Replication 2: Abadie (2005)

Hannah Jones

4/14/2021

Propensity Scores using Logit and OLS

I brought in the data already combined using Stata.

First, we estimate a Logit model using up to quadratic covariates, and a second Logit model using up to cubic covariates. We then store the propensity scores (predictions of treatment) in new variables called quadpscore and cubepscore.

```
quad_logit_nsw <- glm(treat ~ age + agesq + educ + educsq +
                    marr + nodegree + black + hisp + re74 +
                    re74sq + re75 + re75sq + u74 + u75,
                  family = binomial(link = "logit"),
                  data = nsw_dw_cpscontrol)

cube_logit_nsw <- glm(treat ~ age + agesq + agecube + educ + educsq + educcube +
                        marr + nodegree + black + hisp + re74 +
                        re74sq + re74cube + re75 + re75sq + re75cube + u74 + u75,
                      family = binomial(link = "logit"),
                      data = nsw_dw_cpscontrol)

logit_nsw_dw_cpscontrol <- nsw_dw_cpscontrol %>%
  mutate(quadpscore = quad_logit_nsw$fitted.values, cubepscore = cube_logit_nsw$fitted.values)
```

Next we take the mean pscore for the treated and untreated observations for each model. Those values are reported below:

```
# mean pscore

logit_quad_sumstat<- logit_nsw_dw_cpscontrol %>%
  group_by(treat)%>%
  summarize(logit_quad_mean = mean(quadpscore),
            logit_quad_max = max(quadpscore),
            logit_quad_min = min(quadpscore))
logit_quad_sumstat
```

```
## # A tibble: 2 x 4
##   treat logit_quad_mean logit_quad_max logit_quad_min
##   <dbl>           <dbl>          <dbl>          <dbl>
## 1     0         0.00691          0.891 0.000000000283
## 2     1         0.402            0.902 0.000737
```

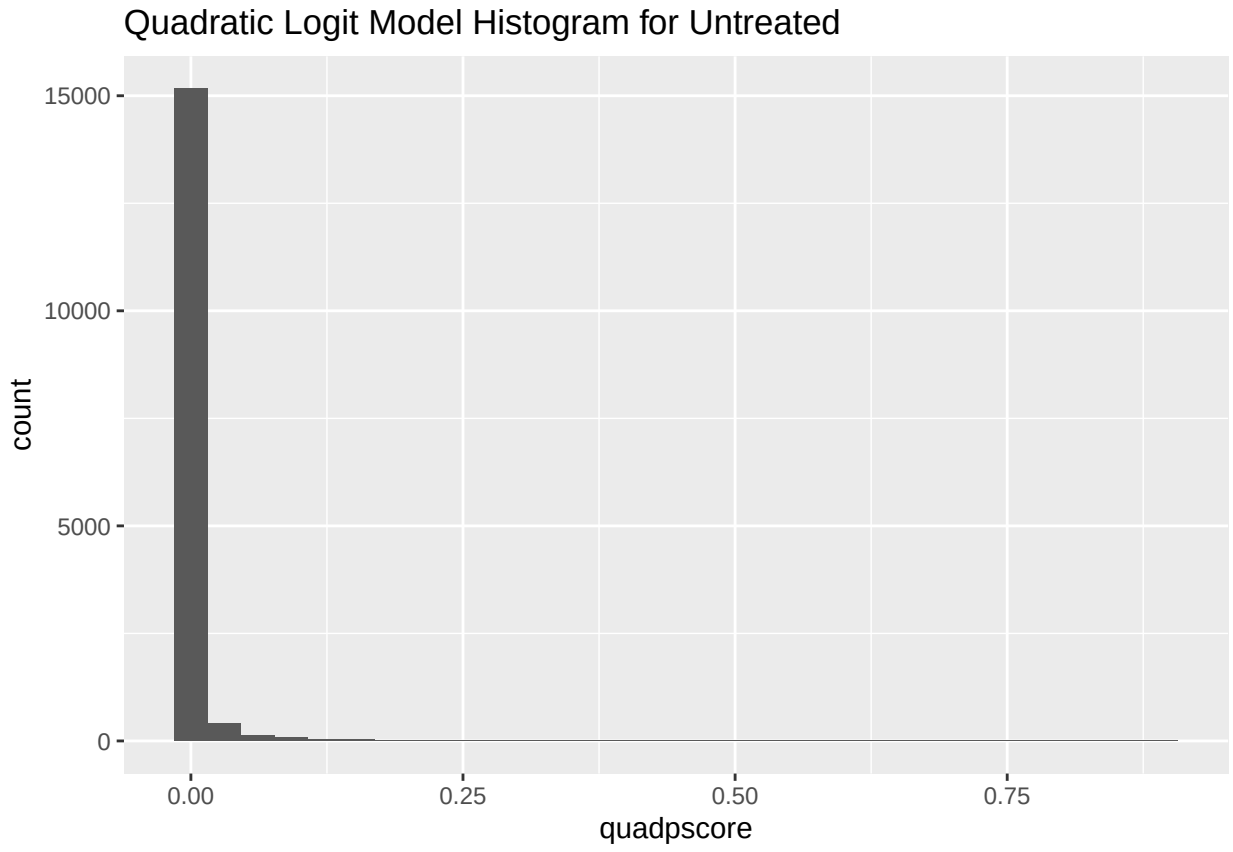
```
logit_cube_sumstat<- logit_nsw_dw_cpscontrol %>%
  group_by(treat)%>%
  summarize(logit_cube_mean = mean(cubepscore),
```

```
            logit_cube_max = max(cubepscore),
             logit_cube_min = min(cubepscore))
logit_cube_sumstat
```

