PS3_JTung

Tung, Joanna

May 12, 2017

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Import Packages

```
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
library(forcats)
library(broom)
library(modelr)
```

```
##
## Attaching package: 'modelr'
```

```
## The following object is masked from 'package:broom':
##
##     bootstrap
```

```
library(stringr)
library(ISLR)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(haven)
```

```
##
## Attaching package: 'haven'

## The following object is masked from 'package:forcats':
##
##     as_factor
```

```
library(plotly)
```

```
##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(coefplot)
library(car)
```

```
##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:purrr':
##
##     some
```

```
library(MVN)
```

```
## sROC 0.1-2 loaded
```

```
options(digits = 3)
set.seed(1234)
theme_set(theme_minimal())
```

Read in the data

```
biden_data <- read_csv("data/biden.csv") %>%
  na.omit() %>%
  mutate(ID = row_number())
```

```
## Parsed with column specification:
## cols(
##   biden = col_integer(),
##   female = col_integer(),
##   age = col_integer(),
##   educ = col_integer(),
##   dem = col_integer(),
##   rep = col_integer()
## )
```

```
biden_data
```

```
## # A tibble: 1,807 × 7
##    biden female   age  educ   dem   rep    ID
##    <int>  <int> <int> <int> <int> <int> <int>
## 1     90      0    19    12     1     0     1
## 2     70      1    51    14     1     0     2
## 3     60      0    27    14     0     0     3
## 4     50      1    43    14     1     0     4
## 5     60      1    38    14     0     1     5
## 6     85      1    27    16     1     0     6
## 7     60      1    28    12     0     0     7
## 8     50      0    31    15     1     0     8
## 9     50      1    32    13     0     0     9
## 10    70      0    51    14     1     0    10
## # ... with 1,797 more rows
```

PART ONE: Regression Diagnostics

For this exercise we consider the following functional form:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$$

where Y is the Joe Biden feeling thermometer, X1 is age, X2 is gender, X3 is education

We estimate the parameters and standard errors for this linear regression model below:

```
# Estimate the parameters for linear regression model
biden_lm <- lm(biden ~ age + female + educ, data = biden_data)
tidy(biden_lm)
```

```
##          term estimate std.error statistic  p.value
## 1 (Intercept)  68.6210    3.5960     19.08 4.34e-74
## 2         age   0.0419    0.0325      1.29 1.98e-01
## 3      female   6.1961    1.0967      5.65 1.86e-08
## 4        educ  -0.8887    0.2247     -3.96 7.94e-05
```

Question 1 First, we examine the model for any unusual and/or influential observations. Each observation was identified with a numerical ID from 1 through 1807 in order to more easily identify potential outliers (accomplished during the readin process). Observations were assessed for outlying leverage, discrepancy and influence effects on the final regression model.

```
# calculate cutoff value for Influence measure
cutoff <- 4 / (nrow(biden_data) - (length(coef(biden_lm)) - 1) - 1)

# flag values that violate influence, discrepancy and leverage measures (1 = violated, 0 = normal)
biden_augment <- biden_data %>%
  mutate(hat = hatvalues(biden_lm),
         student = rstudent(biden_lm),
         cooksd = cooks.distance(biden_lm)) %>%
  mutate(flagcd = ifelse(cooksd > cutoff, 1,0),
         flaglev = ifelse(hat > 2 * mean(hat), 1,0),
         flagstd = ifelse(abs(student) > 2, 1,0))
```

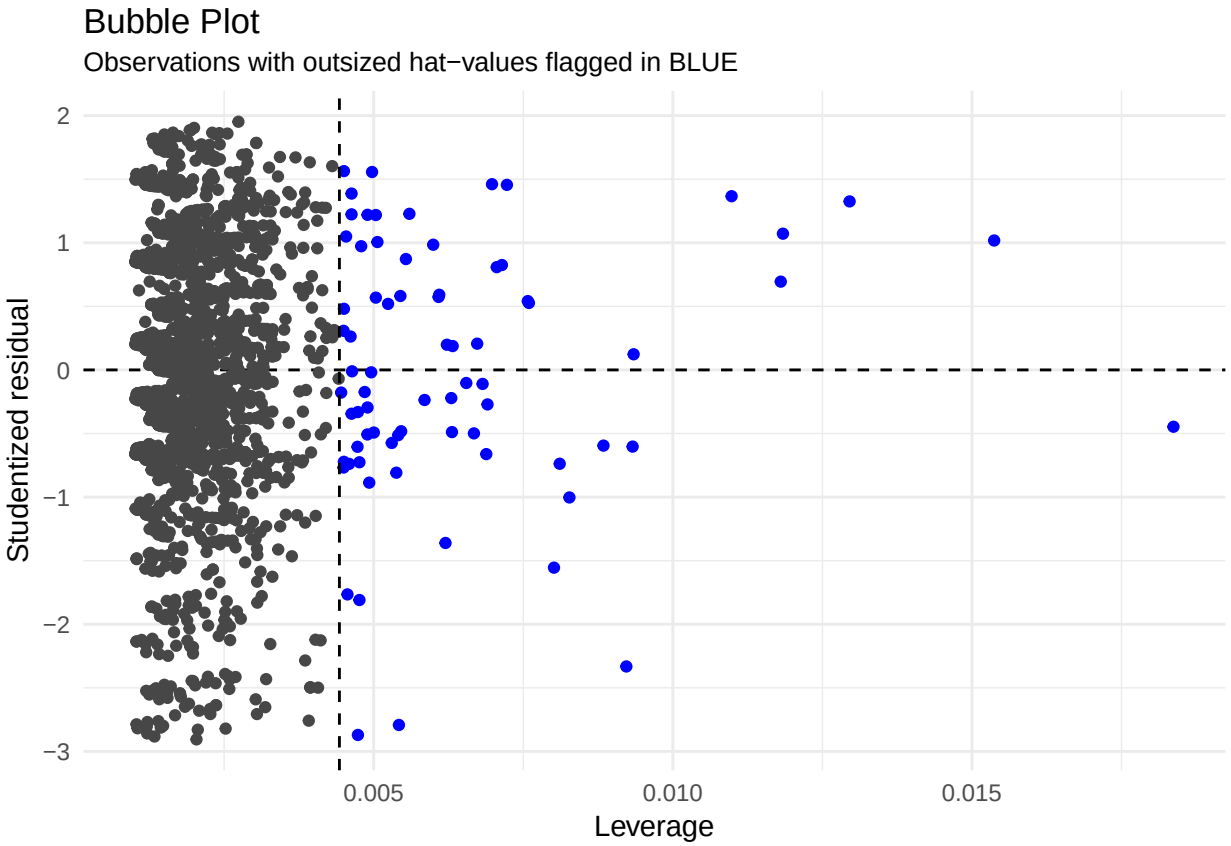
Leverage Hat-values give us a measure of the influence that a given observation may have on the coefficient estimates in the regression model. Observations that are far away from the bulk of observations will have higher calculated hat-values.

Hat-values were calculated to assess each observation's leverage on the biden_lm regression model. A total of 74 observations were larger than 2 times the mean hat value for the entire dataset and may have undue leverage on the final regression model. Observations were plotted; values that lay outide of the normal leverage range were flagged in blue and lay to the right of the vertical dashed line (x = 2 * mean(hat)).

```
biden_augment %>%
  filter(hat > 2 * mean(hat))
```

```
## # A tibble: 74 × 13
##    biden female   age  educ   dem   rep    ID     hat student   cooksd
##    <int>  <int> <int> <int> <int> <int> <int>   <dbl>   <dbl>    <dbl>
## 1     70      0    80    17     0     0    48 0.00504   0.569 4.09e-04
## 2     70      1    44     7     1     0   100 0.00496  -0.019 4.49e-07
## 3    100      1    64     1     1     0   151 0.01537   1.018 4.04e-03
## 4    100      1    76     3     1     0   250 0.01184   1.071 3.44e-03
## 5     60      1    84    16     0     0   253 0.00446  -0.178 3.55e-05
## 6     60      1    63     4     0     0   274 0.00933  -0.603 8.56e-04
## 7     85      0    18     8     1     0   282 0.00600   0.985 1.46e-03
## 8     70      0    79     9     1     0   289 0.00461   0.263 8.00e-05
## 9     50      1    22     9     0     0   296 0.00450  -0.768 6.66e-04
## 10    50      1    23     8     0     0   318 0.00538  -0.808 8.83e-04
## # ... with 64 more rows, and 3 more variables: flagcd <dbl>,
## #   flaglev <dbl>, flagstd <dbl>
```

```
mhat <- mean(biden_augment$hat)

ggplot(biden_augment, aes(hat, student)) +
  geom_hline(yintercept = 0, linetype = 2) +
  geom_point(aes(color = factor(flaglev)), shape = 19) +
  scale_size_continuous(range = c(1, 20)) +
  scale_color_manual(values = c("gray28", "blue")) +
  geom_vline(xintercept = 2*mhat, linetype = 2) +
  labs(x = "Leverage",
       y = "Studentized residual",
       title = "Bubble Plot",
       subtitle = "Observations with outsized hat-values flagged in BLUE") +
  theme(legend.position = "none")
```



Discepancy Discrepancy is commonly measured using studentized residuals, the fraction of a given observation's residual over its estimated standard deviation. Studentized residuals account for the variation in the standard deviation of predicted values' residuals, thus giving us a "scaled" version of the residuals that help us more accurately identify outlier observations. Roughly 95% of studentized residuals values fall within the range [-2,2].

