CS112 Assignment 2

Question 1

The original data-generating equation

The idea behind generating this data is that a hypothetical case in which people that eat more processed meat (in grams) are more likely to be less healthy (in an arbitrary unit).

```
processed_meat <- rnorm(998, mean = 150, sd = 65)
noise <- rnorm(998, mean = 0, sd = 120)
healthy <- (noise - processed_meat) / 100
healthy_meat_relationship <- data.frame(healthy, processed_meat)
reg <- lm(healthy ~ processed_meat)
plot(processed_meat, healthy, xlab = "Processed Meat Per Day (In Grams)", ylab = "Health Level",
     ylim = c(-10, 15), xlim = c(0, 2000))
lines(processed_meat, reg$fitted.values, col = "red")
summary(reg)
```

Regression results for the original 998 (“summary” output)

```
Call:
lm(formula = healthy ~ processed_meat)

Residuals:
    Min      1Q  Median      3Q     Max
-3.5855 -0.8190 -0.0447  0.8059  3.5580

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.1747519  0.0968248   1.805   0.0714 .
processed_meat -0.0109864  0.0005877 -18.694   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.202 on 996 degrees of freedom
Multiple R-squared:  0.2597,    Adjusted R-squared:  0.259
F-statistic: 349.5 on 1 and 996 DF,  p-value: < 2.2e-16
```

Regression results with the outliers included (“summary” output)

```
Call:
lm(formula = healthy ~ processed_meat, data = health_relationship_outliers)

Residuals:
    Min      1Q  Median      3Q     Max
-4.0776 -0.9896 -0.0429  0.9583 17.7031

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -1.6831468  0.0963342 -17.472  < 2e-16 ***
processed_meat  0.0014900  0.0005276   2.824  0.00483 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.618 on 998 degrees of freedom
Multiple R-squared:  0.007929,  Adjusted R-squared:  0.006935
F-statistic: 7.976 on 1 and 998 DF,  p-value: 0.004835
```

A data visualization that shows the regression line based on the original 998 points, and another differentiated regression line based on 1000 points

The final figure is Figure 4. Other figures are for scaling and illustration purposes.

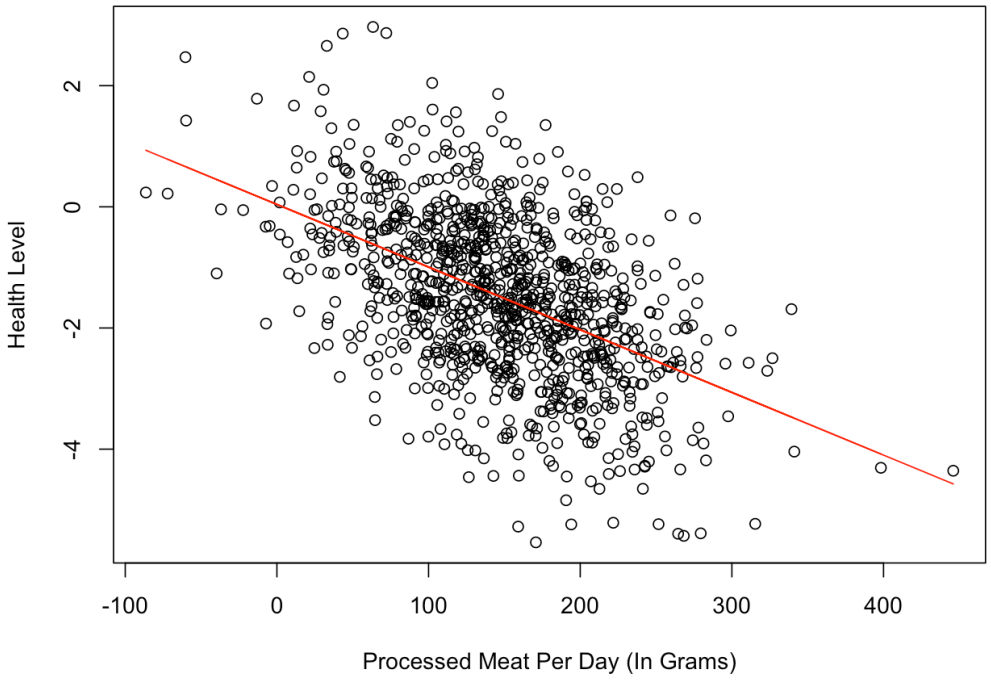


Figure 1. The linear regression that shows the negative relationship between eating meat and health level.