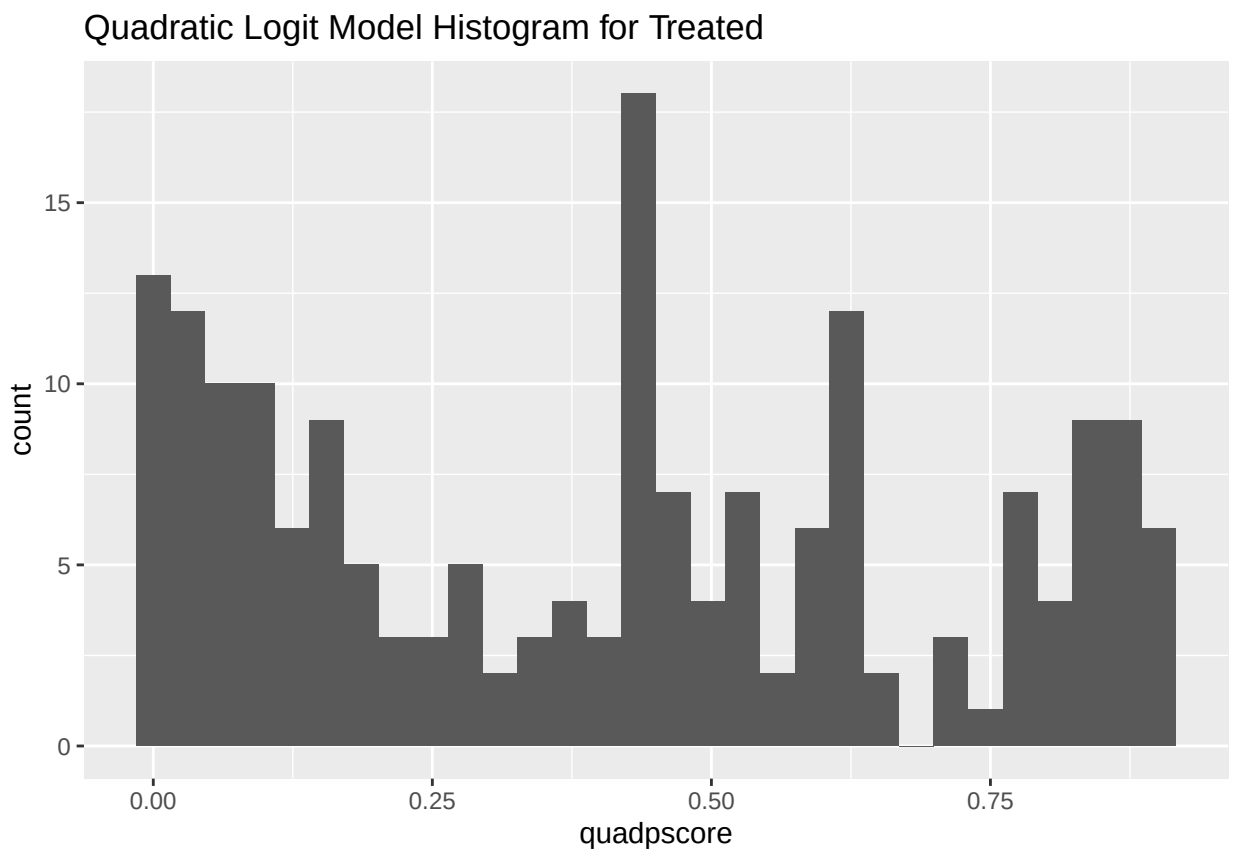
```
## # A tibble: 2 x 4
##   treat logit_cube_mean logit_cube_max logit_cube_min
##   <dbl>           <dbl>          <dbl>          <dbl>
## 1     0         0.00658          0.915  0.00000000576
## 2     1         0.431            0.923  0.00116
```

We then look at histograms of propensity scores for each model for the treated and untreated observations.

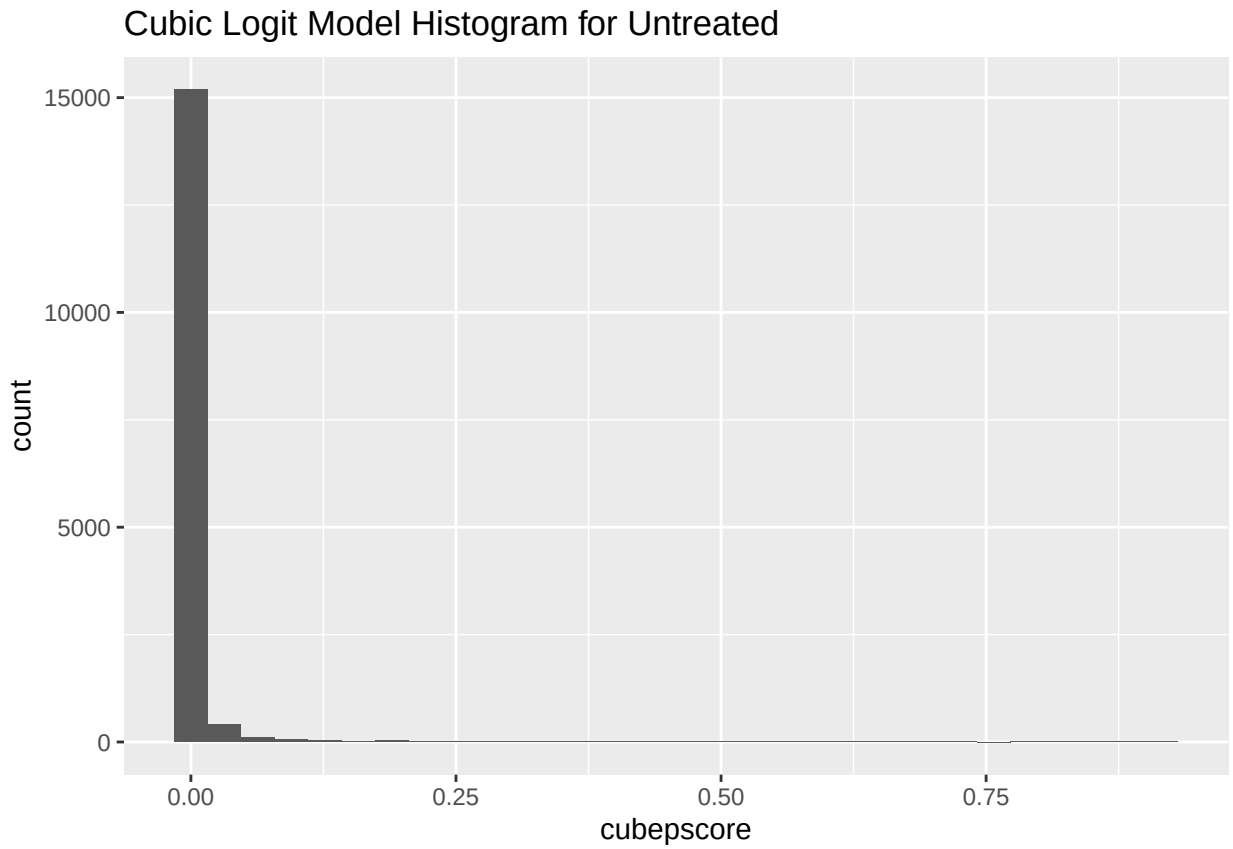
```
logit_nsw_dw_cpscontrol %>%
  filter(treat == 0) %>%
  ggplot() +
  geom_histogram(aes(x = quadpscore))+
  labs(title = "Quadratic Logit Model Histogram for Untreated")
```



