turk_results

Gurdit Chahal, Adam Letcher, Carlos Castro, Noah Randolph

8/7/2018

Load Data

```
dt_raw <- fread('musicdata.8.11.2018.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.   :0.000   Min.   :0.000   A:  1   A: 8   A:16   A:75   A:17
##  1st Qu.:1.000   1st Qu.:1.000   B:  6   B:98   B: 3   B: 5   B:26
##  Median :1.000   Median :1.000   C:132   C:15   C:36   C:42   C:15
##  Mean   :0.986   Mean   :0.986   D:  2   D: 4   D: 3   D: 6   D:75
##  3rd Qu.:1.000   3rd Qu.:1.000   E:  2   E:18   E:85   E:14   E: 9
##  Max.   :1.000   Max.   :1.000                          N: 1   N: 1
##
##       age         gender      own_dog                        education
##  25-34  :64   male   :79   Min.   :0.0000   bachelors             :54
##  18-24  :30   female :63   1st Qu.:0.0000   some college          :27
##  35-44  :29   other  : 0   Median :0.0000   associates            :19
##  45-54  :13   decline: 0   Mean   :0.4577   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:143         Min.   :0.0000   I'm a barbie girl   :  0
##  Class :character   1st Qu.:1.0000   Rocket Man          :118
```

```
##  Mode  :character   Median :1.0000   Don't stop believing:  0
##                     Mean   :0.9085   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.   :0.0000   Min.   :  68.0   Min.   :0.000   Min.   :0.0000
##  1st Qu.:1.0000   1st Qu.: 303.5   1st Qu.:2.000   1st Qu.:0.0000
##  Median :1.0000   Median : 409.0   Median :3.000   Median :1.0000
##  Mean   :0.7552   Mean   : 450.3   Mean   :3.252   Mean   :0.5315
##  3rd Qu.:1.0000   3rd Qu.: 522.5   3rd Qu.:4.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1478.0   Max.   :5.000   Max.   :1.0000
##
##    lyrics_factor
##  lyrics   :76
##  no lyrics:67
##
##
##
##
##
```

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

Table 1:

Statistic	N	Mean	St. Dev.	Min	Max
hear_song	143	0.986	0.118	0	1
piano_playing	143	0.986	0.118	0	1
own_dog	142	0.458	0.500	0	1
native_english	142	0.908	0.289	0	1
is_turk	143	0.755	0.431	0	1
time	143	450.259	210.034	68	1,478
correct_count	143	3.252	1.297	0	5
assigned_lyrics	143	0.531	0.501	0	1

