jh8186-solutionRMDHW4

Jerry Huang

2024-07-23

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Problem 1 - Political Efficacy in China and Mexico

1

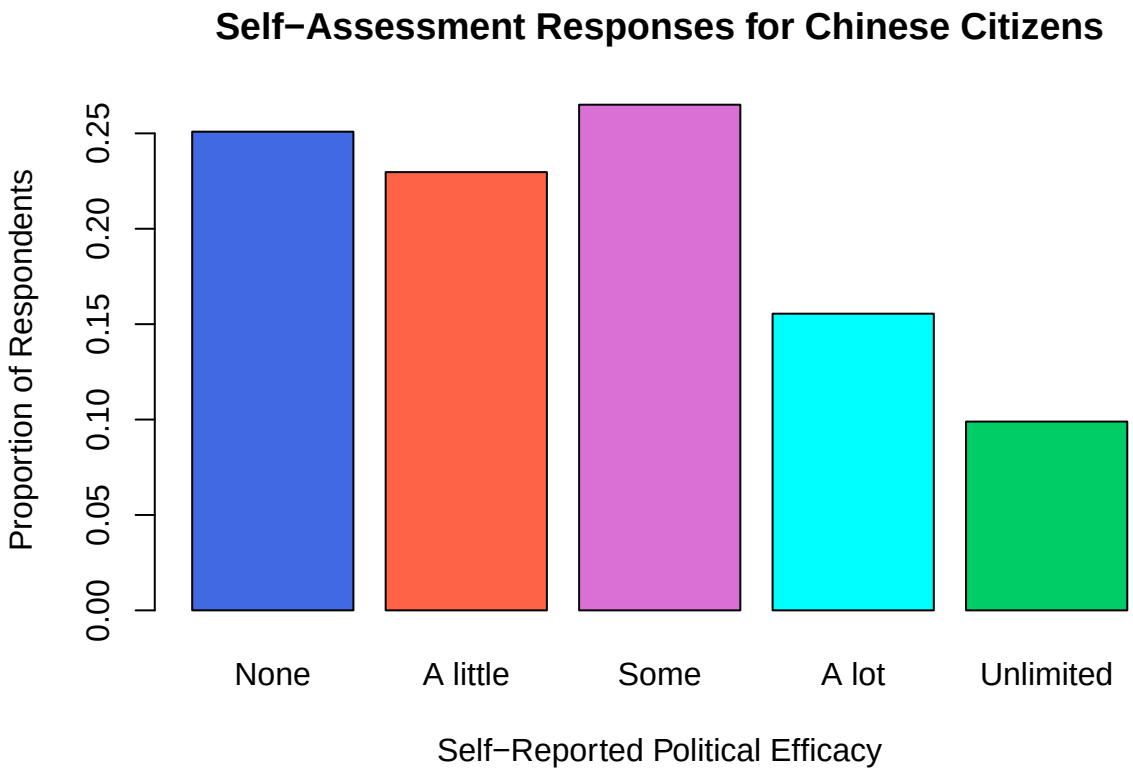
```
vignettes <- read.csv("vignettes.csv")

has_na <- anyNA(vignettes) # False, we have no Nah value

China <- vignettes %>%
  filter(china == 1) %>%
  mutate(
    total = self + alison + jane + moses
  )

Mexico <- vignettes %>%
  filter(china == 0) %>%
  mutate(
    total = self + alison + jane + moses
  )

China_mean <- mean(China$self)
Mexico_mean <- mean(Mexico$self)
China_mean
```

```
## [1] 2.621908
```

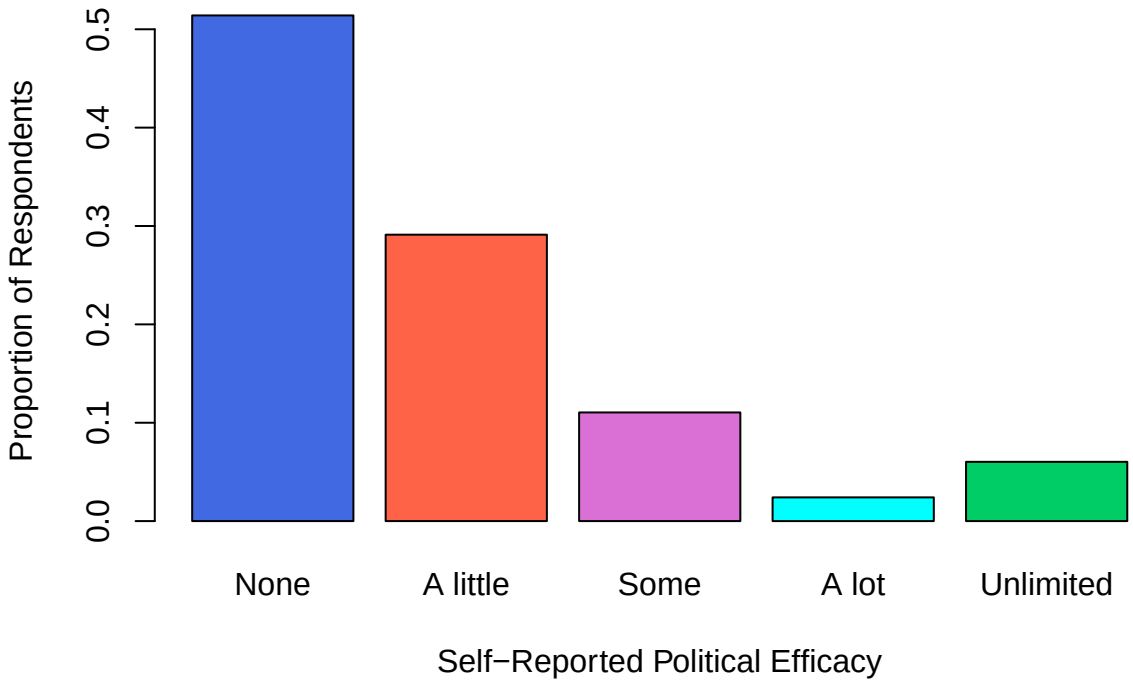
```
Mexico_mean
```

```
## [1] 1.825301
```

```r
# Colours
colours = c("royalblue", "tomato", "orchid", "cyan","springgreen3", "purple")

barplot(prop.table(table(China$self)),
names = c("None", "A little", "Some", "A lot", "Unlimited"),
xlab = "Self-Reported Political Efficacy",
ylab = "Proportion of Respondents", main = "Self-Assessment Responses for Chinese Citizens", col = colou
```



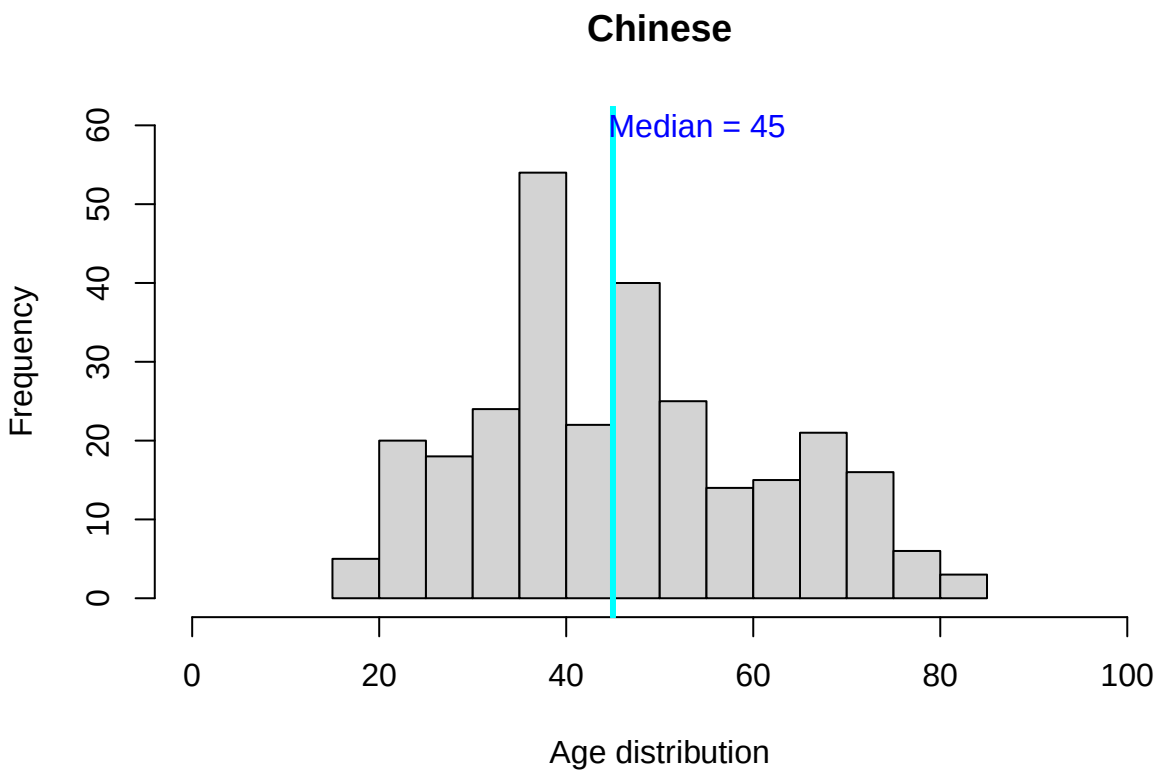

```r
barplot(prop.table(table(Mexico$self)),
names = c("None", "A little", "Some", "A lot", "Unlimited"),
xlab = "Self-Reported Political Efficacy",
ylab = "Proportion of Respondents", main = "Self-Assessment Responses for Mexican Citizens", col = colou
```



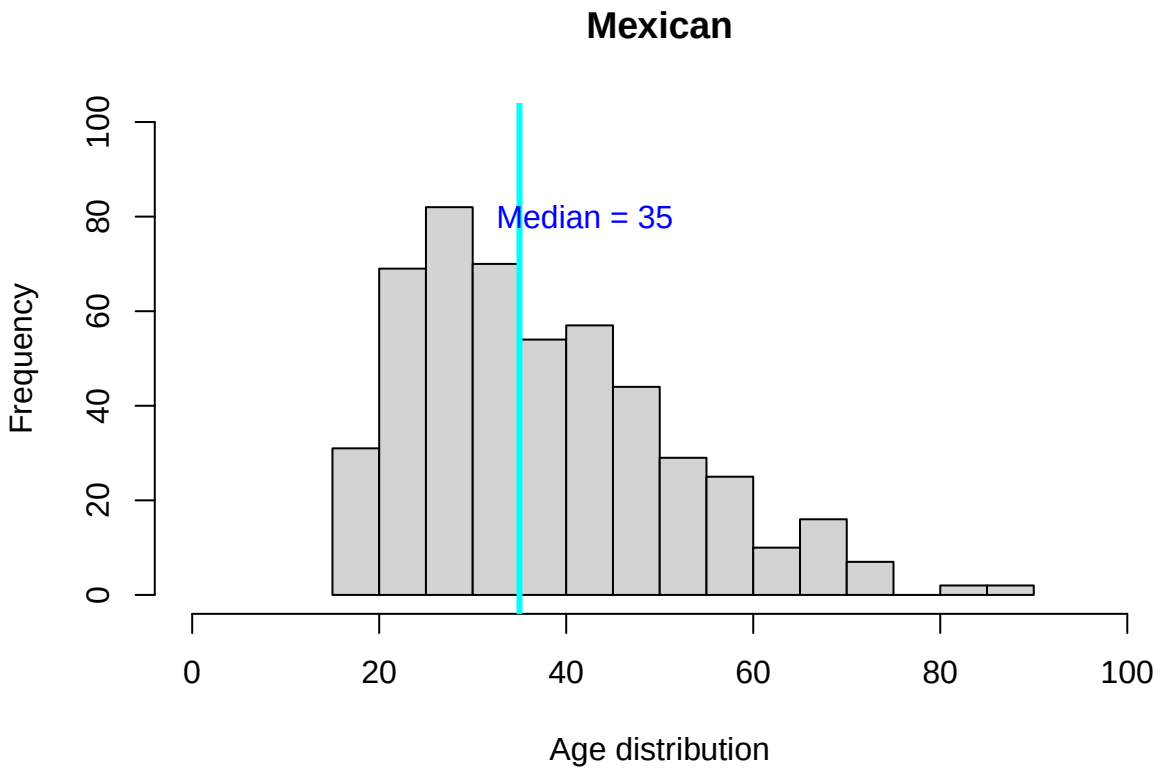
The mean response of self-assessment for China and Mexico are 2.621908 and 1.825301 respectively, which shows that China appears to have a higher political efficacy.
However, this is somewhat different with the historical fact of these two countries. On one hand, the Mexicans were able to vote out the ruling Institutional Revolutionary Party (PRI) who had governed the country for more than 80 years during the 2000 election. On the other hand, Chinese citizens have not been able to vote in a free and fair election. Therefore, this contrast might show that the political efficacy does not align with the political realities all the time and there are other factors influencing how citizens in these two countries perceive their political efficacy.

