turk_results

Gurdit Chahal, Adam Letcher, Carlos Castro, Noah Randolph

8/7/2018

Load Data

```
dt_raw <- fread('musicdata.8.8.2018.csv')

head(dt_raw)
```

```
##    PartitionKey                                    RowKey
## 1:   musictests              1a76b09f-c01f-4bfe-8f83-9f70774e6782
## 2:   musictests              5dcc4cd9-b794-4f3e-862d-c05df05936f1
## 3:   musictests A18TCR555RWUZVb376e672-98e0-4658-b1ce-185374c7e935
## 4:   musictests A1EBQ9X6IN5OZC05d429a2-e1ca-4139-b0dd-f2739d874bb5
## 5:   musictests A1PUHCEBSOWETV5ab6e0ce-75a4-4e7e-887f-9ed0a47c15e6
## 6:   musictests A1VC6F0FYG1L5I9d672728-9457-431b-a8f9-b688efc87efb
##                   Timestamp Check1 Check2 Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10
## 1: 2018-08-07T00:29:39.285Z      A      A  C  D  A  C  A  E  A  B  E   A
## 2: 2018-08-07T00:31:13.773Z      B      A  C  B  E  A  A  G  A  A  D   A
## 3: 2018-08-07T00:23:44.949Z      A      A  C  B  E  A  D  D  A  A  E   A
## 4: 2018-08-07T00:24:44.489Z      A      A  C  B  C  A  D  D  B  A  H   A
## 5: 2018-08-07T00:29:11.786Z      A      A  C  B  E  A  D  C  A  B  G   A
## 6: 2018-08-07T00:26:48.495Z      A      A  C  B  C  A  D  E  B  A  H   B
##    Q11 Q13 isTurk clickedPlay time correctCount lyrics Q12
## 1:   B   B   true        true -405            1   true
## 2:   A   B   true        true -659            4   true
## 3:   A   B   true        true -295            5  false
## 4:   A   B   true        true -194            4  false
## 5:   A   B   true        true -273            5   true
## 6:   A   B   true        true -382            4  false
```

Clean Up Columns

```
dt <- dt_raw[, .(hear_song = as.integer(as.character(factor(Check1, levels = c('A', 'B'),
                                                            labels = c(1, 0)))),
                 piano_playing = as.integer(as.character(factor(Check2,
                                                                levels = c('A', 'B'),
                                                                labels = c(1, 0)))),
                 q1 = as.factor(Q1),
                 q2 = as.factor(Q2),
                 q3 = as.factor(Q3),
                 q4 = as.factor(Q4),
                 q5 = as.factor(Q5),
                 age = factor(Q6, levels = c('A', 'B', 'C', 'D', 'E',
                                             'F', 'G', 'H', 'I', 'J'),
                              labels = c('<12', '12-17', '18-24', '25-34', '35-44',
                                         '45-54', '55-64', '65-74', '>75', 'decline')),
                 gender = factor(Q7, levels = c('A', 'B', 'C', 'D'),
```

```
                                labels = c('male', 'female', 'other', 'decline')),
                own_dog = as.integer(as.character(factor(Q8, levels = c('A', 'B'),
                                                         labels = c(1, 0)))),
                education = factor(Q9, levels = c('A', 'B', 'C', 'D', 'E',
                                                  'F', 'G', 'H', 'I', 'J'),
                                   labels = c('none', '8th grade', 'some high school',
                                              'high school completed', 'some college',
                                              'vocational', 'associates', 'bachelors',
                                              'masters', 'doctorate')),
                occupation = Q10,
                native_english = as.integer(as.character(factor(Q11,
                                                                 levels = c('A', 'B'),
                                                                 labels = c(1, 0)))),
                heard_lyrics = factor(Q13, levels = c('A', 'B', 'C', 'D', 'E', 'F'),
                                                                   labels = c('I\'m a barbie girl',
                                                                              'Rocket Man',
                                                                              'Don\'t stop believing',
                                                                              'Hakuna Matata',
                                                                              'Lyrics but not sure',
                                                                              'No lyrics')),
                is_turk = as.integer(as.character(factor(isTurk,
                                                         levels = c('true', 'null'),
                                                         labels = c(1, 0)))),
                time = time * -1,
                correct_count = correctCount,
                assigned_lyrics = as.integer(as.character(factor(lyrics, levels = c('true', 'false'),
                                                                 labels = c(1, 0)))),
                lyrics_factor = factor(lyrics, levels = c('true', 'false'), labels = c("lyrics", "no ly
```