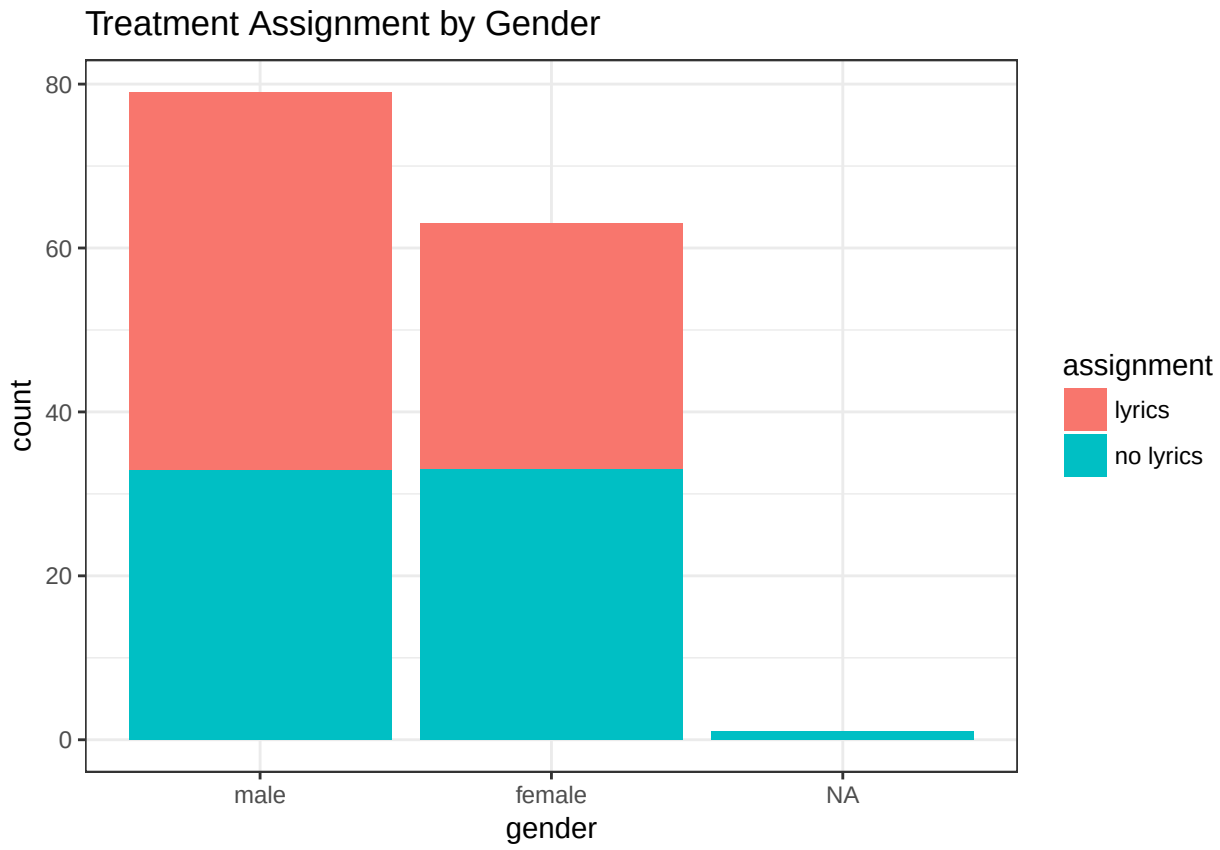
Gender counts of treatment and control:

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

```
##    assigned_lyrics gender  N
## 1:               1   male 46
## 2:               0   male 33
## 3:               0 female 33
## 4:               1 female 30
## 5:               0     NA  1
```

```
ggplot(data = dt, aes(x = gender, group = lyrics_factor, fill = lyrics_factor)) +
  geom_bar() +
  theme_bw() +
  guides(fill=guide_legend(title="assignment")) +
  labs(
    title = "Treatment Assignment by Gender",
    x = "gender",
```

```
    y = "count"
    )
```




```
ggsave("gender_treatment_assignment.png")
```

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

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 65
## 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:       0 573.6286
```

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

```
##
##  Welch Two Sample t-test
```

```
##
## data:  time by is_turk
## t = 3.178, df = 40.772, p-value = 0.002828
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   59.52673 267.17485
## sample estimates:
## mean in group 0 mean in group 1
##        573.6286        410.2778
```

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

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

```
##    assigned_lyrics       V1
## 1:               1 474.8158
## 2:               0 422.4030
```

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

