Final Project

Tina Huang, Gunnar Mein, Jeff Li

7/27/2020

Preprocessing the data

```
#import files
rote_cov_original <- read.csv(file="/Users/jeff/Documents/MIDS/W241/logs/rote_cov.csv", sep=',') #covar
rote_test_original <- read.csv(file="/Users/jeff/Documents/MIDS/W241/logs/rote_test.csv", sep=',') #tes

rote_cov_original$gender
```

```
##   [1] <NA>  M     F     F     F     F     F     M     F     F     F
##  [12] M     M     F     M     M     F     M     M     F     F     M
##  [23] F     F     F     M     F     F     M     F     F     F     F
##  [34] M     F     M     F     F     F     F     F     F     other F
##  [45] F     F     M     M     F     F     M     F     F     F     M
##  [56] F     F     F     F     F     F     F     M     F     M     M
##  [67] other M     F     F     F     F     <NA>  M     F     F     F
##  [78] F     F     F     M     M     F     M     F     <NA>  F     F
##  [89] F     F     F     M     M     <NA>  M     M     <NA>  M     F
## [100] M     F     M     M     F     F     F     M     M     F     F
## [111] F     F     F     M     F     F     M     F     M     F     F
## [122] F     F     M     F     F     M     <NA>  F     M     M     M
## [133] F     F     F     F     F     M     M     F     M     M     F
## [144] M     F     F     F     M     M     F     M     F     <NA>  M
## [155] F     F     M     F     F     M     F     F     F     F     M
## [166] F     M     M     M     F     F     F     F     M     F     F
## [177] M     M     F     F
## Levels: F M other
```

```
#create copies
rote_cov <- rote_cov_original
rote_test <- rote_test_original

#convert unix epoch time to datetime
#cov
rote_cov$session_start_time <- as.POSIXct(rote_cov$session_start_time/1000, origin="1970-01-01")
rote_cov$cov_submit_time <- as.POSIXct(rote_cov$cov_submit_time/1000, origin="1970-01-01")

#test
rote_test$session_start_time <- as.POSIXct(rote_test$session_start_time/1000, origin="1970-01-01")
rote_test$test_submit_time <- as.POSIXct(rote_test$test_submit_time/1000, origin="1970-01-01")
rote_test$test_time <- rote_test$test_submit_time - rote_test$session_start_time

###remove rows with any NAs in the answers(indicates they did not complete the test)
rote_test <- rote_test %>% drop_na("a11", "a12", "a13", "a14", "a21", "a22", "a23","a24")

rote_test_baseline <- rote_test %>% filter(test == 'baseline')
rote_test_experiment <- rote_test %>% filter(test == 'experiment') %>% select('session_id','item_id1','a

rote_test <- inner_join(rote_test_baseline, rote_test_experiment, by="session_id" )
```

```
###remove rows with missing covariates
#covert age, remove outliers for age
rote_cov$age <- ifelse(as.numeric(as.character(rote_cov$age)) >100, NA, as.numeric(as.character(rote_cov
```

```
## Warning in ifelse(as.numeric(as.character(rote_cov$age)) > 100, NA,
## as.numeric(as.character(rote_cov$age))): NAs introduced by coercion
```

```
## Warning in ifelse(as.numeric(as.character(rote_cov$age)) > 100, NA,
## as.numeric(as.character(rote_cov$age))): NAs introduced by coercion
```

```
rote_cov <- rote_cov %>% drop_na("age")

#fix gender
rote_cov$gender <- ifelse(rote_cov$gender == 'other', NA, rote_cov$gender)


#on test file, establish if testee answer matches actual answer. if it matches, set var to 1, else 0.
#notation will be o11, o12, etc (o is for outcome)
#x = baseline
#y = treatment
rote_test$o11x <- ifelse(rote_test$a11.x == rote_test$c11.x, 1, 0)
rote_test$o12x <- ifelse(rote_test$a12.x == rote_test$c12.x, 1, 0)
rote_test$o13x <- ifelse(rote_test$a13.x == rote_test$c13.x, 1, 0)
rote_test$o14x <- ifelse(rote_test$a14.x == rote_test$c14.x, 1, 0)
rote_test$o21x <- ifelse(rote_test$a21.x == rote_test$c21.x, 1, 0)
rote_test$o22x <- ifelse(rote_test$a22.x == rote_test$c22.x, 1, 0)
rote_test$o23x <- ifelse(rote_test$a23.x == rote_test$c23.x, 1, 0)
rote_test$o24x <- ifelse(rote_test$a24.x == rote_test$c24.x, 1, 0)

