# ProjectDataset

AP VoteCast is a survey of the American electorate conducted in all 50 states by NORC at the University of Chicago for The Associated Press and Fox News. The survey is funded by AP. The survey of 138,929 registered voters was conducted October 29 to November 6, 2018, concluding as polls closed on Election Day. Interviews were conducted via phone and web, with 11,059 completing by phone and 127,870 completing by web.

```
data <- read.csv("dataset_CD.csv")
```

```
data$RACETH5 <- recode(data$RACETH5, "(1) White" = 1, "(2) African American or Black" = 2, "(3) Latino o
data$EDUC <- recode(data$EDUC, "(1) High school or less" = 1, "(2) Some college/assoc. degree" = 2, "(3)
data$INCOME <- recode(data$INCOME, "(1) Under $25,000" = 1, "(2) $25,000-$49,999" = 2, "(3) $50,000-$74
data$SEX <- recode(data$SEX, "(1) Men" = 1, "(2) Women" =2, "(99) DON'T KNOW/SKIPPED/REFUSED (VOL)" = 99
data$AGE <- recode(data$AGE65, "(1) 18-24" = 1, "(2) 25-29" = 2, "(3) 30-39" = 3, "(4) 40-49" = 4, "(5)
data$AGE65 <- NULL
data$PARTY <- recode(data$PARTYFULL, "(1) Democrat/Lean Dem" = 1, "(2) Republican/Lean Rep" =2, "(3) Ind
data$PARTYFULL <- NULL
data$IDEO <- recode(data$IDEO, "(1) Very liberal" = 1, "(2) Somewhat liberal" = 2, "(3) Moderate" = 3,
data$RELIG <- recode(data$RELIG4, "(1) Protestant/Other Christian" = 1, "(2) Catholic"= 2, "(3) Other" =
data$RELIG4 <- NULL
data$PLACE <- recode(data$SIZEPLACE, "(1) Urban" = 1, "(2) Suburban" = 2, "(3) Small town" = 3, "(4) Rur
data$SIZEPLACE <- NULL
data$TRACK <- recode(data$TRACK, "(1) Right direction" = 1, "(2) Wrong direction" = 2, "(99) DON'T KNOW/
data$APP <- recode(data$APP, "(1) Approve strongly" = 1, "(2) Approve somewhat" = 2, "(3) Disapprove som
data$STATE <- gsub("\\ .*","",data$STATE)
data$STATE <- substring(data$STATE, 2)
data$STATE <- gsub("\\).*","",data$STATE)
data$SU_ID <- NULL
data$RACE <- data$RACETH5
data$RACETH5 <- NULL
data$PLACE <- NULL
data$APP <- NULL
data$TRACK <- NULL
```

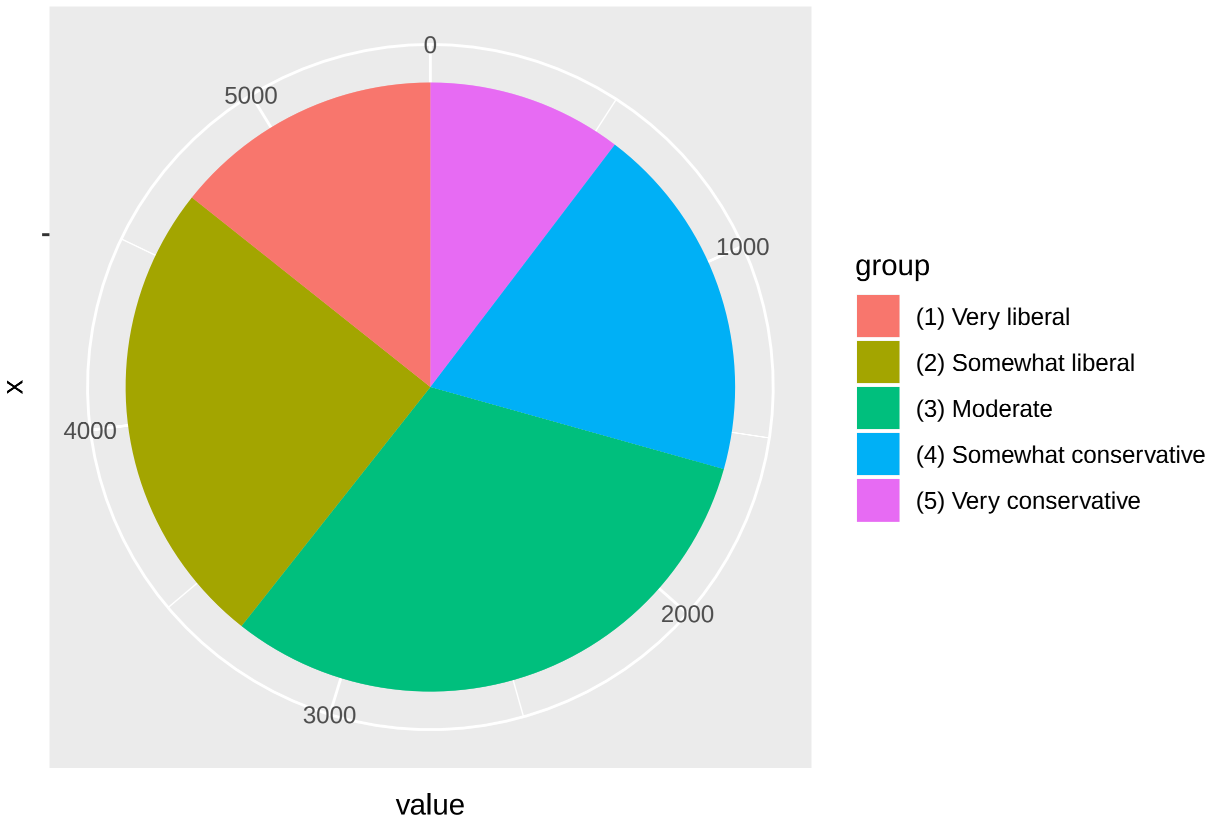
```
data[1:5,]
```

```
##   STATE SEX EDUC INCOME IDEO AGE PARTY RELIG RACE
## 1    26   2    2      2    2   6     3     1    1
## 2    16   1    4      4    4   4     2     1    1
## 3     9   1    1      5   99   5     1     3    2
## 4    26   1    4      3    1   6     1     2    1
## 5    30   1    4      5    4   5     2     1    3
```

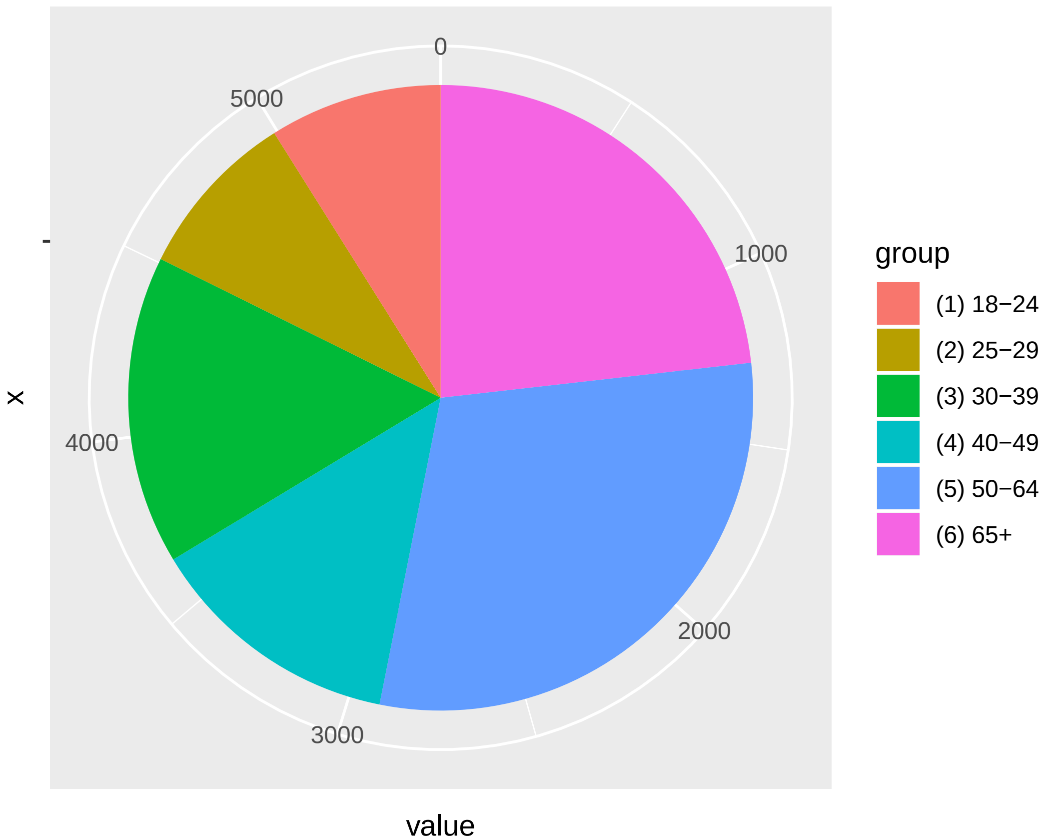
```
data <- filter_all(data, all_vars(. < 60))
```

```
IDEO_data <- c()
for (i in 1:5){
  IDEO_data[i] <- sum(data$IDEO==i)
}
lbls <- c("(1) Very liberal", "(2) Somewhat liberal", "(3) Moderate", "(4) Somewhat conservative", "(5)
df <- data.frame(
  group = lbls,
  value = IDEO_data
```