2

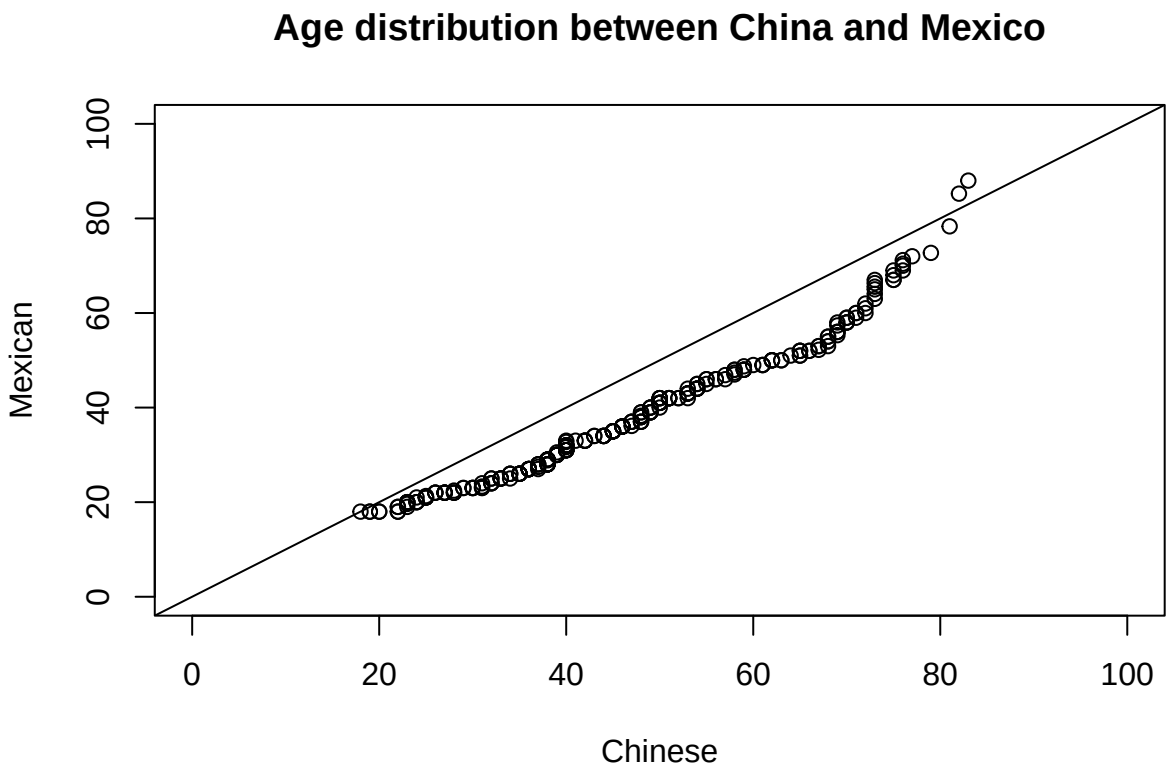
```
median_age_China <- median(China$age)
median_age_Mexico <- median(Mexico$age)

hist(China$age, freq = TRUE, main = "Chinese",
     xlim = c(0, 100), ylim = c(0, 60),
     xlab = "Age distribution")
abline(v = median_age_China, col = "cyan", lwd = 3)
text(x = median_age_China * 1.2, y = 60, paste("Median =", round(median_age_China, 2)), col = "blue")
```




```
hist(Mexico$age, freq = TRUE, main = "Mexican",
     xlim = c(0, 100), ylim = c(0, 100),
     xlab = "Age distribution")
abline(v = median_age_Mexico, col = "cyan", lwd = 3)
text(x = median_age_Mexico * 1.2, y = 80, paste("Median =", round(median_age_Mexico, 2)), col = "blue")
```



```r
# Quantile-quantile plot
qqplot(China$age, Mexico$age, xlab = "Chinese",
       ylab = "Mexican", xlim = c(0, 100), ylim = c(0, 100),
       main = "Age distribution between China and Mexico")
abline(0, 1) # 45 degree line
```



According to the age distribution of two countries, we can see that Mexico appears to have more young subjects with a median of 35 in the survey compared to the age distribution of Chinese respondents which appears more spread out with a higher number of respondents aged between 30 and 50.
The Q-Q plot compares the age distribution of Chinese respondents (x-axis) to Mexican respondents (y-axis). Points lying along the diagonal line indicate that the age distributions of the two countries are similar at those quantiles. Most points lie near the diagonal line for ages below 60, indicating similar distributions for these ages. For ages above 60, the points deviate from the line, suggesting differences in the upper age distributions.

3.

```r
# Calculate the proportion of samples who have lower Self scores than Moses
China$self_Moses <- China$self < China$moses
Mexico$self_Moses <- Mexico$self < Mexico$moses

prop_lower_China <- sum(China$self_Moses == TRUE)/nrow(China)
prop_lower_Mexico <- sum(Mexico$self_Moses == TRUE)/nrow(Mexico)

# Print the proportions
print(prop_lower_China)
```

```
## [1] 0.5618375
```

```r
print(prop_lower_Mexico)
```

```
## [1] 0.248996
```

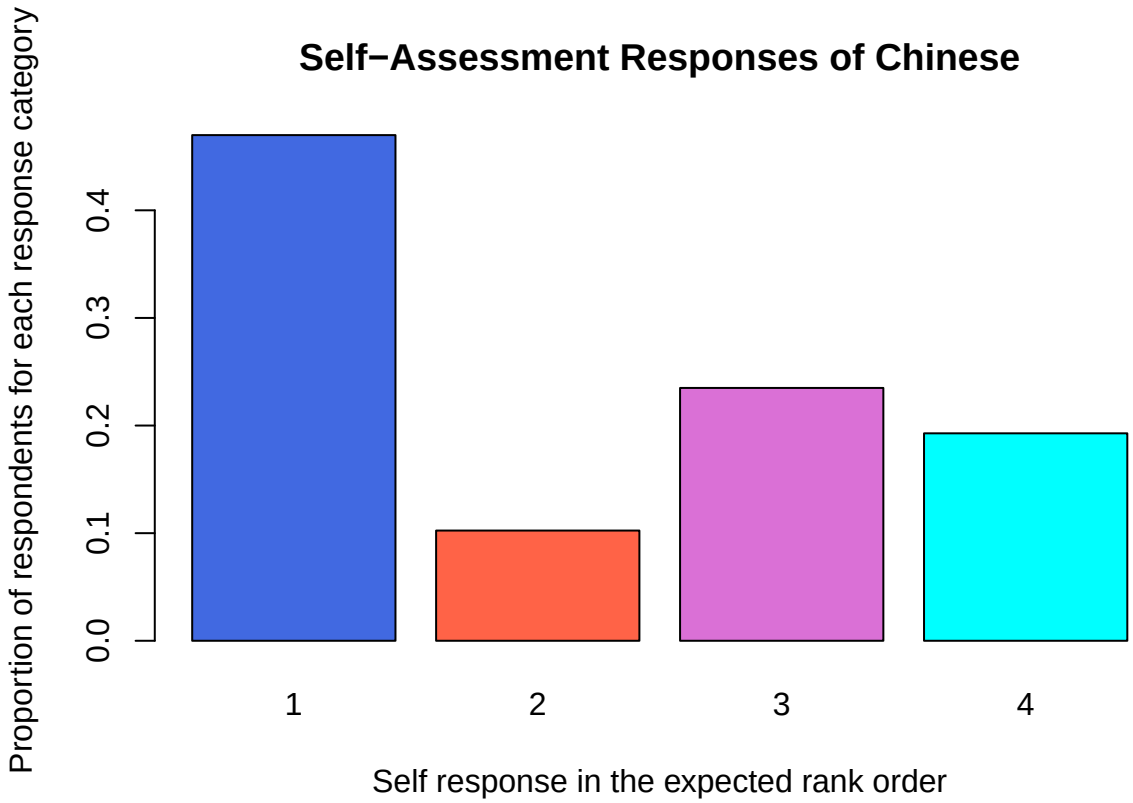
From the proportion result, 56% of respondents in China rank themselves as having less say than Moses in influencing government decisions, while only about 25% of respondents in Mexico feel the same way.

If we compare this result to the self-assessment response alone, Chinese citizens actually feel they will have more to say to get the government to address issues that interest them according to the graph in question 1. However, in this case where we filter the data about subjects who have lower self score than Moses, we can see more Chinese citizens feel they can say more about getting the government to address the issue than Moses' case.

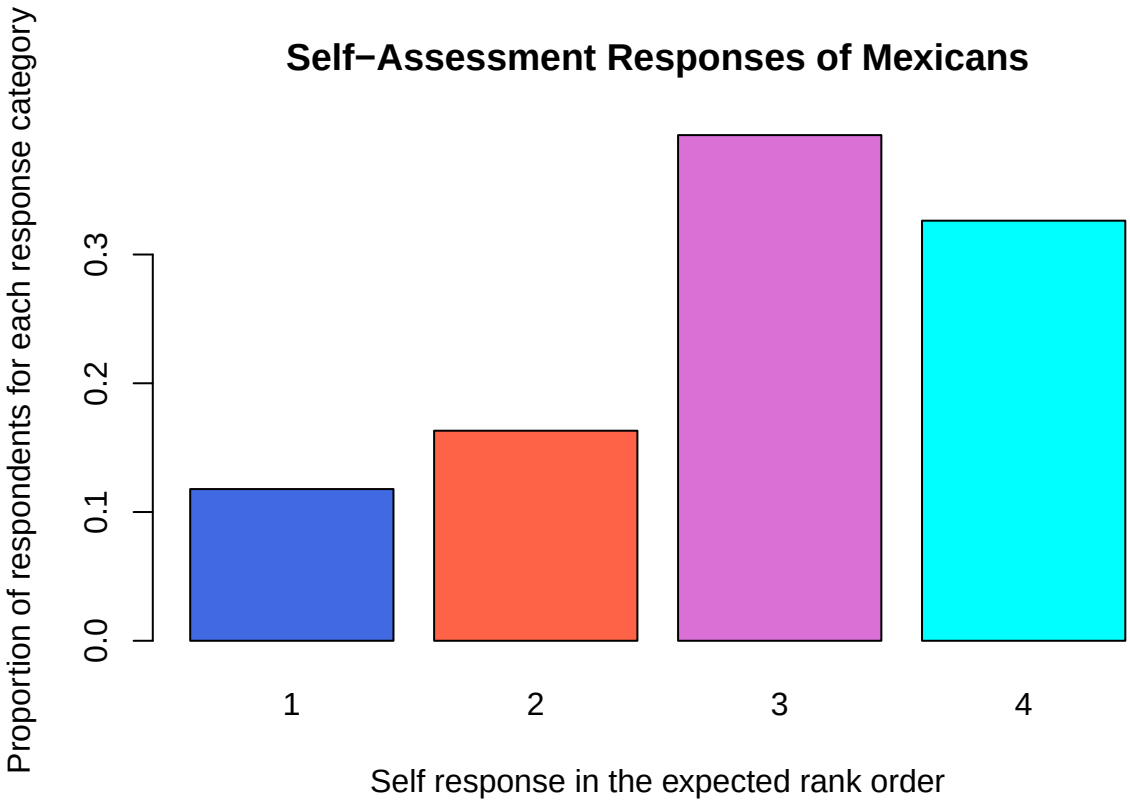
The discrepancy between the self-assessment and the vignette-based ranking might indicate that Chinese respondents interpret their political efficacy differently when comparing their situation directly to a vignette. The self-assessment may be influenced by general optimism or social desirability bias, whereas the vignette comparison provides a more grounded assessment of their perceived influence.