rote_test$o11y <- ifelse(rote_test$a11.y == rote_test$c11.y, 1, 0)
rote_test$o12y <- ifelse(rote_test$a12.y == rote_test$c12.y, 1, 0)
rote_test$o13y <- ifelse(rote_test$a13.y == rote_test$c13.y, 1, 0)
rote_test$o14y <- ifelse(rote_test$a14.y == rote_test$c14.y, 1, 0)
rote_test$o21y <- ifelse(rote_test$a21.y == rote_test$c21.y, 1, 0)
rote_test$o22y <- ifelse(rote_test$a22.y == rote_test$c22.y, 1, 0)
rote_test$o23y <- ifelse(rote_test$a23.y == rote_test$c23.y, 1, 0)
rote_test$o24y <- ifelse(rote_test$a24.y == rote_test$c24.y, 1, 0)
```

```
rote_test$score_pre <- rote_test$o11x + rote_test$o12x + rote_test$o13x + rote_test$o14x + rote_test$o2
rote_test$score_post <- rote_test$o11y + rote_test$o12y + rote_test$o13y + rote_test$o14y + rote_test$o

rote_test$score =  rote_test$score_post - rote_test$score_pre


#on covariates file, sum knowledge scores together
rote_cov$knowledge_cov_pre <- rote_cov$knowledge1 + rote_cov$knowledge2
rote_cov$knowledge_cov_post <- rote_cov$knowledge3 + rote_cov$knowledge4
rote_cov$prior_knowledge <- rote_cov$knowledge_cov_post + rote_cov$knowledge_cov_pre
```

```r
#convert necessary columns to boolean
rote_test$treat <- ifelse(rote_test$treat == "false", 0 , 1)
rote_cov$treat <- ifelse(rote_cov$treat == "false", 0 , 1)


#get distinct ids
rote_test <- rote_test[!duplicated(rote_test$session_id),]
rote_cov <- rote_cov[!duplicated(rote_cov$session_id),]

#inner join 2 datasets
#dataset <- merge(rote_test, rote_cov, by="session_id")

dataset <- inner_join(rote_test, rote_cov, by="session_id" )

#set date filter based on start time in test file
#dataset <- dataset %>% filter(session_start_time.x > as.POSIXct("2020-07-24 20:00:00", tz="UTC"))


#names(dataset)

dataset$treat <- dataset$treat.x
dataset$cluster <- dataset$cluster.x
dataset$gender <- as.factor(dataset$gender)
dataset <- dataset %>% filter %>% select("session_id", "score", "cluster", "gender", "age", "prior_know

dataset
```

```
##                          session_id score cluster gender age
## 1  8080E2E61BA04074F123155741AC29DC     1       2      2  45
## 2  BE7C20F73505C684DB5613B8702BD522     1       1      1  28
## 3  064D45ABDE08A0D54486ED13C1D68AF8     1       3      1  16
## 4  6B9C07A476F694235718EA94C6508183     0       4      2  16
## 5  92420B60CDC088006B4EBD6A987EBC96     4       4      1  12
## 6  8D2E34E1E483BFF773B251ECFDE62B5F     1       5      1  47
## 7  2325C32B409A77D94A567A30AC10CA5E     0       8      2  65
## 8  FAB03D46DC91AF5B99DE426D9AEA0EC0     1       9      2  15
## 9  1928EFED669DCDA43C1BD334080236A2     0       9      1  14
## 10 6745ECD9C1277AEA26BDA95E665CAF21     5       9      2  17
## 11 63E05BB4DD40D44D7DE5CFFA3153AED5     1      12      1  14
## 12 2ADDBD8896E5C047C98C535DD7CDF999    -2      13      1  17
## 13 4AF077B5D24EDDF4E4D49A12BB9C6177     3      14      1  41
## 14 DD33C6F35C1E827B3F9AD06788BCA9BB     3      15      1  63
## 15 3CFC9161282449FC7BD90DC05F901DAB    -1      17      1  76
## 16 49BB81A74413924A03557B74E29F2806     1      17      2  14
## 17 04C1F3054F87E75D4D363A9951111D4A    -2      18      1  25
## 18 3485ED0DBAF733278A3BFB83DA653F04     0      18      1  53
## 19 E2459677B12BAE2E408B762EF2B0CC8E     0      20      2  51
## 20 475194FA6304F78932F5284E1F640486    -1      20      1  66
## 21 38E6A2DFF102F9455E176F258167C4FE     4      22      1  57
## 22 E372B1841FD94EEBECA6507EBC14DDC6     2      22      1  47
## 23 4E2735ED90C9046575C677CA36330E05    -1      23      1  71
## 24 1476FF85858326B5E0EFA45AA6317ADE    -1      25      1  49
## 25 905FE7667B2194324F94799B10373867     1      26   <NA>  63
```

```