EDA

```
summary(dt)
```

```
##    hear_song      piano_playing    q1      q2     q3     q4     q5
##  Min.   :0.0000   Min.   :0.0000   A:  1   A: 8   A:15   A:74   A:17
##  1st Qu.:1.0000   1st Qu.:1.0000   B:  6   B:96   B: 3   B: 5   B:25
##  Median :1.0000   Median :1.0000   C:130   C:15   C:36   C:41   C:15
##  Mean   :0.9929   Mean   :0.9929   D:  2   D: 4   D: 3   D: 6   D:74
##  3rd Qu.:1.0000   3rd Qu.:1.0000   E:  2   E:18   E:84   E:14   E: 9
##  Max.   :1.0000   Max.   :1.0000                          N: 1   N: 1
##
##       age         gender      own_dog                       education
##  25-34  :63   male   :78   Min.   :0.0000   bachelors             :52
##  18-24  :30   female :62   1st Qu.:0.0000   some college          :27
##  35-44  :28   other  : 0   Median :0.0000   associates            :19
##  45-54  :13   decline: 0   Mean   :0.4571   masters               :18
##  55-64  : 3   NA's   : 1   3rd Qu.:1.0000   high school completed:15
##  (Other): 3                Max.   :1.0000   (Other)               : 9
##  NA's   : 1                NA's   :1        NA's                  : 1
##   occupation        native_english                heard_lyrics
##  Length:141         Min.   :0.0000   I'm a barbie girl   :  0
##  Class :character   1st Qu.:1.0000   Rocket Man          :116
```

```
##  Mode  :character   Median :1.0000   Don't stop believing:  0
##                     Mean   :0.9143   Hakuna Matata       :  0
##                     3rd Qu.:1.0000   Lyrics but not sure :  1
##                     Max.   :1.0000   No lyrics           :  0
##                     NA's   :1        NA's                : 24
##     is_turk           time         correct_count   assigned_lyrics
##  Min.   :0.000   Min.   :  68.0   Min.   :0.000   Min.   :0.0000
##  1st Qu.:1.000   1st Qu.: 301.0   1st Qu.:2.000   1st Qu.:0.0000
##  Median :1.000   Median : 405.0   Median :3.000   Median :1.0000
##  Mean   :0.766   Mean   : 448.3   Mean   :3.248   Mean   :0.5248
##  3rd Qu.:1.000   3rd Qu.: 520.0   3rd Qu.:4.000   3rd Qu.:1.0000
##  Max.   :1.000   Max.   :1478.0   Max.   :5.000   Max.   :1.0000
##
##    lyrics_factor
##  lyrics   :74
##  no lyrics:67
##
##
##
##
##
```

```
stargazer(dt, header=FALSE, type='latex')
```

Table 1:

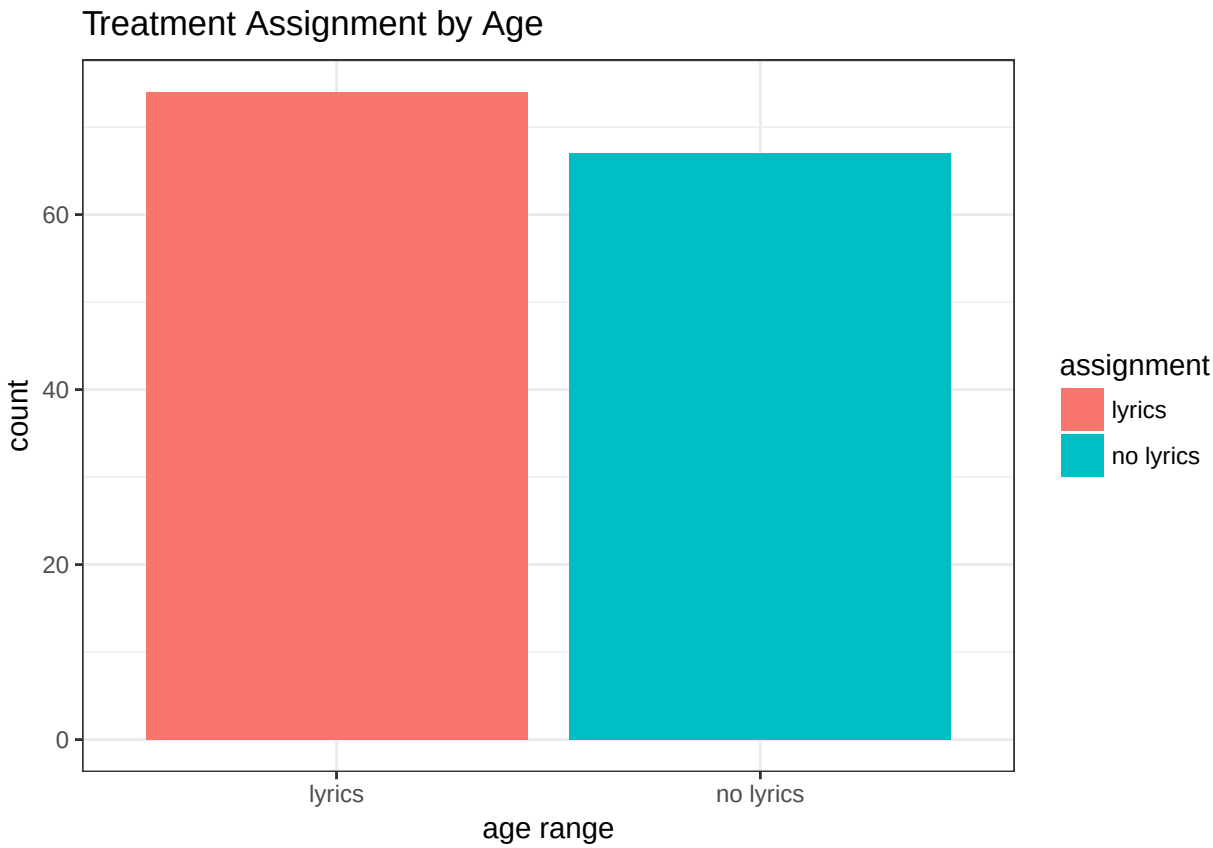
Statistic	N	Mean	St. Dev.	Min	Max
hear_song	141	0.993	0.084	0	1
piano_playing	141	0.993	0.084	0	1
own_dog	140	0.457	0.500	0	1
native_english	140	0.914	0.281	0	1
is_turk	141	0.766	0.425	0	1
time	141	448.348	210.102	68	1,478
correct_count	141	3.248	1.305	0	5
assigned_lyrics	141	0.525	0.501	0	1