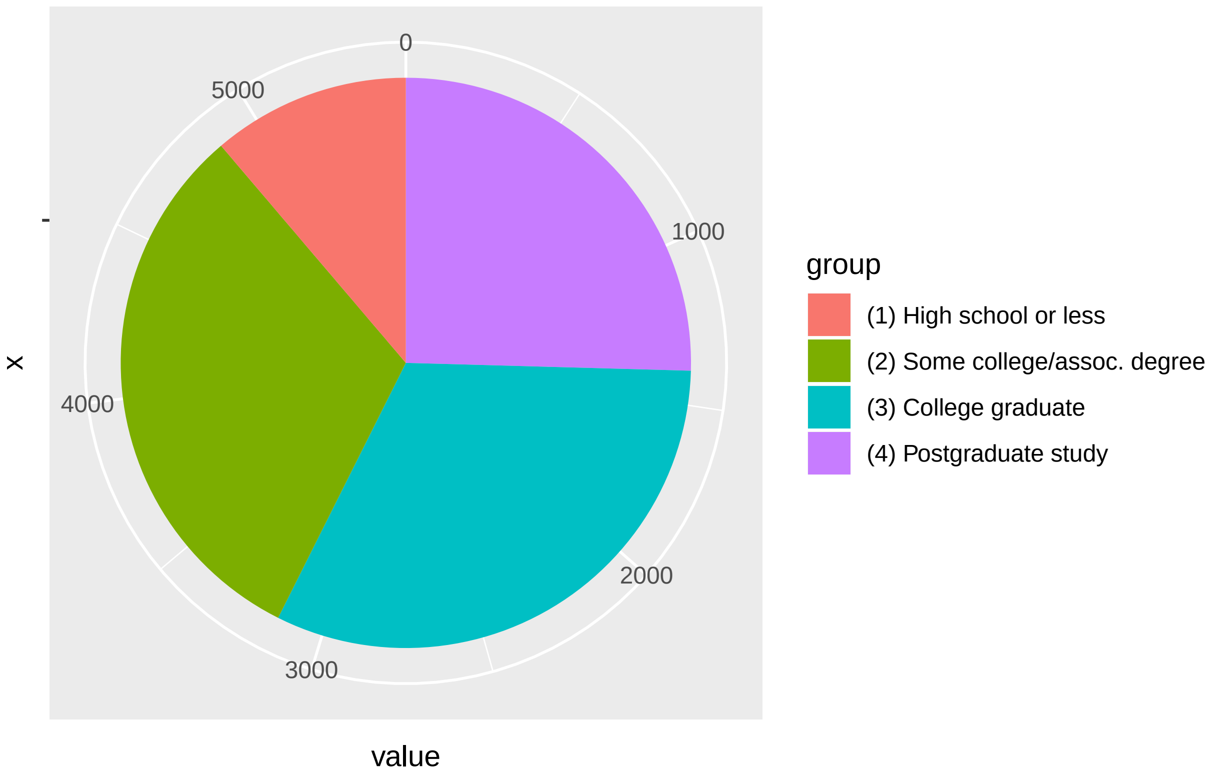
```
  )
ggplot(df, aes(x="", y=value, fill=group))+ geom_bar(width = 1, stat = "identity")+ coord_polar("y", sta
```



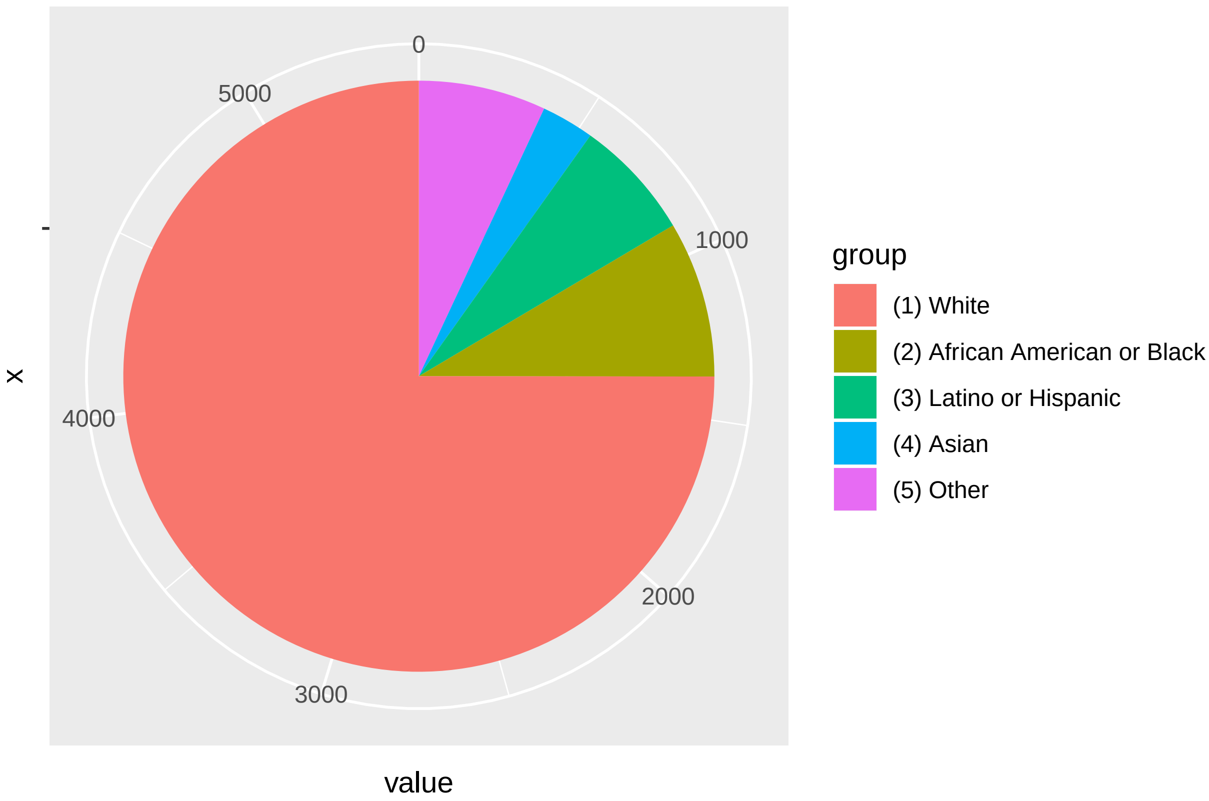

```
AGE_data <- c()
for (i in 1:6){
  AGE_data[i] <- sum(data$AGE==i)
}
lbls <- c("(1) 18-24",
"(2) 25-29",
"(3) 30-39",
"(4) 40-49",
"(5) 50-64",
"(6) 65+")
df <- data.frame(
  group = lbls,
  value = AGE_data
  )
ggplot(df, aes(x="", y=value, fill=group))+ geom_bar(width = 1, stat = "identity")+ coord_polar("y", sta
```