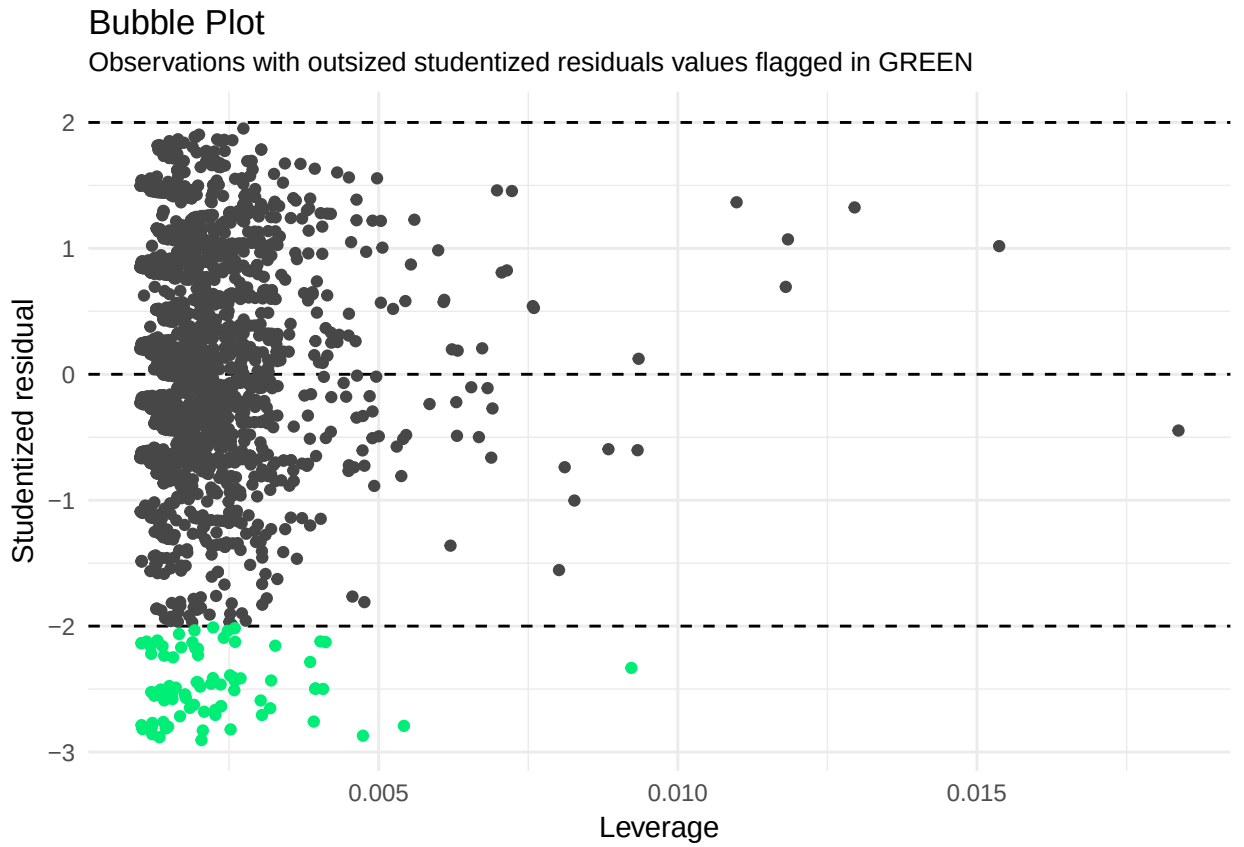
Studentized residuals were also calculated for each of the observations in the biden dataset. 82 of the observations fall outside of the [-2,2] range, and may be sufficiently discrepant from the remaining observations to unduly skew the regression model. Observations were plotted; values that lay outide of the 95% range of observations were flagged in green. Interestingly, all of the observations in green have negative studentized residuals, indicating a distinct pattern to the error in the regression model vis-a-vis these high discrepancy observations.

```
biden_augment %>%
  filter(abs(student) > 2)
```

```
## # A tibble: 82 × 13
##    biden female   age  educ   dem   rep    ID     hat student  cooksd
##    <int>  <int> <int> <int> <int> <int> <int>   <dbl>   <dbl>   <dbl>
## 1      0      1    70    12     0     1    26 0.00204   -2.91 0.00429
## 2      0      0    45    12     0     1    32 0.00142   -2.59 0.00237
## 3      0      0    40    14     0     0    71 0.00136   -2.50 0.00213
## 4     15      0    62     8     0     1   127 0.00411   -2.13 0.00466
## 5     15      1    20    13     0     0   137 0.00260   -2.12 0.00294
## 6      0      1    38    14     1     0   195 0.00122   -2.77 0.00233
## 7      0      0    34    12     0     0   327 0.00178   -2.57 0.00293
## 8      0      0    21    13     0     1   344 0.00259   -2.51 0.00407
```

```
## 9     15      1    29    12     0     1   376 0.00198   -2.18 0.00235
## 10     0      0    36    13     0     1   379 0.00149   -2.53 0.00239
## # ... with 72 more rows, and 3 more variables: flagcd <dbl>,
## #   flaglev <dbl>, flagstd <dbl>
```

```
ggplot(biden_augment, aes(hat, student)) +
  geom_hline(yintercept = 0, linetype = 2) +
  geom_hline(yintercept = 2, linetype = 2) +
  geom_hline(yintercept = -2, linetype = 2) +
  geom_point(aes(color = factor(flagstd)), shape = 19) +
  scale_size_continuous(range = c(1, 20)) +
  scale_color_manual(values = c("gray28", "springgreen2")) +
  labs(x = "Leverage",
       y = "Studentized residual",
       title = "Bubble Plot",
       subtitle = "Observations with outsized studentized residuals values flagged in GREEN") +
  theme(legend.position = "none")
```



Influence Influence accounts for an observation's leverage and discrepancy. The cook's D value is a measurement of influence. Observations with undue influence are commonly identified as those observations with Cook's D values that meet the following criteria:

$$D_i > \frac{4}{n-k-1}$$

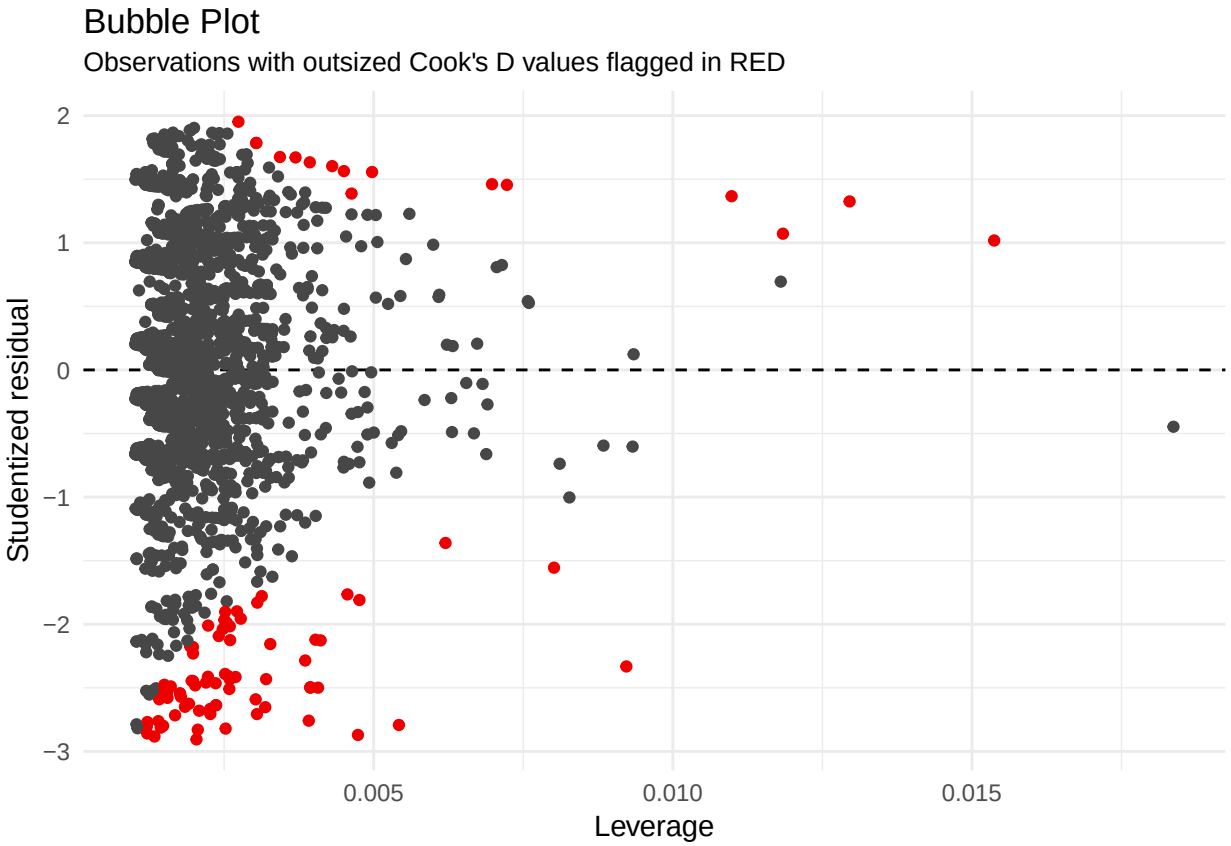
A total of 90 observations had Cook's D values that exceeded this criteria. These observations may have outsized influence over the coefficient estimates of this regression model. Observations were plotted; values that lay outide of the normal Cook's D value range were flagged in red. AGain, we notice that the majority