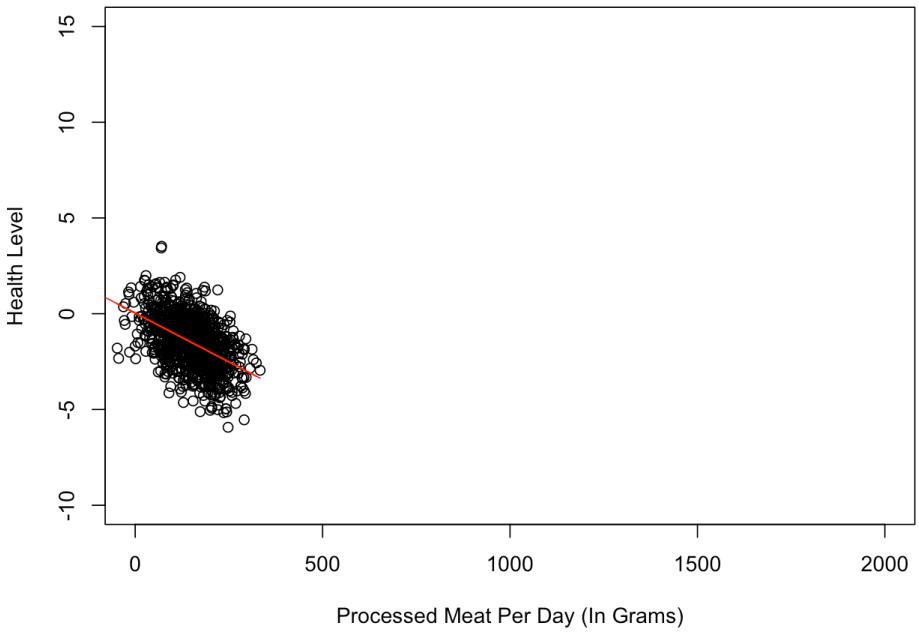


Figure 2. The linear regression that shows the negative relationship between eating meat and health level in a bigger scale.

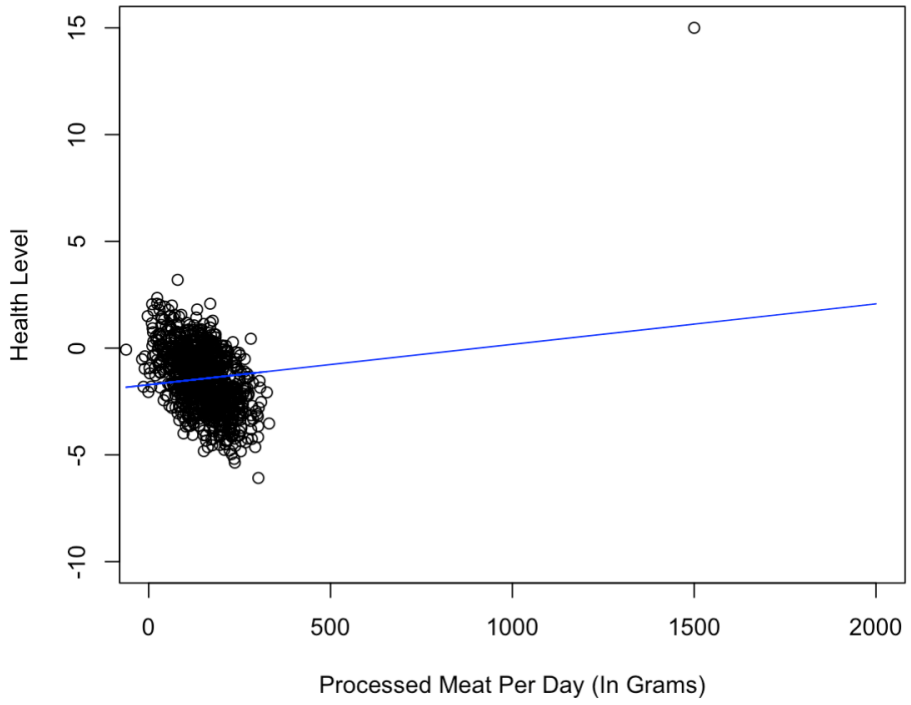


Figure 3. The linear regression that shows the *positive* relationship between eating meat and health level with the outliers.

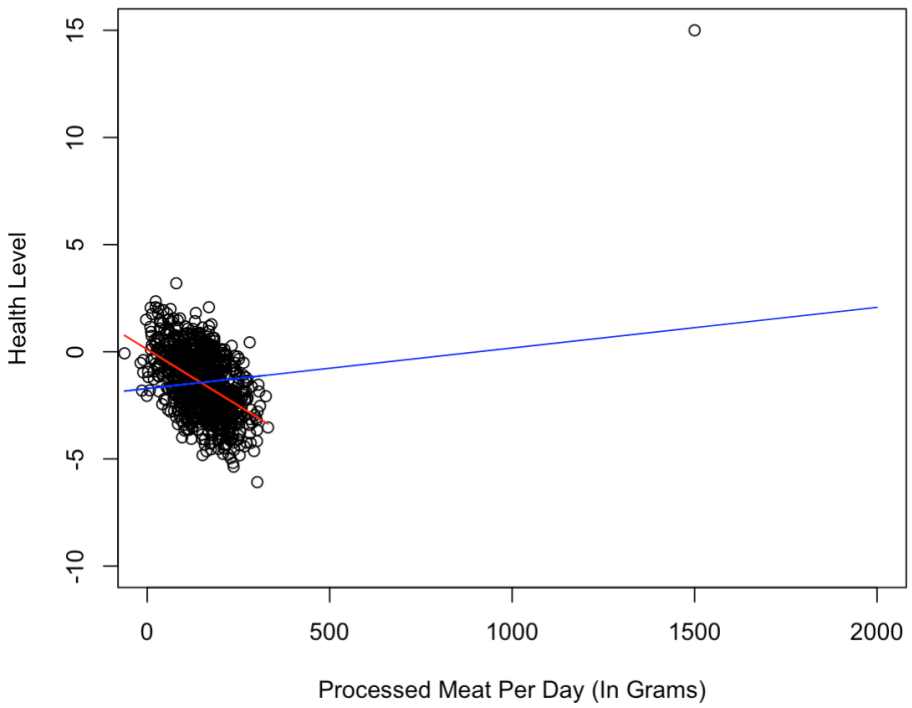


Figure 4. A graph with two lines where red line shows the negative relationship between eating processed meat and health, while the blue line suggests the opposite.