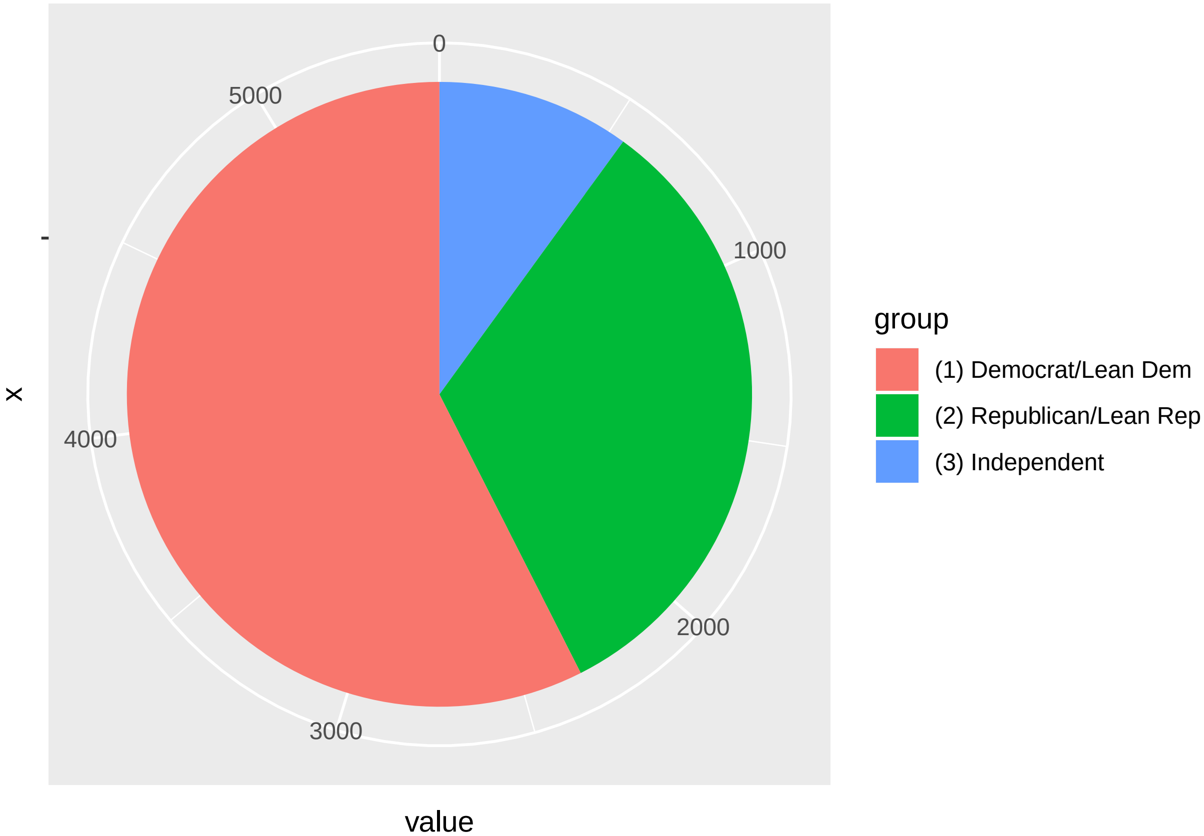

```
data <- data %>% filter(EDUC<88)
EDUC_data <- c()
for (i in 1:4){
  EDUC_data[i] <- sum(data$EDUC==i)
}
lbls <- c("(1) High school or less",
"(2) Some college/assoc. degree",
"(3) College graduate",
"(4) Postgraduate study")
df <- data.frame(
  group = lbls,
  value = EDUC_data
  )
ggplot(df, aes(x="", y=value, fill=group))+ geom_bar(width = 1, stat = "identity")+ coord_polar("y", sta
```



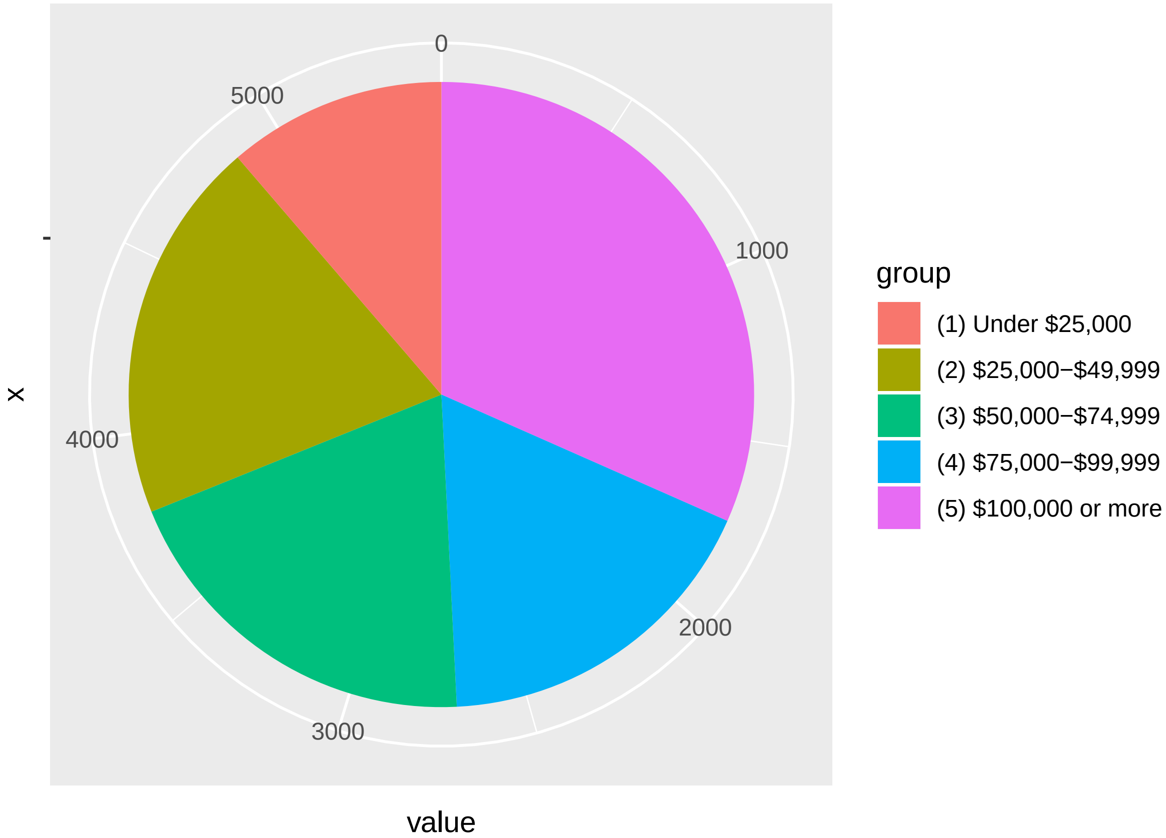

```
data <- data %>% filter(RACE<88)
RACE_data <- c()
for (i in 1:5){
  RACE_data[i] <- sum(data$RACE==i)
}
lbls <- c("(1) White",
"(2) African American or Black",
"(3) Latino or Hispanic",
"(4) Asian",
"(5) Other")
df <- data.frame(
  group = lbls,
  value = RACE_data
  )
ggplot(df, aes(x="", y=value, fill=group))+ geom_bar(width = 1, stat = "identity")+ coord_polar("y", sta
```