of observations that fall outside of the normal Cook's D value range are in the lower left hand corner of the graph, low leverage, but high discrepancy observations.

```
biden_augment %>%
  filter(cooksd > 4 / (nrow(.) - (length(coef(biden_lm)) - 1) - 1))
```

```
## # A tibble: 90 × 13
##    biden female   age  educ   dem   rep    ID     hat student  cooksd
##    <int>  <int> <int> <int> <int> <int> <int>   <dbl>   <dbl>   <dbl>
## 1      0      1    70    12     0     1    26 0.00204   -2.91 0.00429
## 2      0      0    45    12     0     1    32 0.00142   -2.59 0.00237
## 3     15      0    62     8     0     1   127 0.00411   -2.13 0.00466
## 4     15      1    20    13     0     0   137 0.00260   -2.12 0.00294
## 5    100      1    64     1     1     0   151 0.01537    1.02 0.00404
## 6    100      0    19    12     0     0   154 0.00304    1.78 0.00242
## 7    100      0    19    12     1     0   185 0.00304    1.78 0.00242
## 8      0      1    38    14     1     0   195 0.00122   -2.77 0.00233
## 9    100      1    76     3     1     0   250 0.01184    1.07 0.00344
## 10     0      0    34    12     0     0   327 0.00178   -2.57 0.00293
## # ... with 80 more rows, and 3 more variables: flagcd <dbl>,
## #   flaglev <dbl>, flagstd <dbl>
```

```
ggplot(biden_augment, aes(hat, student)) +
  geom_hline(yintercept = 0, linetype = 2) +
  geom_point(aes(color = factor(flagcd)), shape = 19) +
  scale_size_continuous(range = c(1, 20)) +
  scale_color_manual(values = c("gray28", "red2")) +
  labs(x = "Leverage",
       y = "Studentized residual",
       title = "Bubble Plot",
       subtitle = "Observations with outsized Cook's D values flagged in RED") +
  theme(legend.position = "none")
```



Next Steps Our examination of potential outliers identified several observations that may have outsized influence on our regression model. To adjust for these effects, we could selectively remove observations that appear to have undue influence over the coefficient estimates to see if their removal alters our regression model in a significant way.

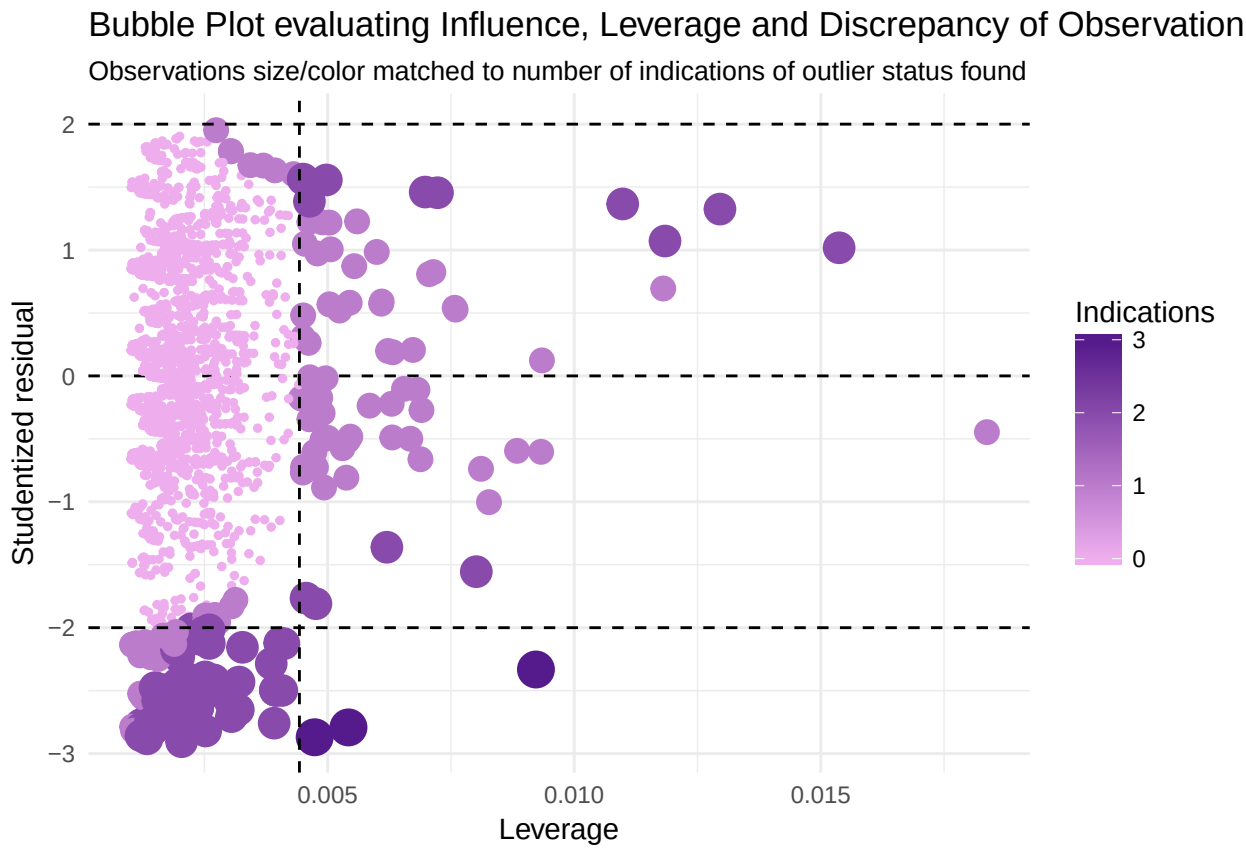
One way to identify such observations is to use graphical methods. Below, we've plotted each observation on the x and y axes by its hat-value (x) and studentized residual (y), adjusting the size and color of the markers by the number of indications of "outlier" status observed, where "outlier" status can be indicated by the leverage (hat-value), discrepancy (studentized residuals), or influence (Cook's D) measures discussed above. Again, we observe that the majority of outliers are low leverage, high discrepancy observations in the lower left hand side of the plot.

Finally, we filter the dataset to return only those 167 values with unusual influence on the coefficient estimates. To test whether or not the influence is meaningful, we could remove these potential outliers stepwise to see which observations most affect our coefficient estimates. Likely, we would start by removing those observations with the most indications of outlier status (Indcations = 3), then progressively omit more observations as we attempt to determine the effect of these observations on the estimated regression model.

```
biden_augment <- biden_augment %>%
  mutate(Indications = flagcd + flaglev + flagstd)


ggplot(biden_augment, aes(hat, student)) +
  geom_hline(yintercept = 0, linetype = 2) +
  geom_point(aes(size = Indications, color = Indications), shape = 19) +
  geom_hline(yintercept = 2, linetype = 2) +
  geom_hline(yintercept = -2, linetype = 2) +
  geom_vline(xintercept = 2*mhat, linetype = 2) +
```

```
  scale_size_continuous(range = c(1, 6)) +
  scale_colour_gradient(low='plum2', high='purple4') +
  labs(x = "Leverage",
       y = "Studentized residual",
       title = "Bubble Plot evaluating Influence, Leverage and Discrepancy of Observations",
       subtitle = "Observations size/color matched to number of indications of outlier status found") +
  theme(legend.position = "right") +
  scale_size(guide = "none")
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```




```
biden_problem <- biden_augment %>%
  filter(hat >= 2 * mean(hat) |
           abs(student) > 2 |
           cooksd > cutoff)