The dangers of extrapolation

In Figure 4., the red line show that eating more processed meat negatively impact health level, while the blue line — incorporating two additional data points as outliers — suggest the opposite. Figure 4 shows when we use the same "*known*" equation that works correctly with certain data set, it might not be suitable with future data set as it might contain outliers that alter the end result. Therefore, although the initial equation might have been correct, it should not be the only point of reference.

Question 2

2a. Predict re78 for every TREATED unit of age for every set of coefficients holding at the means

	Mean.PI.Lower.Bound	Mean.PI.Upper.Bound	mean.educ	mean.re74	mean.re75
17	3720.253	7161.813	10.34595	2095.574	1532.056
18	4027.430	7064.537	10.34595	2095.574	1532.056
19	4306.299	7022.033	10.34595	2095.574	1532.056
20	4557.249	7000.646	10.34595	2095.574	1532.056
21	4761.655	7014.894	10.34595	2095.574	1532.056
22	4945.732	7056.697	10.34595	2095.574	1532.056
23	5087.345	7118.943	10.34595	2095.574	1532.056
24	5215.939	7209.173	10.34595	2095.574	1532.056
25	5313.446	7313.187	10.34595	2095.574	1532.056
26	5388.664	7444.561	10.34595	2095.574	1532.056
27	5453.162	7586.456	10.34595	2095.574	1532.056
28	5498.155	7720.395	10.34595	2095.574	1532.056
29	5562.548	7867.324	10.34595	2095.574	1532.056
30	5606.252	8021.948	10.34595	2095.574	1532.056
31	5632.608	8170.139	10.34595	2095.574	1532.056
32	5660.021	8332.175	10.34595	2095.574	1532.056
33	5667.199	8490.643	10.34595	2095.574	1532.056
34	5668.728	8655.700	10.34595	2095.574	1532.056
35	5665.534	8832.957	10.34595	2095.574	1532.056
36	5679.062	9003.492	10.34595	2095.574	1532.056
37	5672.388	9176.576	10.34595	2095.574	1532.056
38	5648.769	9344.482	10.34595	2095.574	1532.056
39	5603.046	9541.345	10.34595	2095.574	1532.056
40	5540.191	9759.775	10.34595	2095.574	1532.056
41	5481.637	9986.690	10.34595	2095.574	1532.056
42	5397.329	10222.441	10.34595	2095.574	1532.056
43	5277.864	10492.866	10.34595	2095.574	1532.056
44	5145.220	10753.652	10.34595	2095.574	1532.056
45	5016.370	11040.152	10.34595	2095.574	1532.056
46	4853.781	11343.409	10.34595	2095.574	1532.056
47	4660.704	11685.483	10.34595	2095.574	1532.056
48	4442.869	12027.044	10.34595	2095.574	1532.056

Figure 5. A table that contains the upper and lower bounds for the Re78 (income) for each age when other variables — education, Re74, and 75 — are held at their means for the treated units.