## 26 5A7B6CA83415D68180997B9530C3B322     3      26      1  54
## 27 F02C166B557695209A66291BD8C7C856     1      28      2  20
## 28 32CBD7078FE07E4B8047646C22B16649     1      28      1  78
## 29 A3FC53816D85398412B197BA7049F0D2     2      31      1  16
## 30 E81DC439415506E8E3549FA62B2F413B     0      31      2  13
## 31 824DA446738C86622832369AA638CA6D     3      32      1  57
## 32 1D9BE9C59ADC2D0A7EEFEFB101DF1529     2      33      1  68
## 33 786F66640765B5AF05D5E5720B5F2047     0      34      1  49
## 34 1B15E0E12856D0A39DBCB263813BBD12    -1      34      2  51
## 35 1F535047E91A4CB67D3340F9DD0CB9AE     3      34      1  52
## 36 BCE5E44F9C43A861C04B6588ECFC159F     2      35      1  46
## 37 5F65C924B87622582E813461BEA6CEF7     2      36      1  36
## 38 1351282BCD808558B1E6ED75055B156B     1      37      1  15
## 39 DA2E595A20C90A8106C55D061656FD58     0      38      1  59
## 40 25BE1983FAB0B2D27249D82D3BAA78D5     0      47      1  58
## 41 7EB4375173CE1D60754C268126BBC383     1      51      2  28
## 42 D3E2C29452079030CB1AE27528E9933B     0      54      1  26
## 43 1F2A7B12E38C83E42A9388667AFD0B09     1      57      1  56
## 44 BFD05A52CFE2448237DCD8D7D16D5AD3     1      59   <NA>  14
## 45 08F3AA64DF701A59ACA0EBC47BD43AC3     2      62      2  59
## 46 058A942D5B7BCA26FA21022250B4215D     3      63      2  15
## 47 0035AF289E4C2D1138C7604D6E6F38DD    -2      64      2  51
## 48 31BC9CCB20791DD7DC6B4D6949B1DD7A     2      65      2  14
## 49 D27D45D54928E2B91A731DFA05F87FA3    -2      69      2  49
## 50 28442876477B81BEE8215E779404D7F1     1      70      2  15
## 51 7C5A6167643790BEB8977A45C1A3E9DD     1      71      2  13
## 52 EE6B083787EB32506639F606EA93E485     2      71      1  58
## 53 4B4D37F587E02FEE9670368747ECA84E    -1      71      1  17
## 54 A8B09DB6C72A091211A69C6ACA8BF623     2      71      2  15
## 55 98E33A0EB89EA5474C49DC29A0FBD3C2     0      71      1  16
## 56 B5D24933BA864ECA7E7D063DC13A6E83     2      72      1  16
## 57 B28F476BBE6F22F97C0D6F17BEFF05AE     2      71      1  17
## 58 3B618E1B91A00E0C3C93D3D98BCEC573    -1      73      1  33
## 59 6A26FA4336DDCACF235AF69A4BA25EF2     0      74      2  13
## 60 DC61715E42F7E30B684FD46BACC2CE0B     2      75      1  14
## 61 F952A839AADD99D260358AE8867F3B49     4      76      2  13
## 62 80D039C450254B0E4F77E4F6C6E14063    -1      77      1  17
## 63 617BAE843ECEF4C89670EC204701CC2C     1      78      2  12
## 64 72280B554FB00F24A4CAF7559D745E0C    -1      79      1  15
## 65 470C0E8DEB9300796607BB6FBBCFABCE    -1      80      1  49
## 66 147E49AE5FA0B9DFEFA5E6D48687C82A     0      81      1  12
## 67 C855834D7D77F3FED4DE90146D1C7430     1      82      2  17
## 68 6F50CCE57A708A9B6B6569AF0B302867     1      84      1  57
## 69 1294B3E8C9BF3D9071052B15366F26C3     5      85      2  62
## 70 5C956E2AA2D0E4A2E467CDDAC0D1C8E9     1      88      2  15
## 71 41B5C35517F5E65B1A0D728832EB32C3     0      94      1  62
## 72 9C142756B1D62EFCD9D9ED185098E182     4      96      2  64
## 73 586C315A34BF6C8A7F4AA55CE4803082     3      99      2  39
## 74 023E66AA19197AC5F855675BAFDC6781    -1     100      2  17
## 75 62190AA63C5593940AE9EC43DBC06465     0     103      1  40
## 76 2D4940D3E41DF00C8FBBCBCE23B2AF1B     2     104      1  38
## 77 3FAA16956772693DEB7C33E7811B766A     2     104      1  15
## 78 0D8207EAD34EF0F6EEED6EF57481BC63     2     105      2  15
## 79 46CFBB8B8224FB327515F1484D70B6D5     2     106      2  14
```

```
## 80 684C06508AE4768A4DA9C94A2F6C5181    -1     107      1  14
## 81 48FCFA6DB841FA3CBFA0D0DB43BABFA7    -1     108      2  44
## 82 C6060740C5A96867E0D2E835991D998D    -1     112      1  45
## 83 821EAE2C2697881ACB782A2571B030BC     0     113      1  27
## 84 E1574C8B0BDBDDB3372D3CEE38AD0B88     3     114      2  43
## 85 ABD65579A9903B3991E1B56A96A4B920     2     119      1  46
## 86 30F28A2B7C942EA878B0232151CC466E     0     120      1  55
## 87 2B50C2EC262722F110F74002E4D85FA8     0     121      1  49
## 88 24B23FEA12428D6A08CA47A19B8125C5     1     123      2  50
## 89 29750BB5C4AD429B38B8FAADB9448D6F     3     124      2  14
## 90 5AC3FFA5552AC1E1C603516409291440    -2     125      1  54
## 