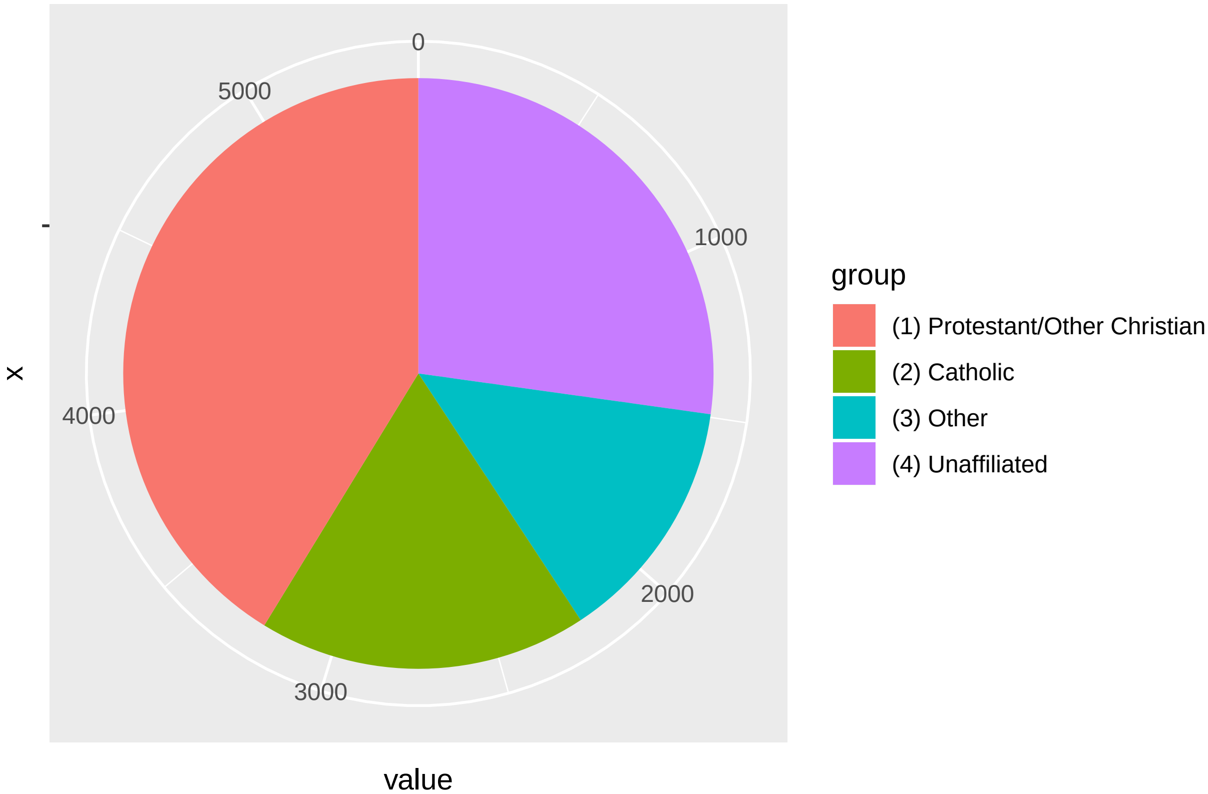

```
data <- data %>% filter(PARTY<88)
PAR_data <- c()
for (i in 1:3){
  PAR_data[i] <- sum(data$PARTY==i)
}
lbls <- c("(1) Democrat/Lean Dem", "(2) Republican/Lean Rep", "(3) Independent")
df <- data.frame(
  group = lbls,
  value = PAR_data
  )
ggplot(df, aes(x="", y=value, fill=group))+ geom_bar(width = 1, stat = "identity")+ coord_polar("y", sta
```




```
data <- data %>% filter(INCOME<88)
INC_data <- c()
for (i in 1:5){
  INC_data[i] <- sum(data$INCOME==i)
}
lbls <- c("(1) Under $25,000",
"(2) $25,000-$49,999",
"(3) $50,000-$74,999",
"(4) $75,000-$99,999",
"(5) $100,000 or more")
df <- data.frame(
  group = lbls,
  value = INC_data
  )
ggplot(df, aes(x="", y=value, fill=group))+ geom_bar(width = 1, stat = "identity")+ coord_polar("y", st
```




```
data <- data %>% filter(RELIG<88)
REL_data <- c()
for (i in 1:4){
  REL_data[i] <- sum(data$RELIG==i)
}
lbls <- c("(1) Protestant/Other Christian",
"(2) Catholic",
"(3) Other",
"(4) Unaffiliated")
df <- data.frame(
  group = lbls,
  value = REL_data
  )
ggplot(df, aes(x="", y=value, fill=group))+ geom_bar(width = 1, stat = "identity")+ coord_polar("y", sta
```




```r
table(data$IDEO, data$RACE)
```

```
##
##         1    2    3    4    5
##    1  608   53   59   23   41
##    2 1037  128   82   55   71
##    3 1168  211  148   50  138
##    4  833   41   53   26   88
##    5  463   35   21    4   44
```

```r
table(data$IDEO, data$AGE)
```

```
##
##       1   2   3   4   5   6
##    1 107  89 170  95 171 152
##    2 123 152 242 170 373 313
##    3 146 143 275 233 555 363
##    4  75  63 121 159 347 276
##    5  39  32  67  67 194 168
```

```r
table(data$IDEO, data$EDUC)
```

```
##
##       1   2   3   4
##    1  58 190 274 262
##    2 100 385 483 405
##    3 222 592 470 431
##    4 133 345 347 216
##    5 102 208 177  80
```

```r
table(data$IDEO, data$INCOME)
```

```
##
##       1   2   3   4   5
##   1  93 154 150 139 248
##   2 134 252 280 244 463
##   3 232 349 312 295 527
##   4  86 208 205 193 349
##   5  74 123 131  93 146
```

```r
table(data$IDEO, data$STATE)
```

```
##
##      10  15  16  19  20  22  23  24  25  26  28  30  31  33  34  35  38
##   1  26  32  20  46  26  24  58  27  15  34  10  35  32  18  43  31  47
##   2  42  66  52  65  60  66  79  55  28  80  14  56  49  36  62  48  67
##   3  68  68  76  63  92  69  81  63  67  82  21  65  72  48  70  61  75
##   4  41  52  50  34  41  38  38  49  50  52  21  34  34  40  40  38  26
##   5  28  27  29  15  12  17  27  31  36  25  14  20  10  21  14  25  18
##
##       4  42  43  45  47  48  49   5
##   1  30  27  19  27  52  46  24  35
##   2  38  42  62  58  82  85  34  47
##   3  74  64  88  73  64 108  52  51
##   4  50  51  47  50  44  51  47  23
##   5  25  31  25  17  25  34  22  19
```

The group with the most conservative ideology is exposed to the greatest disclosure risk, across differenet demographic variables.

```r
table(data$PARTY, data$IDEO)
```

```
##
##        1    2    3    4    5
##   1  759 1273  961  115   41
##   2    9   43  416  826  489
##   3   16   57  338  100   37
```

### Synthesis Model

First, I would use a binomial model to generate synthesized sex.

*Sex*

```r
modelString_sex <-"
model {
## sampling
for (i in 1:N){
y[i] ~ dbern(p)
}
## priors
p ~ dnorm(0.5, 1)
}
"
```

```r
data$SEX <- data$SEX - 1
y = as.vector(data$SEX)
N = length(y)
```

```r
the_data <- list("y" = y,
                 "N" = N)
```

```r
initsfunction <- function(chain){
  .RNG.seed <- c(1,2)[chain]
  .RNG.name <- c("base::Super-Duper",
                 "base::Wichmann-Hill")[chain]
  return(list(.RNG.seed=.RNG.seed,
              .RNG.name=.RNG.name))
}
```