Roughly 50/50 split for treatment vs. control:

```
dt[, .N, by = 'assigned_lyrics']
```

```
##    assigned_lyrics  N
## 1:               1 74
## 2:               0 67
```

```
ggplot(data = dt, aes(x = lyrics_factor, group = lyrics_factor, fill = lyrics_factor)) +
  geom_bar() +
  theme_bw() +
  guides(fill=guide_legend(title="assignment")) +
  labs(
    title = "Treatment Assignment by Age",
    x = "age range",
    y = "count"
    )
```




```
# ggsave("treatment_by_age.png")
```

Most people recognized the song regardless of being assigned lyrics:

```
dt[, .N, by = 'heard_lyrics,assigned_lyrics']
```

```
##           heard_lyrics assigned_lyrics  N
## 1:          Rocket Man               1 63
## 2:          Rocket Man               0 53
## 3:                  NA               1 11
## 4:                  NA               0 13
## 5: Lyrics but not sure               0  1
```

Turkers took roughly 40% less time to complete the survey than non-turkers:

```
dt[, mean(time), by = 'is_turk']
```

```
##    is_turk       V1
## 1:       1 410.2778
## 2:       0 572.9394
```

```
dt[ , t.test(time ~ is_turk)]
```

```
##
##  Welch Two Sample t-test
##
## data:  time by is_turk
## t = 3.0192, df = 37.743, p-value = 0.004526
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##   53.57157 271.75166
## sample estimates:
## mean in group 0 mean in group 1
##        572.9394        410.2778
```

No significant difference in time taken based on treatment vs. control assignment:

```
dt[, mean(time), by = 'assigned_lyrics']
```

```
##    assigned_lyrics       V1
## 1:               1 471.8378
## 2:               0 422.4030
```

```
dt[ , t.test(time ~ assigned_lyrics)]
```

```
##
##  Welch Two Sample t-test
##
## data:  time by assigned_lyrics
## t = -1.3895, df = 130.85, p-value = 0.167
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -119.81669   20.94699
## sample estimates:
## mean in group 0 mean in group 1
##        422.4030        471.8378
```

