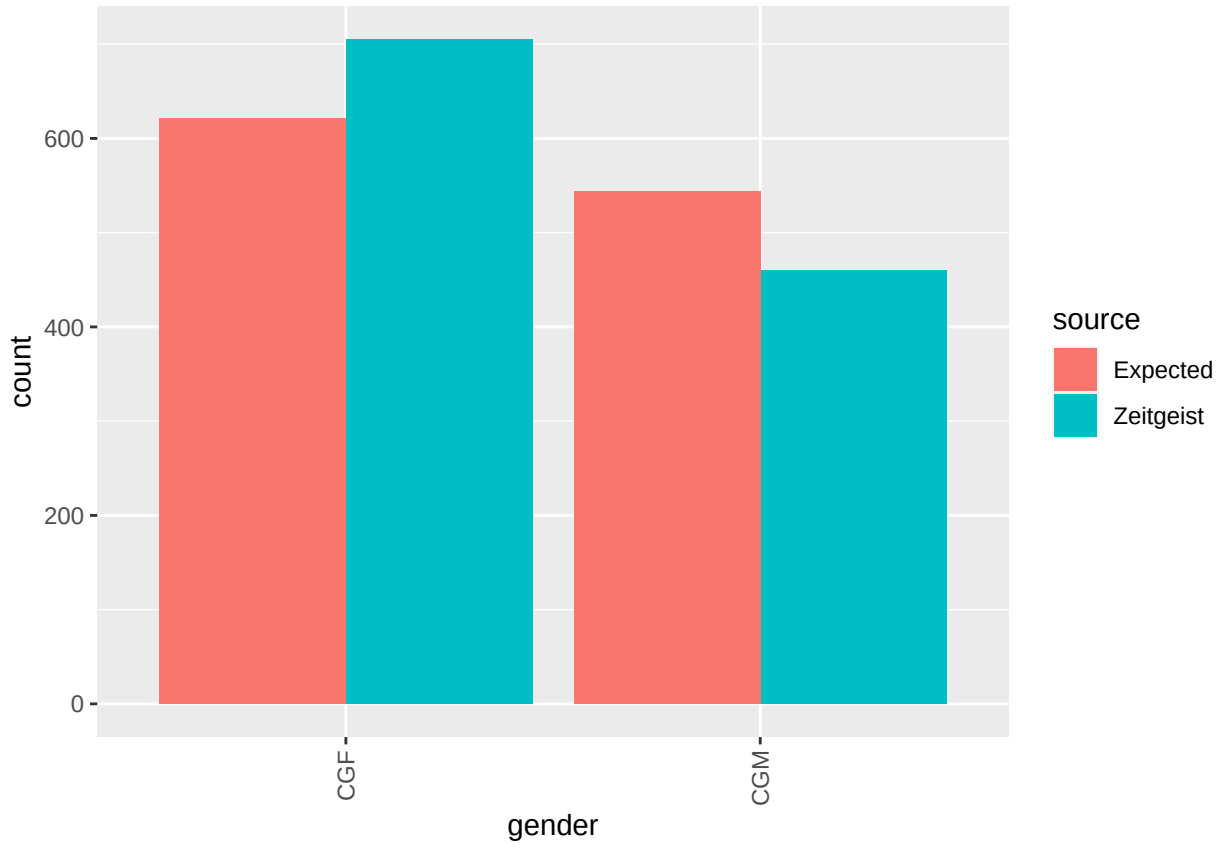
```
## [1] "Alpha-Star: 0.0102062183130115"
```

```
print(paste0("P-Value: ", p.value.race))
```

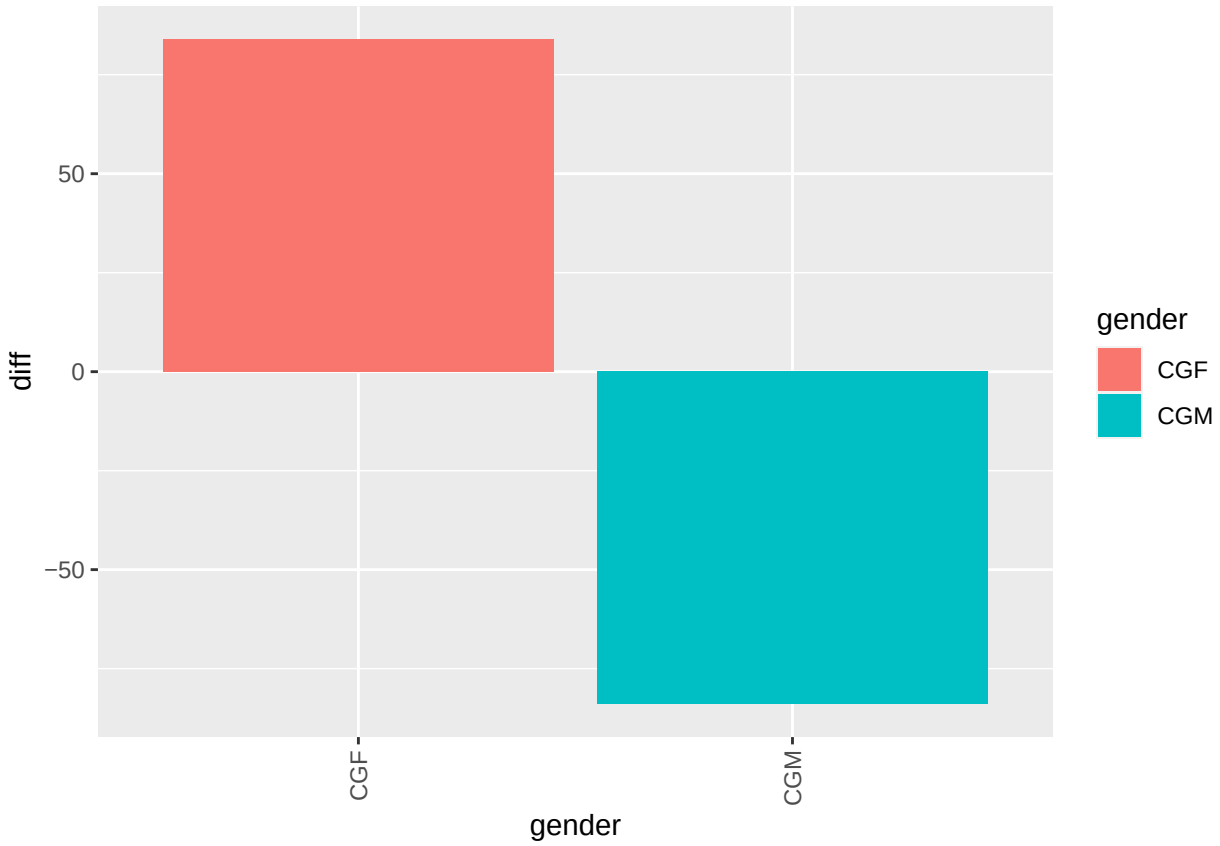
```
## [1] "P-Value: 0"
```

L. Conclusion Charts

```
library(ggplot2)

##############################################################################
# Gender Plots
# 1 is Cisgender Female, 0 is Cisgender Male
n <- sum(Gender.Data.Phase2$Zeitgeist)

# Expected number of cisgender females in Zeitgeist sample
expected.prob.f <- (Gender.Data.Phase2$Middlebury[1] /
                    sum(Gender.Data.Phase2$Middlebury))
expected.count.f <- expected.prob.f * n
# Expected number of cisgender males in Zeitgeist sample
expected.prob.m <- (Gender.Data.Phase2$Middlebury[2] /
                    sum(Gender.Data.Phase2$Middlebury))
expected.count.m <- expected.prob.m * n

expected.count.genders <- c(expected.count.f,
                            expected.count.m)