biden_problem
```

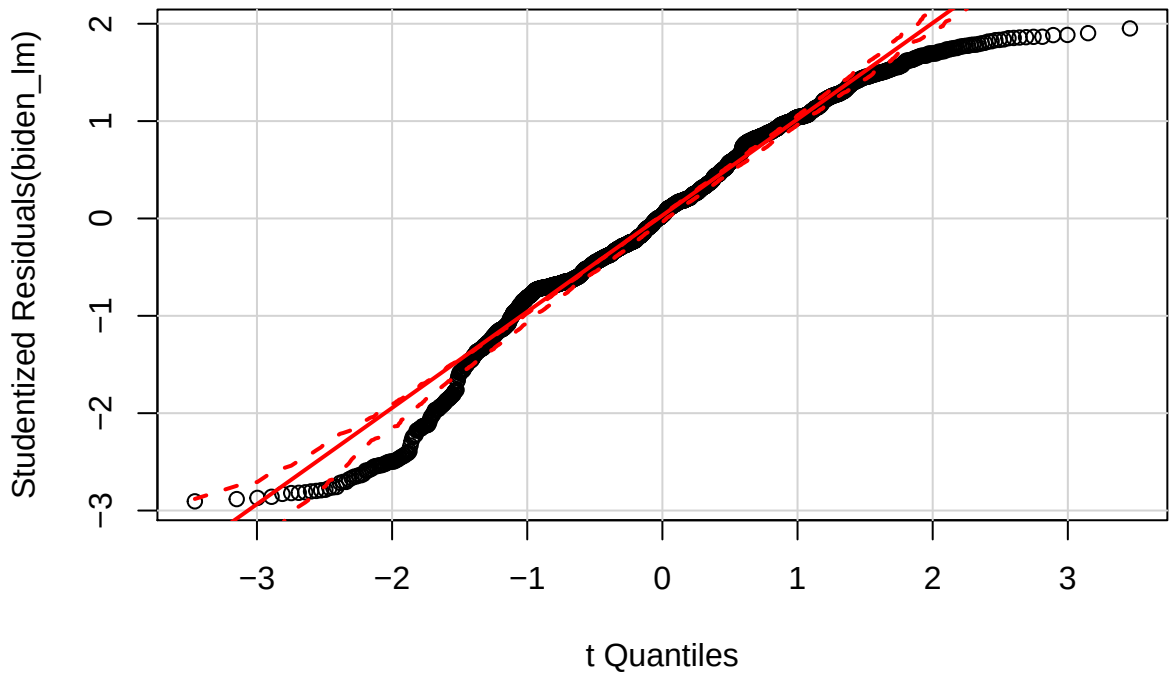
```
## # A tibble: 167 × 14
##    biden female   age  educ   dem   rep    ID     hat student   cooksd
##    <int>  <int> <int> <int> <int> <int> <int>   <dbl>   <dbl>    <dbl>
## 1      0      1    70    12     0     1    26 0.00204  -2.906 4.29e-03
## 2      0      0    45    12     0     1    32 0.00142  -2.590 2.37e-03
## 3     70      0    80    17     0     0    48 0.00504   0.569 4.09e-04
## 4      0      0    40    14     0     0    71 0.00136  -2.503 2.13e-03
```

```
## 5     70      1    44     7     1     0   100 0.00496  -0.019 4.49e-07
## 6     15      0    62     8     0     1   127 0.00411  -2.127 4.66e-03
## 7     15      1    20    13     0     0   137 0.00260  -2.125 2.94e-03
## 8    100      1    64     1     1     0   151 0.01537   1.018 4.04e-03
## 9    100      0    19    12     0     0   154 0.00304   1.785 2.42e-03
## 10   100      0    19    12     1     0   185 0.00304   1.785 2.42e-03
## # ... with 157 more rows, and 4 more variables: flagcd <dbl>,
## #   flaglev <dbl>, flagstd <dbl>, Indications <dbl>
```

Question 2 Next, the data was examiend for non-normally distributed errors. A quantile-projection plot was created to investigate whether data violate our assumption of normal distribution. Because a considerable number of observations fall outside the 95% confidence interval range that assumes errors are normallly distributed, we must conclude that the data is not normally distributed.

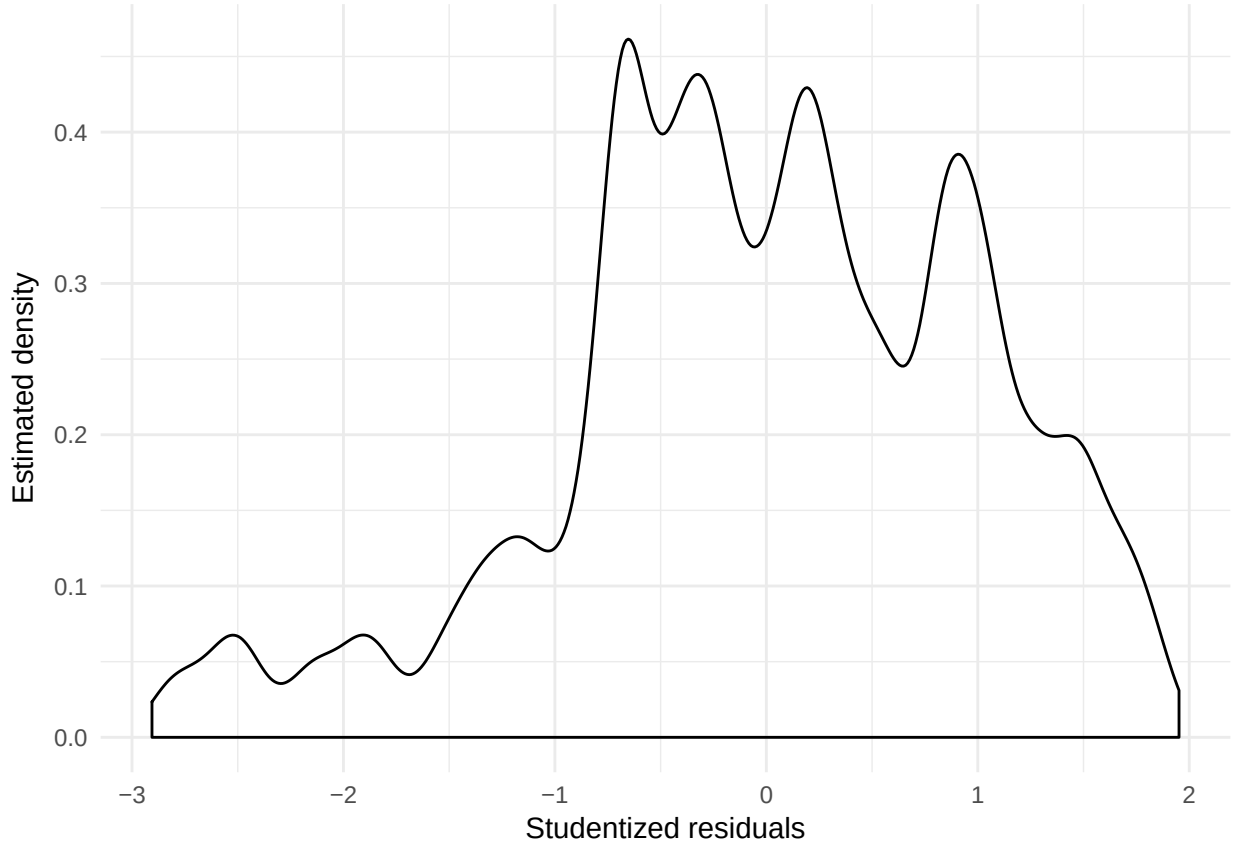
There are ways to correct for the non-normal distribution of the data. Commonly, we would attempt to transform some of the independent and dependent variables to investigate whether this results in a more closely normally distributed sets of observations. By plotting the resultant quantile-projection plot for each transformed dataset, we might determine the transformations that will provide the most normal-distribution of data. Because OLS assumes that the data are normally distributed, the closer we get to a normal distribution of observations, the more accurate our interpreations of the regression model become.

```
car::qqPlot(biden_lm)
```




```
augment(biden_lm, biden_data) %>%
  mutate(.student = rstudent(biden_lm)) %>%
  ggplot(aes(.student)) +
  geom_density(adjust = .5) +
  labs(x = "Studentized residuals",
```

```
       y = "Estimated density")
```



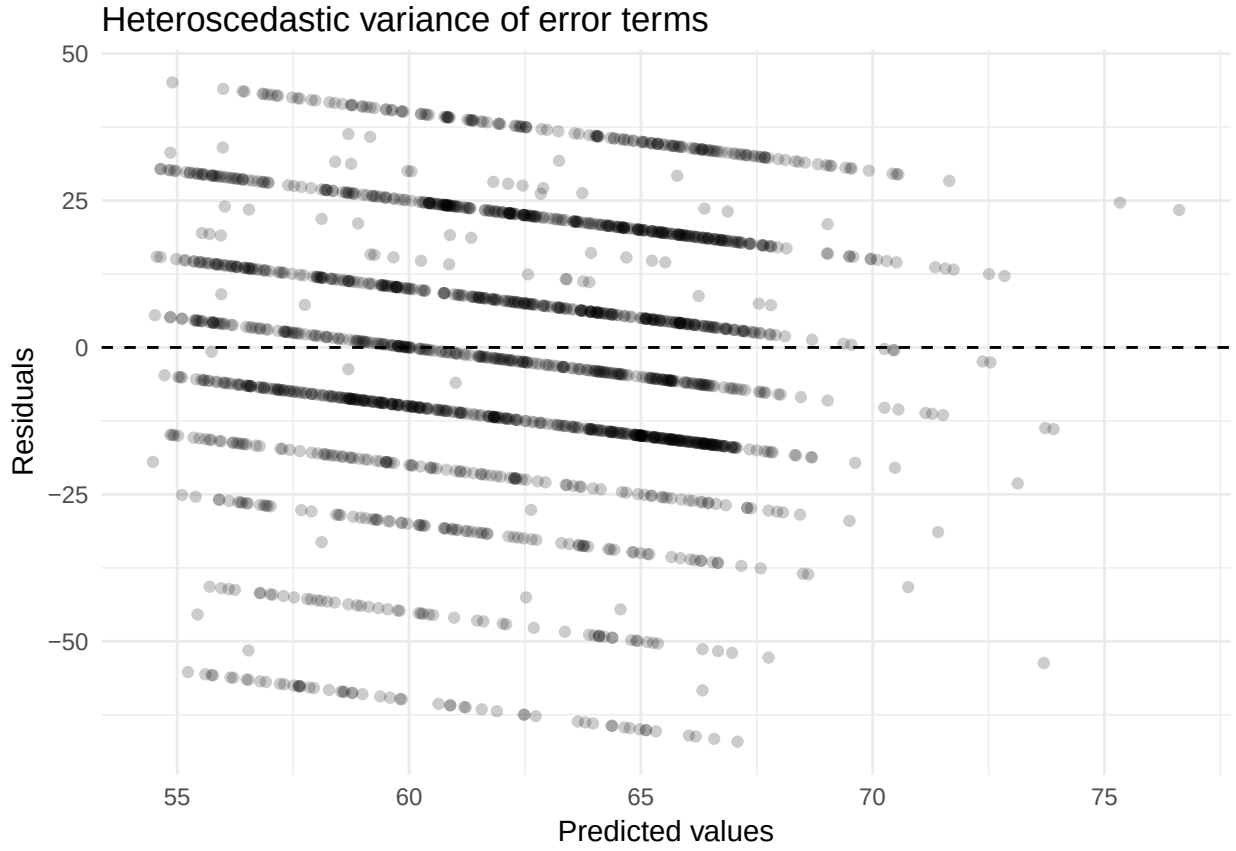
Question 3 OLS assumes that error terms have a constant variance. However, this is not always the case. Now, we examine the data for non-constant variance in the data, aka heteroscedasticity. We investigate the data using first a graphical method and secondly the Breusch-Pagan test.

The difference between the predicted values from the regression model and the residuals (difference between the predicted and actual values) is a measure of error that can be used to detect heteroscedaticity. The plot of predicted values and residuals from the biden regression model is provided below. As the resultant plot clearly indicates, there is a distinct pattern to the error: errors are not distributed evenly around 0 and thus indicate the presence of heteroscedasticity.

We verify this using the Bresuch-Pagan test. Results produce a significant p-value (less than 0.05), which indicates that the data is indeed, heteroscedastic. Because the errors do not have a constant variance, this means that our regression model will have biased estimates for the coefficient estimates, which will in turn result in skewed predictions from regression model.