```
##
##  Welch Two Sample t-test
##
## data:  time by assigned_lyrics
## t = -1.4813, df = 130.98, p-value = 0.1409
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -122.40904   17.58343
## sample estimates:
## mean in group 0 mean in group 1
##        422.4030        474.8158
```

Covariate Balance Check

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

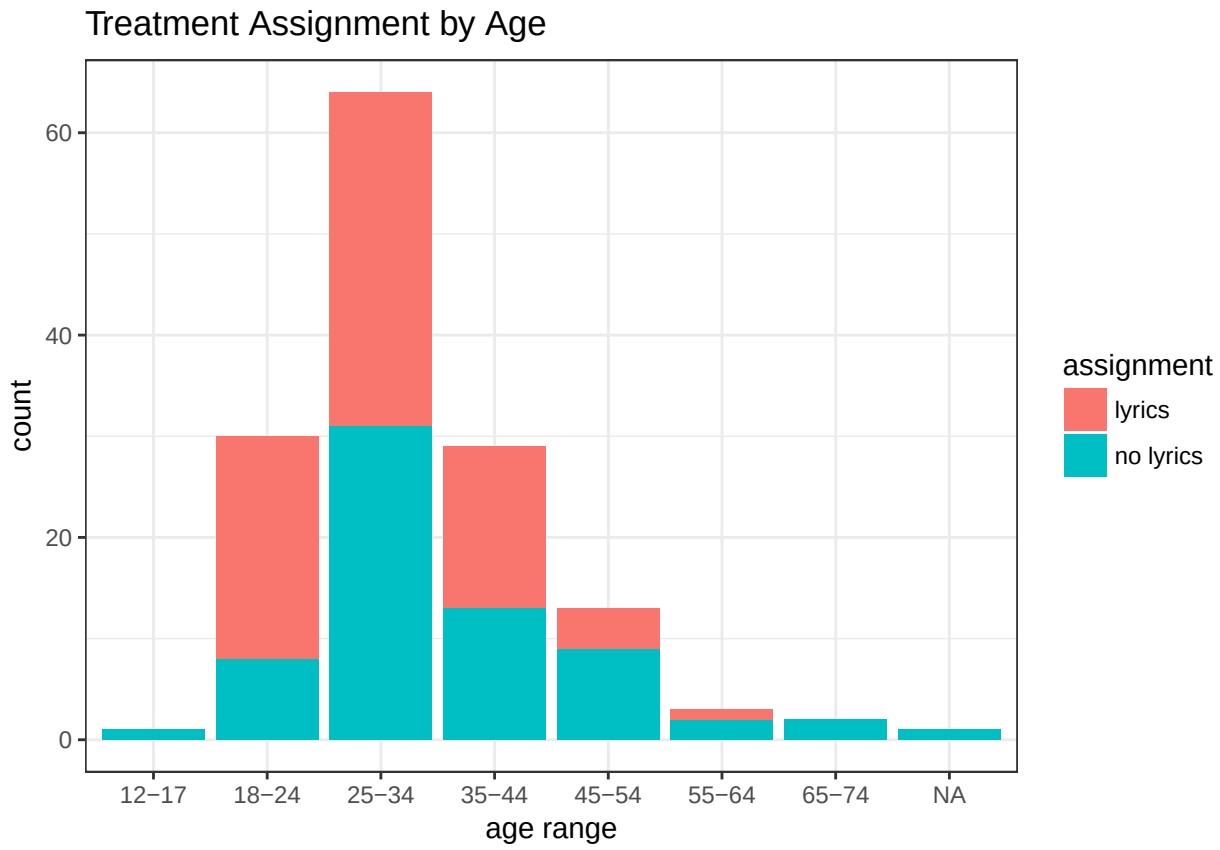
```
##
##     <12 12-17 18-24 25-34 35-44 45-54 55-64 65-74 >75 decline
##   0   0     1     8    31    13     9     2     2   0       0
##   1   0     0    22    33    16     4     1     0   0       0
```