4.

```r
# Samples who ranked the three vignettes in the following order
expected_order_china <- China %>%
  filter(alison >= jane & jane >= moses) %>%
  mutate(self_rank = case_when(
    self < moses ~ 1, # 1 less than moses
    self < jane ~ 2, # 2 equal to moses but less than jane
    self < alison ~ 3, # 3 same as jane but less than alison
    TRUE ~ 4
  ))

expected_order_mexico <- Mexico %>%
  filter(alison >= jane & jane >= moses) %>%
  mutate(self_rank = case_when(
    self < moses ~ 1,
    self < jane ~ 2,
    self < alison ~ 3,
    TRUE ~ 4
  ))

# Create bar plots for the new variable
barplot(prop.table(table(expected_order_china$self_rank)),
        names = c("1", "2", "3", "4"),
        xlab = "Self response in the expected rank order",
        ylab = "Proportion of respondents for each response category",
        main = "Self-Assessment Responses of Chinese",
        col = colours)
```




```
barplot(prop.table(table(expected_order_mexico$self_rank)),
        names = c("1", "2", "3", "4"),
        xlab = "Self response in the expected rank order",
        ylab = "Proportion of respondents for each response category",
        main = "Self-Assessment Responses of Mexicans",
        col = colours)
```




```
# Compute the mean value of the new variable self_rank
mean_self_rank_China <- mean(expected_order_china$self_rank)
mean_self_rank_Mexico <- mean(expected_order_mexico$self_rank)

print(mean_self_rank_China)
```

```
## [1] 2.150602
```

```
print(mean_self_rank_Mexico)
```

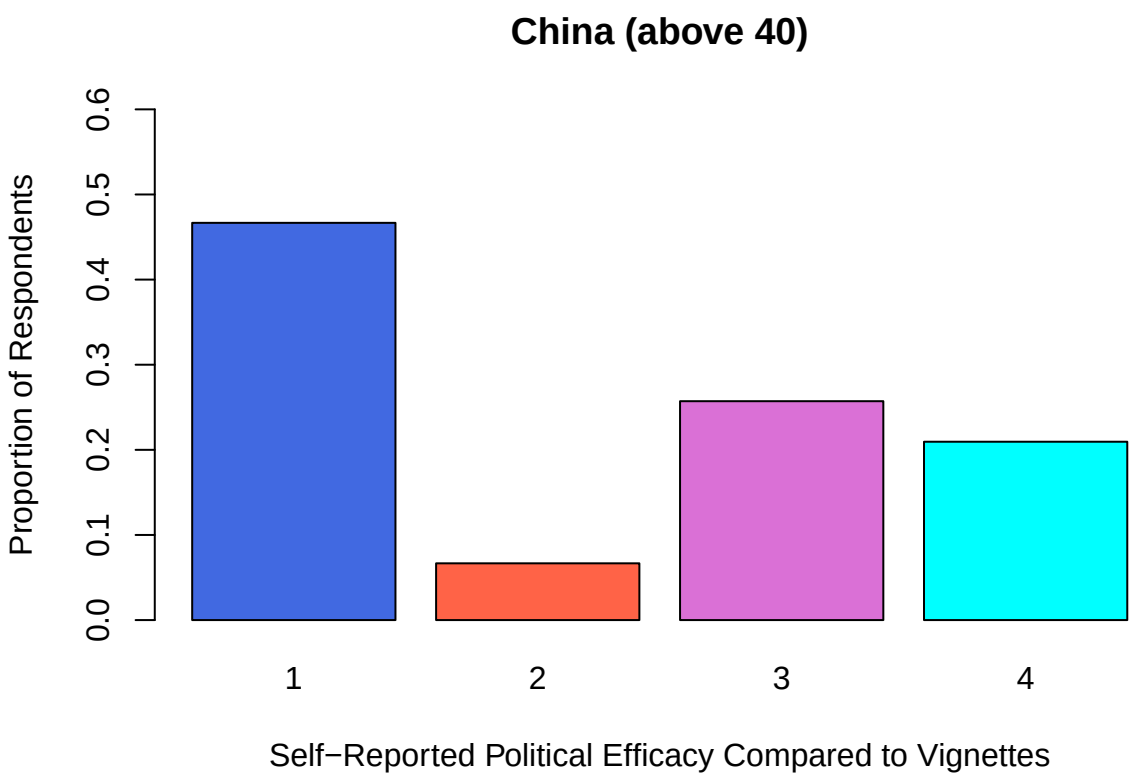
```
## [1] 2.927492
```

The mean value of self_rank this time is quite different from what we have observed in question 1. In question 1, we have the mean of self-assessment response equal to 2.621908 and 1.825301 for China and Mexico respectively. Here, we have 2.150602 and 2.927492 for China and Mexico respectively. Mexican citizens report a higher mean of political efficacy score according to the expected order, with the majority citizens report their self scores at level 3 (self score is ranked the same as Jane or between Jane and Alison). On the other hand, the majority of Chinese citizens report their self scores at level 1 (respondents rank themselves less than Moses).
The differences in means and distributions indicate that Mexican respondents generally feel they have more political efficacy compared to Chinese respondents. This pattern aligns with the historical context provided: Mexican citizens were able to vote out a long-standing party in 2000, indicating a sense of political agency, while Chinese citizens have not had the same opportunities for political expression.

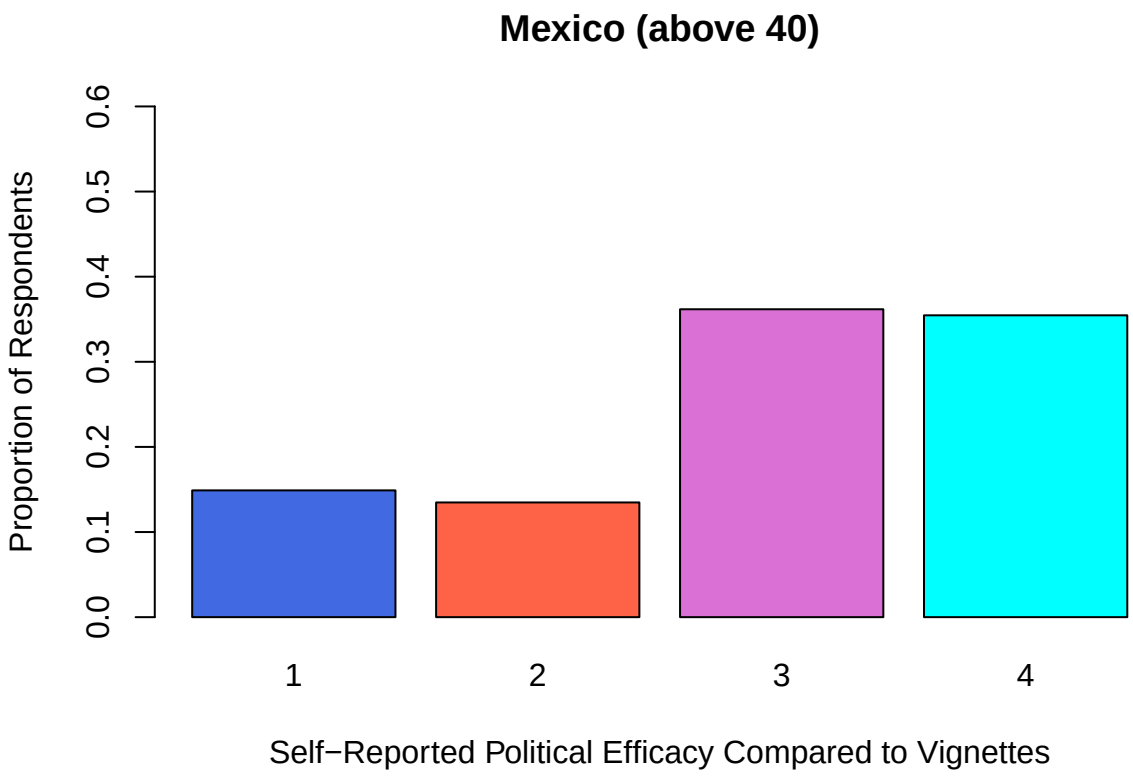
5.

```
# Filter data for age groups
older_respondents_China <- expected_order_china %>% filter(age >= 40)
older_respondents_Mexico <- expected_order_mexico %>% filter(age >= 40)
younger_respondents_China <- expected_order_china %>% filter(age < 40)
```

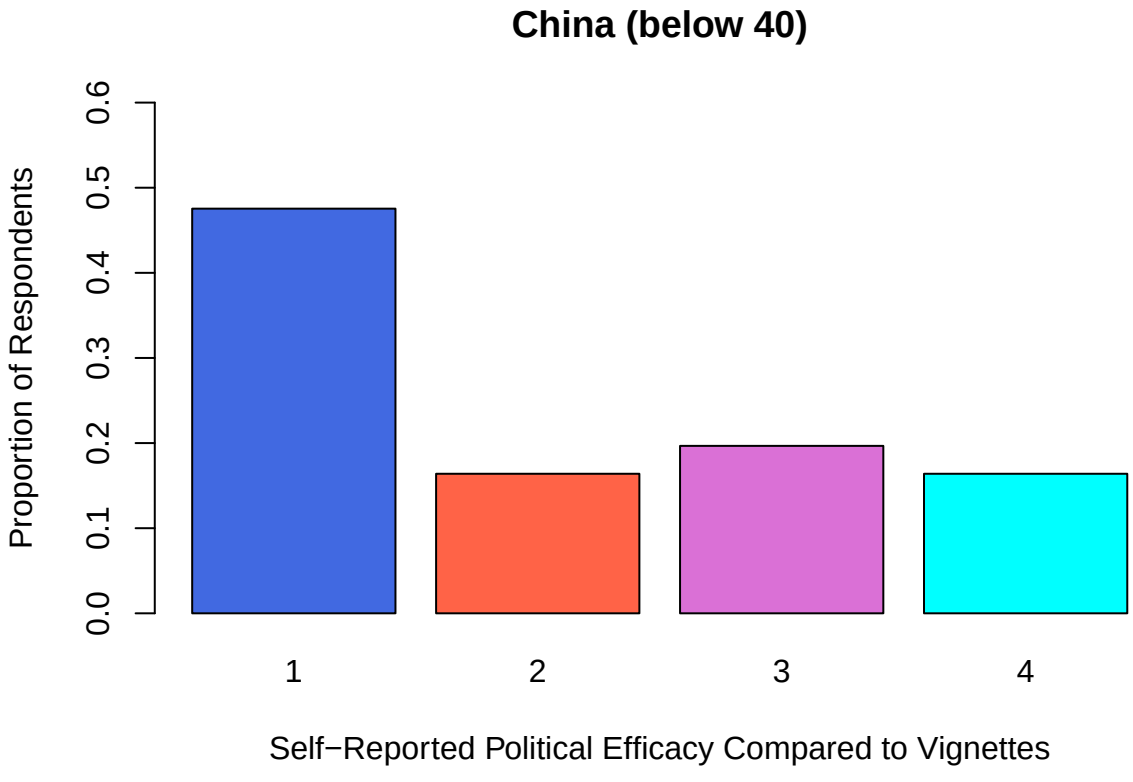
```r
younger_respondents_Mexico <- expected_order_mexico %>% filter(age < 40)
xlab = "Self-Reported Political Efficacy Compared to Vignettes"
ylab = "Proportion of Respondents"
names <- c("1", "2", "3", "4")

barplot(prop.table(table(older_respondents_China$self_rank)), ylim = c(0, 0.6), xlab = xlab, ylab = ylab
```