observed.count.genders <- Gender.Data.Phase2$Zeitgeist
gender.plot.data1 <- data.frame(gender = c("CGF", "CGM", "CGF", "CGM"),
                                count = c(observed.count.genders, expected.count.genders),
                                source = c(rep("Zeitgeist", 2), rep("Expected", 2)))
gender.plot.data2 <- data.frame(gender = c("CGF", "CGM"),
                                diff = observed.count.genders - expected.count.genders)

ggplot(gender.plot.data1, aes(fill=source, y=count, x=gender)) +
  geom_bar(position="dodge", stat="identity") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

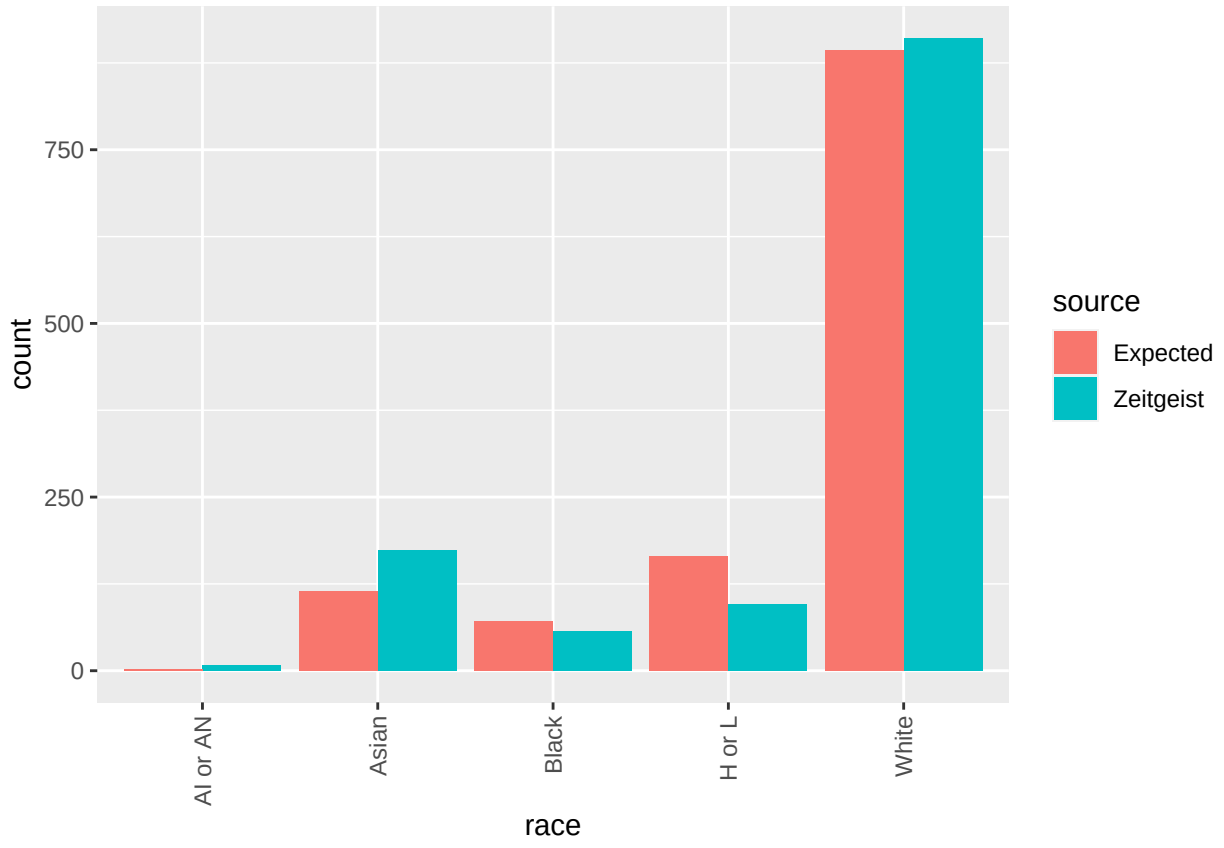



```
ggplot(gender.plot.data2, aes(fill = gender, x = gender, y = diff)) +
  geom_col(position = "identity") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

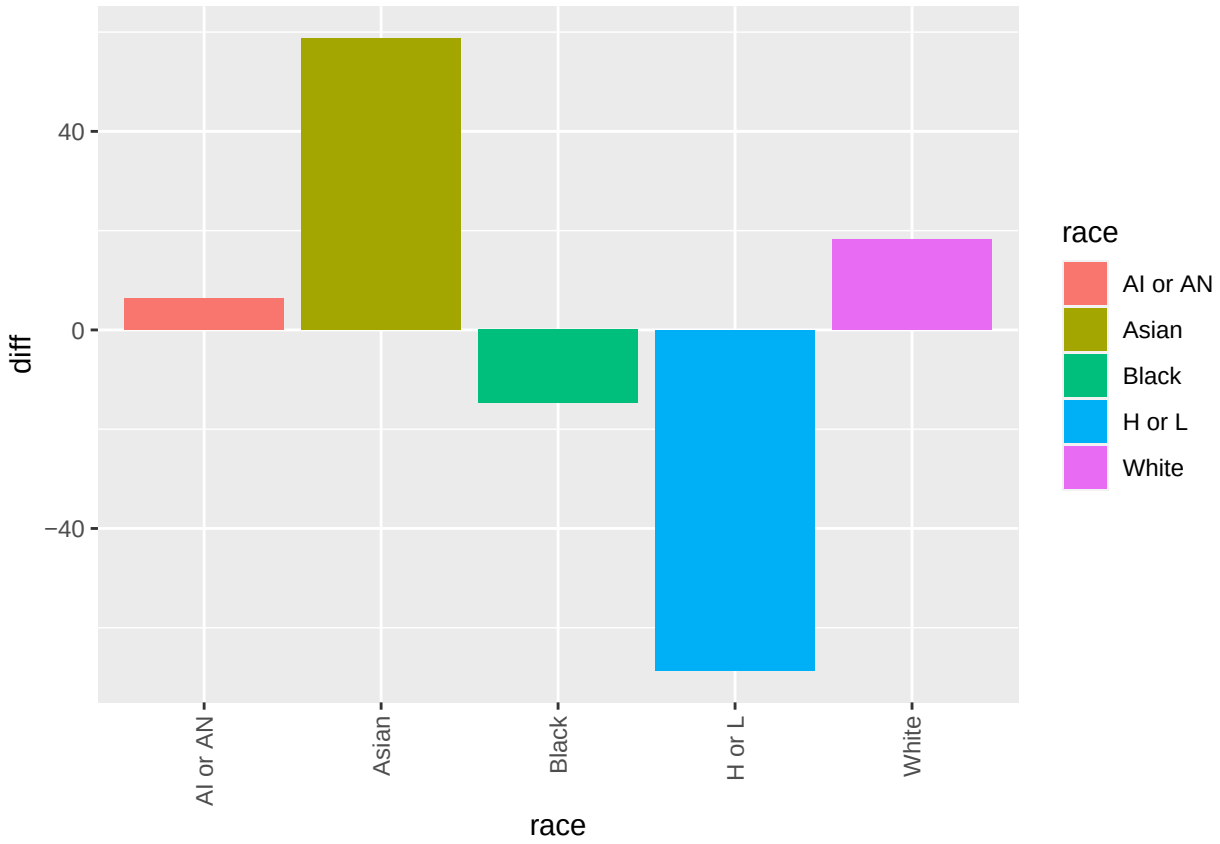



```
################################################################################