```
# 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"
    )
```



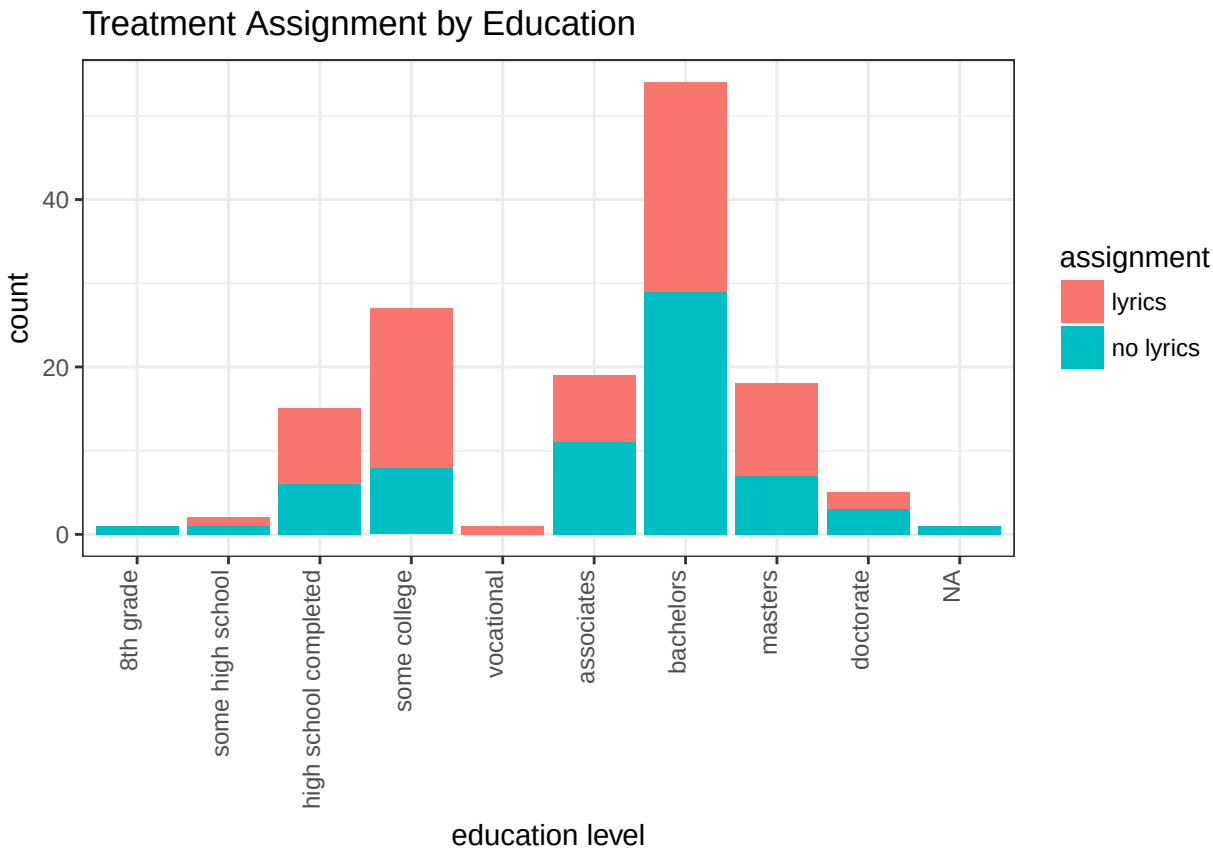

```
ggsave("treatment_by_age.png")
```

```
## 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        25      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 = 11.515, 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 = 8.2772, df = NA, p-value = 0.4353
```

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

```
##
##  Welch Two Sample t-test
##
## data:  native_english by assigned_lyrics
## t = 1.8399, df = 126.99, p-value = 0.06811
```