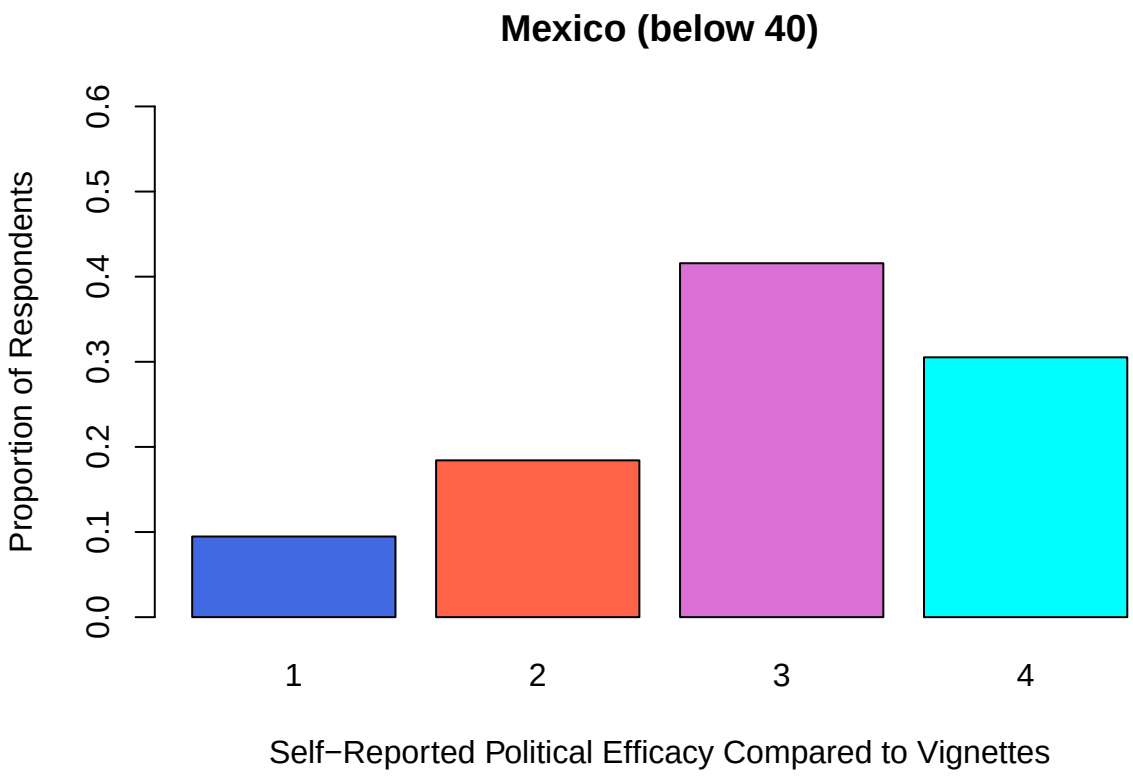

```r
barplot(prop.table(table(older_respondents_Mexico$self_rank)), ylim = c(0, 0.6), xlab = xlab, ylab = yla
```




```
barplot(prop.table(table(younger_respondents_China$self_rank)), ylim = c(0, 0.6), xlab = xlab, ylab = yl
```




```
barplot(prop.table(table(younger_respondents_Mexico$self_rank)), ylim = c(0, 0.6), xlab = xlab, ylab = y
```




```r
# Calculate the mean ranking for each group
mean_ranking_older_china <- mean(older_respondents_China$self_rank)
mean_ranking_older_mexico <- mean(older_respondents_Mexico$self_rank)
mean_ranking_younger_china <- mean(younger_respondents_China$self_rank)
mean_ranking_younger_mexico <- mean(younger_respondents_Mexico$self_rank)


# Calculate the mean self score for each group
mean_self_older_china <- mean(older_respondents_China$self)
mean_self_older_mexico <- mean(older_respondents_Mexico$self)
mean_self_younger_china <- mean(younger_respondents_China$self)
mean_self_younger_mexico <- mean(younger_respondents_Mexico$self)

results <- data.frame(
  Group = c("Older_China", "Older_Mexico", "Younger_China", "Younger_Mexico"),
  Mean_Ranking = c(mean_ranking_older_china, mean_ranking_older_mexico, mean_ranking_younger_china, mean
  Mean_Self_Score = c(mean_self_older_china, mean_self_older_mexico, mean_self_younger_china, mean_self
)

results
```

```
##            Group Mean_Ranking Mean_Self_Score
## 1    Older_China     2.209524        2.714286
## 2   Older_Mexico     2.921986        1.744681
## 3  Younger_China     2.049180        2.409836
## 4 Younger_Mexico     2.931579        1.773684
```

The older Chinese respondents have a slightly higher mean self-score (2.714286) compared to the younger respondents (2.409836), indicating they feel they have more say in government decisions. However, the mean ranking is higher for the older group (2.209524) compared to the younger group (2.049180), suggesting the

older group tends to rank themselves less favorably compared to the vignettes.

The older Mexican respondents have a slightly lower mean self-score (1.744681) compared to the younger respondents (1.773684), indicating both groups feel similarly about their influence on government decisions. The mean ranking is similar between the older (2.921986) and younger (2.931579) groups, showing little difference in how they rank themselves compared to the vignettes.

For both age groups, respondents in China have higher mean self-scores than those in Mexico, suggesting a higher perceived political efficacy in China. However, the mean rankings indicate that Chinese respondents tend to rank themselves lower compared to the vignettes than Mexican respondents, indicating a potential discrepancy in self-assessment versus vignette-assessment.

The problem identified in previous questions about different interpretations of political efficacy seems to be more or less severe depending on the age group. In China, older respondents feel they have more influence, whereas in Mexico, the difference between age groups is minimal. This suggests that age might play a role in perceived political efficacy, and this difference is more pronounced in China than in Mexico.

Problem 2 - Election and Conditional Cash Transfer in Mexico

```
diff_in_means <- function(treated, control){
  # Point Estimate
  point <- mean(treated) - mean(control)

  # Standard Error
  se <- sqrt(var(treated)/length(treated) + var(control)/length(control))

  # Asymptotic 95% CI
  ci_95 <- c(point - qnorm(.975)*se,
             point + qnorm(.975)*se)

  # P-value
  pval <- 2*pnorm(-abs(point/se))

  # Return as a data frame
  output <- data.frame(meanTreated = mean(treated), meanControl = mean(control), est = point, se = se,

  return(as_tibble(output))
}
```

1.

```
progresa <- read_csv("progresa.csv", show_col_types = FALSE)

treatment <- progresa %>%
  filter(treatment == 1)

control <- progresa %>%
  filter(treatment == 0)

diff_in_means(treatment$t2000, control$t2000)
```

```
## # A tibble: 1 x 8
##   meanTreated meanControl   est    se ci95Lower ci95Upper pvalue     N
##         <dbl>       <dbl> <dbl> <dbl>     <dbl>     <dbl>  <dbl> <int>
```

```
## 1        68.1        63.8  4.27  2.85     -1.32      9.86  0.134   417
```

```
turnout_model <- lm(t2000 ~ treatment, data = progresa)
summary(turnout_model)
```

```
##
## Call:
## lm(formula = t2000 ~ treatment, data = progresa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -66.300 -11.700  -1.841   7.778 303.344
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   63.815      2.631  24.255   <2e-16 ***
## treatment      4.270      3.216   1.327    0.185
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.91 on 415 degrees of freedom
## Multiple R-squared:  0.004228,   Adjusted R-squared:  0.001829
## F-statistic: 1.762 on 1 and 415 DF,  p-value: 0.1851
```

```
support_model <- lm(pri2000s ~ treatment, data = progresa)
summary(support_model)
```

```
##
## Call:
## lm(formula = pri2000s ~ treatment, data = progresa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.370 -11.332  -1.673   9.200 104.111
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   34.489      1.554  22.196   <2e-16 ***
## treatment      3.622      1.900   1.907   0.0572 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.25 on 415 degrees of freedom
## Multiple R-squared:  0.008686,   Adjusted R-squared:  0.006298
## F-statistic: 3.636 on 1 and 415 DF,  p-value: 0.05722
```

Simple ATE estimate: From the difference in means between treatment and control group, we can see that there is a positive treatment effect for sure. However, the 95% confidence interval includes 0 so we fail to reject the null hypothesis that there is no impact of the CCT program on turnout and support for the incumbent party (PRI).

Linear regression: From the linear regression result where we predict the treatment effect on both turnout in the 2000 election as a share of precinct and PRI votes as a share of precinct, we can see a positive treatment effect for both (as the coefficient is positive). However, we fail to reject the null hypothesis in both models because the p-value is greater than 0.05, so we cannot conclude that the CCT program will increase in turnout and support for the incumbent party.