# Race plots

# Total Zeitgeist Responses
n <- sum(Race.Data.Phase2$Zeitgeist)

# Expected number of White in Zeitgeist sample
expected.prob.w <- Race.Data.Phase2$Middlebury[1] /
                   sum(Race.Data.Phase2$Middlebury)
expected.count.w <- expected.prob.w * sum(Race.Data.Phase2$Zeitgeist)

# Expected number of Asian in Zeitgeist sample
expected.prob.a <- Race.Data.Phase2$Middlebury[2] /
                   sum(Race.Data.Phase2$Middlebury)
expected.count.a <- expected.prob.a * sum(Race.Data.Phase2$Zeitgeist)

# Expected number of Hispanic or Latino Origin
expected.prob.h <- Race.Data.Phase2$Middlebury[3] /
                   sum(Race.Data.Phase2$Middlebury)
expected.count.h <- expected.prob.h * sum(Race.Data.Phase2$Zeitgeist)

# Expected number of Black or African American
expected.prob.b <- Race.Data.Phase2$Middlebury[4] /
                   sum(Race.Data.Phase2$Middlebury)
expected.count.b <- expected.prob.b * sum(Race.Data.Phase2$Zeitgeist)

# Expected number of American Indian or Alaskan Native
```

```r
expected.prob.i <- Race.Data.Phase2$Middlebury[5] /
                   sum(Race.Data.Phase2$Middlebury)
expected.count.i <- expected.prob.i * sum(Race.Data.Phase2$Zeitgeist)

expected.count.races <- c(expected.count.w,
                          expected.count.a,
                          expected.count.h,
                          expected.count.b,
                          expected.count.i)
observed.count.races <- Race.Data.Phase2$Zeitgeist
race.plot.data1 <- data.frame(race = Race.Data.Phase2$Race,
                              count = c(observed.count.races, expected.count.races),
                              source = c(rep("Zeitgeist", 5), rep("Expected", 5)))
race.plot.data2 <- data.frame(race = Race.Data.Phase2$Race,
                             diff = observed.count.races - expected.count.races)

race.plot.data1$race[3] <- "H or L"