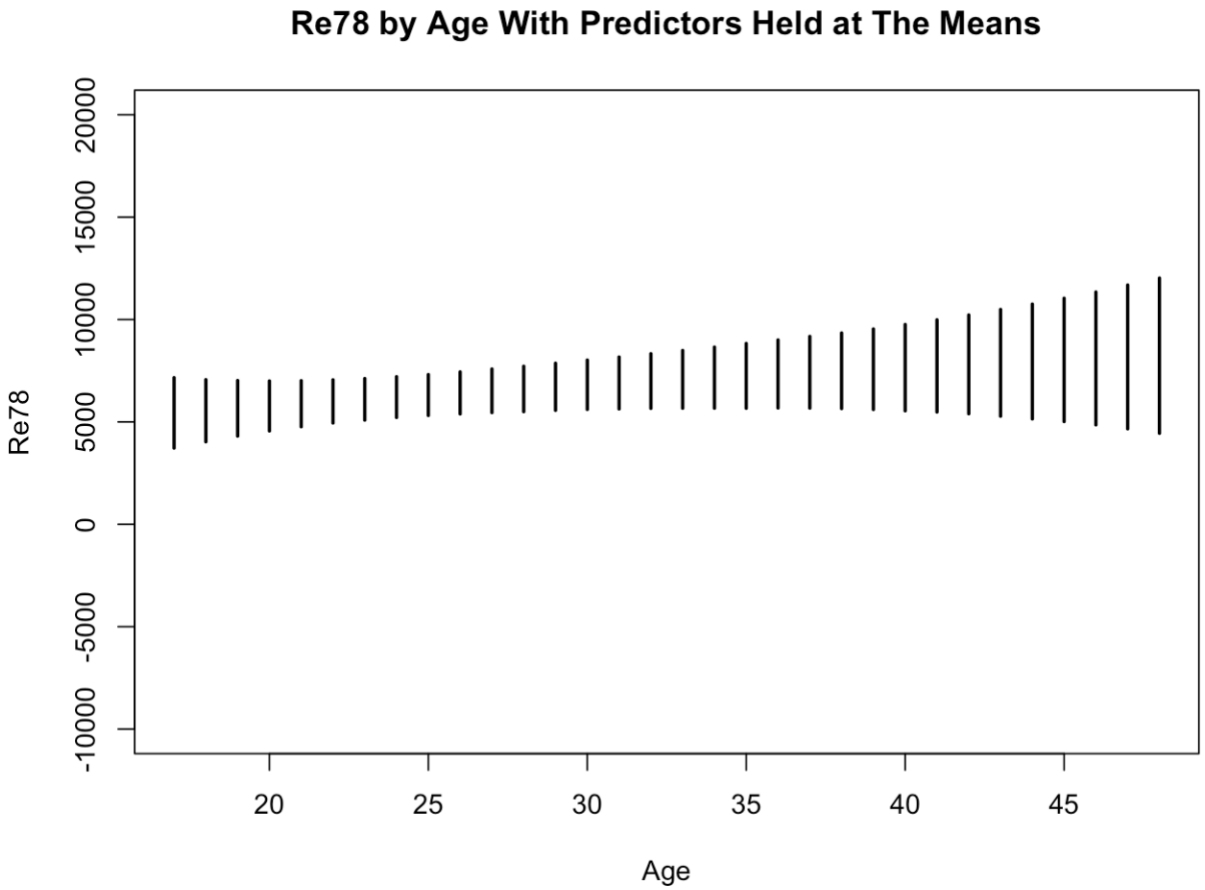


Figure 6. A graph that shows the 95% confidence interval of expected values for re78 for different ages for treated units, where predictors are held at their means. It means that for every age, we can be 95% certain that the confidence interval contains the true mean of the re78.

2b. Predict re78 for every CONTROL unit of age for every set of coefficients holding at the means

	Mean.PI.Lower.Bound	Mean.PI.Upper.Bound	mean.educ	mean.re74	mean.re75
17	2953.1026	5889.842	10.34595	2095.574	1532.056
18	3173.4861	5734.292	10.34595	2095.574	1532.056
19	3339.5503	5611.358	10.34595	2095.574	1532.056
20	3489.9104	5511.207	10.34595	2095.574	1532.056
21	3601.1072	5456.227	10.34595	2095.574	1532.056
22	3680.4786	5416.739	10.34595	2095.574	1532.056
23	3714.8809	5422.569	10.34595	2095.574	1532.056
24	3715.5004	5460.847	10.34595	2095.574	1532.056
25	3695.8076	5504.228	10.34595	2095.574	1532.056
26	3666.6041	5569.326	10.34595	2095.574	1532.056
27	3631.5599	5635.028	10.34595	2095.574	1532.056
28	3594.7230	5688.516	10.34595	2095.574	1532.056
29	3541.6927	5758.675	10.34595	2095.574	1532.056
30	3487.7153	5819.964	10.34595	2095.574	1532.056
31	3447.6735	5883.983	10.34595	2095.574	1532.056
32	3392.9789	5941.352	10.34595	2095.574	1532.056
33	3345.8779	5998.335	10.34595	2095.574	1532.056
34	3282.0495	6057.406	10.34595	2095.574	1532.056
35	3223.8926	6103.835	10.34595	2095.574	1532.056
36	3158.3102	6157.028	10.34595	2095.574	1532.056
37	3075.0907	6221.890	10.34595	2095.574	1532.056
38	2975.5635	6296.039	10.34595	2095.574	1532.056
39	2863.1456	6383.449	10.34595	2095.574	1532.056
40	2749.0952	6480.299	10.34595	2095.574	1532.056
41	2589.3759	6607.048	10.34595	2095.574	1532.056
42	2434.1209	6751.675	10.34595	2095.574	1532.056
43	2233.3059	6896.533	10.34595	2095.574	1532.056
44	2032.3136	7069.604	10.34595	2095.574	1532.056
45	1790.5838	7245.569	10.34595	2095.574	1532.056
46	1494.0361	7467.358	10.34595	2095.574	1532.056
47	1222.7035	7717.920	10.34595	2095.574	1532.056
48	935.6067	7976.412	10.34595	2095.574	1532.056
49	607.7411	8239.596	10.34595	2095.574	1532.056
50	230.6355	8548.008	10.34595	2095.574	1532.056
51	-134.8462	8878.797	10.34595	2095.574	1532.056
52	-528.6151	9197.893	10.34595	2095.574	1532.056
53	-933.0676	9517.106	10.34595	2095.574	1532.056
54	-1369.1166	9890.554	10.34595	2095.574	1532.056
55	-1821.5922	10290.459	10.34595	2095.574	1532.056

Figure 7. A table that contains the upper and lower bounds for the Re78 (income) for each age when other variables — education, Re74, and 75 — are held at their means for the control units.