91 736869DD88381F17F2153057D1D2433F     5     126      1  35
## 92 6A5A142F20A431DFF1B949CF2CB94D6D     0     128      1  41
## 93 68CB5B55A9EEE49C8A2D8D2CC4126854     4     129      2  77
## 94 6BAA9C401DACFB1E1C16F1B887FFF3E5     1     130      1  40
## 95 4526FE620B1603D7D3654EA783484D10     0       1      1  48
## 96 23B6FC073F89C8E953463361A3E343AF     0       2      2  25
## 97 4918A2AD591290EB48F0C87CB5FB9925     1       3      2  49
##    prior_knowledge treat reading practice
## 1                9     0       3        2
## 2                4     0       2        2
## 3                8     0       4        5
## 4                7     1       3        3
## 5                4     1       3        3
## 6                6     1       2        1
## 7                6     0       4        3
## 8                5     1       2        3
## 9                5     1       5        3
## 10               8     1       4        2
## 11              11     1       4        3
## 12              10     1       3        4
## 13               4     0       2        1
## 14               5     1       4        2
## 15               7     0       4        3
## 16               8     0       3        2
## 17               4     0       3        1
## 18               5     0       5        3
## 19               8     0       3        2
## 20               7     0       4        2
## 21               7     1       3        3
## 22               4     1       2        1
## 23               5     0       4        2
## 24               4     0       4        1
## 25               5     1       4        3
## 26               6     1       4        3
## 27               5     0       2        1
## 28              10     0       3        3
## 29               7     1       2        2
## 30               8     1       2        2
## 31               6     1       3        3
## 32               8     1       4        2
## 33               4     1       4        2
## 34              13     1       4        1
## 35               6     1       5        3
```

```
## 36    4    1    3    3
## 37   10    0    3    3
## 38    5    1    4    4
## 39    5    1    4    2
## 40    5    0    4    3
## 41   10    1    4    1
## 42   12    1    4    3
## 43    4    1    4    3
## 44    7    1    3    3
## 45    4    1    3    4
## 46    8    1    3    1
## 47    8    0    4    2
## 48    9    1    2    1
## 49    9    0    3    1
## 50    8    0    3    3
## 51   13    0    5    3
## 52    7    0    5    3
## 53    5    0    2    3
## 54    9    0    3    3
## 55    9    0    3    3
## 56    9    1    3    4
## 57    6    0    3    2
## 58    4    1    1    3
## 59    9    1    5    1
## 60    4    1    3    1
## 61    9    0    4    3
## 62    8    0    5    3
## 63    8    1    4    3
## 64    6    0    5    2
## 65    8    1    5    2
## 66   10    1    4    3
## 67    4    0    3    1
## 68    7    1    4    1
## 69    4    0    3    1
## 70   10    1    3    1
## 71    7    1    2    2
## 72    8    1    4    1
## 73   13    1    4    3
## 74   10    0    3    2
## 75    4    0    3    1
## 76    8    0    4    3
## 77    7    0    5    3
## 78    7    1    2    3
## 79    9    0    4    3
## 80    4    0    4    3
## 81    7    0    3    2
## 82   NA    0    5    2
## 83    4    0    2    1
## 84   12    0    4    3
## 85    7    0    2    1
## 86    4    0    4    2
## 87    5    0    3    1
## 88    7    0    3    2
## 89    7    1    5    2
```

```
## 90               7     1       3       3
## 91               7     0       3       3
## 92               4     0       3       2
## 93              10     0       4       1
## 94               7     1       3       3
## 95               4     0       3       1
## 96               5     0       2       1
## 97               7     1       4       2
```

```
#nrow(dataset)

#nrow(rote_test)
#nrow(rote_cov)
```