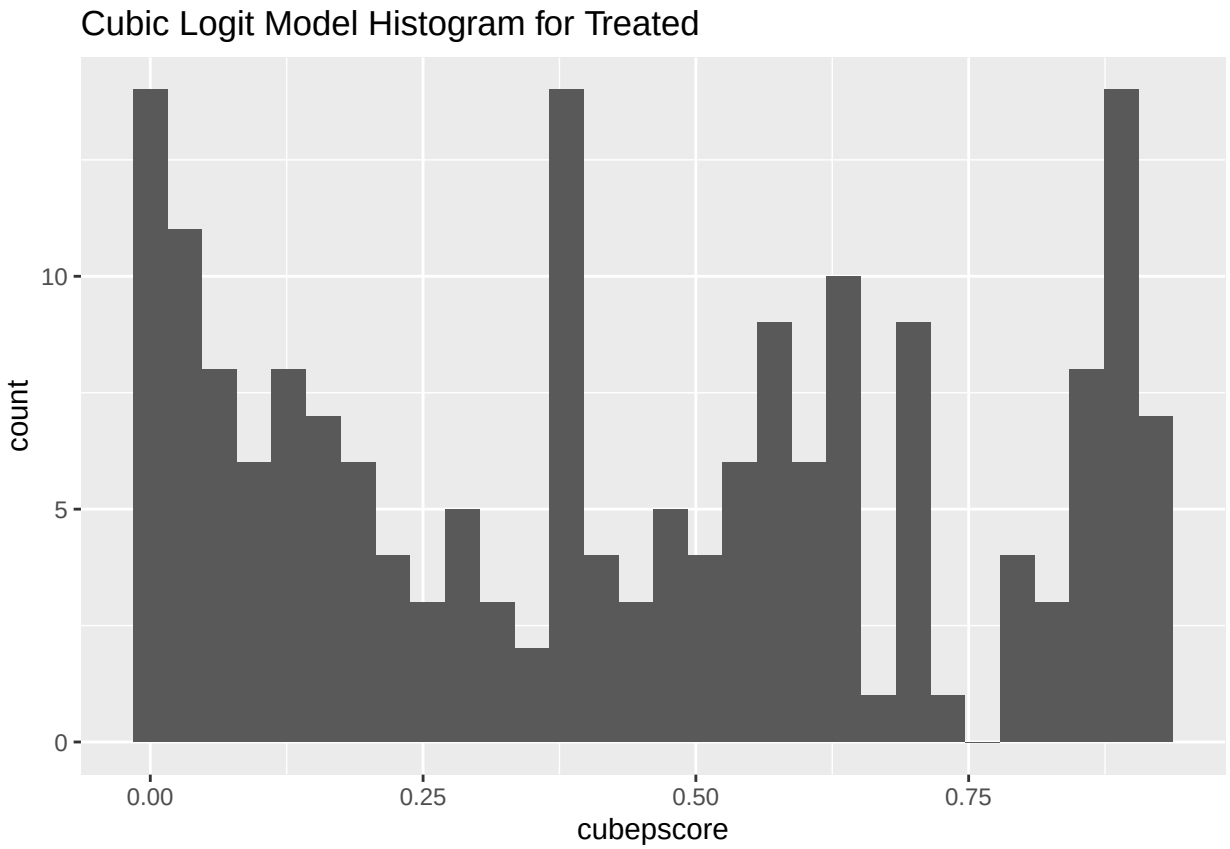

```
logit_nsw_dw_cpscontrol %>%
  filter(treat == 1) %>%
  ggplot() +
  geom_histogram(aes(x = quadpscore))+
  labs(title = "Quadratic Logit Model Histogram for Treated")
```




```
logit_nsw_dw_cpscontrol %>%
  filter(treat == 0) %>%
  ggplot() +
  geom_histogram(aes(x = cubepscore))+
  labs(title = "Cubic Logit Model Histogram for Untreated")
```




```
logit_nsw_dw_cpscontrol %>%
  filter(treat == 1) %>%
  ggplot() +
  geom_histogram(aes(x = cubepscore))+
  labs(title = "Cubic Logit Model Histogram for Treated")
```



Next we go through this whole process again using OLS. We estimate a model using up to quadratic covariates, then up to cubic covariates. Then we will predict and observe the propensity scores for each model for untreated and treated observations.

```
# estimating ols
quad_ols_nsw <- lm(treat ~ age + agesq + educ + educsq +
                       marr + nodegree + black + hisp + re74 +
                       re74sq + re75 + re75sq +
                       u74 + u75, data = nsw_dw_cpscontrol)

cube_ols_nsw <- lm(treat ~ age + agesq + agecube + educ + educsq + educcube +
                           marr + nodegree + black + hisp + re74 +
                           re74sq + re74cube + re75 + re75sq +
                           re75cube + u74 + u75, data = nsw_dw_cpscontrol)

ols_nsw_dw_cpscontrol <- nsw_dw_cpscontrol %>%
  mutate(quadpscore = quad_ols_nsw$fitted.values, cubepscore = cube_ols_nsw$fitted.values)



# mean pscore

ols_quad_sumstat<- ols_nsw_dw_cpscontrol %>%
  group_by(treat)%>%
  summarize(ols_quad_mean = mean(quadpscore),
            ols_quad_max = max(quadpscore),
            ols_quad_min = min(quadpscore))
```

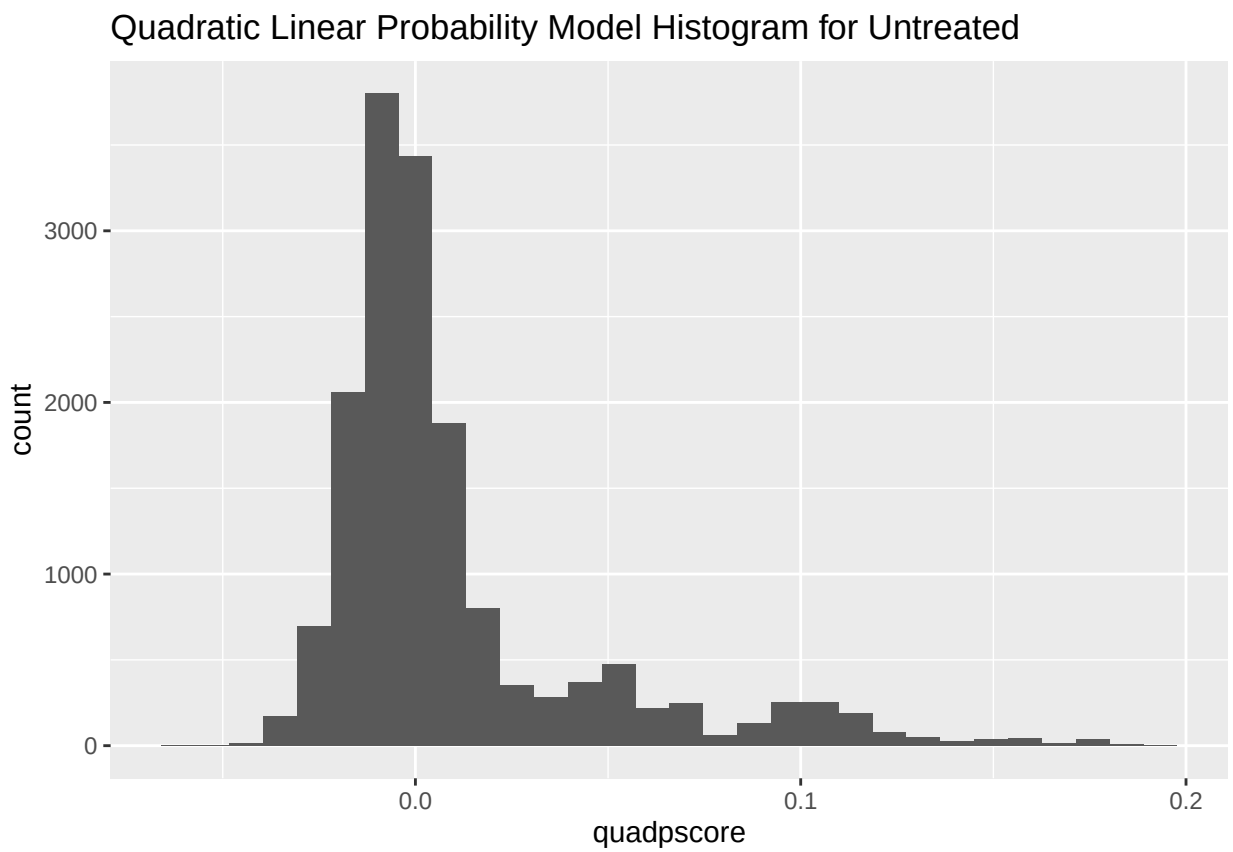
```
ols_quad_sumstat
```

```
## # A tibble: 2 x 4
##   treat ols_quad_mean ols_quad_max ols_quad_min
##   <dbl>         <dbl>        <dbl>        <dbl>
## 1     0       0.00993        0.193      -0.0614
## 2     1       0.141          0.193      -0.0139
```