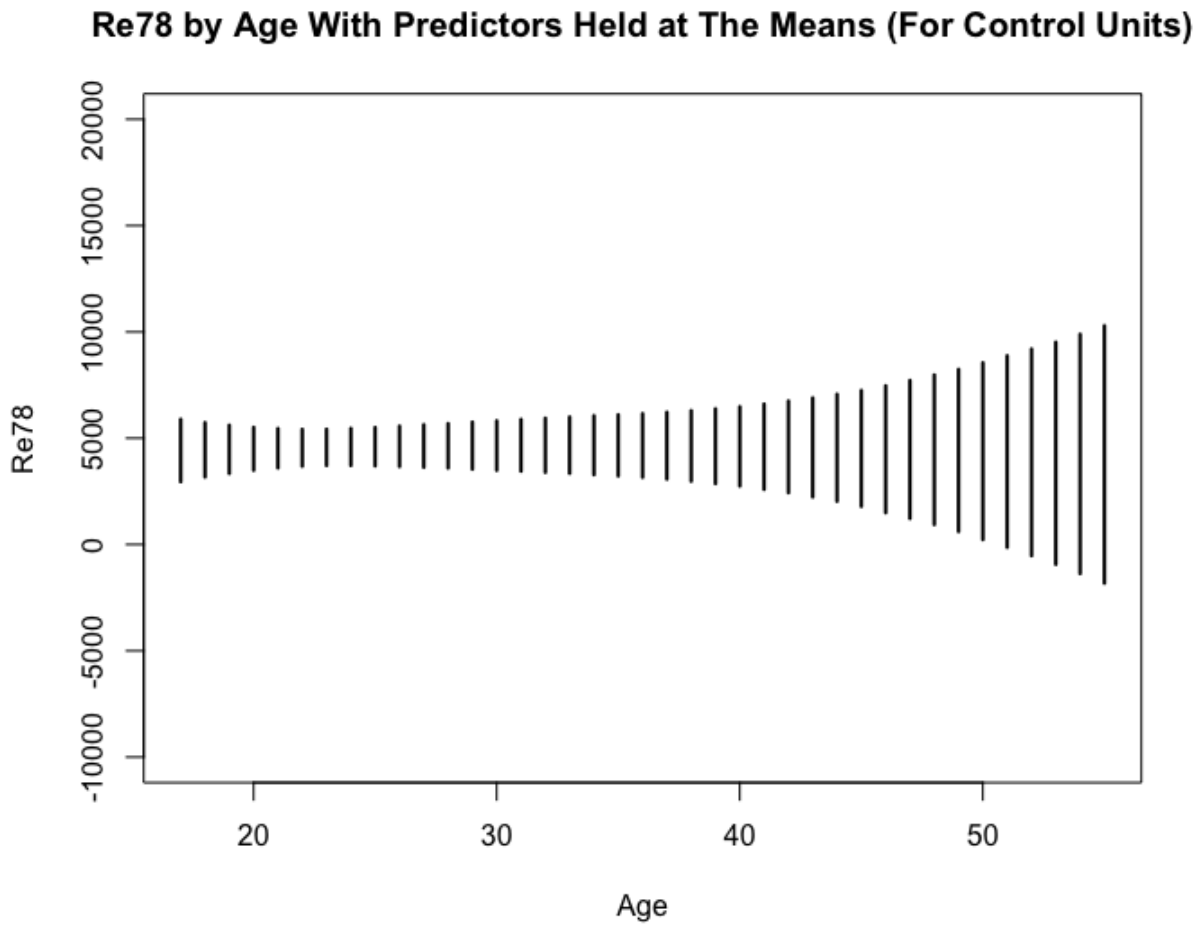


Figure 8. A graph that shows the 95% confidence interval of expected values for re78 for different ages of control units, where predictors are held at their means. It means that for every age, we can be 95% certain that the confidence interval contains the true mean of the re78.

2c. 95% interval of expected values for the treatment effect holding educ, re74, and re75 at their means

	Mean.PI.Lower.Bound	Mean.PI.Upper.Bound
17	-923.169545	2942.013
18	-699.668263	2898.693
19	-487.442705	2867.610
20	-276.731029	2845.204
21	-97.614091	2835.771
22	79.673545	2836.563
23	247.860434	2862.533
24	399.740454	2906.288
25	521.517314	2948.723
26	595.018582	3029.316
27	650.896009	3138.465
28	693.891511	3273.895
29	716.114291	3424.730
30	719.795948	3601.325
31	708.887267	3791.697
32	688.234158	3994.322
33	645.674099	4204.395
34	617.727079	4434.204
35	564.701741	4654.074
36	503.562597	4892.619
37	465.657713	5135.249
38	413.658395	5372.283
39	357.921717	5619.239
40	299.162458	5856.221
41	229.911719	6079.573
42	163.990314	6346.275
43	78.956989	6601.620
44	7.595624	6849.296
45	-69.697579	7110.359
46	-157.334293	7366.157
47	-231.297658	7598.686
48	-298.702518	7855.465
49	-372.789940	8111.573
50	-457.641069	8364.104
51	-546.048528	8626.083
52	-631.070915	8906.761
53	-710.123413	9156.835
54	-778.335048	9432.358
55	-864.166495	9696.139

Figure 9. A table that contains the upper and lower bounds for the treatment effect when — holding educ, re74, and re75 at their means.