```
biden_data %>%
  add_predictions(biden_lm) %>%
  add_residuals(biden_lm) %>%
  ggplot(aes(pred,resid)) +
  geom_point(alpha = .2) +
  geom_hline(yintercept = 0, linetype = 2) +
  geom_quantile(method = 'rqss', lambda = 5, quantiles = c(0.05, 0.95)) +
  labs(title = "Heteroscedastic variance of error terms",
       x = "Predicted values",
       y = "Residuals")
```

```
## Loading required package: SparseM
##
## Attaching package: 'SparseM'
## The following object is masked from 'package:base':
##
##     backsolve
## Smoothing formula not specified. Using: y ~ qss(x, lambda = 5)
## Warning: Computation failed in `stat_quantile()`:
## invalid class "dsparseModelMatrix" object: superclass "replValueSp" not defined in the environment o
```




```
bptest(biden_lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  biden_lm
## BP = 20, df = 3, p-value = 5e-05
```

Question 4 Multicollinearity can also affect the accuracy of a regression model, as it can mask interaction effects between multiple variables. A common way to test for this is to use the variance inflaction factor, which measures the variance of a coefficient estimate when fitted to a full model veresus its own model. Since we have three explanatory variables, we will have to calculate the VIF for each estimated coefficient. This is accomplished by the code below. As a general rule of thumb, because the VIF values are less than 10, we can reasonbly assume that there is no potential multicollinearity in our biden_lm model.

```
car::vif(biden_lm)
```

```
##    age female   educ
##   1.01   1.00   1.01
```

PART TWO: Interaction Terms

Instant Effect Function

```
# function to get point estimates and standard errors
# model - lm object
# mod_var - name of moderating variable in the interaction
instant_effect <- function(model, mod_var){
  # get interaction term name
  int.name <- names(model$coefficients)[[which(str_detect(names(model$coefficients), ":"))]]

  marg_var <- str_split(int.name, ":")[[1]][[which(str_split(int.name, ":")[[1]] != mod_var)]]

  # store coefficients and covariance matrix
  beta.hat <- coef(model)
  cov <- vcov(model)

  # possible set of values for mod_var
  if(class(model)[[1]] == "lm"){
    z <- seq(min(model$model[[mod_var]]), max(model$model[[mod_var]]))
  } else {
    z <- seq(min(model$data[[mod_var]]), max(model$data[[mod_var]]))
  }

  # calculate instantaneous effect
  dy.dx <- beta.hat[[marg_var]] + beta.hat[[int.name]] * z

  # calculate standard errors for instantaeous effect
  se.dy.dx <- sqrt(cov[marg_var, marg_var] +
                     z^2 * cov[int.name, int.name] +
                     2 * z * cov[marg_var, int.name])

  # combine into data frame
  data_frame(z = z,
             dy.dx = dy.dx,
             se = se.dy.dx)
}
```

For this exercise we consider the following functional form:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2$$

where Y is the Joe Biden feeling thermometer, X1 is age and X2 is education

We estimate the parameters and standard errors for this linear regression model:

```
# Estimate the parameters for linear regression model
biden_lmint <- lm(biden ~ age * educ, data = biden_data)
tidy(biden_lmint)
```

```
##          term estimate std.error statistic  p.value
## 1 (Intercept)   38.374    9.5636      4.01 6.25e-05
```

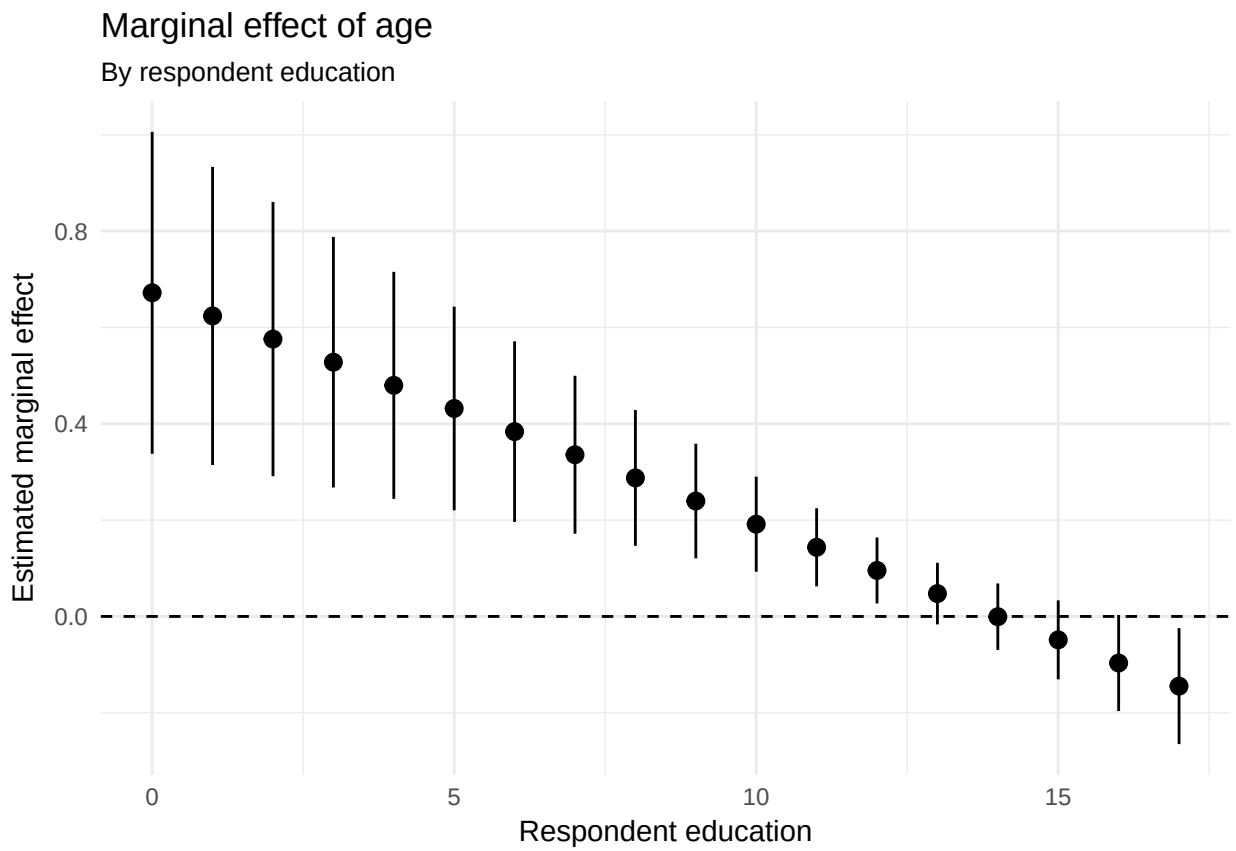
```
## 2         age    0.672    0.1705      3.94 8.43e-05
## 3        educ    1.657    0.7140      2.32 2.04e-02
## 4    age:educ   -0.048    0.0129     -3.72 2.03e-04
```

Question 1 First, we evalute the marginal effect of age on Joe Biden thermometer rating, conditional on education. We do this firstly using a graphical approach, then secondly using the Wald Test (linearHypothesis function).

In the graphical approach below, we have plotted the estimated marginal effect of age by respondent education. This graph tells us that the marginal effect of age on Biden rating steadily decreases as education increases, eventually becoming negative for respondents that reported education levels of 14 or higher. We can clearly see that education has a significant marginal effect on age's impact on Biden Thermometer rating, especially at lower levels of education.

The significance of this finding is confirmed using the results of the Wald Test (linearHypothesis function from the car package). We can see that the p-value for the fit of the unrestricted model compared to the restricted model is less than 0.05: this tells us the marginal effect of education on age's impact on Biden Thermometer rating is significant.