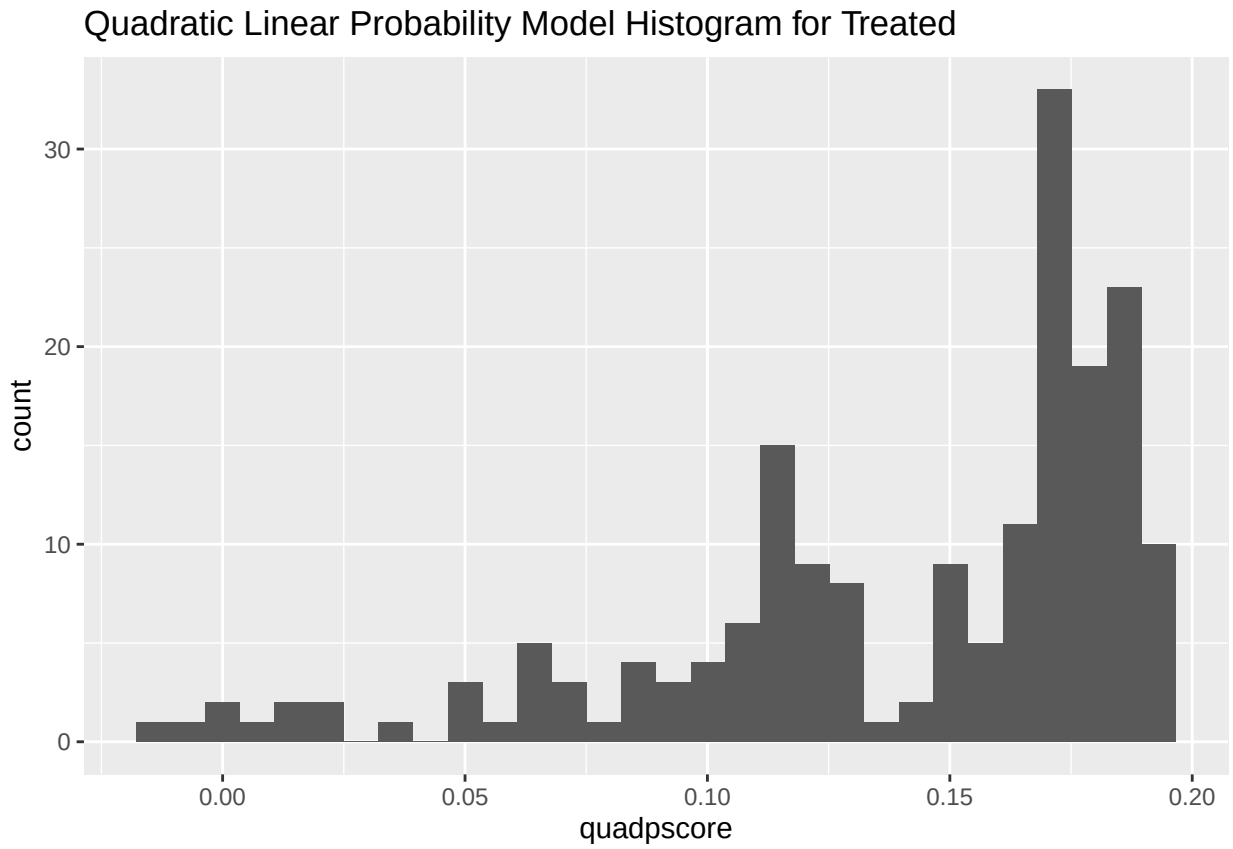
```
ols_cube_sumstat<- ols_nsw_dw_cpscontrol %>%
  group_by(treat)%>%
  summarize(ols_cube_mean = mean(cubepscore),
            ols_cube_max = max(cubepscore),
            ols_cube_min = min(cubepscore))
ols_cube_sumstat
```

```
## # A tibble: 2 x 4
##   treat ols_cube_mean ols_cube_max ols_cube_min
##   <dbl>         <dbl>        <dbl>        <dbl>
## 1     0       0.00986        0.207      -0.0535
## 2     1       0.147          0.208      -0.0153
```

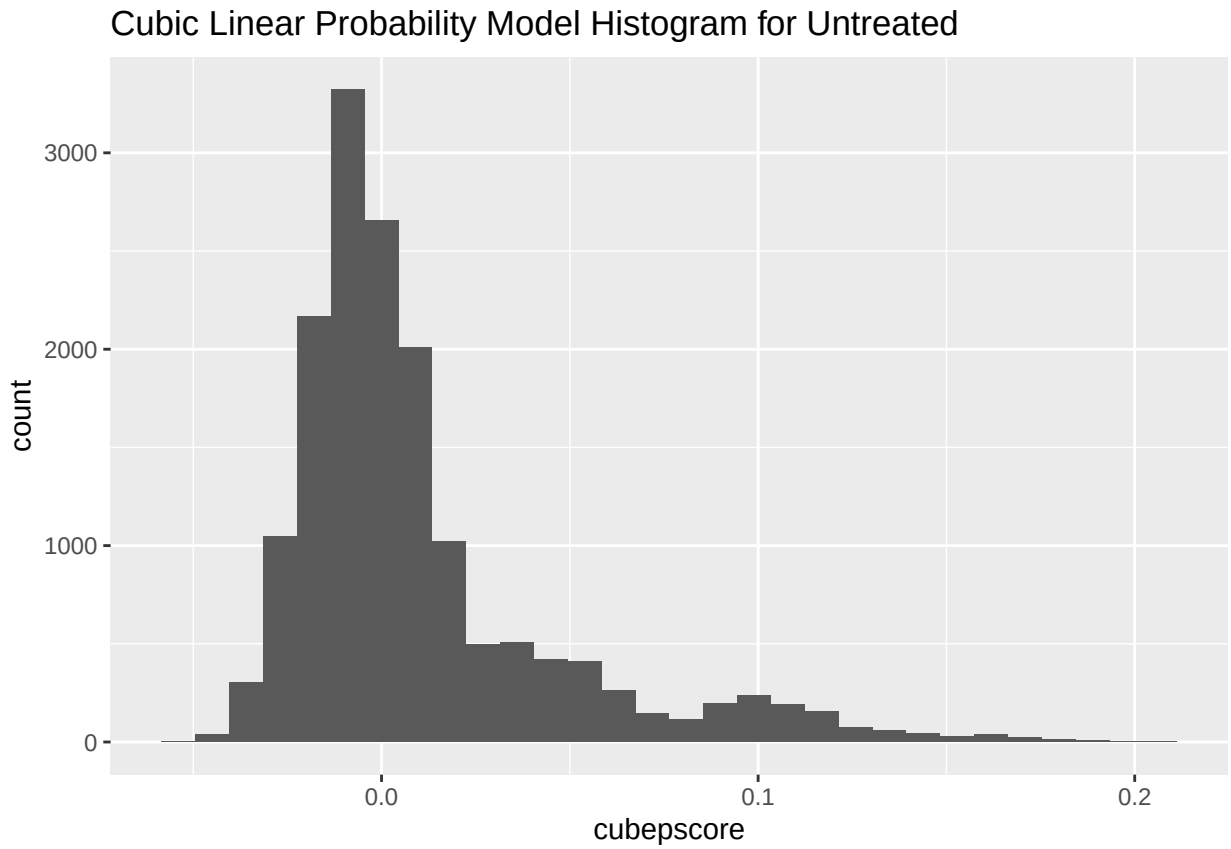
```
# histogram
ols_nsw_dw_cpscontrol %>%
  filter(treat == 0) %>%
  ggplot() +
  geom_histogram(aes(x = quadpscore))+
  labs(title = "Quadratic Linear Probability Model Histogram for Untreated")
```