EDA, Regression modeling

```
#number of rows in the data

print(paste("After cleaning, the number of rows in our dataset is:", toString(nrow(dataset))))
```

[1] "After cleaning, the number of rows in our dataset is: 97"

```
print(paste("After cleaning, the number of observations in treatment is:", toString(sum(dataset$treat))))
```

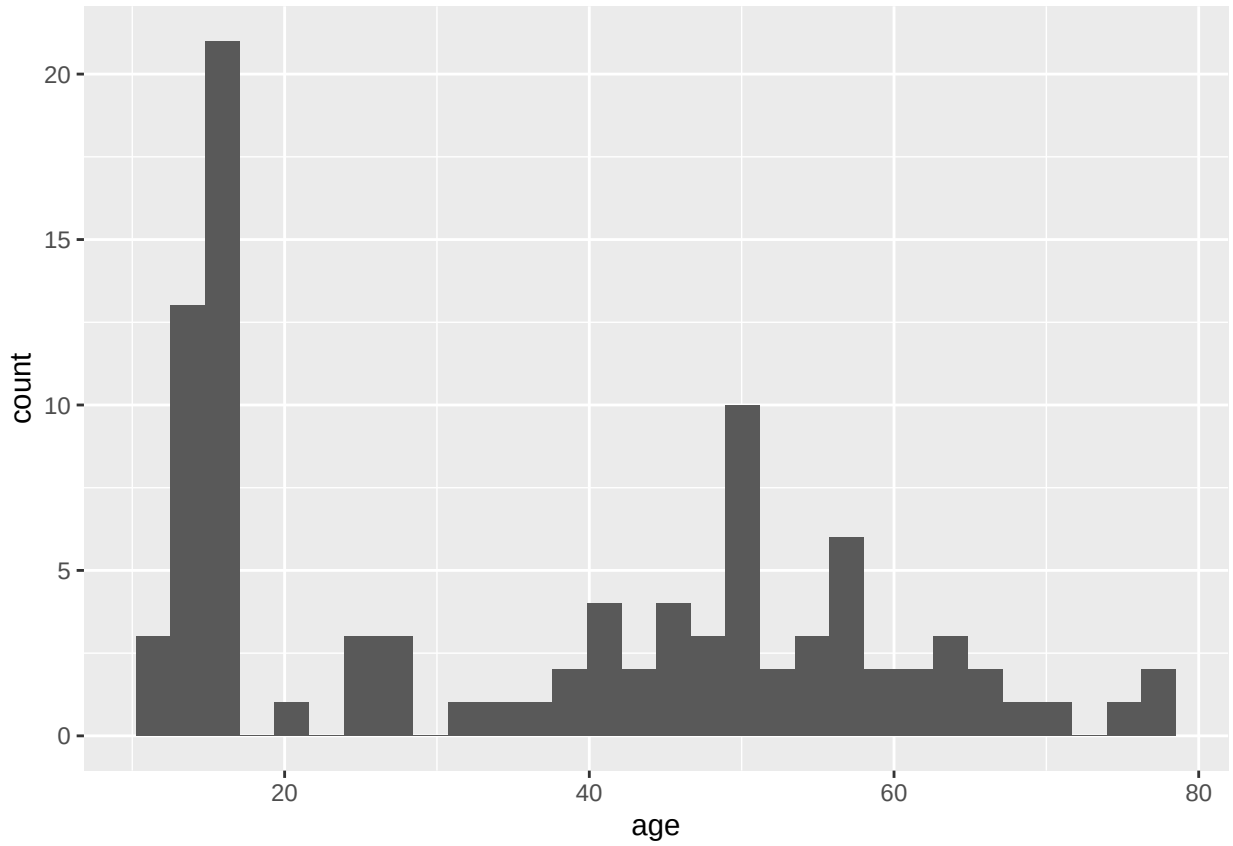
[1] "After cleaning, the number of observations in treatment is: 47"

```
print(paste("After cleaning, the number of observations in control is:", toString(nrow(dataset) - sum(da
```

[1] "After cleaning, the number of observations in control is: 50"