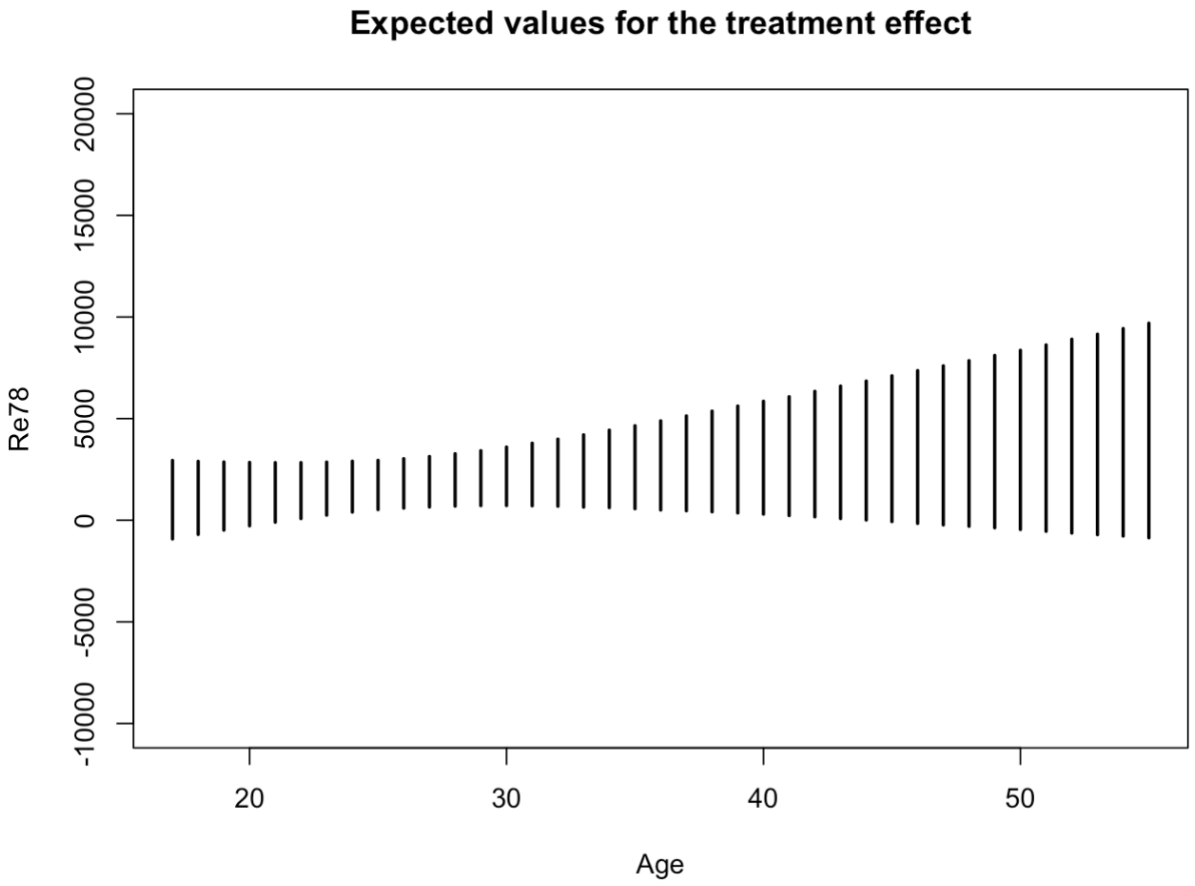


Figure 10. A graph that shows the 95% confidence interval of expected values for treatment effect for different ages, where predictors are held at their means. It means that for every age, we can be 95% certain that the confidence interval contains the true mean of the treatment effect.

2d. 95% interval of expected values for the treatment effect, holding educ, re74, and re75 at their medians and including the sigmas

	Mean.PI.Lower.Bound	Mean.PI.Upper.Bound
17	-100994.45	105372.9
18	-101522.53	106038.2
19	-101765.96	104106.1
20	-100879.36	103162.1
21	-99123.40	105159.5
22	-101798.22	105337.0
23	-103825.93	103846.4
24	-101536.04	105090.2
25	-101116.51	105538.6
26	-98968.23	105408.0
27	-101944.57	105078.0
28	-101647.40	106522.7
29	-100950.31	104420.8
30	-98299.74	103356.3
31	-97552.60	103973.2
32	-100503.17	105733.3
33	-98281.27	107071.3
34	-99112.93	105509.5
35	-100699.63	103743.6
36	-100528.98	106039.9
37	-99189.13	108051.2
38	-98873.06	104038.8
39	-99482.54	104324.8
40	-100191.01	105783.1
41	-99820.97	107672.8
42	-100675.72	105052.5
43	-98620.81	104377.2
44	-101712.10	106552.1
45	-101311.90	106203.6
46	-100271.41	106562.6
47	-101263.72	107672.9
48	-98873.36	107158.6
49	-100908.25	108548.4
50	-100536.31	105940.1
51	-99226.48	106692.8
52	-100536.60	110060.2
53	-101135.73	107985.5
54	-98099.36	107844.8
55	-97178.79	107465.4

Figure 10. A table that contains the upper and lower bounds for the treatment effect when — holding educ, re74, and re75 at their medians.