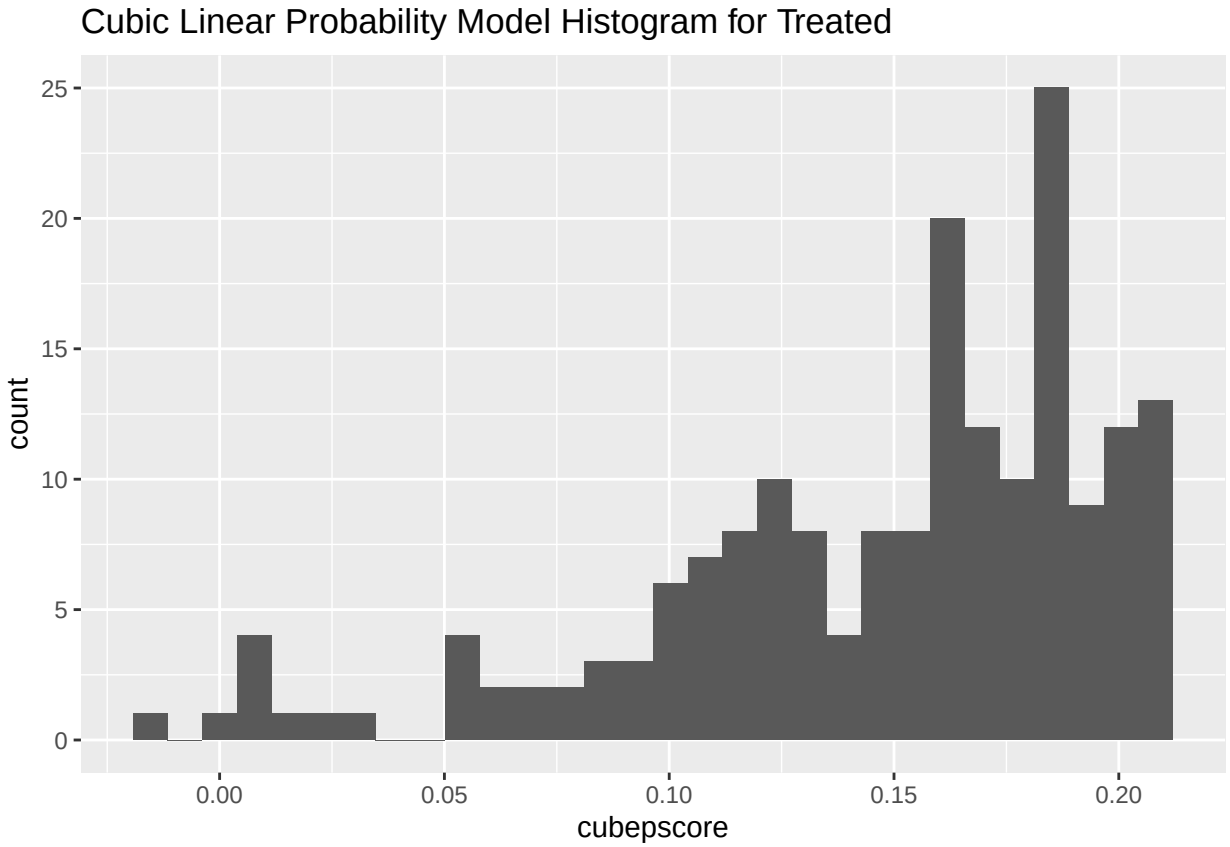
```r
ols_nsw_dw_cpscontrol %>%
  filter(treat == 1) %>%
  ggplot() +
  geom_histogram(aes(x = quadpscore))+
  labs(title = "Quadratic Linear Probability Model Histogram for Treated")
```




```r
ols_nsw_dw_cpscontrol %>%
  filter(treat == 0) %>%
  ggplot() +
  geom_histogram(aes(x = cubepscore))+
  labs(title = "Cubic Linear Probability Model Histogram for Untreated")
```




```
ols_nsw_dw_cpscontrol %>%
  filter(treat == 1) %>%
  ggplot() +
  geom_histogram(aes(x = cubepscore))+
  labs(title = "Cubic Linear Probability Model Histogram for Treated")
```



As you can observe, the average propensity scores for untreated observations is higher using OLS, and the average propensity score for treated observations is much lower using OLS. OLS is giving us results less than zero which is weighting it downwards.

Trim Data and Repeat Histograms

Now we trim the data so that only propensity scores greater that 0.1, and less than 0.9 are included in the histograms.

```
#drop pscores <0.1 and >0.9
quad_cut_logit_data<- logit_nsw_dw_cpscontrol %>%
  filter(quadpscore > 0.1 & quadpscore < 0.9)

cube_cut_logit_data<- logit_nsw_dw_cpscontrol %>%
  filter(cubepscore > 0.1 & cubepscore < 0.9)

quad_cut_ols_data<- ols_nsw_dw_cpscontrol %>%
  filter(quadpscore > 0.1 & quadpscore < 0.9)

cube_cut_ols_data<- ols_nsw_dw_cpscontrol %>%
  filter(cubepscore > 0.1 & cubepscore < 0.9)

#average min max for all 4 models

sumstat_quad_cut_logit_data<- quad_cut_logit_data %>%
  group_by(treat)%>%
  summarize(logit_quad_mean = mean(quadpscore),
            logit_quad_max = max(quadpscore),
```

```
            logit_quad_min = min(quadpscore))
sumstat_quad_cut_logit_data
```

```
## # A tibble: 2 x 4
##   treat logit_quad_mean logit_quad_max logit_quad_min
##   <dbl>           <dbl>          <dbl>          <dbl>
## 1     0           0.289          0.891          0.100
## 2     1           0.512          0.900          0.107
```

```
sumstat_cube_cut_logit_data<- cube_cut_logit_data %>%
  group_by(treat)%>%
  summarize(logit_cube_mean = mean(cubepscore),
            logit_cube_max = max(cubepscore),
            logit_cube_min = min(cubepscore))
sumstat_cube_cut_logit_data
```

```
## # A tibble: 2 x 4
##   treat logit_cube_mean logit_cube_max logit_cube_min
##   <dbl>           <dbl>          <dbl>          <dbl>
## 1     0           0.283          0.877          0.100
## 2     1           0.511          0.899          0.106
```

```
sumstat_quad_cut_ols_data<- quad_cut_ols_data %>%
  group_by(treat)%>%
  summarize(ols_quad_mean = mean(quadpscore),
            ols_quad_max = max(quadpscore),
            ols_quad_min = min(quadpscore))
sumstat_quad_cut_ols_data
```

```
## # A tibble: 2 x 4
##   treat ols_quad_mean ols_quad_max ols_quad_min
##   <dbl>         <dbl>        <dbl>        <dbl>
## 1     0         0.122        0.193        0.100
## 2     1         0.160        0.193        0.102
```

```
sumstat_cube_cut_ols_data<- cube_cut_ols_data %>%
  group_by(treat)%>%
  summarize(ols_cube_mean = mean(cubepscore),
            ols_cube_max = max(cubepscore),
            ols_cube_min = min(cubepscore))
sumstat_cube_cut_ols_data
```