```
# Plot the marginal effect of age on biden thermometer rating, by education
instant_effect(biden_lmint, "educ") %>%
  ggplot(aes(z, dy.dx,
             ymin = dy.dx - 1.96 * se,
             ymax = dy.dx + 1.96 * se)) +
  geom_pointrange() +
  geom_hline(yintercept = 0, linetype = 2) +
  labs(title = "Marginal effect of age",
       subtitle = "By respondent education",
       x = "Respondent education",
       y = "Estimated marginal effect")
```




```
# Run Wald test to check for variable significance of education
linearHypothesis(biden_lmint, "educ + age:educ")
```

```
## Linear hypothesis test
##
## Hypothesis:
## educ  + age:educ = 0
##
## Model 1: restricted model
## Model 2: biden ~ age * educ
##
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1   1804 979537
## 2   1803 976688  1      2849 5.26  0.022 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

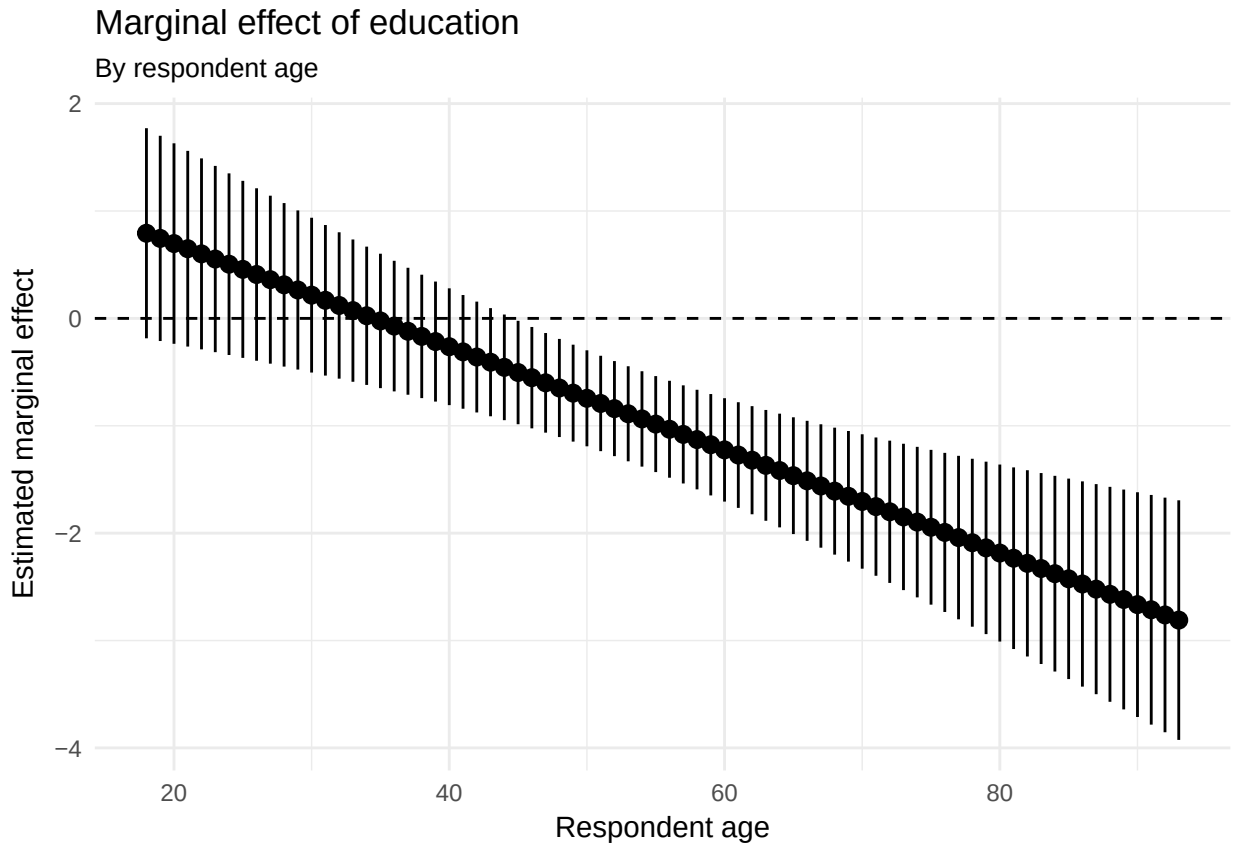
Question 2 Secondly, we evalute the marginal effect of education on Joe Biden thermometer rating, conditional on age. Similarly, we do this firstly using a graphical approach, then secondly using the Wald Test (linearHypothesis function).

In the graphical approach below, we have plotted the estimated marginal effect of education by respondent age, the inverse of the context above. This graph tells us that the marginal effect of education on biden rating steadily decreases as age increases, eventually becoming negative for respondents that reported age levels of 34 or higher. Again, we can clearly see that age has a significant effect on education's impact on Biden Thermometer rating for certain ranges of ages.

The significance of this finding is verified using the results of the Wald Test (linearHypothesis function from

the car package). Because the Wald Test below has produced a p-value less than 0.05, we confirm that the marginal effect from age on education's impact on Biden Thermometer rating is significant.

```
instant_effect(biden_lmint, "age") %>%
  ggplot(aes(z, dy.dx,
             ymin = dy.dx - 1.96 * se,
             ymax = dy.dx + 1.96 * se)) +
  geom_pointrange() +
  geom_hline(yintercept = 0, linetype = 2) +
  labs(title = "Marginal effect of education",
       subtitle = "By respondent age",
       x = "Respondent age",
       y = "Estimated marginal effect")
```




```
# Run Wald test to check for variable significance of education
linearHypothesis(biden_lmint, "age + age:educ")
```

```
## Linear hypothesis test
##
## Hypothesis:
## age  + age:educ = 0
##
## Model 1: restricted model
## Model 2: biden ~ age * educ
##
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1   1804 985149
```

```
## 2   1803 976688  1      8461 15.6  8e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

PART THREE: Missing Data

The data is re-imported in full; missing records were not omitted.

First, we test the data for multivariate normality, using both the Mardia and Henze-Zirkler tests found in the MVN package. Both tests find that the data are not multivariate normal. We plot the q-q and histogram plots for the variables of interest and visually to examine the distribution of observations for each variable. Results suggest that we may be able to transform age or education to get the distribution to more closely resemble normal.

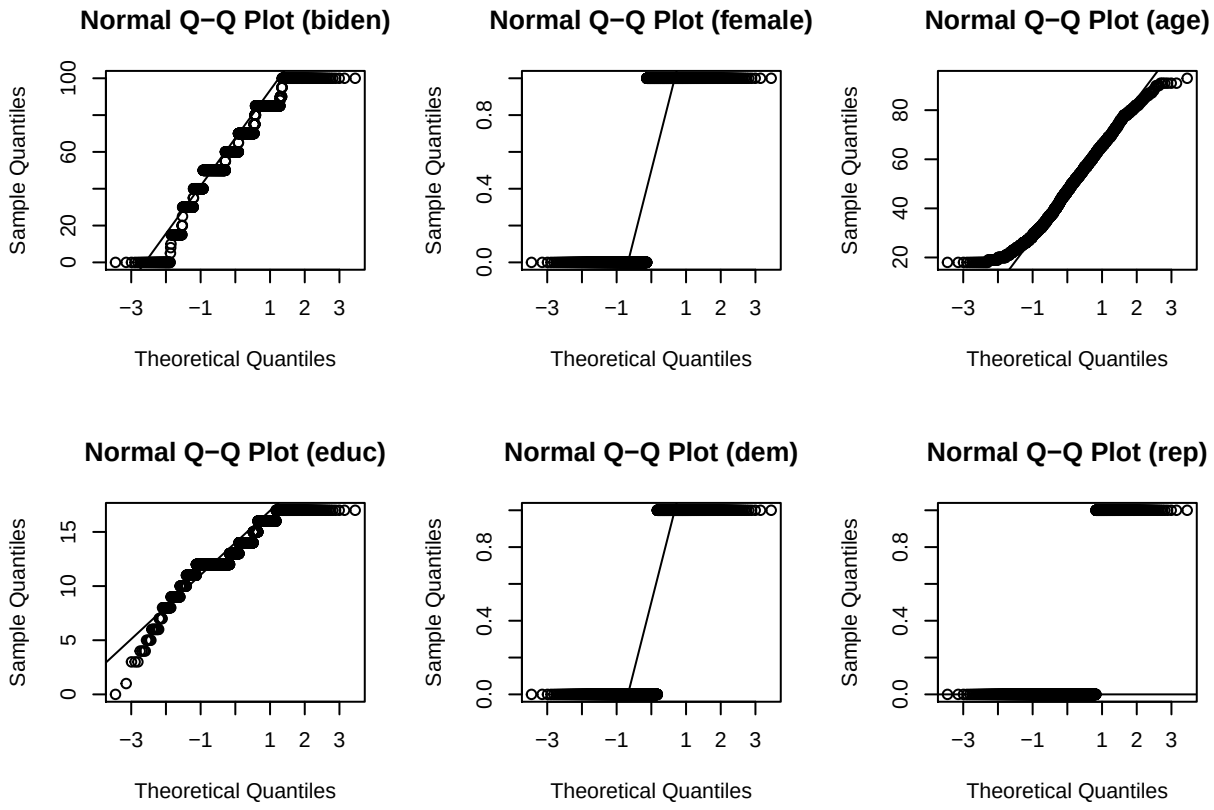
```
# get rid of ID column, unnecessary for Part Three
biden_p3 <- biden_data %>%
  select(-ID)

