Dennis Kageni

CS112 - Spring 2019

Regression and Bootstrapping

GitHub:

https://github.com/kagenidennis/CS112-Spring-19/blob/master/CS112%20Assignment%202_FInal.ipynb

1. Write your own original code that produces a dataset that conforms to the classic univariate regression model. Your data set should have 999 observations and a Normal error term. The slope of the coefficient on your regressor should be positive. Now include a single outlier, such that when you fit a regression to your 1000 data points, the slope of your regression line is negative. Your answer to this question should consist of:

    (a) Your original data-generating equation

b = 2.5*a + 1 + normal error term

    (b) Regression results for the original 999 (copy/paste the "summary" output)

```
Call:
lm(formula = b ~ a, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.2548 -2.0701  0.0363  2.0550  8.5877 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.86793    0.09570   9.069   <2e-16 ***
a            2.53599    0.09262  27.380   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.024 on 997 degrees of freedom
Multiple R-squared:  0.4292,    Adjusted R-squared:  0.4286 
F-statistic: 749.6 on 1 and 997 DF,  p-value: < 2.2e-16
```

Figure 1. Regression results of the original data generating function b = 2.5*a + 1 + *e*

(c) Regression results with the outlier included.

```
Call:
lm(formula = b ~ a, data = data2)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.359  -3.039  -0.049   2.859  63.755 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.98152    0.15110   6.496  1.3e-10 ***
a           -0.70528    0.08068  -8.741  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.776 on 998 degrees of freedom
Multiple R-squared:  0.07112,   Adjusted R-squared:  0.07019 
F-statistic: 76.41 on 1 and 998 DF,  p-value: < 2.2e-16
```

Figure 2. Regression results of the data generating function that includes the outlier

(d) A properly-labeled data visualization that shows the regression line based on the original 999 points, and another differentiated regression line (on the same axes) based on 1000 points.