Covariate Balance Check

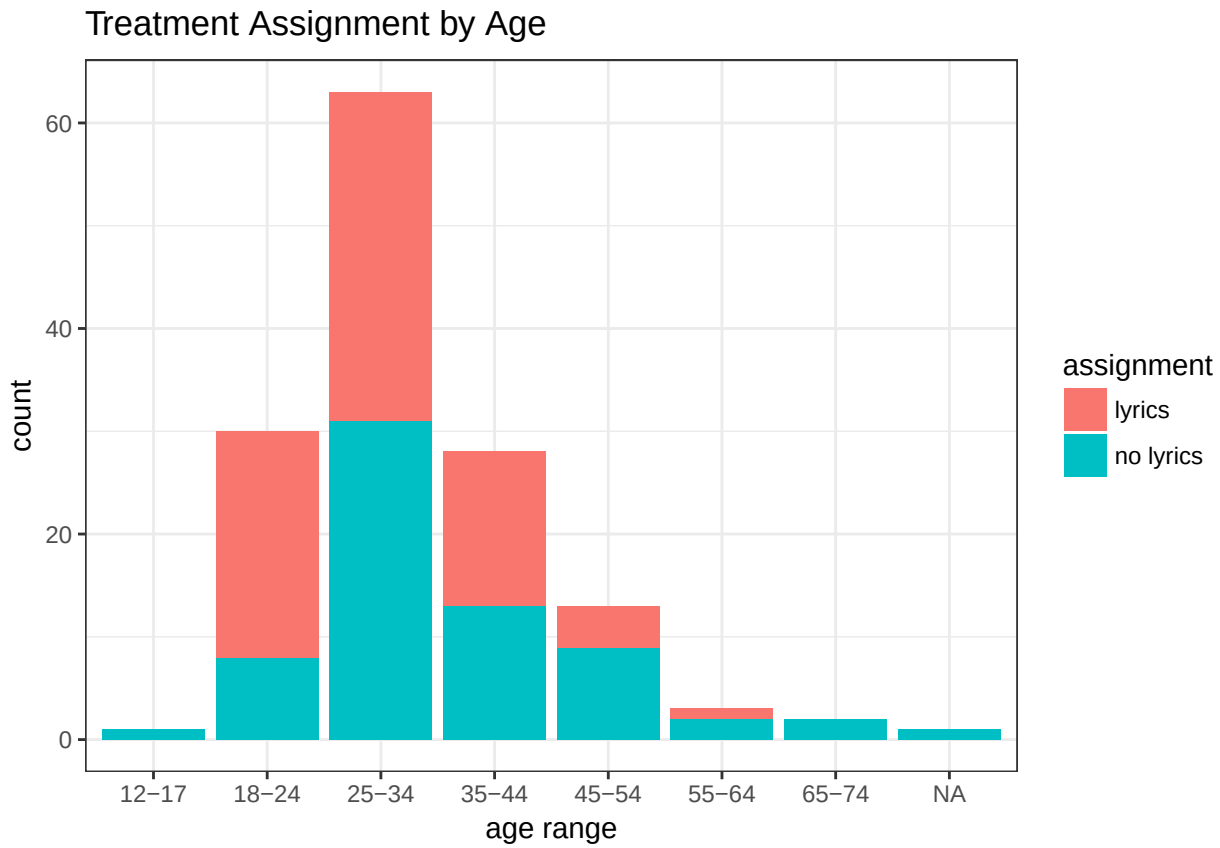
```
table(dt$assigned_lyrics, dt$age)
```

```
##
##     <12 12-17 18-24 25-34 35-44 45-54 55-64 65-74 >75 decline
##   0   0     1     8    31    13     9     2     2   0       0
##   1   0     0    22    32    15     4     1     0   0       0
```

```
# assignment_by_age <- table(dt$assigned_lyrics, dt$age)
# barplot(assignment_by_age, main = 'Treatment Assignment by Age',
#         xlab = "Age Range", col = c('darkblue', 'red'),
#         legend = c('no lyrics', 'lyrics'))

# dt[, .N, keyby = 'age,assigned_lyrics']

ggplot(data = dt, aes(x = age, group = lyrics_factor, fill = lyrics_factor)) +
  geom_bar() +
  theme_bw() +
  guides(fill=guide_legend(title="assignment")) +
  labs(
    title = "Treatment Assignment by Age",
    x = "age range",
    y = "count"
    )
```