```
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.00650176  0.17875056
## sample estimates:
## mean in group 0 mean in group 1
##       0.9545455       0.8684211
```

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

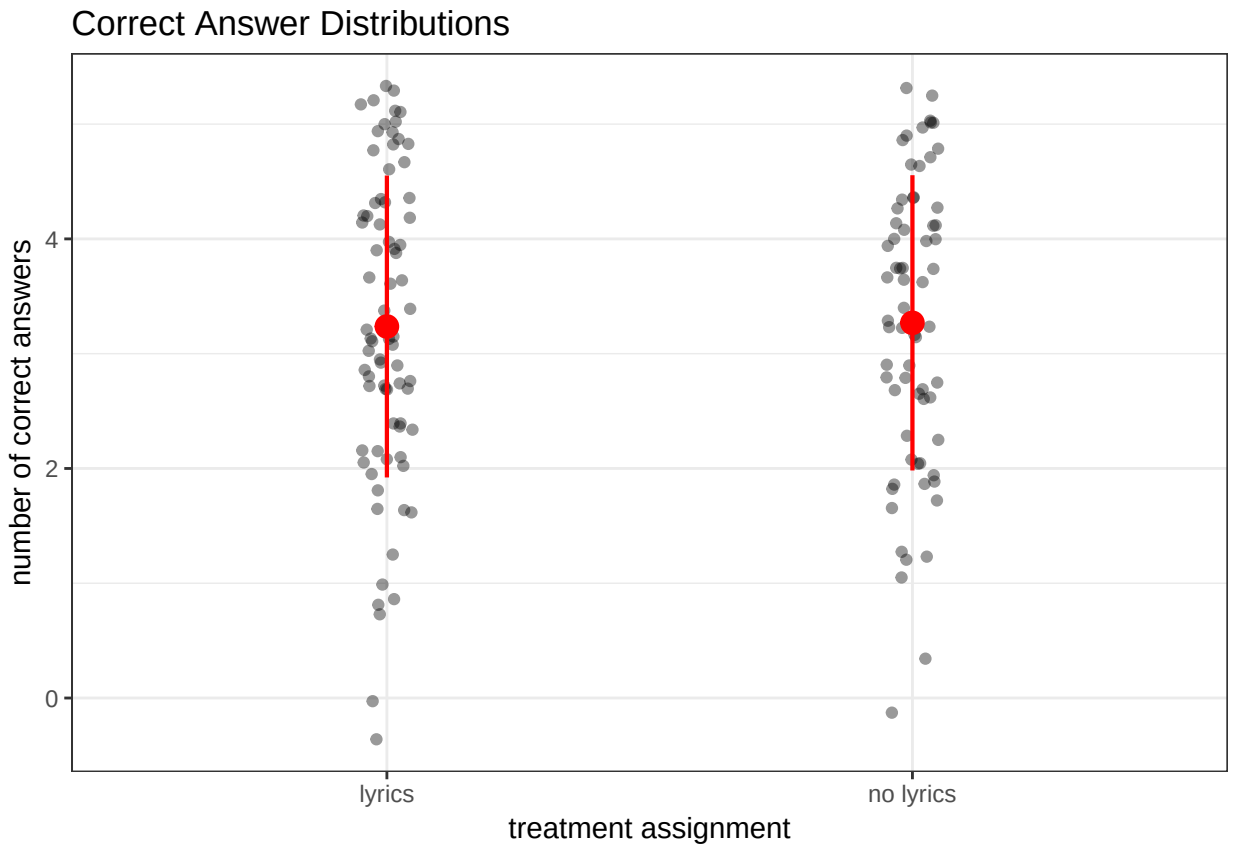
```
##
##  Welch Two Sample t-test
##
## data:  is_turk by assigned_lyrics
## t = 0.15443, df = 139.24, p-value = 0.8775
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1321213  0.1545094
## sample estimates:
## mean in group 0 mean in group 1
##        0.761194        0.750000
```

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"
    )
```