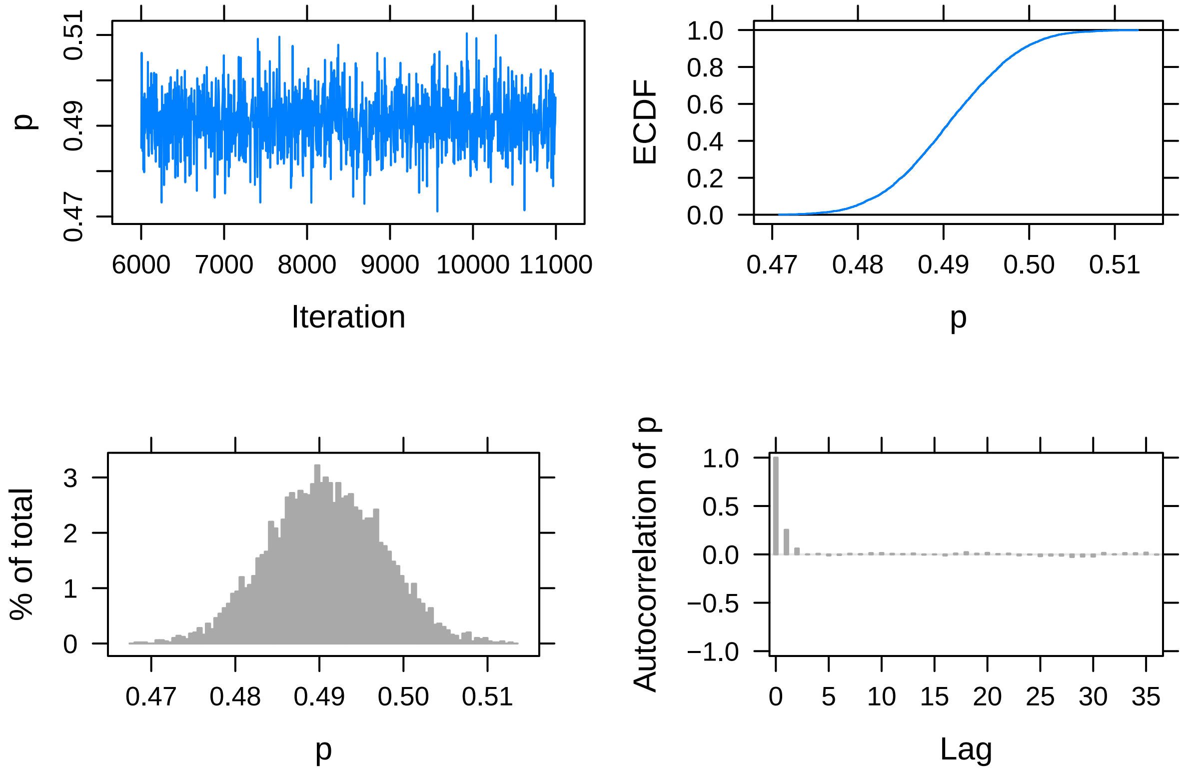
```r
posterior_sex <- run.jags(modelString_sex,
                      n.chains = 1,
                      data = the_data,
                      monitor = c("p"),
                      adapt = 1000,
                      burnin = 5000,
                      sample = 5000,
                      thin = 1,
                      inits = initsfunction)
```

```
## Calling the simulation...
## Welcome to JAGS 4.3.0 on Tue Mar  3 15:49:12 2020
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
## Loading module: bugs: ok
## . . Reading data file data.txt
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 5480
##    Unobserved stochastic nodes: 1
##    Total graph size: 5484
## . Reading parameter file inits1.txt
## . Initializing model
## . Adapting 1000
## -------------------------------------------------| 1000
## ++++++++++++++++++++++++++++++++++++++++++++++++++ 100%
## Adaptation successful
## . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
## . . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
## . . . . Updating 0
## . Deleting model
## .
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Calculating summary statistics...

## Warning: Convergence cannot be assessed with only 1 chain
```

```
## Finished running the simulation
```

```
plot(posterior_sex, vars="p")
```

```
## Generating plots...
```




```
post <- as.mcmc(posterior_sex)
```

```
synthesize <- function(index, n){
  synthetic_sex <- c()
  synthetic_sex <- rbinom(n, 1, post[index,"p"])
  data.frame(synthetic_sex)
}
n <- dim(data)[1]
syn_sex <- data.frame(synthesize(1, n))
data <- cbind(data, syn_sex)
colnames(data)[10] <- "SEX_s"
```

Then, I will use a Dirichlet model to generate age and race.

*Age*

```
modelString_age <-"
model {
## sampling
for (i in 1:N){
y[i] ~ dmulti(p,1)
p ~ ddirch(alpha[])
}
## priors
for (c in 1:C){
alpha[c] <- 1
```

```
}
}
"
```

```
data$AGE <- as.numeric(data$AGE)
y = as.vector(data$AGE)
N = length(y)
C = 5
```

```
the_data <- list("y" = y,
                 "N" = N,
                 "C" = C)
```

```
posterior_sex <- run.jags(modelString_age,
                      n.chains = 1,
                      data = the_data,
                      monitor = c("p"),
                      adapt = 1000,
                      burnin = 5000,
                      sample = 5000,
                      thin = 1,
                      inits = initsfunction)
```

```
post <- as.mcmc(posterior_age)
```

*Race*

```
modelString_race <-"
model {
## sampling
for (i in 1:N){
y[i,] ~ dmulti(p[i,1:C],1)
p[i,] ~ ddirch(alpha[])
}
## priors
for (c in 1:C){
alpha[c] <- 1
}
}
"
```

```
y = as.vector(data$RACE)
N = length(y)
C = 5
```

```
the_data <- list("y" = y,
                 "N" = N,
                 "C" = C)
```

```
posterior_sex <- run.jags(modelString_race,
                      n.chains = 1,
                      data = the_data,
                      monitor = c("p"),
                      adapt = 1000,
                      burnin = 5000,
                      sample = 5000,
                      thin = 1,
```

```
                        inits = initsfunction)
```

Moving on, I will use a multinomial logistic model to synthesize education and income.

```
data$Age1 = fastDummies::dummy_cols(data$AGE)[,names(fastDummies::dummy_cols(data$AGE)) == ".data_1"]
data$Age2 = fastDummies::dummy_cols(data$AGE)[,names(fastDummies::dummy_cols(data$AGE)) == ".data_2"]
data$Age3 = fastDummies::dummy_cols(data$AGE)[,names(fastDummies::dummy_cols(data$AGE)) == ".data_3"]
data$Age4 = fastDummies::dummy_cols(data$AGE)[,names(fastDummies::dummy_cols(data$AGE)) == ".data_4"]
data$Age5 = fastDummies::dummy_cols(data$AGE)[,names(fastDummies::dummy_cols(data$AGE)) == ".data_5"]
data$Age6 = fastDummies::dummy_cols(data$AGE)[,names(fastDummies::dummy_cols(data$AGE)) == ".data_6"]
data$Race1 = fastDummies::dummy_cols(data$RACE)[,names(fastDummies::dummy_cols(data$RACE)) == ".data_1"
data$Race2 = fastDummies::dummy_cols(data$RACE)[,names(fastDummies::dummy_cols(data$RACE)) == ".data_2"
data$Race3 = fastDummies::dummy_cols(data$RACE)[,names(fastDummies::dummy_cols(data$RACE)) == ".data_3"
data$Race4 = fastDummies::dummy_cols(data$RACE)[,names(fastDummies::dummy_cols(data$RACE)) == ".data_4"
data$Race5 = fastDummies::dummy_cols(data$RACE)[,names(fastDummies::dummy_cols(data$RACE)) == ".data_5"
```