# MVN tests: Mardia and Henze-Zirkler
mardiaTest(biden_p3, qqplot = FALSE)
```

```
##    Mardia's Multivariate Normality Test
## ---------------------------------------
##    data : biden_p3
##
##    g1p            : 4.22
##    chi.skew       : 1270
##    p.value.skew   : 7.96e-229
##
##    g2p            : 44.7
##    z.kurtosis     : -7.14
##    p.value.kurt   : 9.02e-13
##
##    chi.small.skew : 1273
##    p.value.small  : 2.17e-229
##
##    Result          : Data are not multivariate normal.
## ---------------------------------------
```

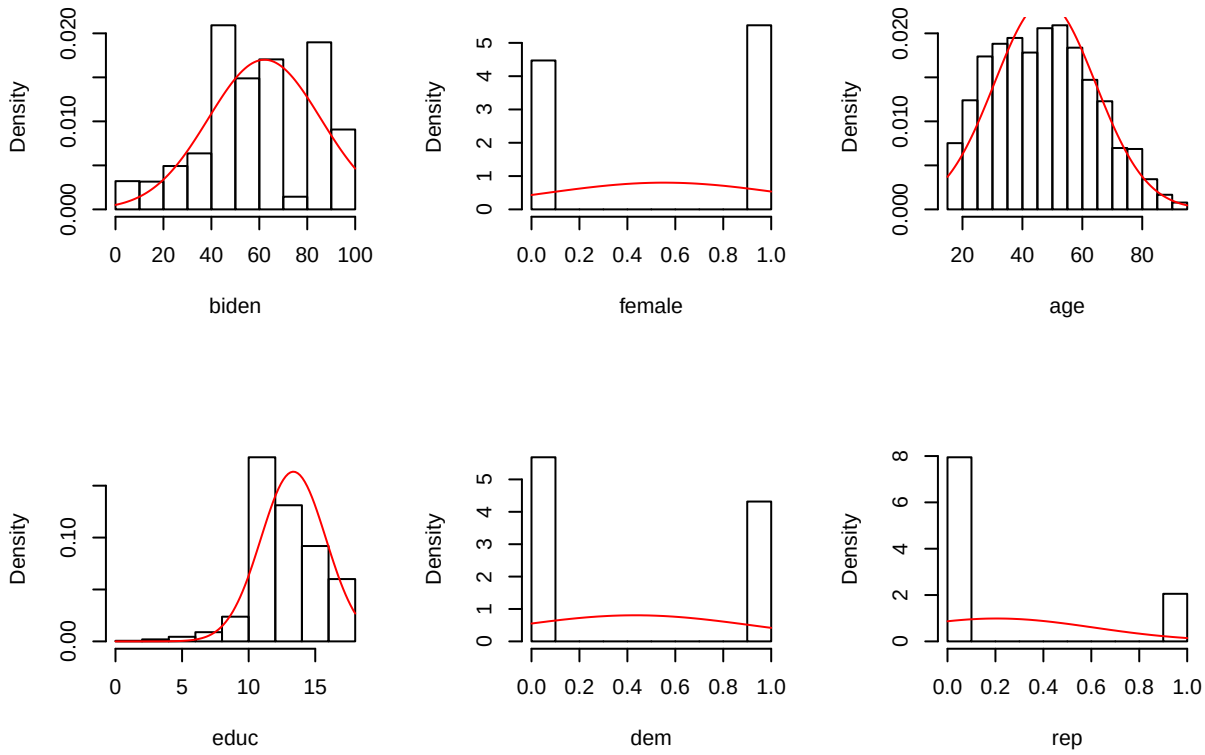
```
hzTest(biden_p3, qqplot = FALSE)
```

```
##   Henze-Zirkler's Multivariate Normality Test
## ---------------------------------------------
##   data : biden_p3
##
##   HZ      : 19
##   p-value : 0
##
##   Result  : Data are not multivariate normal.
## ---------------------------------------------
```

```
# Plot Q-Q and histogram plots
uniPlot(biden_p3, type = "qqplot") # creates univariate Q-Q plots
```

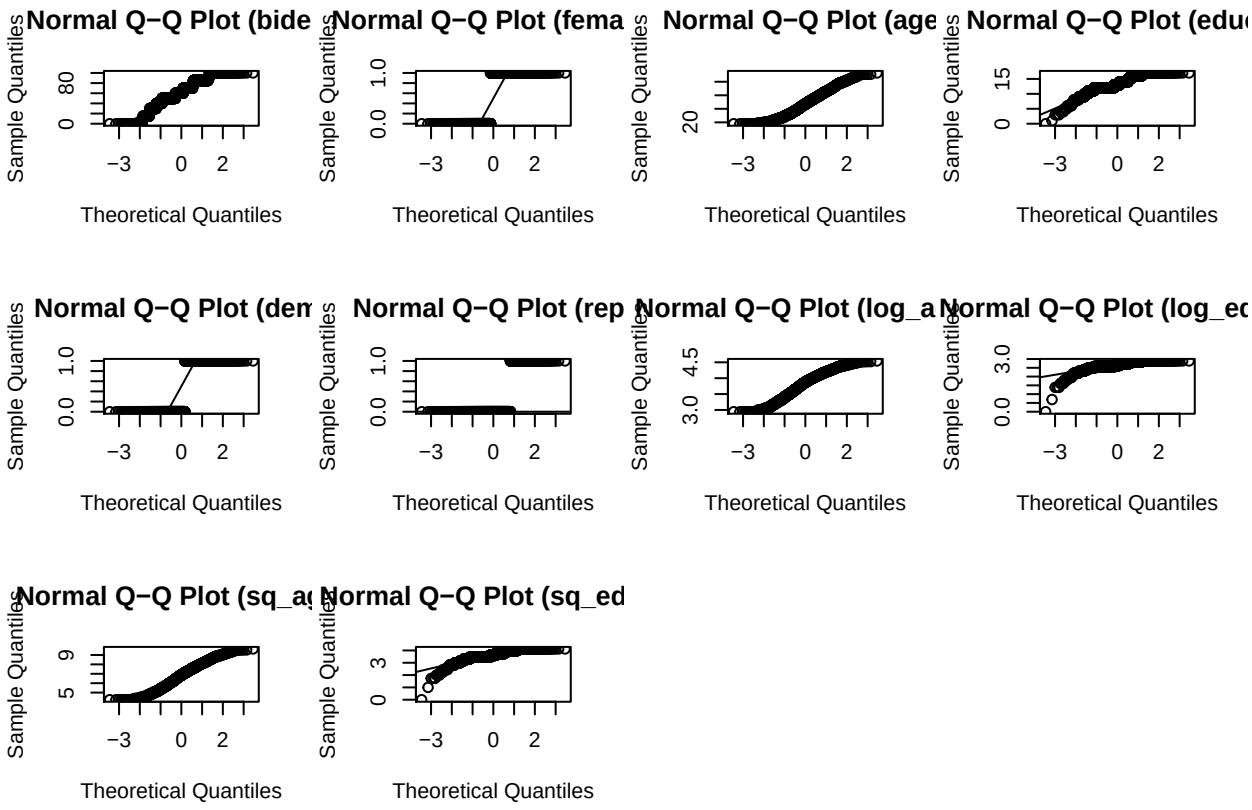



```
uniPlot(biden_p3, type = "histogram") # creates univariate histograms
```



We try taking the log and the square root of age and education variables, then replot the q-q and histogram plots. The square root transformations appear to adjust the distribution best. Unfortunately, when the square-root transformed model was tested for Multivariate Normality using the Mardia and Henze-Zirkler tests, data were still found to violate multivariate normality. Since the transformations fail to restore multivariate normality, no transformation will use in the multiple imputation analysis to follow.