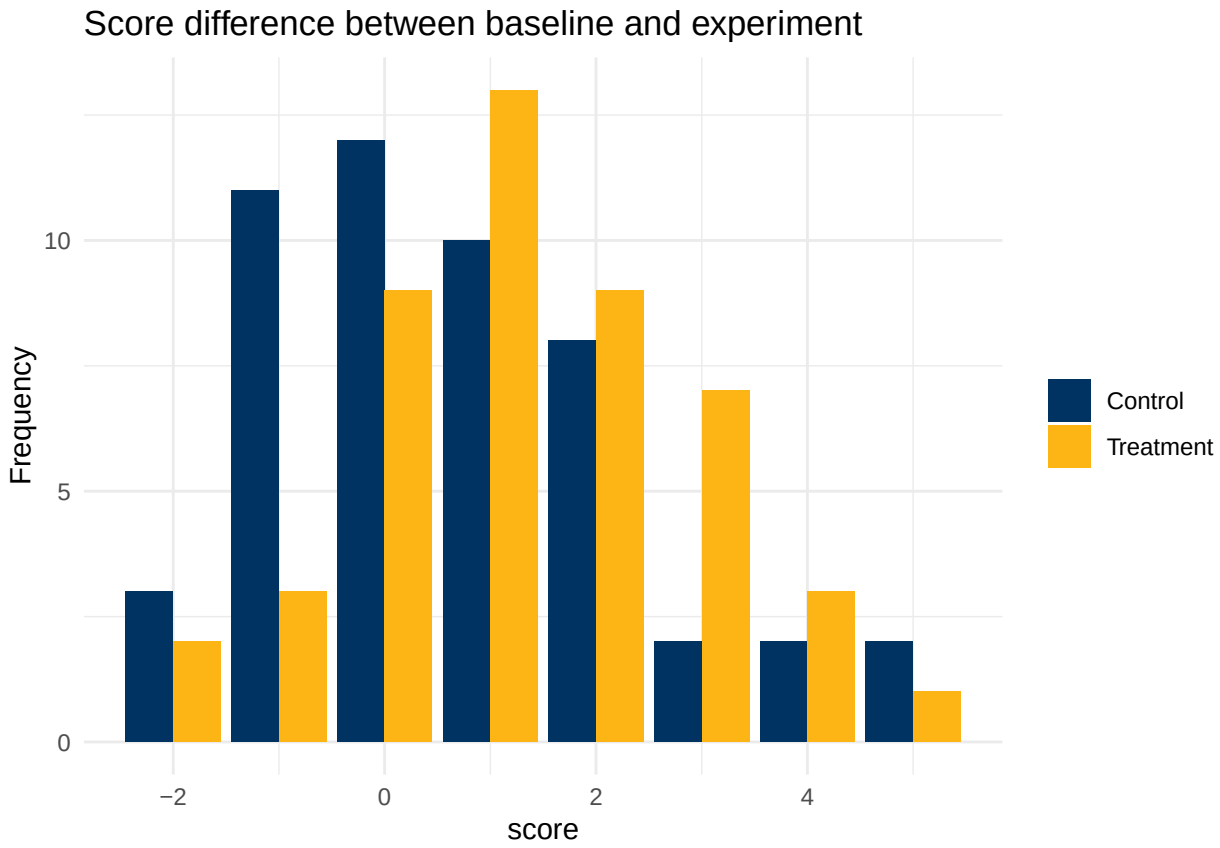
```
ggplot(dataset, aes(x=age)