race.plot.data1$race[4] <- "Black"
race.plot.data1$race[5] <- "AI or AN"
race.plot.data1$race[8] <- "H or L"
race.plot.data1$race[9] <- "Black"
race.plot.data1$race[10] <- "AI or AN"

race.plot.data2$race[3] <- "H or L"
race.plot.data2$race[4] <- "Black"
race.plot.data2$race[5] <- "AI or AN"

ggplot(race.plot.data1, aes(fill=source, y=count, x=race)) +
  geom_bar(position="dodge", stat="identity") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

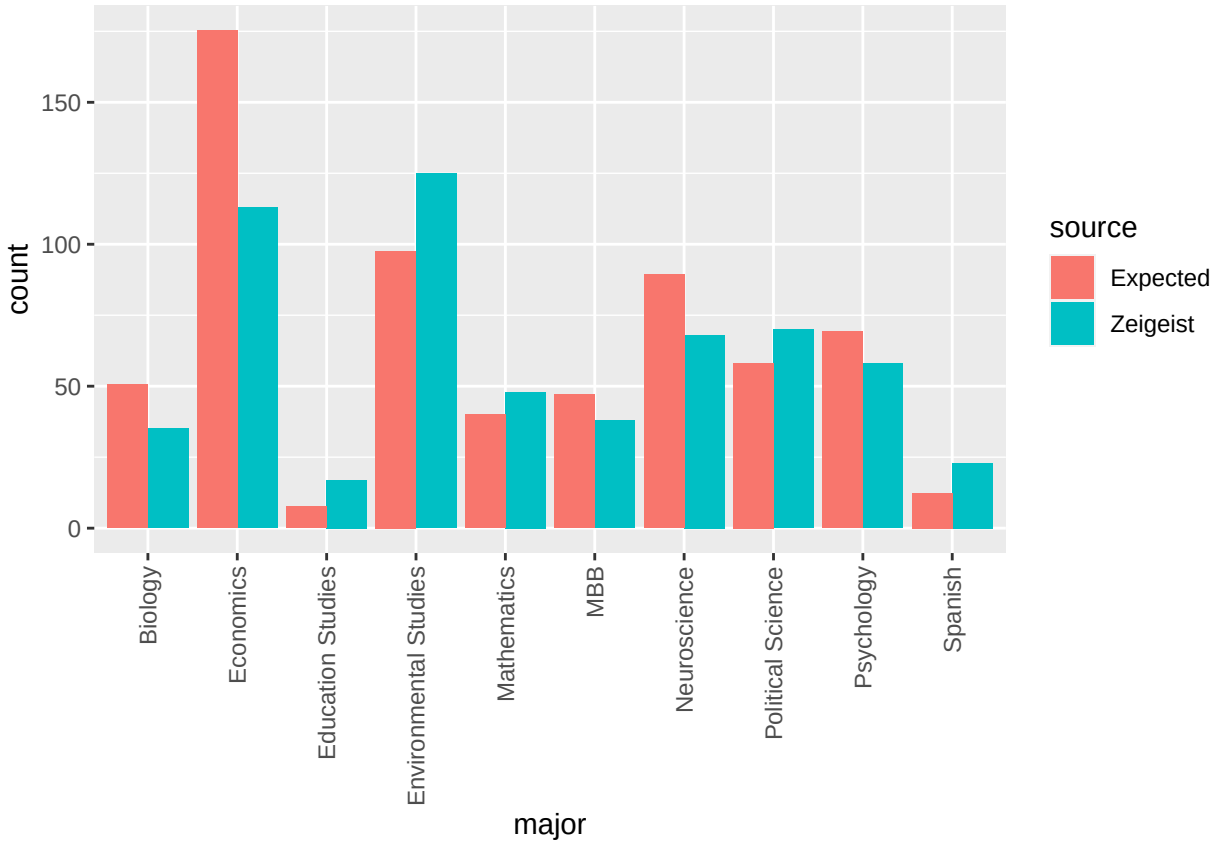



```
ggplot(race.plot.data2, aes(fill = race, x = race, y = diff)) +
  geom_col(position = "identity") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

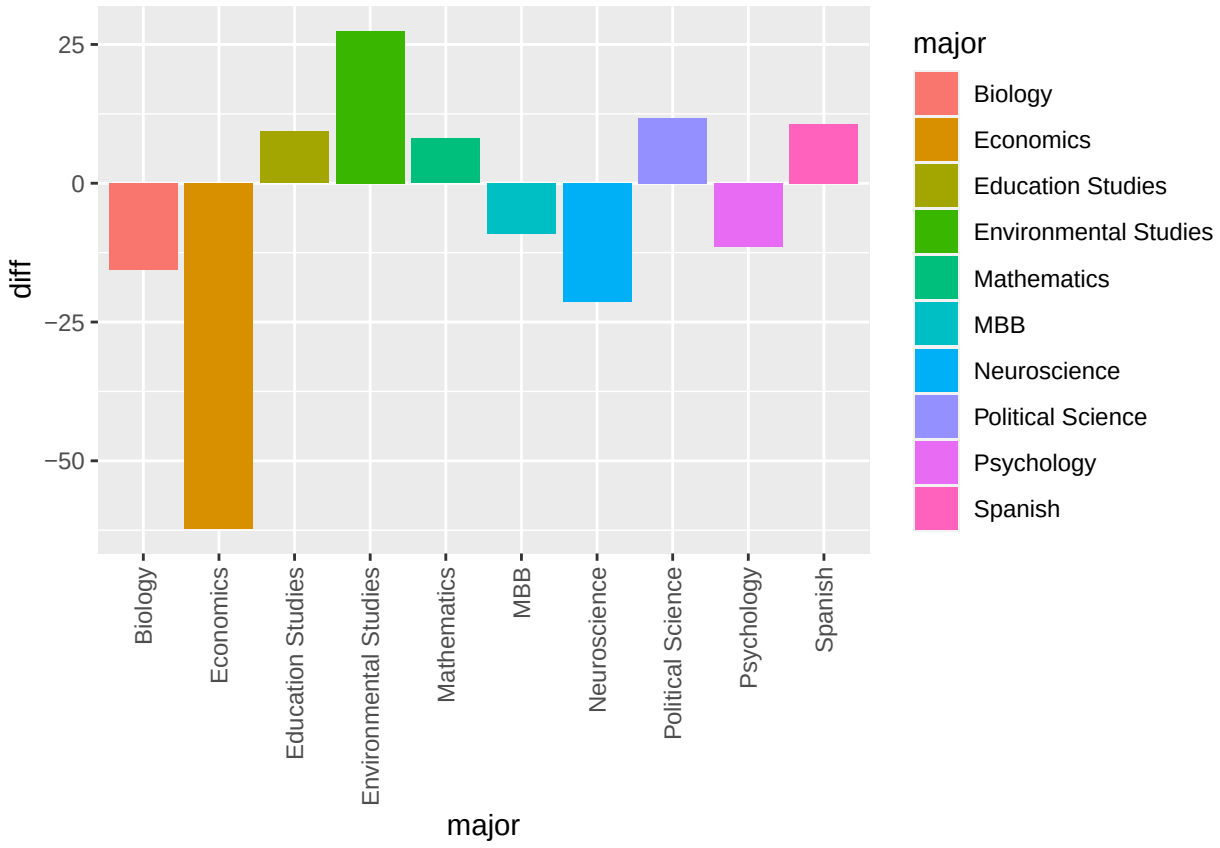



```
################################################################################
# Major plots
n <- sum(Major.Data.Phase2$Zeitgeist)
expected.prob.majors <- NULL
expected.count.majors <- NULL