```
biden_test <- biden_p3 %>%
  mutate(log_age = log(age +1),
         log_educ = log(educ+1),
         sq_age = sqrt(age),
         sq_educ = sqrt(educ))

uniPlot(biden_test, type = "qqplot") # creates univariate Q-Q plots
uniPlot(biden_test, type = "histogram") # creates univariate histograms
```




```
biden_sq <- biden_test %>%
  select(-log_age, -log_educ, -age, -educ)

biden_sq
```

```
## # A tibble: 1,807 × 6
##    biden female   dem   rep sq_age sq_educ
##    <int>  <int> <int> <int>  <dbl>   <dbl>
## 1     90      0     1     0   4.36    3.46
## 2     70      1     1     0   7.14    3.74
## 3     60      0     0     0   5.20    3.74
## 4     50      1     1     0   6.56    3.74
## 5     60      1     0     1   6.16    3.74
## 6     85      1     1     0   5.20    4.00
## 7     60      1     0     0   5.29    3.46
## 8     50      0     1     0   5.57    3.87
## 9     50      1     0     0   5.66    3.61
## 10    70      0     1     0   7.14    3.74
## # ... with 1,797 more rows
```

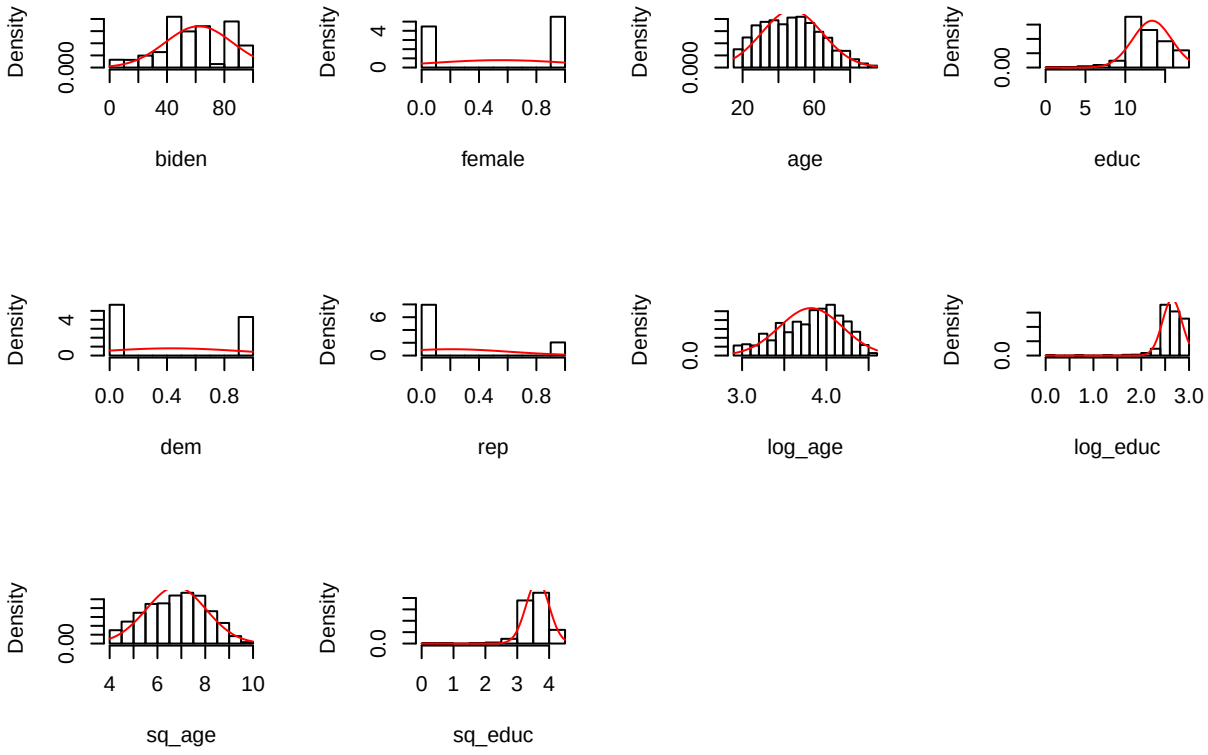
```
# MVN tests: Mardia and Henze-Zirkler
mardiaTest(biden_sq, qqplot = FALSE)
```

```
##    Mardia's Multivariate Normality Test
## ---------------------------------------
##    data : biden_sq
##
```

```
##    g1p            : 6.45
##    chi.skew       : 1944
##    p.value.skew   : 0
##
##    g2p            : 52.2
##    z.kurtosis     : 9.2
##    p.value.kurt   : 0
##
##    chi.small.skew : 1948
##    p.value.small  : 0
##
##    Result          : Data are not multivariate normal.
## ---------------------------------------
```

```
hzTest(biden_sq, qqplot = FALSE)
```

```
##   Henze-Zirkler's Multivariate Normality Test
## ---------------------------------------------
##   data : biden_sq
##
##   HZ      : 19.7
##   p-value : 0
##
##   Result  : Data are not multivariate normal.
## ---------------------------------------------
```



We use the Amelia package to create 5 imputations of the data, then estimate the parameters and calculate the standard errors for each imputed dataset. Per the multiple imputation method, the final parameters

and standard errors reported in the table below (statistics with the label ".mi") are simply the mean of the given statistic from each of the 5 imputed datasets. We compare this imputed model with the original linear regression model (biden_lm) generated from the dataset with missing records removed. The sign and magnitude of the coefficients and standard errors are the same across the imputed and original models. There are some differences: the standard errors are slightly lower for the imputed model and the coefficients differ slightly.

```
library(Amelia)
```

```
## Warning: package 'Amelia' was built under R version 3.3.3
```

```
## Loading required package: Rcpp
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.4, built: 2015-12-05)
## ## Copyright (C) 2005-2017 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
biden_full <- read_csv("data/biden.csv")
```

```
## Parsed with column specification:
## cols(
##   biden = col_integer(),
##   female = col_integer(),
##   age = col_integer(),
##   educ = col_integer(),
##   dem = col_integer(),
##   rep = col_integer()
## )
```

```
biden_full.out <- amelia(as.data.frame(biden_full), m = 5)
```

```
## -- Imputation 1 --
##
##   1  2  3  4  5  6
##
## -- Imputation 2 --
##
##   1  2  3  4  5  6
##
## -- Imputation 3 --
##
##   1  2  3  4  5  6
##
## -- Imputation 4 --
##
##   1  2  3  4  5
##
## -- Imputation 5 --
##
##   1  2  3  4  5  6
```

```
models_imp <- data_frame(data = biden_full.out$imputations) %>%
  mutate(model = map(data, ~ lm(biden ~ age +
                                  female + educ,
                                data = .x)),
```

```
        coef = map(model, tidy)) %>%
  unnest(coef, .id = "id")
models_imp
```

```
## # A tibble: 20 × 6
##       id        term estimate std.error statistic   p.value
##    <chr>       <chr>    <dbl>     <dbl>     <dbl>     <dbl>
## 1   imp1 (Intercept) 61.83369    2.9970    20.632  5.77e-87
## 2   imp1         age  0.06305    0.0278     2.267  2.35e-02
## 3   imp1      female  5.56240    0.9675     5.749  1.01e-08
## 4   imp1        educ -0.46651    0.1863    -2.504  1.24e-02
## 5   imp2 (Intercept) 66.87247    2.9974    22.311 4.74e-100
## 6   imp2         age  0.00572    0.0278     0.206  8.37e-01
## 7   imp2      female  5.88058    0.9688     6.070  1.49e-09
## 8   imp2        educ -0.63381    0.1863    -3.401  6.82e-04
## 9   imp3 (Intercept) 66.80089    3.0277    22.063  4.44e-98
## 10  imp3         age  0.05100    0.0280     1.820  6.89e-02
## 11  imp3      female  5.91164    0.9733     6.074  1.45e-09
## 12  imp3        educ -0.78892    0.1875    -4.208  2.68e-05
## 13  imp4 (Intercept) 65.43896    2.9854    21.920  6.05e-97
## 14  imp4         age  0.03523    0.0277     1.272  2.03e-01
## 15  imp4      female  5.29741    0.9632     5.500  4.21e-08
## 16  imp4        educ -0.61571    0.1853    -3.322  9.07e-04
## 17  imp5 (Intercept) 66.16179    2.9867    22.152  8.69e-99
## 18  imp5         age  0.04624    0.0277     1.670  9.50e-02
## 19  imp5      female  4.93615    0.9637     5.122  3.27e-07
## 20  imp5        educ -0.69513    0.1855    -3.746  1.84e-04
```

```
mi.meld.plus <- function(df_tidy){
  # transform data into appropriate matrix shape
  coef.out <- df_tidy %>%
    select(id:estimate) %>%
    spread(term, estimate) %>%
    select(-id)

  se.out <- df_tidy %>%
    select(id, term, std.error) %>%
    spread(term, std.error) %>%
    select(-id)

  combined.results <- mi.meld(q = coef.out, se = se.out)

  data_frame(term = colnames(combined.results$q.mi),
             estimate.mi = combined.results$q.mi[1, ],
             std.error.mi = combined.results$se.mi[1, ])
}

# compare results
tidy(biden_lm) %>%
  left_join(mi.meld.plus(models_imp)) %>%
  select(-statistic, -p.value)
```

```
## Joining, by = "term"
```

```
##          term estimate std.error estimate.mi std.error.mi
```

```
## 1 (Intercept)  68.6210    3.5960     65.4216       3.7712
## 2         age   0.0419    0.0325      0.0402       0.0366
## 3      female   6.1961    1.0967      5.5176       1.0669
## 4        educ  -0.8887    0.2247     -0.6400       0.2268
```

The imputed model may not differ considerably from the original model (biden_lm) because there is relatively little data missing from the original dataset. Below, we've plotted a graphic illustrating the number of missing data. As the graphic clearly shows, there is very little "missingness" in the data, so the posterior distribution of the data that is used to impute the data should be fairly accurate.

```
missmap(biden_full.out)
```