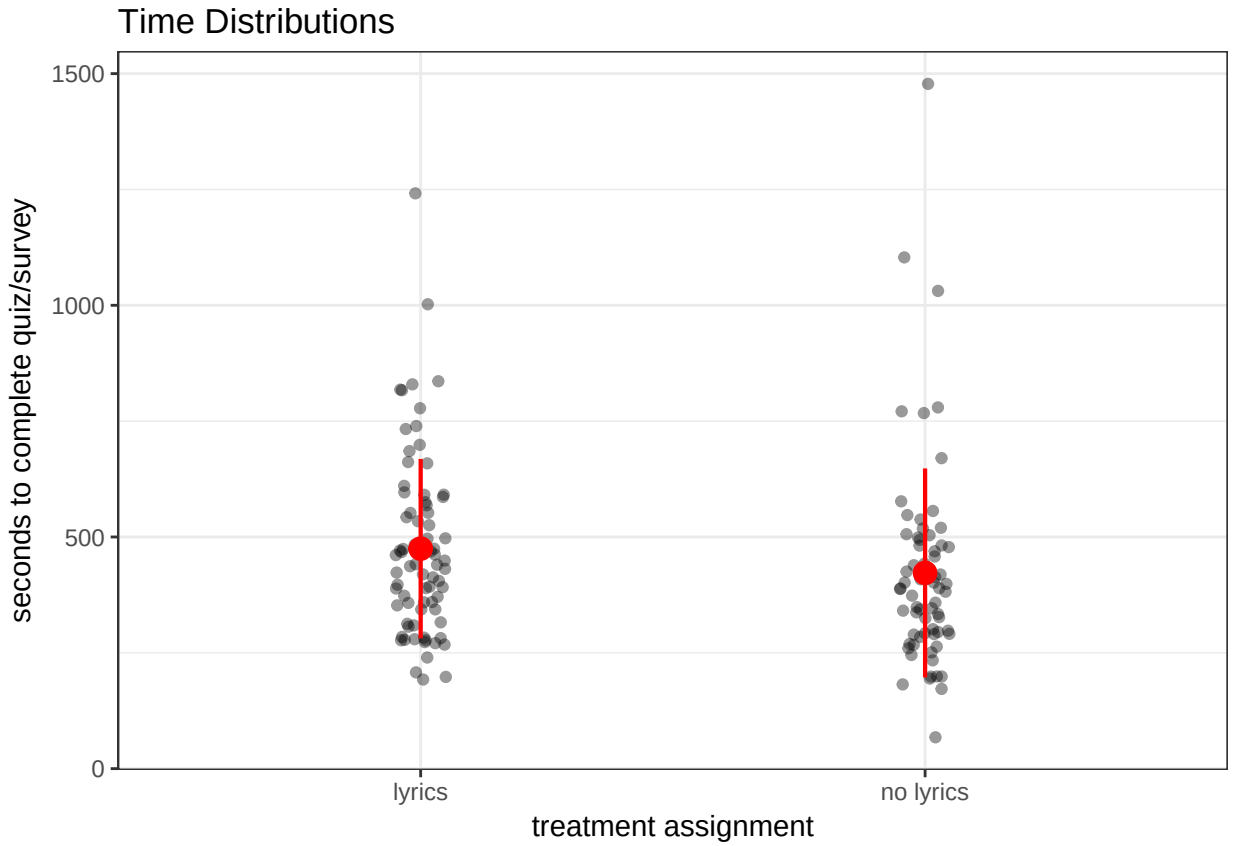



```
ggsave("answers_individual_values.png", width = 3)
```

```
## 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_simple <- lm(correct_count ~ assigned_lyrics, dt)
cov_simple <- vcovHC(fit_simple, type = 'HC')
robust.se_simple <- sqrt(diag(cov_simple))
```

With covariates

```
fit_with_covariates <- lm(correct_count ~ assigned_lyrics + gender + native_english + own_dog + is_turk
cov_with_covariates <- vcovHC(fit_with_covariates, type = 'HC')
robust.se_with_covariates <- sqrt(diag(cov_with_covariates))
```

```
stargazer(fit_simple, fit_with_covariates,
          se=list(robust.se_simple, robust.se_with_covariates),
          dep.var.labels=c("correct answer count"),
          covariate.labels=c("assigned lyrics","female", "native english speaker","owns dog","mechanica
          keep.stat="n")
```

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Sat, Aug 11, 2018 - 16:20:35

Table 2:

	Dependent variable:	
	correct answer count	
	(1)	(2)
assigned lyrics	−0.032	−0.006
	(0.216)	(0.217)
female		0.064
		(0.212)
native english speaker		0.994***
		(0.260)
owns dog		−0.419**
		(0.207)
mechanical turk		−0.935***
		(0.248)
Constant	3.269***	3.238***
	(0.156)	(0.312)
Observations	143	141
Note:	*p<0.1; **p<0.05; ***p<0.01	

```r
fit_with_education <- lm(correct_count ~ assigned_lyrics + education, dt)
cov_with_education <- vcovHC(fit_with_education, type = 'HC')
robust.se_with_education <- sqrt(diag(cov_with_education))
```

```
## Warning in sqrt(diag(cov_with_education)): NaNs produced
```

```r
stargazer(fit_with_education,
          se=list(robust.se_with_education),
          dep.var.labels=c("correct answer count"),
          covariate.labels=c("assigned lyrics","some high school", "high school completed","some college
                             "vocational", "associate", "bachelor", "master", "doctorate"),
          keep.stat="n")
```