2.

```
#
turnout_model <- lm(t2000 ~ treatment + avgpoverty + pobtot1994 + votos1994 + pri1994 + pan1994 + prd199
summary(turnout_model)
```

```
##
## Call:
## lm(formula = t2000 ~ treatment + avgpoverty + pobtot1994 + votos1994 +
##     pri1994 + pan1994 + prd1994, data = progresa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.173 -11.924  -2.938   6.972 302.183
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 64.0117350 17.3115236   3.698 0.000247 ***
## treatment    4.5494445  3.1346655   1.451 0.147454
## avgpoverty   0.3102553  3.5223779   0.088 0.929855
## pobtot1994  -0.0012128  0.0002316  -5.236 2.63e-07 ***
## votos1994   -0.0261518  0.0354456  -0.738 0.461059
## pri1994      0.0360555  0.0409173   0.881 0.378739
## pan1994      0.0265376  0.0595018   0.446 0.655836
## prd1994      0.0175753  0.0426669   0.412 0.680615
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 29.95 on 409 degrees of freedom
## Multiple R-squared:  0.0785, Adjusted R-squared:  0.06273
## F-statistic: 4.978 on 7 and 409 DF,  p-value: 2.01e-05
```

```
support_model <- lm(pri2000s ~ treatment + avgpoverty + pobtot1994 + votos1994 + pri1994 + pan1994 + pr
summary(support_model)
```

```
##
## Call:
## lm(formula = pri2000s ~ treatment + avgpoverty + pobtot1994 +
##     votos1994 + pri1994 + pan1994 + prd1994, data = progresa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.297  -9.854  -1.322   6.468  94.918
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 37.9500862  9.4239516   4.027 6.73e-05 ***
## treatment    2.9277395  1.7064319   1.716  0.08697 .
## avgpoverty   0.5329801  1.9174926   0.278  0.78119
## pobtot1994  -0.0004996  0.0001261  -3.962 8.77e-05 ***
## votos1994   -0.0417278  0.0192957  -2.163  0.03116 *
## pri1994      0.0624589  0.0222744   2.804  0.00529 **
## pan1994     -0.0487349  0.0323913  -1.505  0.13321
## prd1994     -0.0287363  0.0232268  -1.237  0.21672
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.3 on 409 degrees of freedom
## Multiple R-squared:  0.2206, Adjusted R-squared:  0.2073
## F-statistic: 16.54 on 7 and 409 DF,  p-value: < 2.2e-16
```

Turnout (t2000): The Progresa treatment has a positive treatment effect (4.5494445) but not statistically significant effect on turnout (p = 0.147454). Support for PRI (pri2000s): The Progresa treatment has a positive treatment effect (2.9277395) and marginally significant effect on PRI support (p = 0.08697). These results from question 2 are different from question 1 numerically, which has a treatment effect of 3.622 and p-value of 0.0572. Both cases suggest a positive treatment effect on support for PRI, but neither provides strong statistical evidence of significance at the conventional 0.05 level. The additional covariates in the models for Question 2 provide a more nuanced understanding of the treatment effects but do not substantially change the overall interpretation regarding statistical significance.

3.

```
turnout_model <- lm(t2000 ~ treatment + avgpoverty + log(pobtot1994) + t1994 + pri1994s + pan1994s + pr
summary(turnout_model)
```

```
##
## Call:
## lm(formula = t2000 ~ treatment + avgpoverty + log(pobtot1994) +
##     t1994 + pri1994s + pan1994s + prd1994s, data = progresa)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -119.044   -7.248   -1.180    5.579  176.967
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      19.7984    14.4292   1.372  0.17078
## treatment        -0.1530     1.8182  -0.084  0.93299
## avgpoverty        2.8621     1.8954   1.510  0.13180
## log(pobtot1994)  -3.2471     1.1678  -2.780  0.00568 **
## t1994             0.6605     0.1292   5.112 4.91e-07 ***
## pri1994s          0.1943     0.1371   1.417  0.15720
## pan1994s          0.6374     0.2121   3.005  0.00282 **
## prd1994s          0.3065     0.1471   2.084  0.03780 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.31 on 409 degrees of freedom
## Multiple R-squared:  0.6921, Adjusted R-squared:  0.6868
## F-statistic: 131.3 on 7 and 409 DF,  p-value: < 2.2e-16
```

```
support_model <- lm(pri2000s ~ treatment + avgpoverty + log(pobtot1994) + t1994 + pri1994s + pan1994s +
summary(support_model)
```

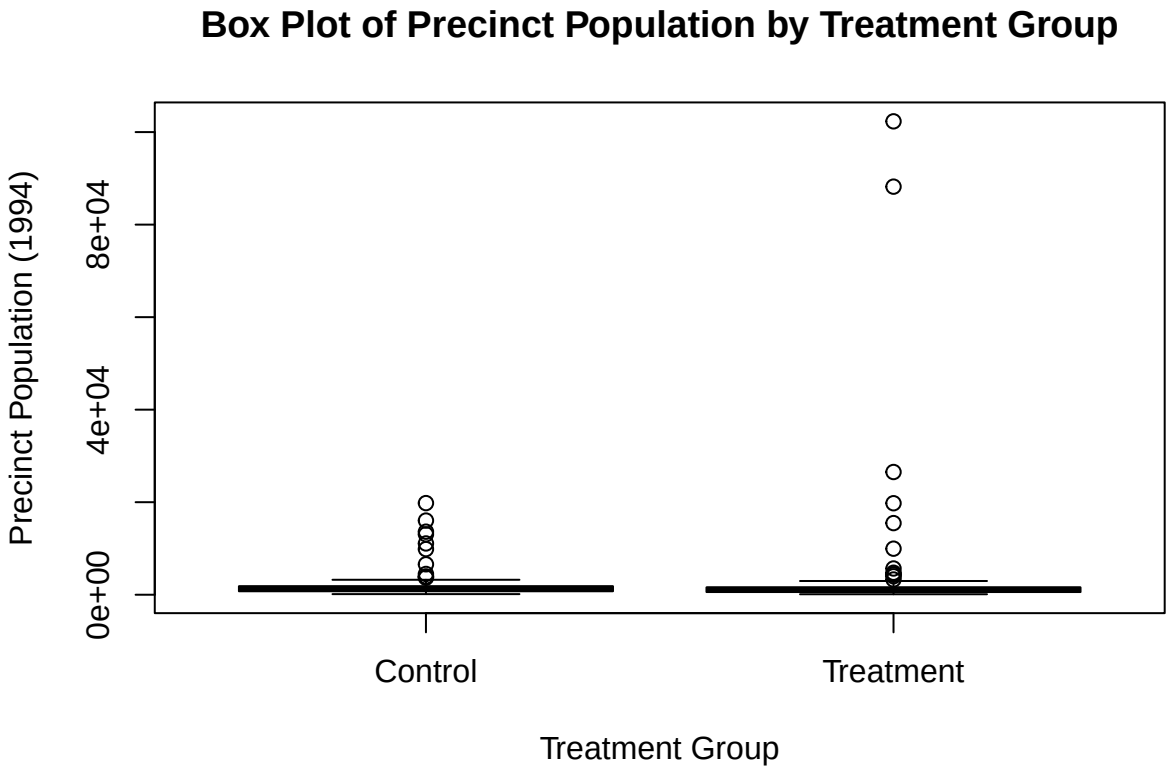
```
##
## Call:
## lm(formula = pri2000s ~ treatment + avgpoverty + log(pobtot1994) +
##     t1994 + pri1994s + pan1994s + prd1994s, data = progresa)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -61.330  -7.237  -0.430   6.116  55.869
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     35.85174    9.98288   3.591 0.000369 ***
## treatment        0.23547    1.25790   0.187 0.851602
## avgpoverty       2.47163    1.31133   1.885 0.060161 .
## log(pobtot1994) -4.62934    0.80798  -5.730 1.96e-08 ***
## t1994            0.03257    0.08939   0.364 0.715792
## pri1994s         0.51047    0.09488   5.380 1.26e-07 ***
## pan1994s        -0.18384    0.14676  -1.253 0.211052
## prd1994s        -0.05293    0.10175  -0.520 0.603192
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.98 on 409 degrees of freedom
## Multiple R-squared:  0.5794, Adjusted R-squared:  0.5722
## F-statistic: 80.48 on 7 and 409 DF,  p-value: < 2.2e-16
```

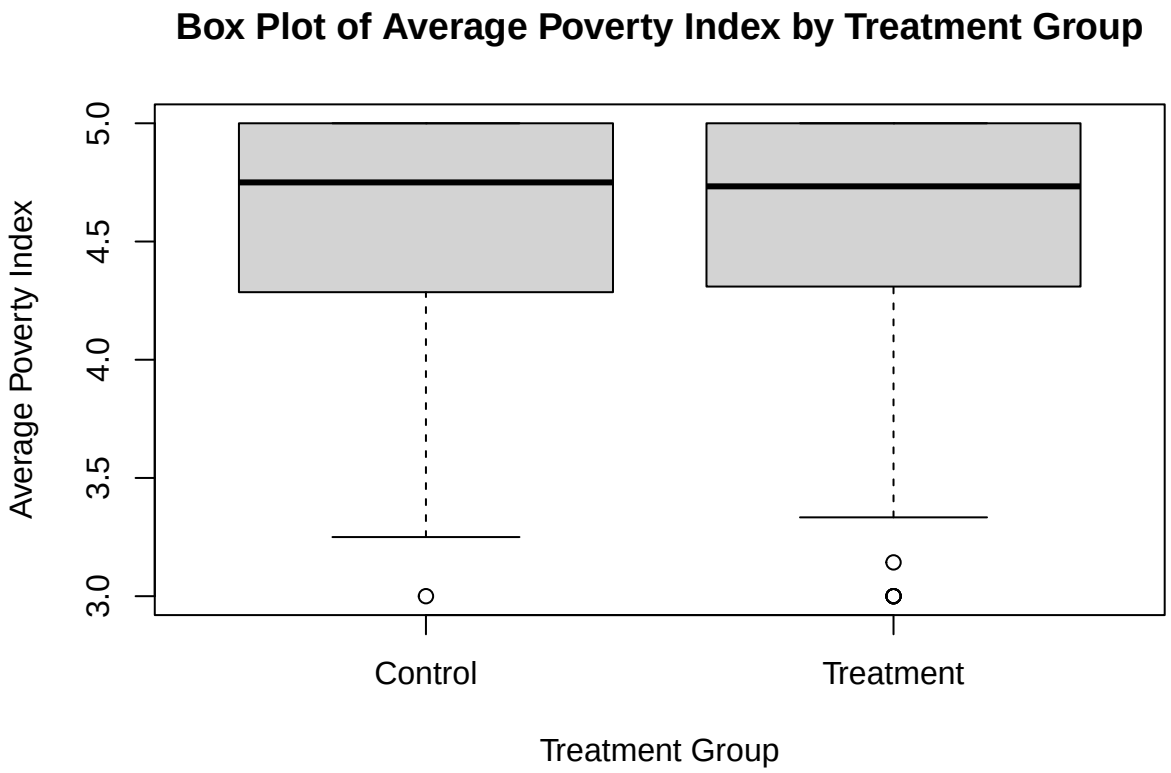
Turnout (t2000): The treatment effect changes from positive (4.5494445) in the previous model to a small negative (-0.1530) in the new model. The new model shows no significant effect of treatment on turnout, and it improves the model fit significantly with a higher R-squared value (0.6921 compared to 0.0785).
Support for PRI (pri2000s): The treatment effect remains positive but is reduced from 2.9277395 to 0.23547 in the new model. The new model shows no significant effect of treatment on PRI support, but the fit of the model improves significantly with a higher R-squared value (0.5794 compared to 0.2206).
The new model specifications, which include the log-transformed precinct population and previous election outcomes as shares of the voting age population, provide a better fit to the data as indicated by higher R-squared values. The results indicate that while the Progresa treatment does not have a significant effect on turnout or PRI support, the inclusion of additional predictors provides a more nuanced understanding of the factors influencing these outcomes. The new model fits the data better compared to the previous models.

4.