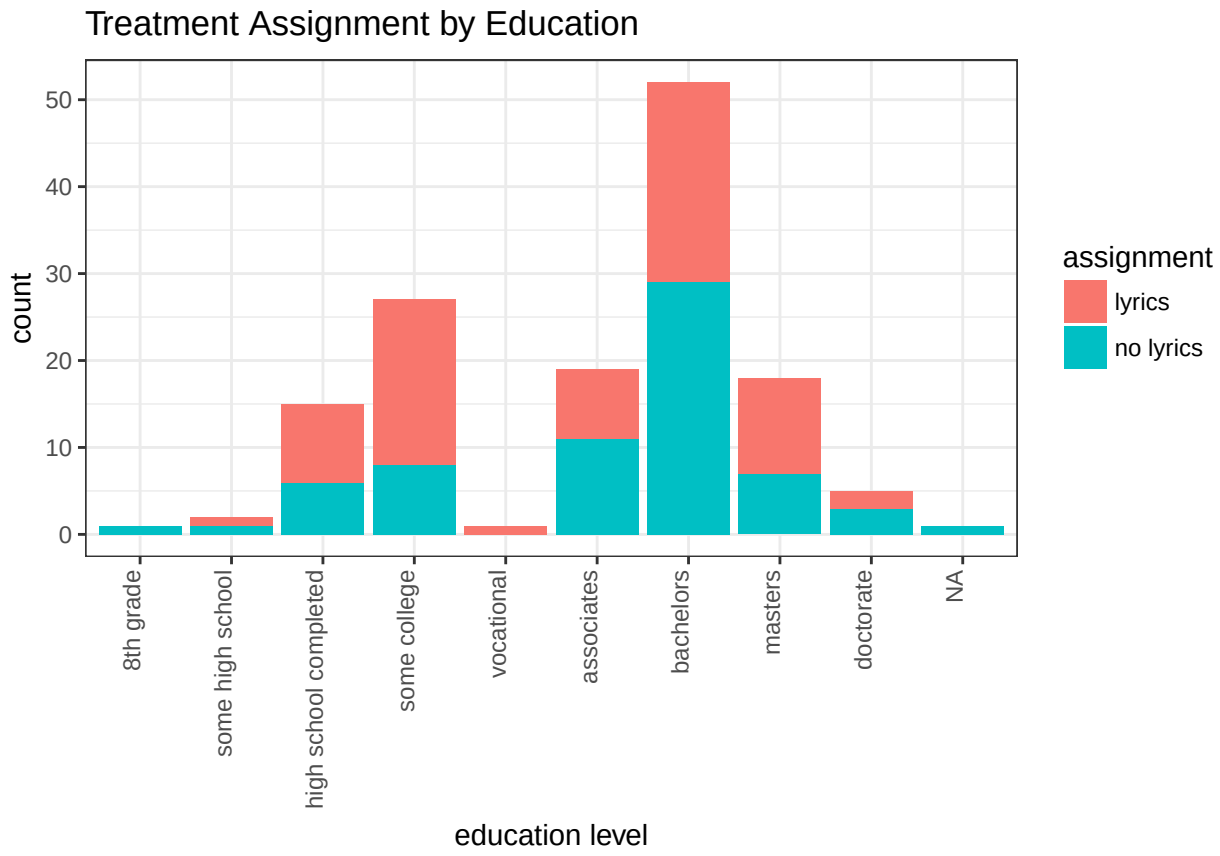

```
ggsave("treatment_by_age.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
table(dt$assigned_lyrics, dt$education)
```

```
##
##     none 8th grade some high school high school completed some college
##   0    0         1                1                      6            8
##   1    0         0                1                      9           19
##
##     vocational associates bachelors masters doctorate
##   0          0         11        29       7         3
##   1          1          8        23      11         2
```

```
ggplot(data = dt, aes(x = education, group = lyrics_factor, fill = lyrics_factor)) +
  geom_bar() +
  theme_bw() +
  guides(fill=guide_legend(title="assignment")) +
  labs(
    title = "Treatment Assignment by Education",
    x = "education level",
    y = "count"
    ) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0))
```




```
ggsave("treatment_by_education.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
dt[ , chisq.test(assigned_lyrics, age, simulate.p.value = TRUE)]
```

```
##
##  Pearson's Chi-squared test with simulated p-value (based on 2000
##  replicates)
##
## data:  assigned_lyrics and age
## X-squared = 11.529, df = NA, p-value = 0.05197
```

```
dt[ , chisq.test(assigned_lyrics, education, simulate.p.value = TRUE)]
```

```
##
##  Pearson's Chi-squared test with simulated p-value (based on 2000
##  replicates)
##
## data:  assigned_lyrics and education
## X-squared = 8.9083, df = NA, p-value = 0.3593
```

```
dt[ , t.test(native_english ~ assigned_lyrics)]
```

```
##
##  Welch Two Sample t-test
##
## data:  native_english by assigned_lyrics
## t = 1.65, df = 125.47, p-value = 0.1014
```

```
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.01519006  0.16752421
## sample estimates:
## mean in group 0 mean in group 1
##       0.9545455       0.8783784
```

```
dt[ , t.test(is_turk ~ assigned_lyrics)]
```