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

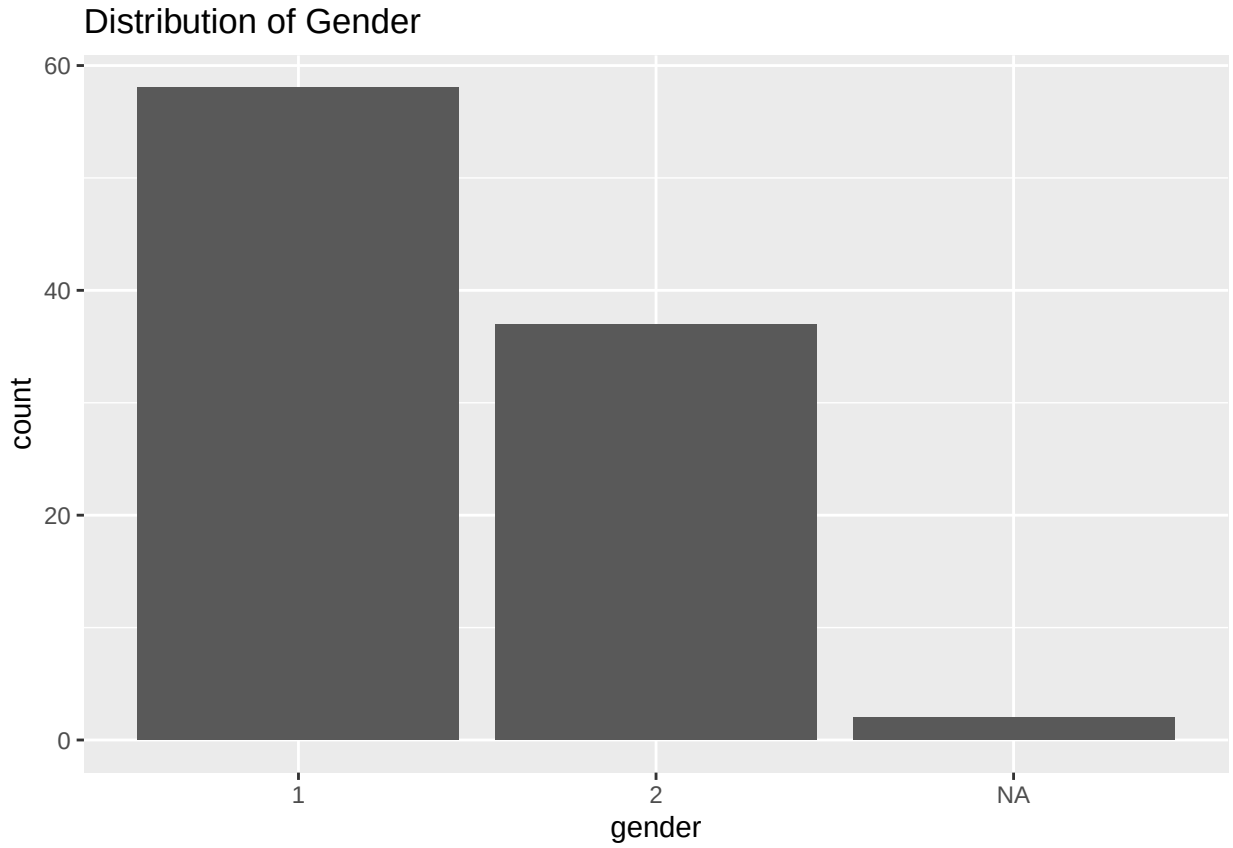