*Education*

```
modelString_educ <-"
model {
## sampling
for (i in 1:N){
y[i] ~ dnorm(beta0 + beta1*Age1[i] + beta2*Age2[i] + beta3*Age3[i] + beta4*Age4[i] + beta5*Age5[i] + beta6*Race1[i] + beta7*Race
}
## priors
beta0 ~ dnorm(0, 0.00001)
beta1 ~ dnorm(0, 0.00001)
beta2 ~ dnorm(0, 0.00001)
beta3 ~ dnorm(0, 0.00001)
beta4 ~ dnorm(0, 0.00001)
beta5 ~ dnorm(0, 0.00001)
beta6 ~ dnorm(0, 0.00001)
beta7 ~ dnorm(0, 0.00001)
beta8 ~ dnorm(0, 0.00001)
beta9 ~ dnorm(0, 0.00001)
beta10 ~ dnorm(0, 0.00001)
invsigma2 ~ dgamma(a, b)
sigma <- sqrt(pow(invsigma2, -1))
}
"
```

```
y = as.vector(data$EDUC)
Age1 = as.vector(data$Age1)
Age2 = as.vector(data$Age2)
Age3 = as.vector(data$Age3)
Age4 = as.vector(data$Age4)
Age5 = as.vector(data$Age5)
Race1 = as.vector(data$Race1)
Race2 = as.vector(data$Race2)
Race3 = as.vector(data$Race3)
Race4 = as.vector(data$Race4)
Woman = as.vector(data$SEX_s)
N = length(y)
```

```
the_data <- list("y" = y,
                 "N" = N,
                 "Age1" = Age1,
                 "Age2" = Age2,
                 "Age3" = Age3,
                 "Age4" = Age4,
                 "Age5" = Age5,
                 "Race1" = Race1,
```

```r
                 "Race2" = Race2,
                 "Race3" = Race3,
                 "Race4" = Race4,
                 "Woman" = Woman,
                 "a" = 1, "b" = 1)
```

```r
posterior <- run.jags(modelString_educ,
                      n.chains = 1,
                      data = the_data,
                      monitor = c("beta0", "beta1", "beta2", "beta3", "beta4", "beta5", "beta6", "beta7
                      adapt = 1000,
                      burnin = 5000,
                      sample = 5000,
                      thin = 1,
                      inits = initsfunction)
```

```
## Calling the simulation...
## Welcome to JAGS 4.3.0 on Tue Mar  3 15:49:31 2020
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
## Loading module: bugs: ok
## . . Reading data file data.txt
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 5480
##    Unobserved stochastic nodes: 12
##    Total graph size: 60381
## . Reading parameter file inits1.txt
## . Initializing model
## . Adaptation skipped: model is not in adaptive mode.
## . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
## . . . . . . . . . . . . . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
## . . . . Updating 0
## . Deleting model
## Note: the model did not require adaptation
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Calculating summary statistics...

## Warning: Convergence cannot be assessed with only 1 chain

## Finished running the simulation
```

```r
post <- as.mcmc(posterior)
```

```r
synf_educ <- function(Age1, Age2, Age3, Age4, Age5, Race1, Race2, Race3, Race4, Woman, index, n){
  mean_Y <- post[index, "beta0"] +  Age1 * post[index, "beta1"] + Age2 * post[index, "beta2"] + Age3 * post[index, "beta3"] + Ag
  synthetic_educ <- rnorm(n, mean_Y, post[index, "sigma"])
  data.frame(synthetic_educ)
}
n <- dim(data)[1]
syn_educ <- data.frame(synf_educ(data$Age1, data$Age2, data$Age3, data$Age4, data$Age5, data$Race1, data$Race2, data$Race3, data
syn_educ <- round(syn_educ$synthetic_educ,0)
```

```r
data <- cbind(data, syn_educ)
colnames(data)[22] <- "EDUC_s"
data <- data %>% filter(EDUC_s >=1, EDUC_s <=4)
```

*Income*

```r
data$Educ1 = fastDummies::dummy_cols(data$EDUC_s)[,names(fastDummies::dummy_cols(data$EDUC_s)) == ".data
data$Educ2 = fastDummies::dummy_cols(data$EDUC_s)[,names(fastDummies::dummy_cols(data$EDUC_s)) == ".data
data$Educ3 = fastDummies::dummy_cols(data$EDUC_s)[,names(fastDummies::dummy_cols(data$EDUC_s)) == ".data
```

```r
modelString_inc <-"
model {
## sampling
for (i in 1:N){
y[i] ~ dnorm(beta0 + beta1*Age1[i] + beta2*Age2[i] + beta3*Age3[i] + beta4*Age4[i] + beta5*Age5[i] + beta6*Race1[i] + beta7*Race
}
## priors
beta0 ~ dnorm(0, 0.00001)
beta1 ~ dnorm(0, 0.00001)
beta2 ~ dnorm(0, 0.00001)
beta3 ~ dnorm(0, 0.00001)
beta4 ~ dnorm(0, 0.00001)
beta5 ~ dnorm(0, 0.00001)
beta6 ~ dnorm(0, 0.00001)
beta7 ~ dnorm(0, 0.00001)
beta8 ~ dnorm(0, 0.00001)
beta9 ~ dnorm(0, 0.00001)
beta10 ~ dnorm(0, 0.00001)
beta11 ~ dnorm(0, 0.00001)
beta12 ~ dnorm(0, 0.00001)
beta13 ~ dnorm(0, 0.00001)
invsigma2 ~ dgamma(a, b)
sigma <- sqrt(pow(invsigma2, -1))
}
"
```

```r
y = as.vector(data$INCOME)
Age1 = as.vector(data$Age1)
Age2 = as.vector(data$Age2)
Age3 = as.vector(data$Age3)
Age4 = as.vector(data$Age4)
Age5 = as.vector(data$Age5)
Race1 = as.vector(data$Race1)
Race2 = as.vector(data$Race2)
Race3 = as.vector(data$Race3)
Race4 = as.vector(data$Race4)
Woman = as.vector(data$SEX_s)
Educ1 = as.vector(data$Educ1)
Educ2 = as.vector(data$Educ2)
Educ3 = as.vector(data$Educ3)
N = length(y)
```

```r
the_data <- list("y" = y,
                 "N" = N,
                 "Age1" = Age1,
                 "Age2" = Age2,
                 "Age3" = Age3,
                 "Age4" = Age4,
                 "Age5" = Age5,
                 "Race1" = Race1,
                 "Race2" = Race2,
                 "Race3" = Race3,
                 "Race4" = Race4,
                 "Woman" = Woman,
```

```r
                  "Educ1" = Educ1,
                  "Educ2" = Educ2,
                  "Educ3" = Educ3,
                  "a" = 1, "b" = 1)
```

```r
posterior <- run.jags(modelString_inc,
                      n.chains = 1,
                      data = the_data,
                      monitor = c("beta0", "beta1", "beta2", "beta3", "beta4", "beta5", "beta6", "beta7
                      adapt = 1000,
                      burnin = 5000,
                      sample = 5000,
                      thin = 1,
                      inits = initsfunction)
```

```
## Calling the simulation...
## Welcome to JAGS 4.3.0 on Tue Mar  3 15:50:03 2020
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
## Loading module: bugs: ok
## . . Reading data file data.txt
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 5281
##    Unobserved stochastic nodes: 15
##    Total graph size: 74210
## . Reading parameter file inits1.txt
## . Initializing model
## . Adaptation skipped: model is not in adaptive mode.
## . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
## . . . . . . . . . . . . . . . . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
## . . . . Updating 0
## . Deleting model
## .
## Note: the model did not require adaptation
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Calculating summary statistics...
```

```
## Warning: Convergence cannot be assessed with only 1 chain
```

```
## Finished running the simulation
```

```r
post <- as.mcmc(posterior)
```

```r
synf_inc <- function(Age1, Age2, Age3, Age4, Age5, Race1, Race2, Race3, Race4, Woman,Educ1, Educ2, Educ3, index, n){
  mean_Y <- post[index, "beta0"] +  Age1 * post[index, "beta1"] + Age2 * post[index, "beta2"] + Age3 * post[index, "beta3"] + Ag
  synthetic_inc <- rnorm(n, mean_Y, post[index, "sigma"])
  data.frame(synthetic_inc)
}
n <- dim(data)[1]
syn_inc <- data.frame(synf_inc(data$Age1, data$Age2, data$Age3, data$Age4, data$Age5, data$Race1, data$Race2, data$Race3, data$R
syn_inc <- round(syn_inc$synthetic_inc,0)
```

```
data <- cbind(data, syn_inc)
colnames(data)[26] <- "INC_s"
data <- data %>% filter(INC_s >=1, INC_s <=5)
```