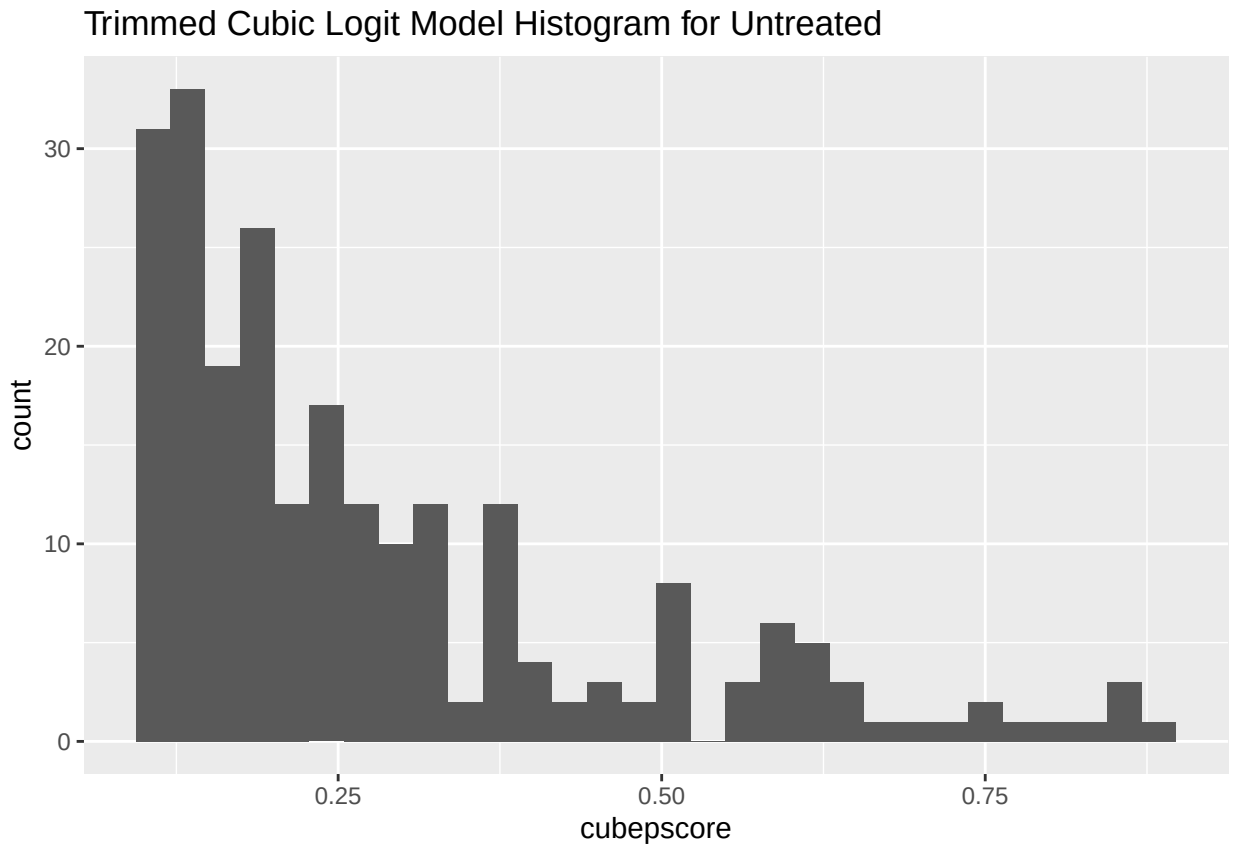
```
## # A tibble: 2 x 4
##   treat ols_cube_mean ols_cube_max ols_cube_min
##   <dbl>         <dbl>        <dbl>        <dbl>
## 1     0         0.125        0.207        0.100
## 2     1         0.164        0.208        0.101
```

Now we repeat the histograms using the trimmed data. Moving forward I will only use the Logit model with the cubic covariates, and the OLS model with the quadratic covariates.

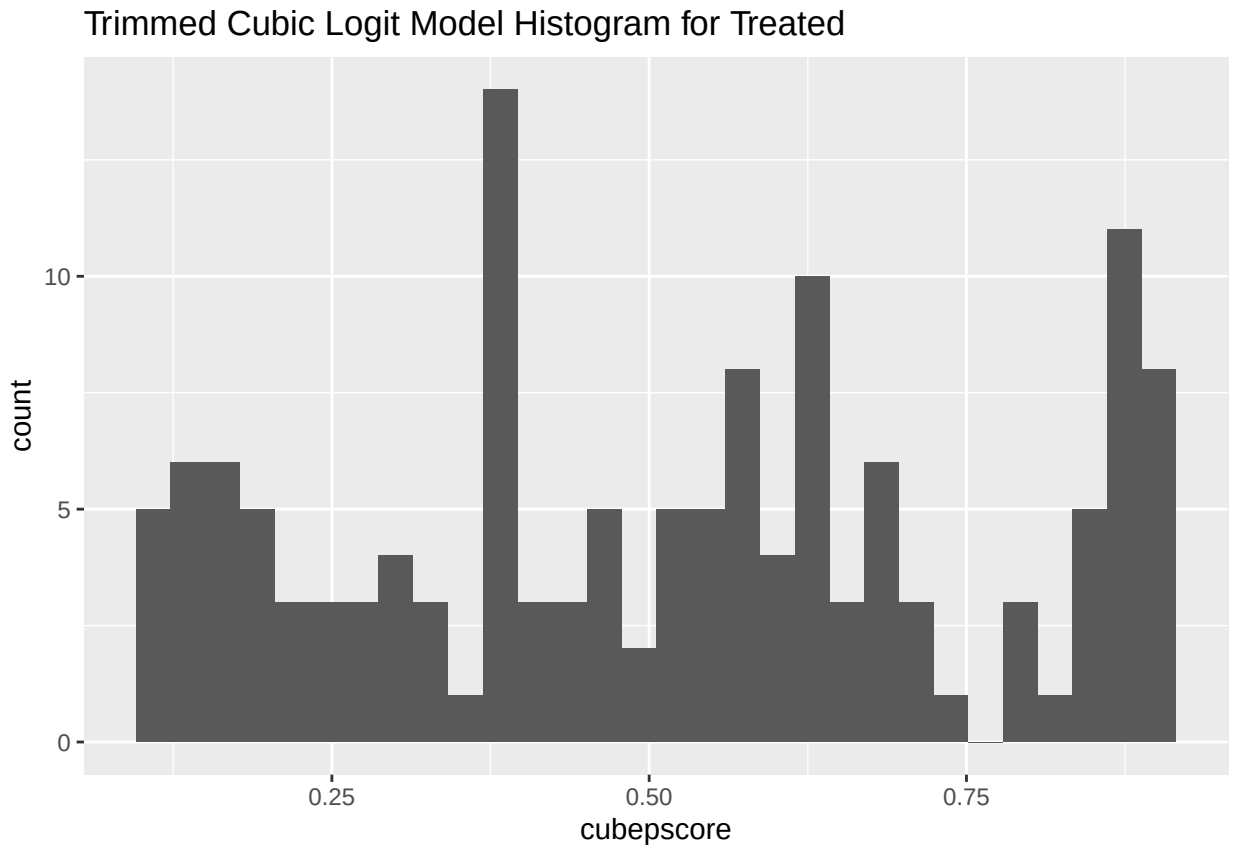
```
#repeat 1c


cube_cut_logit_data %>%
  filter(treat == 0) %>%
  ggplot() +
```