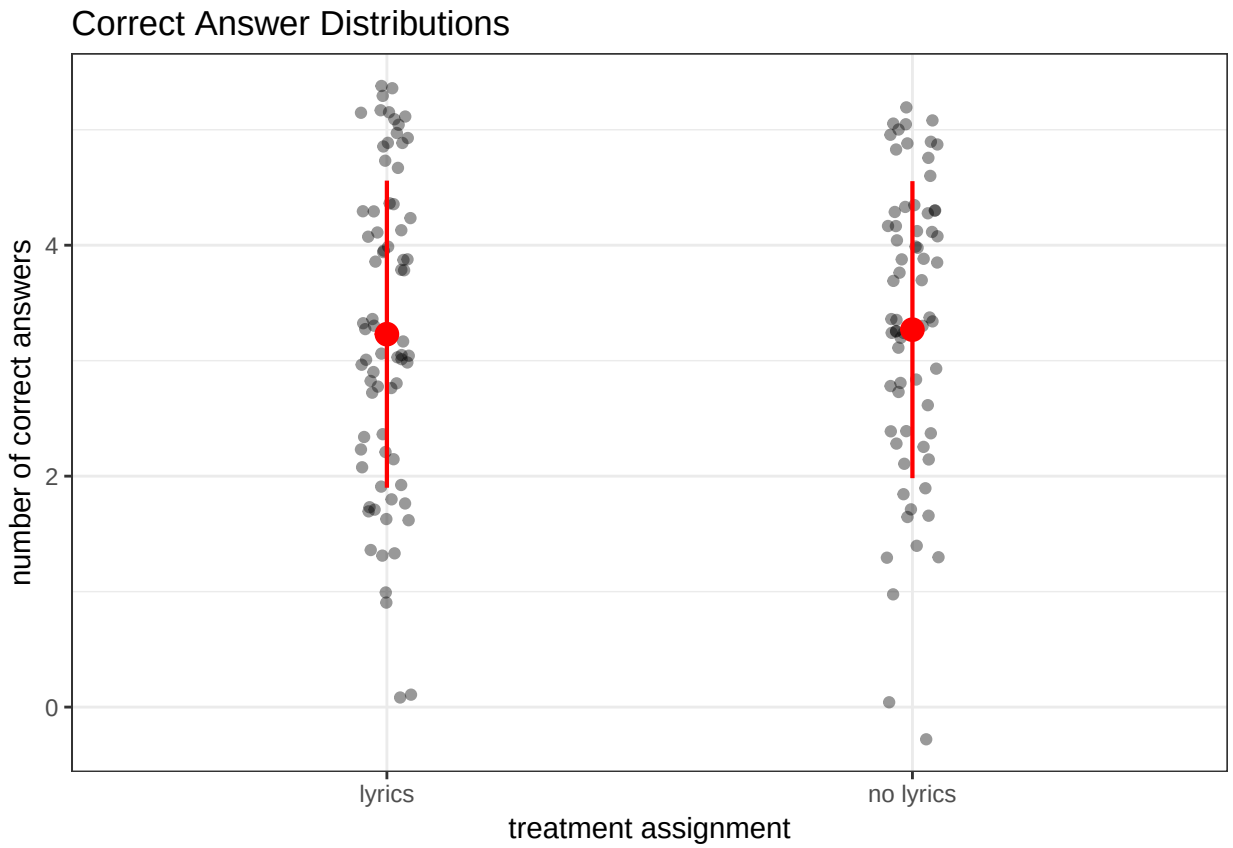
```
##
##  Welch Two Sample t-test
##
## data:  is_turk by assigned_lyrics
## t = -0.12613, df = 137.21, p-value = 0.8998
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1513702  0.1332177
## sample estimates:
## mean in group 0 mean in group 1
##       0.7611940       0.7702703
```

Results

```
# correct_count_by_assignment <- table(dt$assigned_lyrics, dt$correct_count)
# x <- barplot(correct_count_by_assignment, main = 'Correct Count by Treatment Assignment',
#         col = c('orange', 'purple'), beside = TRUE, space = c(0, 0.2),
#         legend = c('no lyrics', 'lyrics'), args.legend = c(xjust = 5))

# ggplot(data = dt, aes(x = correct_count, group = lyrics_factor, fill = lyrics_factor)) +
#   geom_bar(position = "dodge") +
#   theme_bw() +
#   guides(fill=guide_legend(title="assignment")) +
#   labs(
#     title = "Correct Answers by Treatment Assignment",
#     x = "number of correct answers",
#     y = "count"
#     )

ggplot(data = dt, aes(x = lyrics_factor, y = correct_count,
                      group = lyrics_factor, fill = lyrics_factor)) +
  geom_jitter(width = .05, alpha = .4) +
  stat_summary(fun.data="mean_sdl", colour = 'red', size = .75, fun.args = 1) +
  guides(fill = "none") +
    theme_bw() +
    labs(
      title = "Correct Answer Distributions",
      x = "treatment assignment",
      y = "number of correct answers"
    )
```