Finally! I will use age, race, sex, education, and income to synthesize ideology.

```
data$Inc1 = fastDummies::dummy_cols(data$INC_s)[,names(fastDummies::dummy_cols(data$INC_s)) == ".data_1
data$Inc2 = fastDummies::dummy_cols(data$INC_s)[,names(fastDummies::dummy_cols(data$INC_s)) == ".data_2
data$Inc3 = fastDummies::dummy_cols(data$INC_s)[,names(fastDummies::dummy_cols(data$INC_s)) == ".data_3
data$Inc4 = fastDummies::dummy_cols(data$INC_s)[,names(fastDummies::dummy_cols(data$INC_s)) == ".data_4
```

```
modelString_ideo <-"
model {
## sampling
for (i in 1:N){
y[i] ~ dnorm(beta0 + beta1*Age1[i] + beta2*Age2[i] + beta3*Age3[i] + beta4*Age4[i] + beta5*Age5[i] + beta6*Race1[i] + beta7*Race
}
## priors
beta0 ~ dnorm(0, 0.00001)
beta1 ~ dnorm(0, 0.00001)
beta2 ~ dnorm(0, 0.00001)
beta3 ~ dnorm(0, 0.00001)
beta4 ~ dnorm(0, 0.00001)
beta5 ~ dnorm(0, 0.00001)
beta6 ~ dnorm(0, 0.00001)
beta7 ~ dnorm(0, 0.00001)
beta8 ~ dnorm(0, 0.00001)
beta9 ~ dnorm(0, 0.00001)
beta10 ~ dnorm(0, 0.00001)
beta11 ~ dnorm(0, 0.00001)
beta12 ~ dnorm(0, 0.00001)
beta13 ~ dnorm(0, 0.00001)
beta14 ~ dnorm(0, 0.00001)
beta15 ~ dnorm(0, 0.00001)
beta16 ~ dnorm(0, 0.00001)
beta17 ~ dnorm(0, 0.00001)
invsigma2 ~ dgamma(a, b)
sigma <- sqrt(pow(invsigma2, -1))
}
"
```

```
y = as.vector(data$IDEO)
Age1 = as.vector(data$Age1)
Age2 = as.vector(data$Age2)
Age3 = as.vector(data$Age3)
Age4 = as.vector(data$Age4)
Age5 = as.vector(data$Age5)
Race1 = as.vector(data$Race1)
Race2 = as.vector(data$Race2)
Race3 = as.vector(data$Race3)
Race4 = as.vector(data$Race4)
Woman = as.vector(data$SEX_s)
Educ1 = as.vector(data$Educ1)
Educ2 = as.vector(data$Educ2)
Educ3 = as.vector(data$Educ3)
Inc1 = as.vector(data$Inc1)
Inc2 = as.vector(data$Inc2)
Inc3 = as.vector(data$Inc3)
Inc4 = as.vector(data$Inc4)
N = length(y)
```

```r
the_data <- list("y" = y,
                 "N" = N,
                 "Age1" = Age1,
                 "Age2" = Age2,
                 "Age3" = Age3,
                 "Age4" = Age4,
                 "Age5" = Age5,
                 "Race1" = Race1,
                 "Race2" = Race2,
                 "Race3" = Race3,
                 "Race4" = Race4,
                 "Woman" = Woman,
                 "Educ1" = Educ1,
                 "Educ2" = Educ2,
                 "Educ3" = Educ3,
                 "Inc1" = Inc1,
                 "Inc2" = Inc2,
                 "Inc3" = Inc3,
                 "Inc4" = Inc4,
                 "a" = 1, "b" = 1)
```

```r
posterior <- run.jags(modelString_ideo,
                      n.chains = 1,
                      data = the_data,
                      monitor = c("beta0", "beta1", "beta2", "beta3", "beta4", "beta5", "beta6", "beta7
                      adapt = 1000,
                      burnin = 5000,
                      sample = 5000,
                      thin = 1,
                      inits = initsfunction)
```

```
## Calling the simulation...
## Welcome to JAGS 4.3.0 on Tue Mar  3 15:50:41 2020
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
## Loading module: bugs: ok
## . . Reading data file data.txt
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 4865
##    Unobserved stochastic nodes: 19
##    Total graph size: 88427
## . Reading parameter file inits1.txt
## . Initializing model
## . Adaptation skipped: model is not in adaptive mode.
## . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
## . . . . . . . . . . . . . . . . . . . . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
## . . . . Updating 0
## . Deleting model
## Note: the model did not require adaptation
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
```

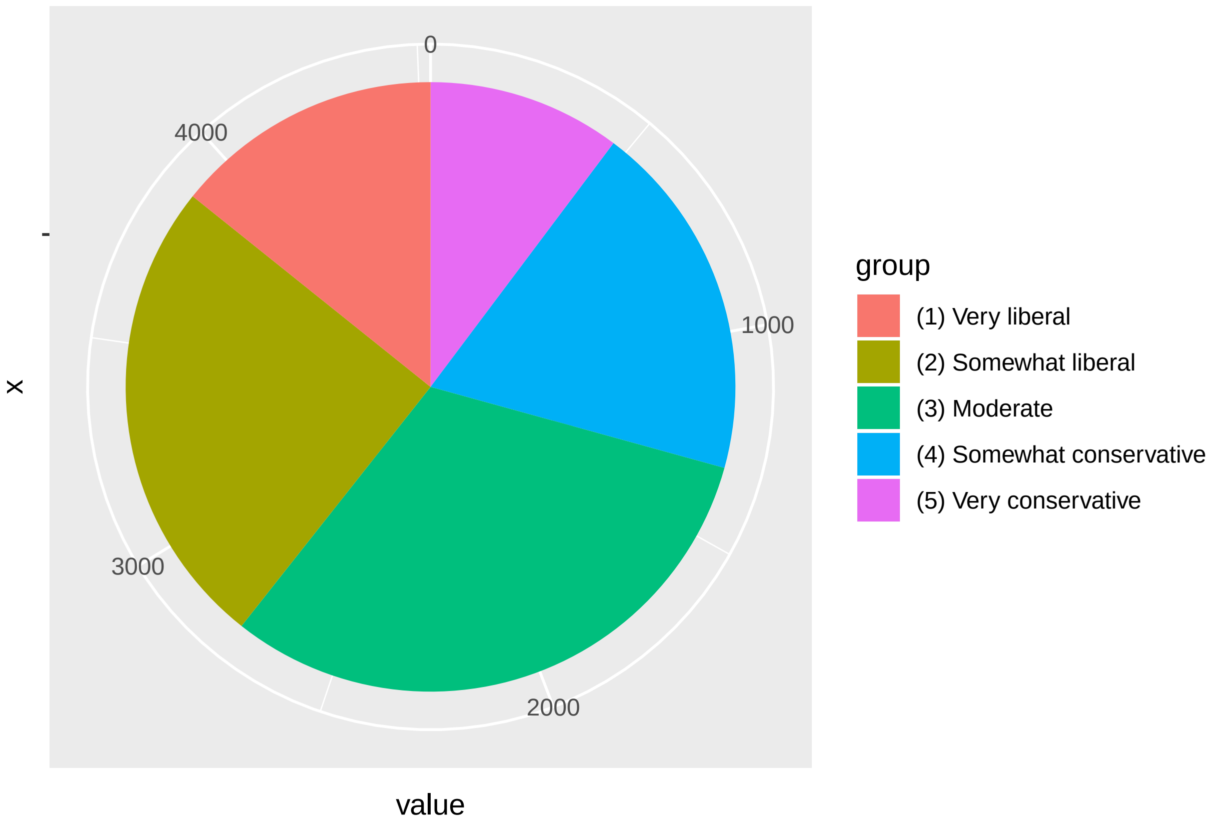
```
## Calculating summary statistics...

## Warning: Convergence cannot be assessed with only 1 chain

## Finished running the simulation
```

```
post <- as.mcmc(posterior)
```