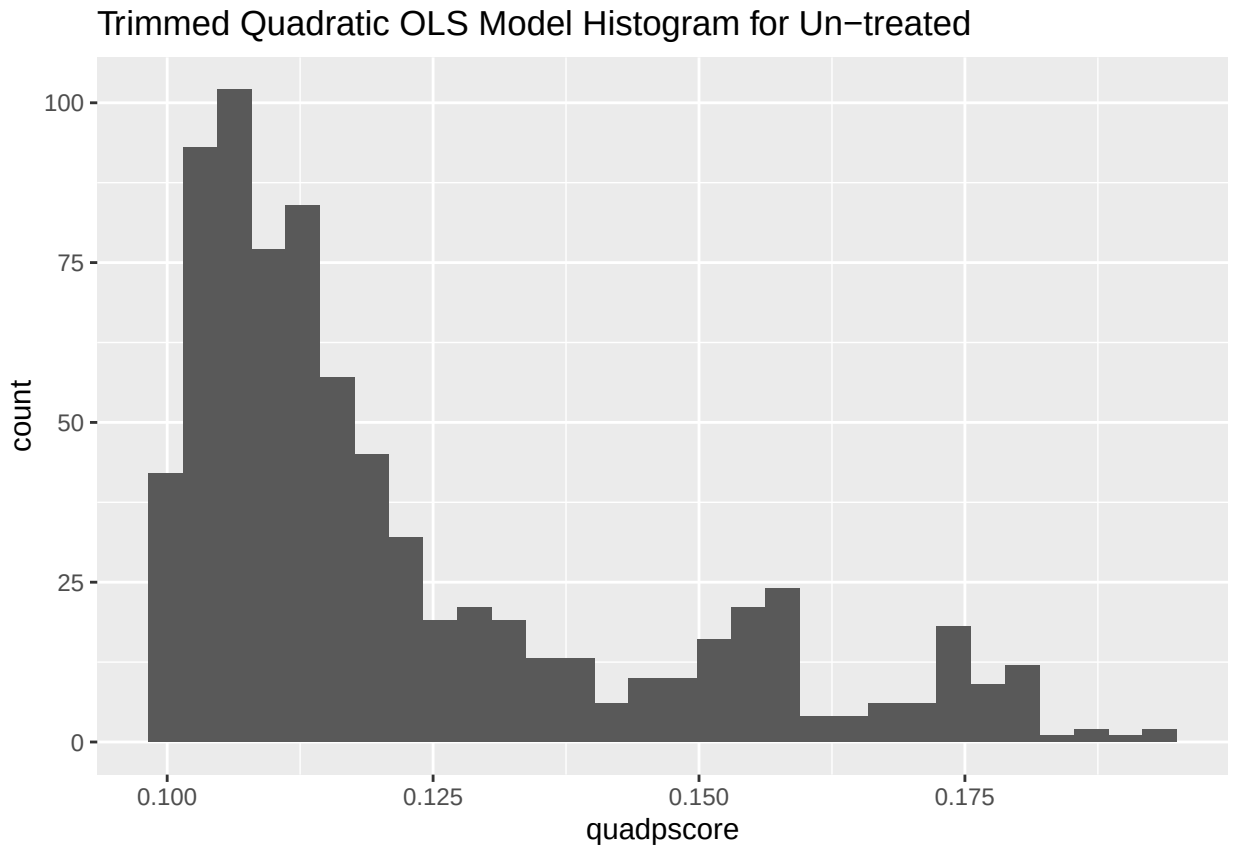
```
  geom_histogram(aes(x = cubepscore))+
  labs(title = "Trimmed Cubic Logit Model Histogram for Untreated" )
```