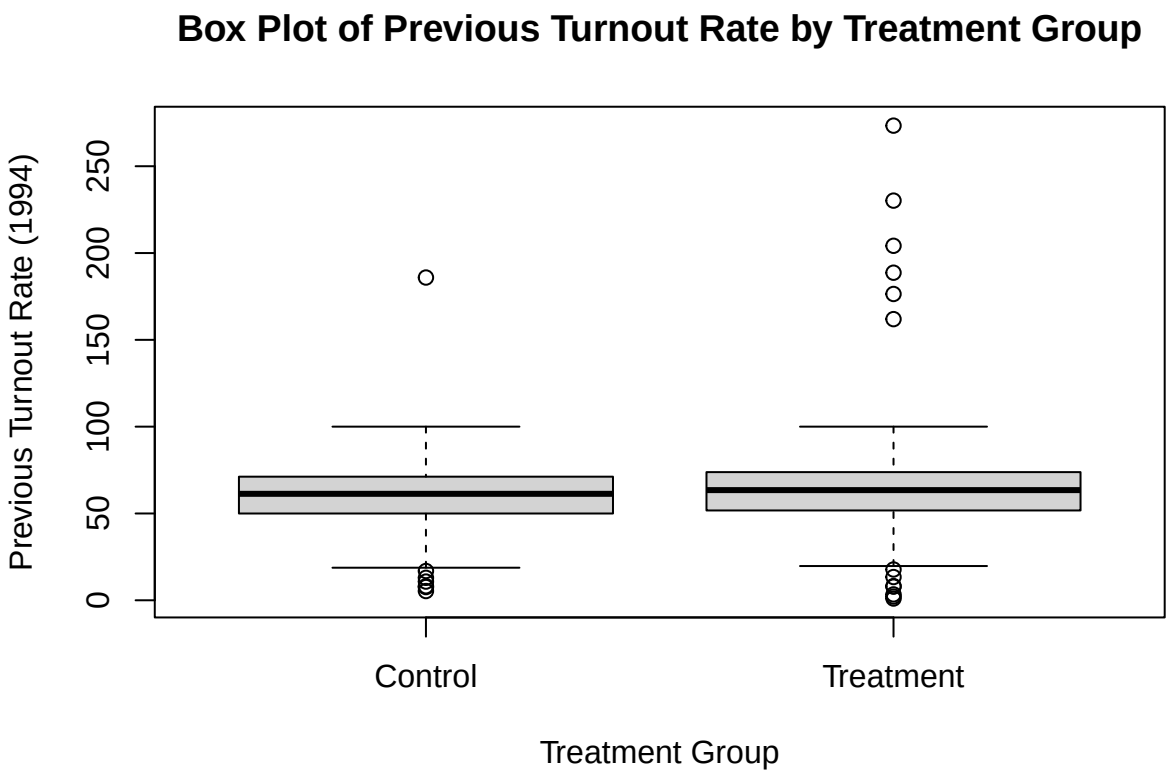
```r
# Standardize covariates
# progresa <- progresa %>%
#   mutate(treatment_std = treatment/sd(treatment),
#          avgpoverty_std = avgpoverty/sd(avgpoverty),
#          pobtot1994_std = pobtot1994/sd(pobtot1994),
#          votos1994_std = votos1994/sd(votos1994),
#          t1994_std = t1994/sd(t1994),
#          pri1994s_std = pri1994s/sd(pri1994s),
#          pan1994s_std = pan1994s/sd(pan1994s),
#          prd1994s_std = prd1994s/sd(prd1994s))

# Box plot for precinct population
boxplot(pobtot1994 ~ treatment, data = progresa,
        main = "Box Plot of Precinct Population by Treatment Group",
        xlab = "Treatment Group", ylab = "Precinct Population (1994)",
        names = c("Control", "Treatment"))
```



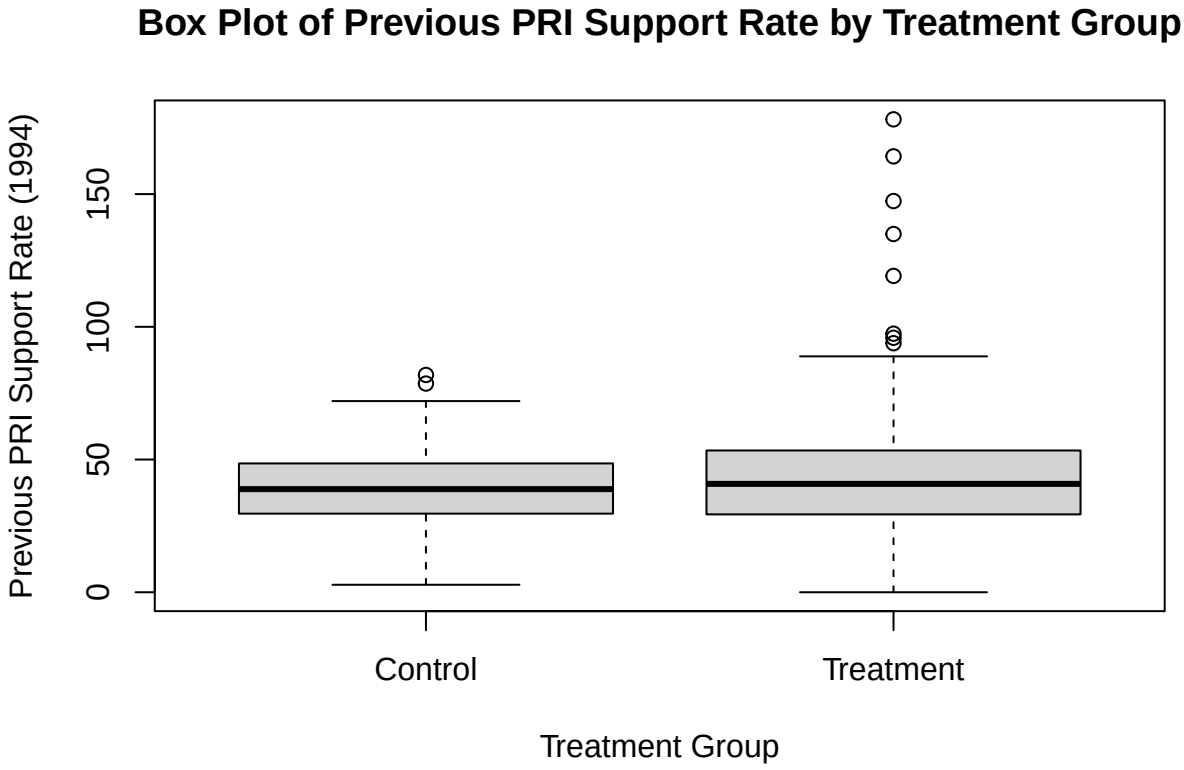

```r
# Box plot for average poverty index
boxplot(avgpoverty ~ treatment, data = progresa,
        main = "Box Plot of Average Poverty Index by Treatment Group",
        xlab = "Treatment Group", ylab = "Average Poverty Index",
        names = c("Control", "Treatment"))
```



```r
# Box plot for previous turnout rate
boxplot(t1994 ~ treatment, data = progresa,
        main = "Box Plot of Previous Turnout Rate by Treatment Group",
        xlab = "Treatment Group", ylab = "Previous Turnout Rate (1994)",
        names = c("Control", "Treatment"))
```




```r
# Box plot for previous PRI support rate
boxplot(pri1994s ~ treatment, data = progresa,
        main = "Box Plot of Previous PRI Support Rate by Treatment Group",
        xlab = "Treatment Group", ylab = "Previous PRI Support Rate (1994)",
        names = c("Control", "Treatment"))
```



The box plots indicate that the key pretreatment variables (precinct population, average poverty index, previous turnout rate, and previous PRI support rate) are generally balanced between the treatment and control groups. This suggests that the randomization process in the Progresa study was effective in creating comparable groups with respect to these covariates. The presence of some outliers, especially in precinct population, does not appear to significantly affect the overall balance between the groups.

5.

```
turnout_model_official <- lm(t2000r ~ treatment + avgpoverty + log(pobtot1994) + t1994r + pri1994v + pa
summary(turnout_model_official)
```

```
##
## Call:
## lm(formula = t2000r ~ treatment + avgpoverty + log(pobtot1994) +
##     t1994r + pri1994v + pan1994v + prd1994v, data = progresa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.041  -4.555   0.029   5.010  29.069
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     33.47237    8.85553   3.780 0.000180 ***
## treatment       -1.08094    0.81683  -1.323 0.186465
## avgpoverty      -0.27734    0.88768  -0.312 0.754874
## log(pobtot1994) -0.27878    0.47487  -0.587 0.557487
## t1994r           0.22238    0.03102   7.168 3.59e-12 ***
## pri1994v         0.12473    0.05489   2.272 0.023586 *
## pan1994v         0.27356    0.07257   3.770 0.000188 ***
## prd1994v         0.11323    0.05790   1.955 0.051209 .
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.795 on 408 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.1809, Adjusted R-squared:  0.1669
## F-statistic: 12.87 on 7 and 408 DF,  p-value: 5.603e-15
```

```
pri_model_official <- lm(pri2000v ~ treatment + avgpoverty + log(pobtot1994) + t1994r + pri1994v + pan19
summary(pri_model_official)
```

```
##
## Call:
## lm(formula = pri2000v ~ treatment + avgpoverty + log(pobtot1994) +
##     t1994r + pri1994v + pan1994v + prd1994v, data = progresa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.876  -7.686   1.072   7.461  34.758
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     50.22721   14.43904   3.479 0.000558 ***
## treatment        0.80172    1.33185   0.602 0.547536
## avgpoverty       3.19082    1.44737   2.205 0.028042 *
## log(pobtot1994) -2.59489    0.77428  -3.351 0.000879 ***
## t1994r          -0.08612    0.05058  -1.703 0.089421 .
## pri1994v         0.35738    0.08950   3.993 7.73e-05 ***
## pan1994v        -0.48863    0.11832  -4.130 4.41e-05 ***
## prd1994v        -0.24158    0.09441  -2.559 0.010862 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.71 on 408 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.4836, Adjusted R-squared:  0.4747
## F-statistic: 54.58 on 7 and 408 DF,  p-value: < 2.2e-16
```