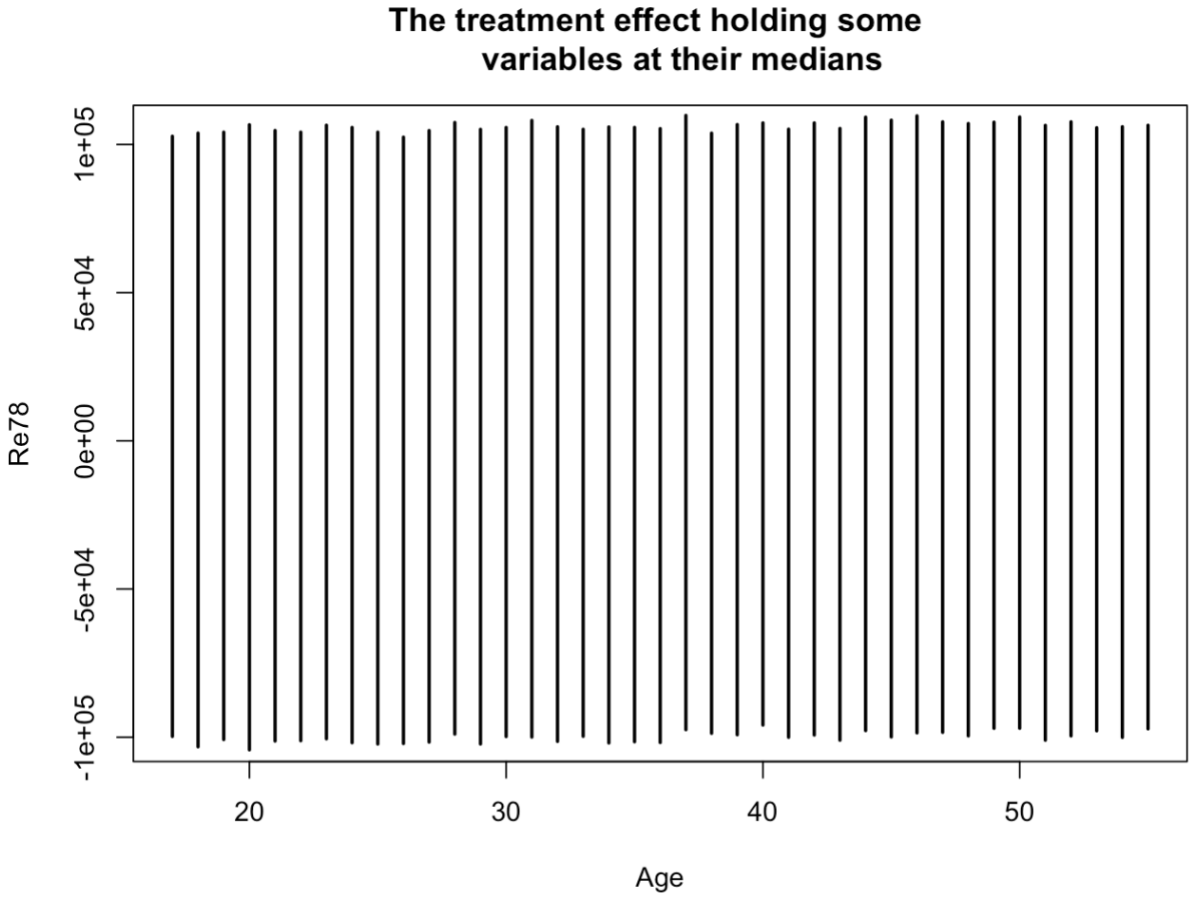
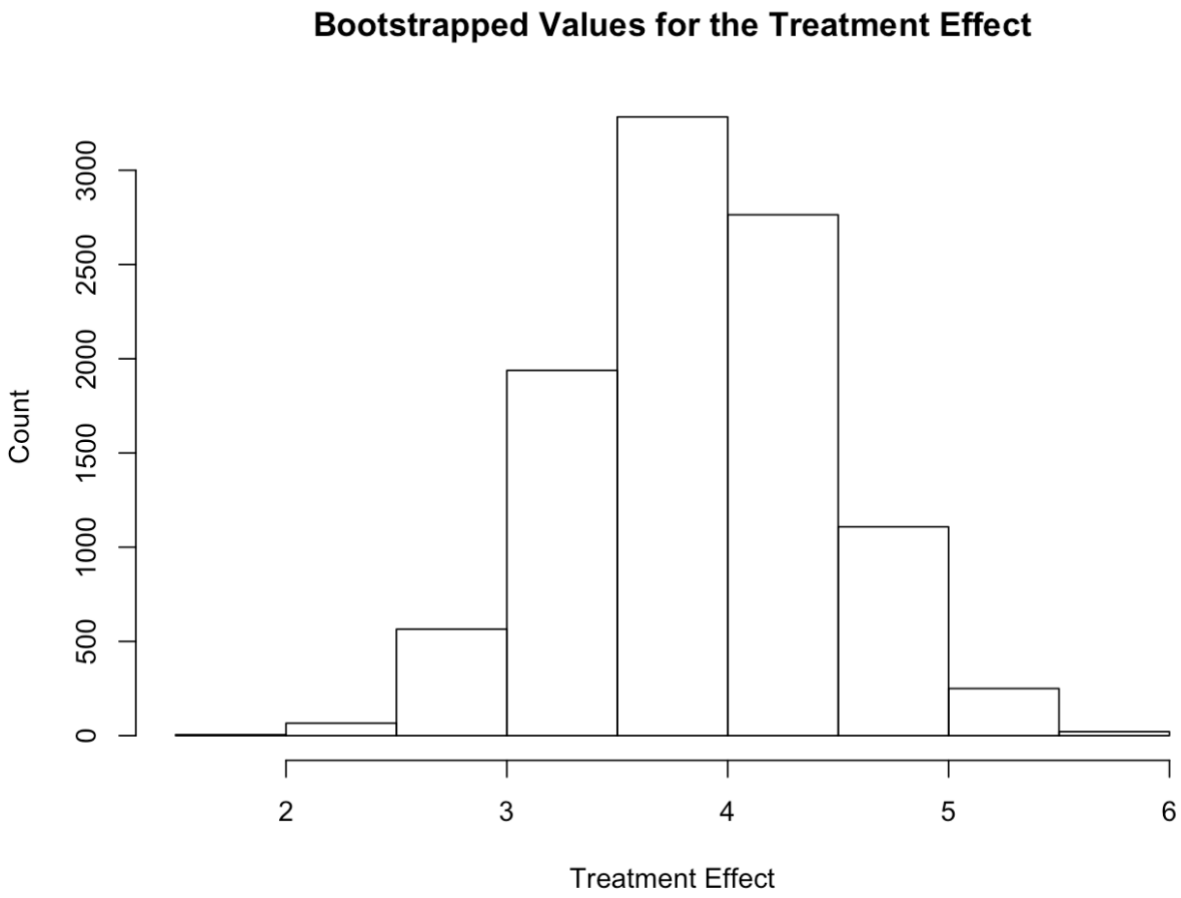