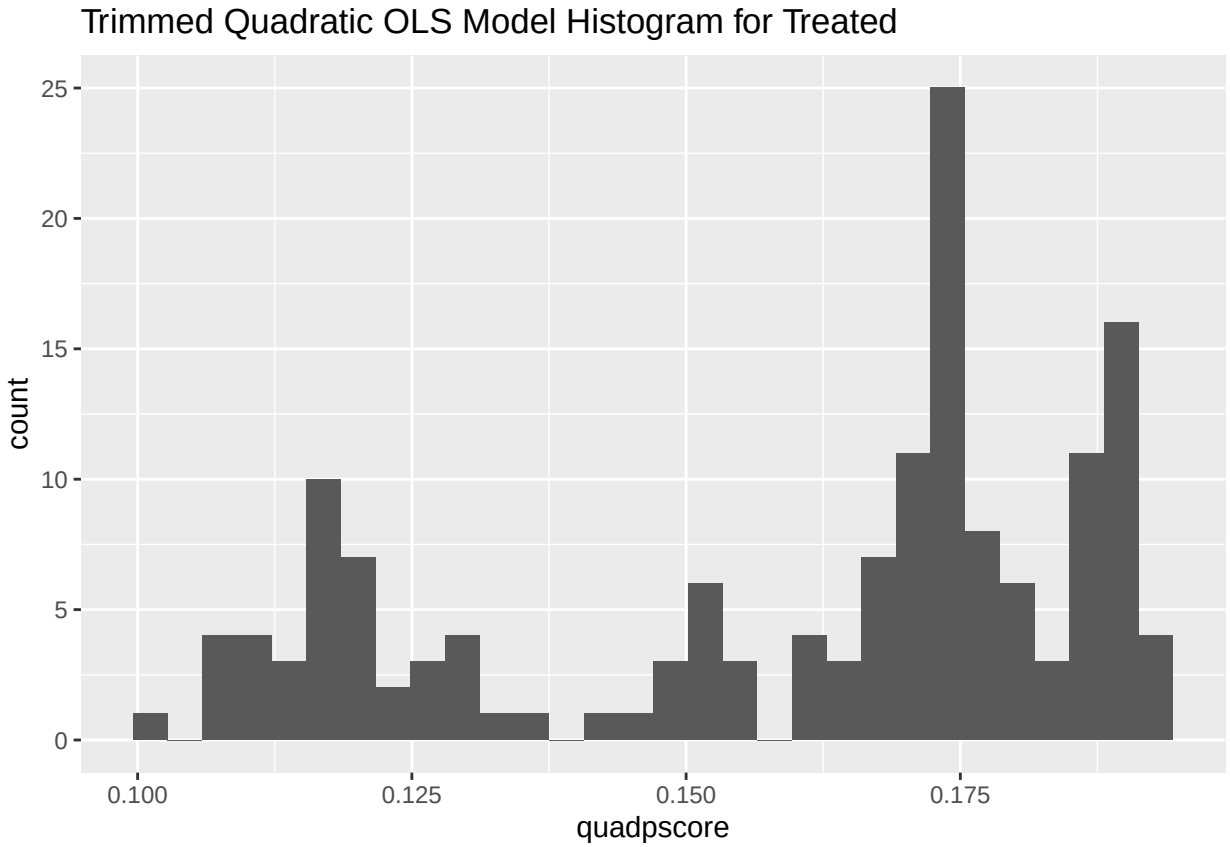

```
cube_cut_logit_data %>%
  filter(treat == 1) %>%
  ggplot() +
  geom_histogram(aes(x = cubepscore))+
  labs(title = "Trimmed Cubic Logit Model Histogram for Treated" )
```




```
quad_cut_ols_data %>%
  filter(treat == 0) %>%
  ggplot() +
  geom_histogram(aes(x = quadpscore))+
  labs(title = "Trimmed Quadratic OLS Model Histogram for Un-treated")
```




```
quad_cut_ols_data %>%
  filter(treat == 1) %>%
  ggplot() +
  geom_histogram(aes(x = quadpscore))+
  labs(title = "Trimmed Quadratic OLS Model Histogram for Treated")
```



Repeating 1C with the trimmed data yields average propensity scores that are higher for the untreated and lower for the treated, as would be expected by removing the highest and lowest propensity scores. The propensity scores are now closer to the mean as desired, and no longer weighted down.

2. Calculate Before and After First Difference for Each Unit

Moving forward, I will only be using the model with cubic covariates for Logit and the model with quadratic covariates for OLS. For both datasets I will use the data trimmed to include on observations with pscores greater that 0.1 and less than 0.9.

```
#LOGIT

mean1 <- cube_cut_logit_data%>%
  filter(treat == 1) %>%
  pull(re78) %>%
  mean()

cube_cut_logit_data$y1 <- mean1

mean0 <- cube_cut_logit_data%>%
  filter(treat == 0) %>%
  pull(re78) %>%
  mean()

cube_cut_logit_data$y0 <- mean0

ate <- unique(cube_cut_logit_data$y1 - cube_cut_logit_data$y0)
```

```
atedf <- data.frame(ate)

atedf
```

```
##        ate
## 1 1724.773
```

```
cube_cut_logit_data <- cube_cut_logit_data%>%
  select(-y1, -y0)
```

Using the logit model without weights, the first difference is ~ $1724.

```
#OLS

mean1 <- quad_cut_ols_data %>%
  filter(treat == 1) %>%
  pull(re78) %>%
  mean()

quad_cut_ols_data$y1 <- mean1

mean0 <- quad_cut_ols_data%>%
  filter(treat == 0) %>%
  pull(re78) %>%
  mean()

quad_cut_ols_data$y0 <- mean0

ate <- unique(quad_cut_ols_data$y1 - quad_cut_ols_data$y0)

atedf <- data.frame(ate)

atedf
```

```
##         ate
## 1 -5014.113
```

```
quad_cut_ols_data <- quad_cut_ols_data %>%
  select(-y1, -y0)
```

Using the OLS model without weights, the first difference is ~ -$5014.

Construct a Weighted Difference in Difference

Now I construct a weighted difference in difference using normalized and non-normalized wieghts based on the predicted propensity scores for the Logit model and OLS model.

```
#number 3: Construct a Weighted Difference in Difference

#LOGIT
N <- nrow(cube_cut_logit_data )
#- Manual with non-normalized weights using all data
  cube_cut_logit_data  <- cube_cut_logit_data  %>%
  mutate(d1 = treat/cubepscore,
         d0 = (1-treat)/(1-cubepscore))

s1 <- sum(cube_cut_logit_data$d1)
```

```r
s0 <- sum(cube_cut_logit_data$d0)


cube_cut_logit_data <- cube_cut_logit_data %>%
  mutate(y1 = treat * re78/cubepscore,
         y0 = (1-treat) * re78/(1-cubepscore),
         ht = y1 - y0)

#- Manual with normalized weights
cube_cut_logit_data  <- cube_cut_logit_data  %>%
  mutate(y1 = (treat*re78/cubepscore)/(s1/N),
         y0 = ((1-treat)*re78/(1-cubepscore))/(s0/N),
         norm = y1 - y0)

Non_normalized_ATT<- cube_cut_logit_data  %>%
  pull(ht) %>%
  mean()

Normalized_ATT<- cube_cut_logit_data  %>%
  pull(norm) %>%
  mean()

df<- data.frame(Non_normalized_ATT, Normalized_ATT)
df
```

```
##   Non_normalized_ATT Normalized_ATT
## 1           1744.356       1635.115
```

Above are the average normalized and non-normalized average treatment effects for the treated observations using the logit propensity score prediction weights. This result suggest that the effect of the treatment is an increase in wages of about $1700.

```r
#OLS
N <- nrow(quad_cut_ols_data )
#- Manual with non-normalized weights using all data
quad_cut_ols_data <- quad_cut_ols_data %>%
  mutate(d1 = treat/quadpscore,
         d0 = (1-treat)/(1-quadpscore))

s1 <- sum(quad_cut_ols_data$d1)
s0 <- sum(quad_cut_ols_data$d0)


quad_cut_ols_data <- quad_cut_ols_data %>%
  mutate(y1 = treat * re78/quadpscore,
         y0 = (1-treat) * re78/(1-quadpscore),
         ht = y1 - y0)

#- Manual with normalized weights
quad_cut_ols_data  <- quad_cut_ols_data  %>%
  mutate(y1 = (treat*re78/quadpscore)/(s1/N),
         y0 = ((1-treat)*re78/(1-quadpscore))/(s0/N),
         norm = y1 - y0)

Non_normalized_ATT<-quad_cut_ols_data %>%
```

```
  pull(ht) %>%
  mean()

Normalized_ATT<-quad_cut_ols_data  %>%
  pull(norm) %>%
  mean()

df<- data.frame(Non_normalized_ATT, Normalized_ATT)
df
```

```
##   Non_normalized_ATT Normalized_ATT
## 1           -4047.76      -4967.571
```

Above are the average normalized and non-normalized average treatment effects for the treated observations using the ols propensity score prediction weights. This result suggest that the effect of the treatment is a decrease in wages of about $5000.