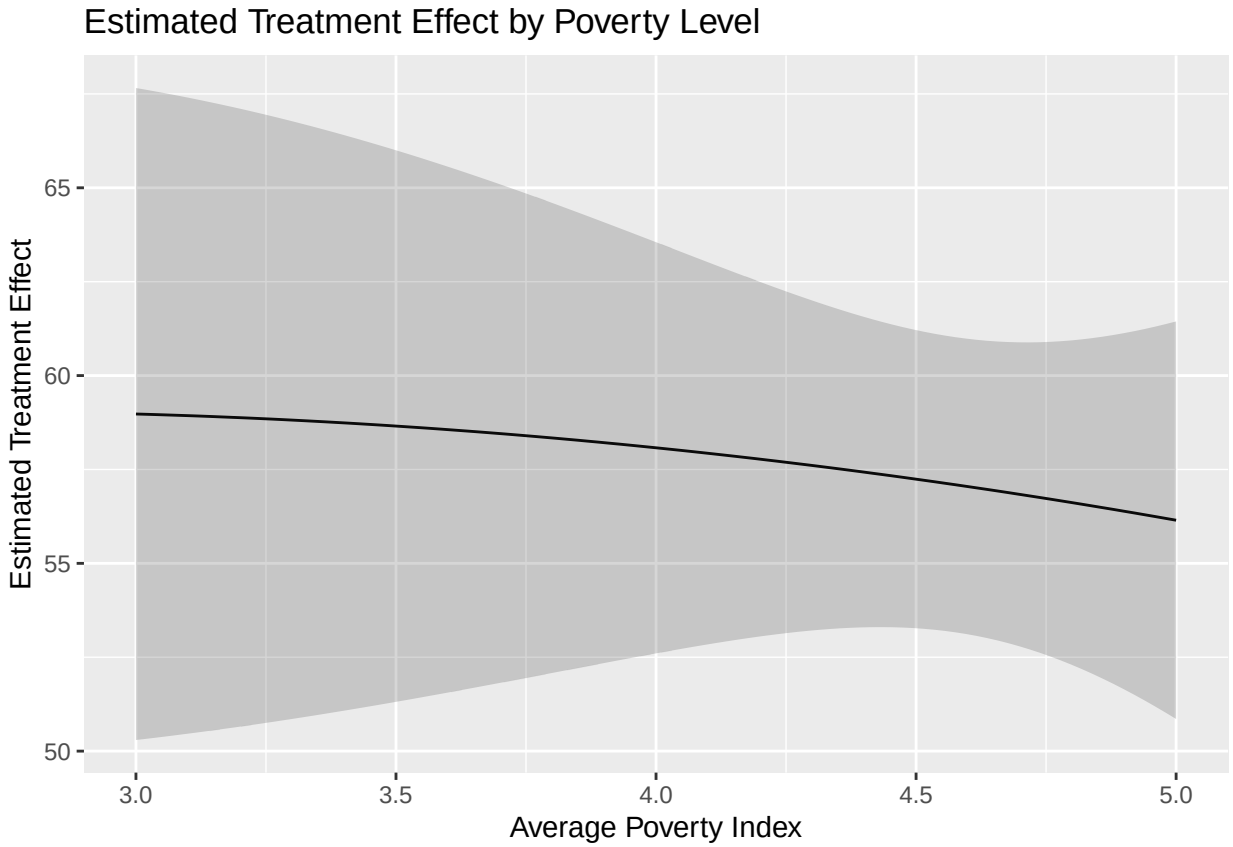
6.

```
progresa <- progresa %>%
  mutate(
    poverty_squared = avgpoverty^2,
    treatment_poverty = treatment * avgpoverty,
    treatment_poverty_squared = treatment * (avgpoverty)^2
  )

poverty_interaction_model <- lm(t2000 ~ treatment_poverty + treatment_poverty_squared + log(pobtot1994)
summary(poverty_interaction_model)
```

```
##
## Call:
## lm(formula = t2000 ~ treatment_poverty + treatment_poverty_squared +
##     log(pobtot1994), data = progresa)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.147 -13.350  -2.064   8.119 280.163
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                177.9165    11.6257  15.304   <2e-16 ***
## treatment_poverty            2.7041     4.2131   0.642    0.521
## treatment_poverty_squared   -0.5148     0.8788  -0.586    0.558
## log(pobtot1994)            -16.0667     1.6014 -10.033   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.71 on 413 degrees of freedom
## Multiple R-squared:  0.2035, Adjusted R-squared:  0.1977
## F-statistic: 35.18 on 3 and 413 DF,  p-value: < 2.2e-16
```

```r
# Define a range of poverty levels for prediction
poverty_levels <- seq(min(progresa$avgpoverty), max(progresa$avgpoverty), length.out = 100)

# Create a data frame for prediction
prediction_data <- data.frame(
  treatment = 1,
  avgpoverty = poverty_levels,
  poverty_squared = poverty_levels^2,
  treatment_poverty = 1 * poverty_levels,
  treatment_poverty_squared = 1 * poverty_levels^2,
  pobtot1994 = mean(progresa$pobtot1994)
)

# Predict the treatment effect at different poverty levels
predicted_effects <- predict(poverty_interaction_model, newdata = prediction_data, se.fit = TRUE)

# Combine predictions with poverty levels
results <- data.frame(
  avgpoverty = poverty_levels,
  treatment_effect = predicted_effects$fit,
  lower_ci = predicted_effects$fit - 1.96 * predicted_effects$se.fit,
  upper_ci = predicted_effects$fit + 1.96 * predicted_effects$se.fit
)

# Plot the predicted treatment effects
ggplot(results, aes(x = avgpoverty, y = treatment_effect)) +
  geom_line() +
  geom_ribbon(aes(ymin = lower_ci, ymax = upper_ci), alpha = 0.2) +
  labs(title = "Estimated Treatment Effect by Poverty Level",
       x = "Average Poverty Index",
       y = "Estimated Treatment Effect")
```



Problem 3 - Data Generating Process

Given the Data Generating Process (DGP):

$$Y_i \mid X_i, D_i = X_i\beta + \tau D_i + \epsilon_i$$

Whenever the condition is met,the binary treatment indicator will be:

$$D_i \mid X_i = I(X_i\gamma + \nu_i > 0)$$

The noise term follows a standard normal distribution with mean of 0 and a variance of 1:

$$\epsilon_i \sim N(0,1)$$

$$\nu_i \sim N(0,1)$$

$$X_i \sim Bernoulli(\pi)$$

3.1 Distribution of $Y_i \mid X_i, D_i$

Initially, since $\epsilon_i \sim N(0,1)$, the outcome will be normally distributed when the linear combination adds with a random variable that follows standard normal distribution $N(0,1)$:

$$Y_i \mid X_i, D_i \sim N(X_i\beta + \tau D_i, 1)$$

Where the mean is $\mu_{Y_i|X_i,D_i} = X_i\beta + \tau D_i$.

Thus, the mean and variance of $Y_i \mid X_i, D_i$ are:
- **Mean**: $E[Y_i \mid X_i, D_i] = X_i\beta + \tau D_i$.
- **Variance**: $\text{Var}(Y_i \mid X_i, D_i) = 1$.

Secondly, since the outcome variable is $E[Y_i \mid X_i, D_i] \sim N(X_i\beta + \tau D_i, 1)$, we can standardize it by the following:
- Let $A = E[Y_i \mid X_i, D_i]$ and since $Z = \frac{A-\mu}{\sigma}$.
- We will have $Z = \frac{A-(X_i\beta+\tau D_i)}{1}$ since the mean and variance of $Y_i \mid X_i, D_i$ are $X_i\beta + \tau D_i$ and 1.
- And therefore $Z \sim N(0, 1)$.

Lastly, given $Z \sim N(0, 1)$ and $Z = A - (X_i\beta + \tau D_i)$, the probability density function (PDF) and cumulative distribution function (CDF) are:
PDF Here we use the PDF of a normal distribution $N(\mu, \sigma^2)$:
- $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$.
We will have the following evaluated at point y with $N(0, 1)$:
- $\phi(y) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y-(X_i\beta+\tau D_i))^2}{2}\right)$.

CDF Here we use the CDF of a normal distribution $N(\mu, \sigma^2)$:
- $F(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) dt$.
We will have the following CDF evaluated at point z with $N(0, 1)$:
- $\Phi(x) = \Pr(Z \leq x) = \int_{-\infty}^{x} \phi(t)dt$.
- Therefore, we will have: $F(y) = \Phi(y - (X_i\beta + \tau D_i))$.

3.2 Expected Values of $D_i \mid X_i$

Initially, D_i is an indicator function that takes the value 1 when $X_i\gamma + \nu_i > 0$, and 0 otherwise. Hence, $\nu_i > -X_i\gamma$. Secondly, X_i, our covariate, is a binary random variable that takes value 1 with probability π.

For the first case when $E[D_i \mid X_i = 1]$:
- $E[D_i \mid X_i = 1] = \Pr(D_i = 1 \mid X_i = 1)$ because the expected value of an indicator function is the probability that the condition is met, which will be $\Pr(\nu_i > -\gamma)$ in this case.
- Since $\nu_i \sim N(0, 1)$:
$\Pr(\nu_i > -\gamma) = 1 - \Pr(\nu_i \leq -\gamma) = \Phi(\gamma)$.
- In conclusion, $E[D_i \mid X_i = 1] = \Phi(\gamma)$.

For the second case when $E[D_i \mid X_i = 0]$:
- Given $X_i = 0$, the condition that $D_i = 1$ is when $\nu_i > 0$.
- $E[D_i \mid X_i = 0] = \Pr(D_i = 1 \mid X_i = 0) = \Pr(\nu_i > 0) = 1 - \Pr(\nu_i \leq 0)$.
- Thus, $\Pr(\nu_i \leq 0) = \Phi(0) = 0.5$ and $E[D_i \mid X_i = 0] = 1 - \Phi(0) = 1 - 0.5 = 0.5$.

In summary, we find that $E[D_i \mid X_i = 1] = \Phi(\gamma)$ and $E[D_i \mid X_i = 0] = 0.5$.

3.3 Applying Bayes' Rule

Bayes' rule states that:

$$\Pr(A = a \mid B = b) = \frac{\Pr(B = b \mid A = a)\Pr(A = a)}{\sum_a \Pr(B = b \mid A = a)\Pr(A = a)}$$

where $\Pr(B = b) = \sum_a \Pr(B = b \mid A = a)\Pr(A = a)$ according to the law of total probability.

From previous questions, we have found out that:
- $E[D_i \mid X_i = 1] = \Phi(\gamma)$ and $E[D_i \mid X_i = 0] = 0.5$.

For the first case, we want $\Pr(X_i = 1 \mid D_i = 1)$:

$$\Pr(X_i = 1 \mid D_i = 1) = \frac{\Pr(D_i = 1 \mid X_i = 1)\Pr(X_i = 1)}{\sum_x \Pr(D_i = 1 \mid X_i = x)\Pr(X_i = x)}$$

Given $\Pr(D_i = 1 \mid X_i = 1) = \Phi(\gamma)$, $\Pr(D_i = 1 \mid X_i = 0) = 0.5$, $\Pr(X_i = 1) = \pi$, and $\Pr(X_i = 0) = 1 - \pi$, we will have:

$$\frac{\Pr(D_i = 1 \mid X_i = 1)\Pr(X_i = 1)}{\sum_x \Pr(D_i = 1 \mid X_i = x)\Pr(X_i = x)} = \frac{\Phi(\gamma)\pi}{\Phi(\gamma)\pi + 0.5(1-\pi)}$$

In conclusion, $\Pr(X_i = 1 \mid D_i = 1) = \frac{\Phi(\gamma)\pi}{\Phi(\gamma)\pi + 0.5(1-\pi)}$

Similarly, for $\Pr(X_i = 1 \mid D_i = 0)$:

$$\Pr(X_i = 1 \mid D_i = 0) = \frac{(1 - \Phi(\gamma))\pi}{1 - (\Phi(\gamma)\pi + 0.5(1-\pi))}$$

3.4 Deriving Bias Formula

For $E[Y_i \mid D_i = 1]$ and $E[Y_i \mid D_i = 0]$, we initially substitute equation (1) from the DGP:

- $E[Y_i \mid D_i = 1] = E[X_i\beta + \tau D_i + \epsilon_i] = X_i\beta + \tau$ as $E[\epsilon_i] = 0$
- $E[Y_i \mid D_i = 0] = E[X_i\beta + \tau D_i + \epsilon_i] = X_i\beta$

Thus, we have:
- $\eta = (X_i\beta + \tau) - X_i\beta = \tau$

Therefore, the formula for the bias $\eta - \tau$ will be $\eta - \tau = \tau - \tau = 0$, which shows that there is no bias in the DGP equations.