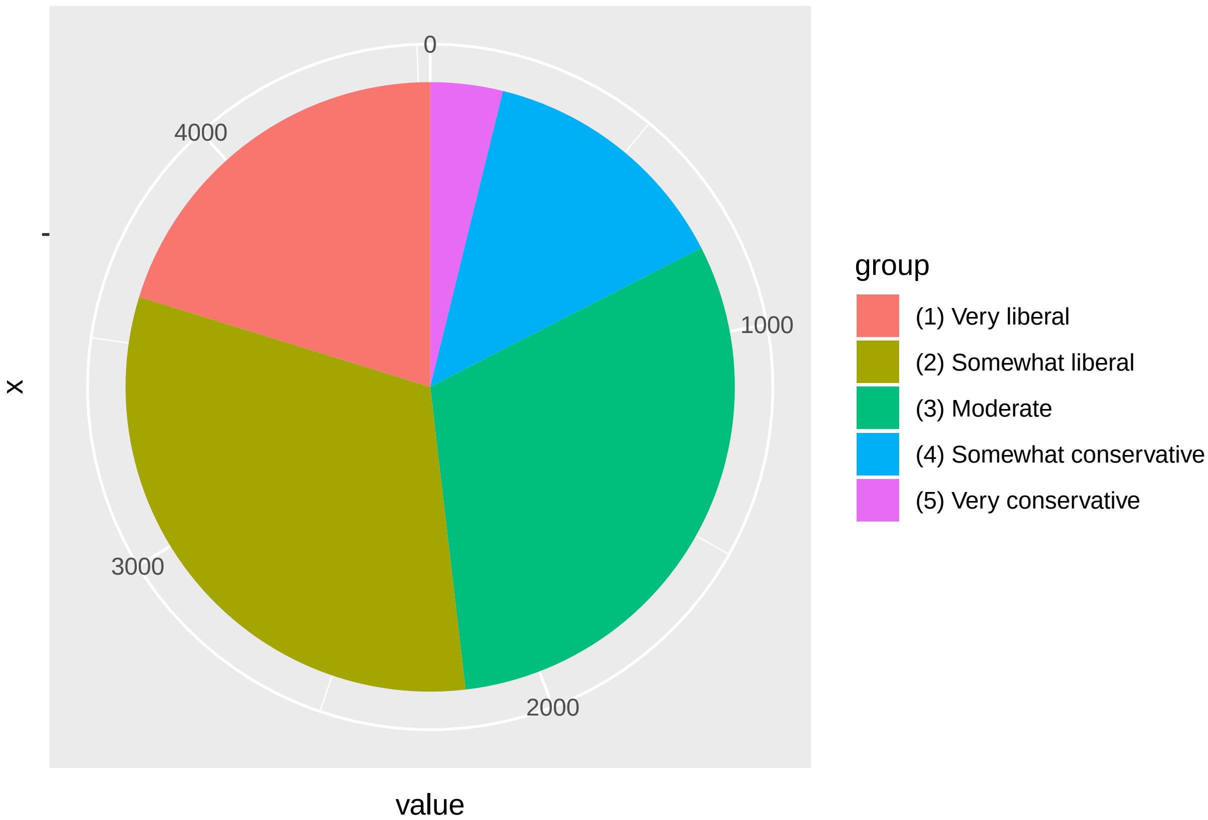
```
synf_ideo <- function(Age1, Age2, Age3, Age4, Age5, Race1, Race2, Race3, Race4, Woman,Educ1, Educ2, Educ3, Inc1, Inc2, Inc3, Inc
  mean_Y <- post[index, "beta0"] +  Age1 * post[index, "beta1"] + Age2 * post[index, "beta2"] + Age3 * post[index, "beta3"] + Ag
  synthetic_ideo <- rnorm(n, mean_Y, post[index, "sigma"])
  data.frame(synthetic_ideo)
}
n <- dim(data)[1]
syn_ideo <- data.frame(synf_ideo(data$Age1, data$Age2, data$Age3, data$Age4, data$Age5, data$Race1, data$Race2, data$Race3, data
syn_ideo <- round(syn_ideo$synthetic_ideo,0)
data <- cbind(data, syn_ideo)
colnames(data)[31] <- "IDEO_s"
data <- data %>% filter(IDEO_s >=1, IDEO_s <=5)
```

```
test <- cbind(data$STATE, data$SEX, data$EDUC, data$INCOME, data$IDEO, data$AGE, data$RACE, data$SEX_s,
test <- data.frame(test)
names(test) <- c("STATE", "SEX_o", "EDUC_o", "INCOME_o", "IDEO_o", "AGE_o", "RACE_o", "SEX_s", "EDUC_s"
```

```
IDEO_o <- c()
for (i in 1:5){
  IDEO_o[i] <- sum(test$IDEO_o==i)
}
lbls <- c("(1) Very liberal", "(2) Somewhat liberal", "(3) Moderate", "(4) Somewhat conservative", "(5)
df <- data.frame(
  group = lbls,
  value = IDEO_o
  )
ggplot(df, aes(x="", y=value, fill=group))+ geom_bar(width = 1, stat = "identity")+ coord_polar("y", sta
```




```
IDEO_s <- c()
for (i in 1:5){
  IDEO_s[i] <- sum(test$IDEO_s==i)
}
lbls <- c("(1) Very liberal", "(2) Somewhat liberal", "(3) Moderate", "(4) Somewhat conservative", "(5)
df <- data.frame(
  group = lbls,
  value = IDEO_s
  )
ggplot(df, aes(x="", y=value, fill=group))+ geom_bar(width = 1, stat = "identity")+ coord_polar("y", sta
```




```
mean(as.numeric(test$IDEO_o))
```

```
## [1] 2.85932
```

```
mean(as.numeric(test$IDEO_s))
```

```
## [1] 2.49227
```

```
median(as.numeric(test$IDEO_o))
```

```
## [1] 3
```

```
median(as.numeric(test$IDEO_s))
```

```
## [1] 2
```

We could see that the moderate groups are better preserved in our synthesized datasets, whereas the groups with more radical ideologies got shrunk a little bit.

```
model_o <- lm(as.numeric(IDEO_o) ~ SEX_o + EDUC_o + INCOME_o, data=test)
model_s <- lm(as.numeric(IDEO_s) ~ SEX_o + EDUC_o + INCOME_o, data=test)
```

```
summary(model_o)
```

```
##
## Call:
## lm(formula = as.numeric(IDEO_o) ~ SEX_o + EDUC_o + INCOME_o,
##     data = test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.45979 -0.86906  0.01495  0.86610  2.72531
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.26276    0.06737  48.429  < 2e-16 ***
## SEX_o1      -0.27551    0.03471  -7.937 2.59e-15 ***
## EDUC_o2     -0.25988    0.06114  -4.250 2.18e-05 ***
## EDUC_o3     -0.47473    0.06352  -7.473 9.34e-14 ***
## EDUC_o4     -0.71256    0.06664 -10.693  < 2e-16 ***
## INCOME_o2    0.13102    0.06519   2.010  0.04450 *
## INCOME_o3    0.14169    0.06661   2.127  0.03347 *
## INCOME_o4    0.17069    0.06880   2.481  0.01314 *
## INCOME_o5    0.19703    0.06516   3.024  0.00251 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.158 on 4519 degrees of freedom
## Multiple R-squared:  0.04445,    Adjusted R-squared:  0.04276
## F-statistic: 26.28 on 8 and 4519 DF,  p-value: < 2.2e-16
```

```r
summary(model_s)
```

```
##
## Call:
## lm(formula = as.numeric(IDEO_s) ~ SEX_o + EDUC_o + INCOME_o,
##     data = test)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6782 -0.5883 -0.3339  0.5736  2.6866
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.50698    0.06243  40.156   <2e-16 ***
## SEX_o1       0.08131    0.03217   2.528   0.0115 *
## EDUC_o2      0.08993    0.05666   1.587   0.1125
## EDUC_o3     -0.09253    0.05886  -1.572   0.1160
## EDUC_o4      0.01554    0.06175   0.252   0.8013
## INCOME_o2   -0.10103    0.06040  -1.673   0.0945 .
## INCOME_o3   -0.08055    0.06173  -1.305   0.1920
## INCOME_o4   -0.06122    0.06375  -0.960   0.3370
## INCOME_o5   -0.03359    0.06038  -0.556   0.5781
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.073 on 4519 degrees of freedom
## Multiple R-squared:  0.006735,   Adjusted R-squared:  0.004977
## F-statistic: 3.831 on 8 and 4519 DF,  p-value: 0.0001682
```

Judging from R-squared, we could see that my synthesized data has lost some utility in terms of the relationship between ideology and other variables in the dataset.