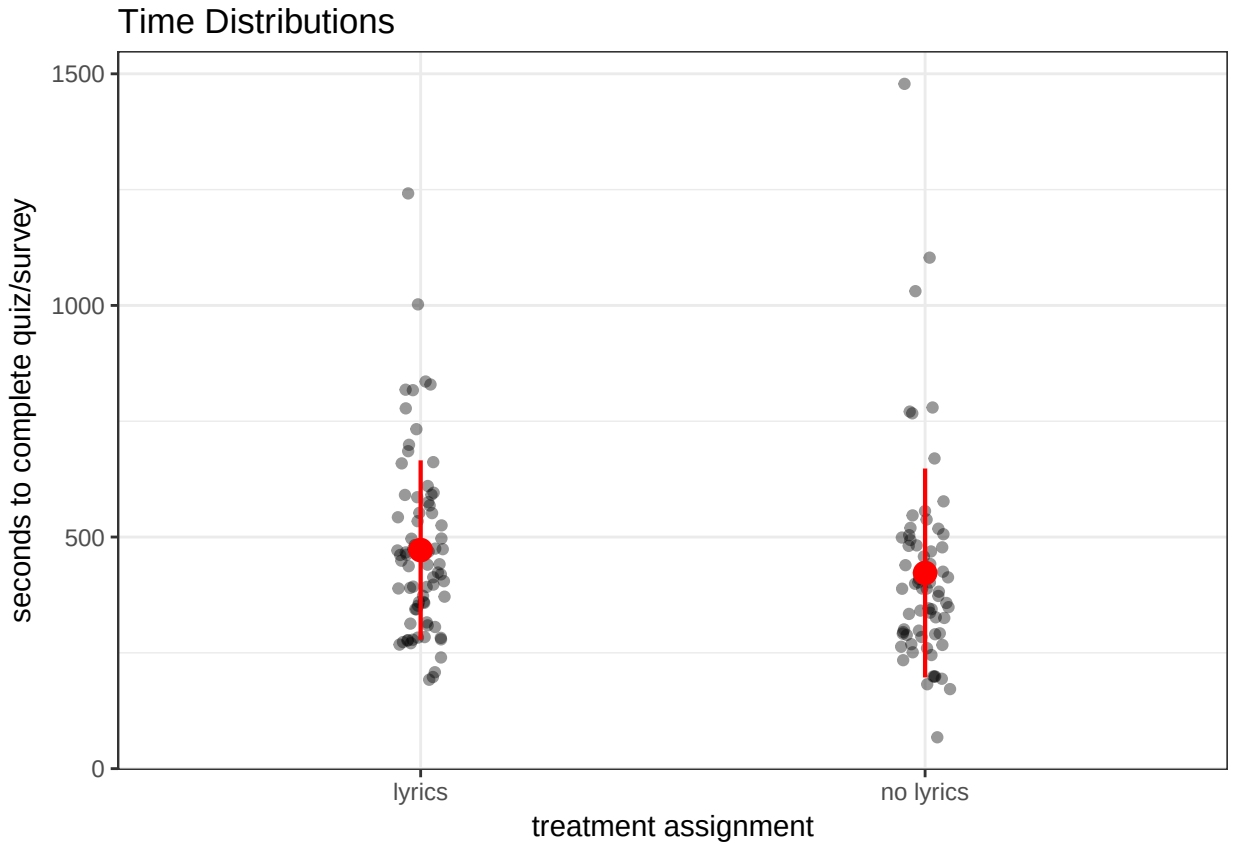



```
ggsave("answers_individual_values.png", width = 3)
```

```
## Saving 3 x 4.5 in image
```

```
# ggplot(data = dt, aes(x = lyrics_factor, y = time, group = lyrics_factor, fill = lyrics_factor)) +
#   geom_boxplot(alpha = .7, varwidth = TRUE) +
#   geom_jitter(width = .05, alpha = .4) +
#   guides(fill = "none") +
#     theme_bw() +
#     labs(
#       title = "Time Distributions Under Treatment and Control",
#       x = "treatment assignment",
#       y = "seconds to complete quiz/survey"
#     )
# ggsave("time_individual_values.png")

ggplot(data = dt, aes(x = lyrics_factor, y = time,
                      group = lyrics_factor, fill = lyrics_factor)) +
  geom_jitter(width = .05, alpha = .4) +
  stat_summary(fun.data="mean_sdl", colour = 'red', size = .75, fun.args = 1) +
  guides(fill = "none") +
    theme_bw() +
    labs(
      title = "Time Distributions",
      x = "treatment assignment",
      y = "seconds to complete quiz/survey"
    )
```




```
ggsave("time_individual_values.png", width = 3)
```

```
## Saving 3 x 4.5 in image
```

Regression

No significant difference in scores between treatment and control groups:

```
fit_all <- lm(correct_count ~ assigned_lyrics, dt)
summary(fit_all)
```

```
##
## Call:
## lm(formula = correct_count ~ assigned_lyrics, data = dt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2687 -1.2297 -0.2297  0.7703  1.7703
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.26866    0.15995  20.436   <2e-16 ***
## assigned_lyrics -0.03893    0.22079  -0.176     0.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.309 on 139 degrees of freedom
```

```
## Multiple R-squared:  0.0002236,  Adjusted R-squared:  -0.006969
## F-statistic: 0.03109 on 1 and 139 DF,  p-value: 0.8603
```