Figure 11. A graph that shows the 95% confidence interval of expected values for treatment effect for different ages, where predictors are held at their medians. It means that for every age, we can be 95% certain that the confidence interval contains the true mean of the treatment effect.

3. Bootstrap the 95% confidence intervals for the value of the coefficient for treatment

The confidence intervals via simulation and formula

	simulation	analytical
2.5%	2.759408	2.758678
97.5%	5.017793	4.998793



Summary

First, the results of the confidence intervals both via simulation and formula are similar (To me, it is helpful as it helps me to be confident that I have done the right thing and that the results are valid). Second, we do the bootstrapping — resampling with replacement — to simulate the real population distribution while we only have the limited data. From this bootstrapping, we can infer that most of the time, in average, the treated students/units improve their math score by 3.285.

4. Bootstrap function that takes Ys and predicted Ys as inputs, and outputs R2

I have two ways to do it and I get similar results.

Bootstrap function #1

```
# Declare bootstrapping function + find the R^2
iterations <- 200000

r_squared2 <- function(true_y, predicted_y) {
```

```
  storage <- rep(NA, iterations)
  for (i in 1:iterations) {
    indices <- sample(1:length(true_y), length(true_y), replace = T)
    new_true_y <- true_y[indices]
    new_pred_y <- predicted_y[indices]
    rss <- sum((new_true_y - new_pred_y)**2)
    tss <- sum((new_true_y - mean(new_true_y))**2)
    storage[i] <- (1 - rss/tss)
  }
  return(storage)
}

storage_rr <- r_squared2(foo.wo.na$MATH_SCORE, lm.foo1$fitted.values)


# Find confidence intervals for the r2
quantile(storage_rr, c(0.025, 0.975))
```

The confidence interval of the R^2 for bootstrap function #1

```
      2.5%       97.5% 
0.008949293 0.034162684 
```

Bootstrap function #2

```
# Create the R^2 function
foo.wo.na <- foo[!is.na(foo$MATH_SCORE), ]
rsquared <- function(ytrue, ypred) {
  rss <- sum((ytrue - ypred)**2)
  tss <- sum((ytrue - mean(ytrue))**2)
  return(1 - rss/tss)
}
lm.foo1 <- lm(MATH_SCORE ~ TREATMENT, data = foo.wo.na)
rsquared(foo.wo.na$MATH_SCORE, lm.foo1$fitted.values)
summary(lm.foo1)$r.sq

# Declare bootstrapping function
iterations <- 20000
storage_r2 <- rep(NA, iterations)

for (i in 1:iterations) {
  boot_idx <- sample(1:nrow(foo.wo.na), nrow(foo.wo.na), replace = T)
  temp_lm <- lm(MATH_SCORE ~ TREATMENT, data = foo.wo.na[boot_idx,])
  fit_y <- temp_lm$fitted.values
  real_y <- foo.wo.na[boot_idx,]$MATH_SCORE
  storage_r2[i] <- rsquared(real_y, fit_y)
}


# Find confidence intervals for the r2
quantile(storage_r2, c(0.025, 0.975))
```

The confidence interval of the R^2 for bootstrap function #2

```
      2.5%      97.5%
0.01148068 0.03647111
```

5. A table that summarizes the 10 different sets of numbers (5 LOOCV estimates, and 5 test set error rates) and summary

Model	Test Error via LOOCV		Mean Squared Error for the Test Set
1	0.01120068	0.01119978	0.007031152
2	0.01583319	0.01583286	0.01039028
3	0.0165831	0.0165828	0.01016406
4	0.01059269	0.01059206	0.008651446
5	0.01439019	0.01438964	0.008969371

I used 1000 random train data set for LOOCV, and generated the test errors above. Comparing the test error from the training data with the test set data, the values are generally similar. The test error LOOCV has slightly higher value than the MSE on test set which we can infer that LOOCV has relatively been effective in not overestimating the test error.

However, to find the error rate for this problem where the *treat* is the dependent variable is less useful as it has binary values. Therefore, instead of looking for test errors, we could also create the confusion matrix where we can check if the model could classify properly. For instance, I tried to create the confusion matrices for the train set and test set for one model (model 1), and they give these results:

The confusion matrix for the train set

```
predicted_ys.1    0    1
             0 1977   16
             1    3    4
```

The confusion matrix for the test set

predicted_yn.1	0	1
0	1973	20
1	6	1

From the confusion matrices above we can compute the misclassification rate for both:

a. misclassification rate for train set: (3+16)/2000 = 0.95%

b. misclassification rate for test set: (6+20)/2000 = 1.3%

Github link:

anggunberlian/cs112
https://github.com/anggunberlian/cs112/blob/master/CS112%20Assignment%202.R