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Sat, Aug 11, 2018 - 16:20:35

Table 3:

	Dependent variable:
	correct answer count
assigned lyrics	−0.028
	(0.229)
some high school	1.014***
	(0.115)
high school completed	2.017***
	(0.398)
some college	2.205***
	(0.302)
vocational	0.028
	(0.229)
associate	2.380***
	(0.264)
bachelor	2.346***
	(0.201)
master	2.628***
	(0.296)
doctorate	2.211***
	(0.530)
Constant	1.000
Observations	142
Note:	*p<0.1; **p<0.05; ***p<0.01

```
fit_with_age <- lm(correct_count ~ assigned_lyrics + age, dt)
cov_with_age <- vcovHC(fit_with_age, type = 'HC')
robust.se_with_age <- sqrt(diag(cov_with_age))
```

```
## Warning in sqrt(diag(cov_with_age)): NaNs produced
```

```
stargazer(fit_with_age,
          se=list(robust.se_with_age),
          dep.var.labels=c("correct answer count"),
          covariate.labels=c("assigned lyrics","ages 18-24", "ages 25-34","ages 35-44",
                             "ages 45-54", "ages 55-64", "ages 65-74"),
          keep.stat="n")
```

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Sat, Aug 11, 2018 - 16:20:35

Table 4:

	Dependent variable:
	correct answer count
assigned lyrics	−0.185
	(0.220)
ages 18-24	0.936***
	(0.247)
ages 25-34	0.095
	(0.217)
ages 35-44	0.481**
	(0.239)
ages 45-54	0.057
	(0.312)
ages 55-64	0.395
	(0.327)
ages 65-74	−0.000
	(0.707)
Constant	3.000
Observations	142
Note:	*p<0.1; **p<0.05; ***p<0.01

Regression with elapsed time as outcome

```
fit_time <- lm(time ~ assigned_lyrics, dt)
cov_time <- vcovHC(fit_time, type = 'HC')
```

```
robust.se_time <- sqrt(diag(cov_time))
```

With covariates

```
fit_time_with_covariates <- lm(time ~ assigned_lyrics + gender + native_english + own_dog + is_turk, dt
cov_time_with_covariates <- vcovHC(fit_time_with_covariates, type = 'HC')
robust.se_time_with_covariates <- sqrt(diag(cov_time_with_covariates))
```

```
stargazer(fit_time, fit_time_with_covariates,
          se=list(robust.se_time, robust.se_time_with_covariates),
          dep.var.labels=c("time to complete (seconds)"),
          covariate.labels=c("assigned lyrics","female", "native english speaker","owns dog",
                             "mechanical turk"),
          keep.stat="n")
```

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Sat, Aug 11, 2018 - 16:20:35

Table 5:

	Dependent variable:	
	time to complete (seconds)	
	(1)	(2)
assigned lyrics	52.413	32.450
	(35.130)	(33.971)
female		−14.663
		(31.400)
native english speaker		−169.416*
		(95.261)
owns dog		24.049
		(29.385)
mechanical turk		−137.290***
		(48.560)
Constant	422.403***	687.803***
	(27.340)	(109.664)
Observations	143	141
Note:	*p<0.1; **p<0.05; ***p<0.01	

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] 0.2506285
```

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

```
##
##      Two-sample t test power calculation
##
##               n = 250.8695
##               d = 0.2506285
##       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] 0.02444146
```

```
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 = 0.02444146
##       sig.level = 0.05
##           power = 0.05237661
##     alternative = two.sided
```