The only significant difference found is that Turkers scored roughly 0.90 out of 5 lower on average:

```
fit_all_with_covariates <- lm(correct_count ~ assigned_lyrics + age + education + is_turk, dt)
summary(fit_all_with_covariates)
```

```
##
## Call:
## lm(formula = correct_count ~ assigned_lyrics + age + education +
##     is_turk, data = dt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.06555 -0.77781  0.04613  0.96618  2.36494
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1.91542    1.82791   1.048  0.29675
## assigned_lyrics                -0.24307    0.22727  -1.070  0.28694
## age18-24                        0.95387    1.30403   0.731  0.46588
## age25-34                        0.03857    1.28843   0.030  0.97617
## age35-44                        0.46906    1.30518   0.359  0.71992
## age45-54                        0.23151    1.32174   0.175  0.86124
## age55-64                        0.51879    1.46818   0.353  0.72443
## age65-74                       -0.79355    1.58686  -0.500  0.61791
## educationsome high school       0.80982    1.55413   0.521  0.60325
## educationhigh school completed  1.64875    1.30140   1.267  0.20758
## educationsome college           2.01576    1.28138   1.573  0.11826
## educationvocational             0.24307    1.77408   0.137  0.89125
## educationassociates             2.03857    1.28843   1.582  0.11617
## educationbachelors              1.87813    1.26906   1.480  0.14145
## educationmasters                2.08222    1.30111   1.600  0.11209
## educationdoctorate              1.16109    1.41232   0.822  0.41260
## is_turk                        -0.95399    0.30940  -3.083  0.00253 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.244 on 123 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.2009, Adjusted R-squared:  0.09695
## F-statistic: 1.933 on 16 and 123 DF,  p-value: 0.02322
```

```
fit_turk_only <- lm(correct_count ~ assigned_lyrics, dt[is_turk == 1])
summary(fit_turk_only)
```

```
##
## Call:
## lm(formula = correct_count ~ assigned_lyrics, data = dt[is_turk ==
##     1])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1765 -0.9825  0.0175  1.0175  2.0175
##
```

```
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.1765     0.1837  17.293   <2e-16 ***
## assigned_lyrics  -0.1940     0.2528  -0.767    0.445
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.312 on 106 degrees of freedom
## Multiple R-squared:  0.005524,   Adjusted R-squared:  -0.003858
## F-statistic: 0.5888 on 1 and 106 DF,  p-value: 0.4446
```

```
fit_no_turks <- lm(correct_count ~ assigned_lyrics, dt[is_turk == 0])
summary(fit_no_turks)
```

```
##
## Call:
## lm(formula = correct_count ~ assigned_lyrics, data = dt[is_turk ==
##     0])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0588 -1.0588  0.4375  0.9412  1.4375
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.5625     0.2800  12.724 7.61e-14 ***
## assigned_lyrics   0.4963     0.3901   1.272    0.213
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.12 on 31 degrees of freedom
## Multiple R-squared:  0.04963,    Adjusted R-squared:  0.01898
## F-statistic: 1.619 on 1 and 31 DF,  p-value: 0.2127
```

```
dt <- dt[, correct_per_second := correct_count/time]

fit_per_time <- lm(correct_per_second ~ assigned_lyrics, dt)
summary(fit_per_time)
```

```
##
## Call:
## lm(formula = correct_per_second ~ assigned_lyrics, data = dt)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.009835 -0.004021 -0.000897  0.002272  0.048989
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.0098350  0.0008025  12.256   <2e-16 ***
## assigned_lyrics -0.0017699  0.0011077  -1.598    0.112
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.006568 on 139 degrees of freedom
```

```
## Multiple R-squared:  0.01804,    Adjusted R-squared:  0.01097
## F-statistic: 2.553 on 1 and 139 DF,  p-value: 0.1123
```

Power Calculation

```
# Calculating number of subjects needed for 80% power (BASED ON TIME AS THE OUTCOME)

cohens_d <- function(x, y) {
    lx <- length(x)- 1
    ly <- length(y)- 1
    md  <- abs(mean(x) - mean(y))        ## mean difference (numerator)
    csd <- lx * var(x) + ly * var(y)
    csd <- csd/(lx + ly)
    csd <- sqrt(csd)                     ## common sd computation

    cd  <- md/csd                        ## cohen's d
}

(effect_size_time <- cohens_d(dt[assigned_lyrics==1, time], dt[assigned_lyrics==0, time]))
```

```
## [1] 0.2360955
```

```
pwr.t.test(power = 0.8, d = effect_size_time, sig.level = 0.05, type = "two.sample")
```

```
##
##      Two-sample t test power calculation
##
##               n = 282.5822
##               d = 0.2360955
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```
# Calculating what power we got for our experiment

(effect_size_correct_count <- cohens_d(dt[assigned_lyrics==1, correct_count], dt[assigned_lyrics==0, co
```

```
## [1] 0.02973266
```

```
pwr.t2n.test(n1 = 74, n2 = 67, d = effect_size_correct_count, sig.level = 0.05)
```

```
##
##      t test power calculation
##
##              n1 = 74
##              n2 = 67
##               d = 0.02973266
##       sig.level = 0.05
##           power = 0.0535194
##     alternative = two.sided
```