Problem 4 - Propensity Score Regression and IPW

4.1

The propensity score is the probability of receiving treatment given the covariate, $e(X_i) = P(D_i = 1 \mid X_i)$. For $X_i = 1$ and $X_i = 0$, the predicted propensity scores can be calculated as follows, given $(X'X)^{-1}X'D = \frac{1}{\sum_{i=1}^n X_i}$ and $X'D = \sum_{i=1}^n X_i D_i$:
- For $X = 1$:
The propensity score is the probability of receiving treatment given the covariate is $X = 1$. And we know from DGP before that X is a binary random variable. Therefore, the propensity score is the sum of treatment given the covariate is 1 divided by the number of units who has the covariate of 1.

$$\hat{e}(1) = E[D_i \mid X_i = 1] = \gamma_i = \frac{\sum_{i=1}^n X_i D_i}{\sum_{i=1}^n X_i}$$

- For $X = 0$:
  Similarly, the propensity score here is $\hat{e}(0) = Pr(D_i = 1 \mid X_i = 0)$, which can be calculated by taking the sum of D_i for all instances where $X_i = 0$ and dividing it by the total number of instances where $X_i = 0$.

$$\hat{e}(0) = E[D_i \mid X_i = 0] = \gamma_0 = \frac{\sum_{i=1}^n (1 - X_i) D_i}{\sum_{i=1}^n (1 - X_i)}$$

4.2

Based on the information provided, we know that:

$$\hat{e}(1) = \frac{\sum_{i=1}^n X_i D_i}{\sum_{i=1}^n X_i}$$

$$\hat{e}(0) = \frac{\sum_{i=1}^{n}(1-X_i)D_i}{\sum_{i=1}^{n}(1-X_i)}$$

- For the stratified estimator on X:
  The stratified estimator is the sum of average treatment effect of all stratas and their corresponding weight (the number of units in the strata divided by the total number of all units).
- For the IPW estimator:
  $\hat{\tau} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{D_iY_i}{e(X_i)} - \frac{(1-D_i)Y_i}{1-e(X_i)}\right)$

Given

$$\hat{e}(1) = \frac{\sum_{i=1}^{n} X_iD_i}{\sum_{i=1}^{n} X_i}$$

,

$$X_i \sim Bernoulli(\pi)$$

, and the IPW estimator, we will have:
- For $X_i = 1$:
$\hat{\tau} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{D_iY_i}{\hat{e}(1)} - \frac{(1-D_i)Y_i}{1-\hat{e}(1)}\right)$

- For $X_i = 0$:
  $\hat{\tau} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{D_iY_i}{\hat{e}(0)} - \frac{(1-D_i)Y_i}{1-\hat{e}(0)}\right)$

Since we know that the propensity score is the sum of D_i for all instances where $X_i = 1$ and dividing it by the total number of instances where $X_i = 1$, we will have:

$$\hat{e}(1) = \frac{\sum_{i:X_i=1} D_i}{N_{X=1}}$$

Similarly, we will also have:

$$\hat{e}(0) = \frac{\sum_{i:X_i=0} D_i}{N_{X=0}}$$

Substitute both equation into $\hat{\tau}$:
- For $X_i = 1$:
$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{D_iY_i}{\frac{\sum_{i:X_i=1} D_i}{N_{X=1}}} - \frac{(1-D_i)Y_i}{1-\frac{\sum_{i:X_i=1} D_i}{N_{X=1}}}\right)$
which is equivalent to:
$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{D_iY_iN_{X=1}}{\sum_{i:X_i=1} D_i} - \frac{(1-D_i)Y_iN_{X=1}}{N_{X=1}-\sum_{i:X_i=1} D_i}\right)$
Therefore, we will have the following after simplification:
$\frac{N_{X=1}}{N}\left(\frac{\sum_{i:X_i=1} D_iY_i}{\sum_{i:X_i=1} D_i} - \frac{\sum_{i:X_i=1}(1-D_i)Y_i}{N_{X=1}-\sum_{i:X_i=1} D_i}\right)$
Eventually:
$\frac{1}{N_{X_i=1D_i=1}}\sum_{i:X_i=1} D_iY_i - \frac{1}{N_{X_i=1D_i=1}}\sum_{i:X_i=0}(1-D_i)Y_i$

- For $X_i = 0$: $\frac{1}{n}\sum_{i=1}^{n}\left(\frac{D_iY_i}{\frac{\sum_{i:X_i=0} D_i}{N_{X=0}}} - \frac{(1-D_i)Y_i}{1-\frac{\sum_{i:X_i=0} D_i}{N_{X=0}}}\right)$
  which is equivalent to:
  $\frac{1}{n}\sum_{i=1}^{n}\left(\frac{D_iY_iN_{X=0}}{\sum_{i:X_i=0} D_i} - \frac{(1-D_i)Y_iN_{X=0}}{N_{X=0}-\sum_{i:X_i=0} D_i}\right)$
  Therefore, we will have the following after simplification:

$\frac{N_{X=0}}{N}\left(\frac{\sum_{i:X_i=0} D_i Y_i}{\sum_{i:X_i=0} D_i} - \frac{\sum_{i:X_i=0}(1-D_i)Y_i}{N_{X=0}-\sum_{i:X_i=0} D_i}\right)$
Eventually:
$\frac{1}{N_{X_i=0 D_i=1}} \sum_{i:X_i=0} D_i Y_i - \frac{1}{N_{X_i=0 D_i=1}} \sum_{i:X_i=0}(1-D_i)Y_i$

Thus, if we combine everything together:
$\hat{\tau} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{D_i Y_i}{e(X_i)} - \frac{(1-D_i)Y_i}{1-e(X_i)}\right) = \frac{1}{N_{X_i=1 D_i=1}} \sum_{i:X_i=1} D_i Y_i - \frac{1}{N_{X_i=1 D_i=1}} \sum_{i:X_i=0}(1-D_i)Y_i + \frac{1}{N_{X_i=0 D_i=1}} \sum_{i:X_i=0} D_i Y_i - \frac{1}{N_{X_i=0 D_i=1}} \sum_{i:X_i=0}(1-D_i)Y_i$ Simplify:

$$= \sum_{x=0}^{1} \frac{N_{X=x}}{n}\left(\frac{1}{N_{X_i=x, D_i=1}} \sum_{i:X_i=x} Y_i D_i - \frac{1}{N_{X_i=x, D_i=0}} \sum_{i:X_i=x} Y_i(1-D_i)\right)$$

4.3

Problem 5 - Matching Approach using TRC

Part a.

```
# Loading library needed for reading dta file and matching
# install.packages("Matching")
library(haven)
library(Matching)
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## ##
## ##  Matching (Version 4.10-14, Build Date: 2023-09-13)
## ##  See https://www.jsekhon.com for additional documentation.
## ##  Please cite software as:
## ##   Jasjeet S. Sekhon. 2011. ``Multivariate and Propensity Score Matching
## ##   Software with Automated Balance Optimization: The Matching package for R.''
## ##   Journal of Statistical Software, 42(7): 1-52.
## ##
```

```
# Reading the data
trc <- read_dta("trc_data.dta")

# Treatment of our interest
treatment <- trc$TRCKNOW

# Outcome of our interest
Y <- trc$RUSTAND

# The covariates we need for matching
covariates <- trc[, c("age", "female", "wealth", "religiosity", "ethsalience", "rcblack", "rcwhite", "r

# Apply matching function in R, Weight = 2 means specifies Mahalanobis distance
match_result <- Match(Y = Y, Tr = treatment, X = covariates, estimand ="ATE", M = 1, Weight = 2)
summary(match_result)
```

```
##
## Estimate...  -0.18157
## AI SE......  0.050924
## T-stat.....  -3.5654
## p.val......  0.00036331
##
## Original number of observations..............  3205
## Original number of treated obs...............  1439
## Matched number of observations...............  3205
## Matched number of observations  (unweighted).  3419
```

```
# The estimate of causal effect
ate <- match_result$est
confidence_interval <- match_result$se * qnorm(0.975)

cat("Estimated ATE:", ate, "\n")
```

```
## Estimated ATE: -0.1815653
```

```
cat("95% Confidence Interval:", ate - confidence_interval, "to", ate + confidence_interval, "\n")
```

```
## 95% Confidence Interval: -0.2813751 to -0.08175546
```

Part b.

```
match_result <- Match(Y = Y, Tr = treatment, X = covariates, estimand ="ATE", M = 3, Weight = 2)
summary(match_result)
```

```
##
## Estimate...  -0.15611
## AI SE......  0.046918
## T-stat.....  -3.3273
## p.val......  0.00087692
##
## Original number of observations..............  3205
## Original number of treated obs...............  1439
## Matched number of observations...............  3205
## Matched number of observations  (unweighted).  9787
```

```
# The estimate of causal effect
ate <- match_result$est
confidence_interval <- match_result$se * qnorm(0.975)

cat("Estimated ATE:", ate, "\n")
```

```
## Estimated ATE: -0.1561117
```

```
cat("95% Confidence Interval:", ate - confidence_interval, "to", ate + confidence_interval, "\n")
```

```
## 95% Confidence Interval: -0.2480702 to -0.06415327
```

The standard error changes and decreases from 0.050924 to 0.046918.

Part c.

```r
match_result <- Match(Y = Y, Tr = treatment, X = covariates, estimand ="ATE", BiasAdjust = TRUE, M = 3,
summary(match_result)
```

```
##
## Estimate...  -0.14832
## AI SE......  0.046892
## T-stat.....  -3.1631
## p.val......  0.0015609
##
## Original number of observations..............  3205
## Original number of treated obs...............  1439
## Matched number of observations...............  3205
## Matched number of observations  (unweighted).  9787
```

```r
# The estimate of causal effect
ate <- match_result$est
confidence_interval <- match_result$se * qnorm(0.975)

cat("Estimated ATE:", ate, "\n")
```

```
## Estimated ATE: -0.1483235
```

```r
cat("95% Confidence Interval:", ate - confidence_interval, "to", ate + confidence_interval, "\n")
```

```
## 95% Confidence Interval: -0.2402291 to -0.05641783
```

The standard error decreases after implementing bias adjust.

Part d.

```r
# # Checking balance between treatment and control group, so that our treatment and control will have t
# MatchBalance(TRCKNOW ~ age + female + wealth + religiosity + ethsalience + rcblack + rcwhite + rccol

# Extract indices of matched pairs
treated_indices <- match_result$index.treated
control_indices <- match_result$index.control

# Compute matching weights
weights <- rep(0, nrow(trc))
weights[treated_indices] <- 1 / length(treated_indices) # Weights for treated units
weights[control_indices] <- 1 / length(control_indices) # Weights for control units

# Add weights to the dataset
trc$weights <- weights

# Function to compute weighted mean and standard deviation
weighted_stats <- function(var, weights) {
  weighted_mean <- sum(var * weights) / sum(weights)
  weighted_sd <- sqrt(sum(weights * (var - weighted_mean)^2) / sum(weights))
  return(c(mean = weighted_mean, sd = weighted_sd))
}

# Compute balance statistics for each covariate
balance_table <- data.frame(Covariate = colnames(covariates),
                            Treated_Mean = NA, Treated_SD = NA,
                            Control_Mean = NA, Control_SD = NA)
```

```r
for (i in 1:ncol(covariates)) {
  covariate <- covariates[, i]
  treated_stats <- weighted_stats(trc[treatment == 1, i], trc$weights[treatment == 1])
  control_stats <- weighted_stats(trc[treatment == 0, i], trc$weights[treatment == 0])

  balance_table$Treated_Mean[i] <- treated_stats["mean"]
  balance_table$Treated_SD[i] <- treated_stats["sd"]
  balance_table$Control_Mean[i] <- control_stats["mean"]
  balance_table$Control_SD[i] <- control_stats["sd"]
}


# Display the balance table
print(balance_table)
```

```
##     Covariate Treated_Mean   Treated_SD Control_Mean   Control_SD
## 1         age   38.9402363   14.7964381   40.4546999   15.9220901
## 2      female    0.5378735    0.4985635    0.4326161    0.4954386
## 3      wealth 6945.1702571 7613.1430526 5792.7746319 7341.1164874
## 4 religiosity    3.8401668    1.8492976    3.9156285    1.8000926
## 5 ethsalience    2.7345379    0.5632931    2.7134768    0.5901318
## 6     rcblack    0.5517721    0.4973124    0.5130238    0.4998304
## 7     rcwhite    0.2696317    0.4437685    0.2531144    0.4347959
## 8       rccol    0.1104934    0.3135038    0.1574179    0.3641943
## 9        EDUC    4.2918694    1.1933959    3.8465459    1.0908385
```