```
dataset_control <- dataset %>% filter(treat == 0)
dataset_treat <- dataset %>% filter(treat == 1)

dataset %>% ggplot(aes(x=score, fill=as.factor(treat)))+
  geom_bar(stat="count", position="dodge") +
  theme_minimal() +
  #scale_fill_brewer(palette="Dark2") +
  scale_fill_manual(values=c("#003262", "#FDB515"),
                    labels=c("Control", "Treatment"))+
 ylab("Frequency") +
  ggtitle("Score difference between baseline and experiment") +
  theme(legend.title=element_blank())
```




```
ggplot(dataset, aes(x=gender)) +
  geom_bar() +
  ggtitle("Distribution of Gender")
```




```
#names(dataset)

#dataset
unique(dataset$gender)
```

[1] 2 1 Levels: 1 2

```
regression1 <- lm(score ~ treat + age + prior_knowledge + reading + gender + practice,data=dataset)
regression2 <- lm(score ~ treat ,data=dataset)


clustered_errors1 <- vcovCL_1c <- vcovCL(regression1, cluster = dataset[ , 'cluster'])
clustered_errors2 <- vcovCL_1c <- vcovCL(regression2, cluster = dataset[ , 'cluster'])

stargazer(dataset,
                  header= F,
                  title = "Summary Table of Data",
                  type="latex") #flip type between text and latex
```

```
stargazer(regression1, regression2,
          header = F,
          type = "latex",
          omit.table.layout= "n",
          keep.stat = c("adj.rsq", "n", "f", "ser", "aic", "wald"),
          se = list(sqrt(diag(clustered_errors1)),sqrt(diag(clustered_errors2))),
          star.cutoffs = c(0.05, 0.01, 0.001),
          title="Regression Results with Clustered Standard Errors")
```

Table 1: Summary Table of Data

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
score	97	0.979	1.671	−2	0	2	5
cluster	97	58.763	39.901	1	22	88	130
age	97	36.309	19.968	12	15	52	78
prior_knowledge	96	6.948	2.373	4.000	5.000	8.000	13.000
treat	97	0.485	0.502	0	0	1	1
reading	97	3.423	0.934	1	3	4	5
practice	97	2.278	0.933	1	1	3	5

Table 2: Regression Results with Clustered Standard Errors

	Dependent variable:	
	score	
	(1)	(2)
treat	0.663*	0.659*
	(0.327)	(0.319)
age	0.006	
	(0.009)	
prior_knowledge	−0.027	
	(0.077)	
reading	−0.025	
	(0.184)	
gender2	0.712	
	(0.399)	
practice	0.153	
	(0.176)	
Constant	0.126	0.660**
	(0.731)	(0.245)
Observations	94	97
Adjusted R^2	0.007	0.029
Residual Std. Error	1.679 (df = 87)	1.646 (df = 95)
F Statistic	1.112 (df = 6; 87)	3.884 (df = 1; 95)