# Calculate probability using loop bc no way I'm doing it by hand
for (i in c(1:length(Major.Data.Phase2$Major))) {
  expected.prob.majors[i] <- Major.Data.Phase2$Middlebury[i] /
    sum(Major.Data.Phase2$Middlebury)
  expected.count.majors[i] <- expected.prob.majors[i] * n
}


all.plot.data <- data.frame(expected = expected.count.majors,
                            observed = Major.Data.Phase2$Zeitgeist,
                            major = Major.Data.Phase2$Major)[-26, ] %>%
  mutate(diff = observed - expected)
plot.data.reduced <- all.plot.data[order(abs(all.plot.data$diff),
                                   decreasing = TRUE),][c(1:10),]
plot1.data <- data.frame(count = c(
                                   plot.data.reduced$observed,
                                   plot.data.reduced$expected),
                         source = c(
                                   rep("Zeigeist", length(plot.data.reduced$observed)),
                                   rep("Expected", length(plot.data.reduced$expected))),
                         major = c(rep(plot.data.reduced, 2))) %>%
```

```
  select(count, source, major.major, major.diff.1)
colnames(plot1.data)[c(3,4)] <- c("major", "diff")
plot2.data <- plot.data.reduced %>%
  select(major, diff)

plot1.data$major[9] <- "MBB"
plot1.data$major[19] <- "MBB"
plot2.data$major[9] <- "MBB"
ggplot(plot1.data, aes(fill=source, y=count, x=major)) +
  geom_bar(position="dodge", stat="identity") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```




```
ggplot(plot2.data, aes(fill = major, x = major, y = diff)) +
  geom_col(position = "identity") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```


25
0
−25
−50
diff
Biology
Economics
Education Studies
Environmental Studies
Mathematics
MBB
Neuroscience
Political Science
Psychology
Spanish
major
major
Biology
Economics
Education Studies
Environmental Studies
Mathematics
MBB
Neuroscience
Political Science
Psychology
Spanish