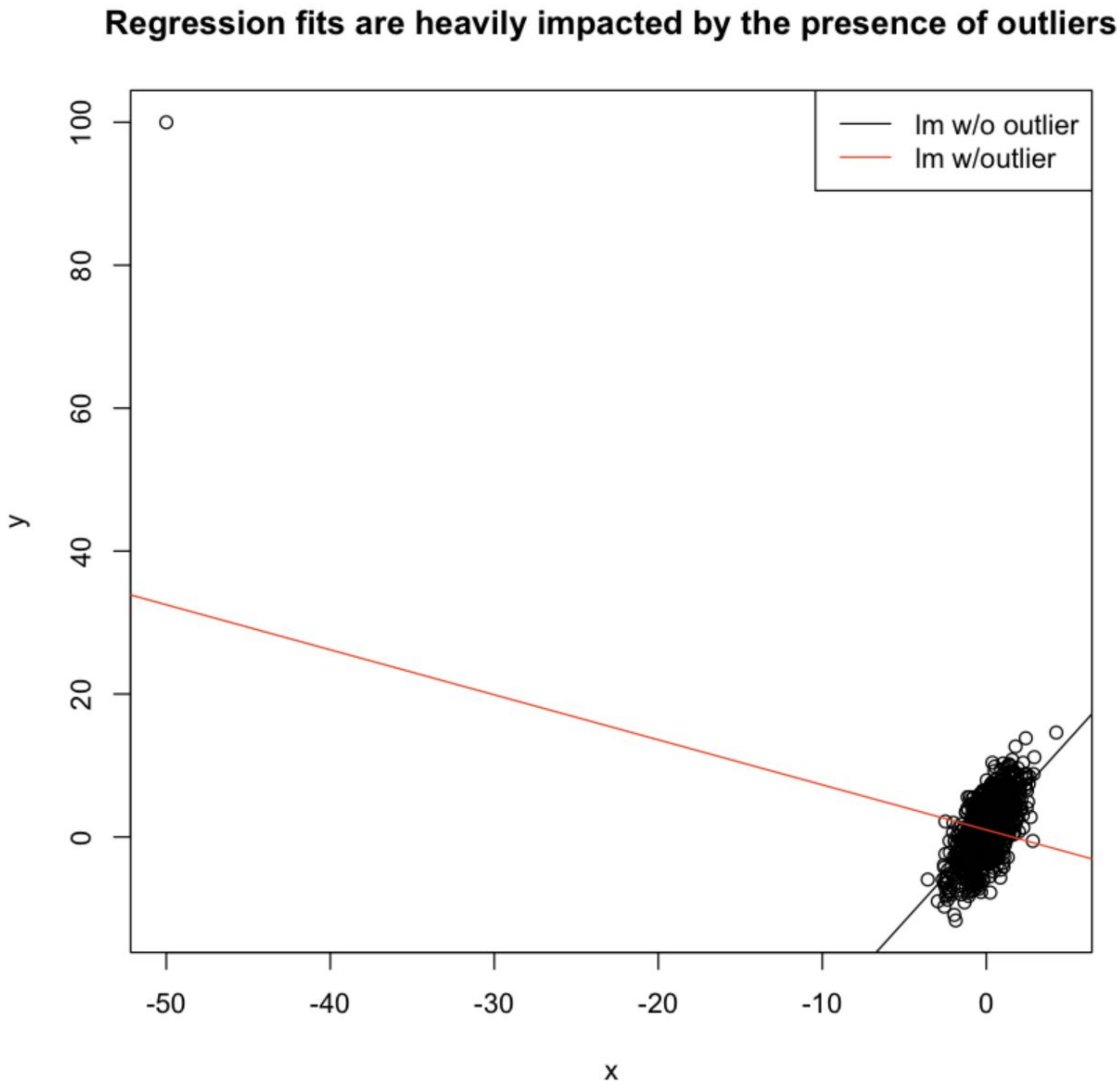


Figure 3. Including the single data point, x=-50 and y=100 has a drastic effect on the regression line and that would be misleading

(e) No more than 3 sentences that would serve as a caption for your figure if it were to be included in an econometrics textbook to illustrate the dangers of extrapolation.

Figure 1 shows the danger of extrapolating a regression to include data that is beyond the range of values in the data set. Since regression is an inferential procedure, the conclusions drawn from such an extrapolation would be inaccurate.

2. *NOTE: FOR THIS PROBLEM (AND THIS PROBLEM ONLY), USE ONLY THE CONTROL GROUP. DO NOT USE ANY UNITS FOR WHICH TREATMENT == 1.*

   Using the Lalonde data set and a linear model that predicts re78 as a linear additive function of age, educ, re74, re75, educ*re74, educ*re75, age*re74, age*re75, age*age, and re74*re75, estimate:

   - the 95% interval of expected values for re78, for every unit (i.e., each age 17-55, spanning the age range in the data set), using simulation (i.e., 10000 simulated predictions for every row from 10000 sets of coefficients). You should not incorporate simulated sigmas, and you should hold educ, re74, and re75 at their medians. Even include ages that are not covered by the data (e.g., 47, 49, etc.).

   - the 95% interval of expected values for re78, for every unit, using simulation (i.e., 10000 simulated predictions for every row from 10000 sets of coefficients). You should not incorporate simulated sigmas, and you should hold educ, re74, and re75 at their 75% quantiles.

   - the 95% prediction interval for re78, for every unit (i.e., each age, spanning the age range in the data set), using simulation (i.e., 10000 simulated predictions for every row from 10000 sets of coefficients). You will need to incorporate simulated sigmas, and you should hold educ, re74, and re75 at their medians.

   - the 95% prediction interval for re78, for every unit, using simulation (i.e., 10000 simulated predictions for every row from 10000 sets of coefficients). You will need to incorporate simulated sigmas, and you should hold educ, re74, and re75 at their 75% quantiles.

Your answer to this question should consist of the following:

(a) A table with the relevant point estimates (e.g., the bounds of the prediction intervals of y for the different ages, and the medians of the other predictors)

(b) 1 figure for the 2 interval analyses with expected values, and 1 figure for the 2 interval analyses with predicted values. The "scatterplots" don't have to show the original data--all I am interested in are the prediction intervals for each age. Each of these figures should show how the intervals change over time (i.e., over the range of ages in the data set). Be sure to label your plot's features (axis, title, etc.).

E.g.: https://gist.github.com/diamonaj/75fef6eb48639c2c36f73c58d54bac2f

Kindly refer to the Appendix.

3. Obtain the PlantGrowth dataset in R.

   Specify a regression model in which the dependent variable is *weight* and the independent variable is an indicator of treatment1 (set the value = 1) or control (set the value = 0). This means you will discard observations associated with treatment2.

   Then, bootstrap the 95% confidence intervals for the value of the coefficient for treatment. Then, obtain the analytical confidence interval for the coefficient value using the standard error that pops out of a regression (or equivalently, in R, you can use the *confint* function). Compare the two confidence intervals--one obtained via simulation, the other via the formula.

   NOTE: Make sure that you don't use a 'canned' bootstrap function -- please code the bootstrap routine manually.

   Your answer to this question should consist of the following:

   (a) A table with the relevant results (bounds on the 2 confidence intervals).

	simulated	analytical
2.5%	-0.5483198	-0.5443617
97.5%	0.1355345	0.1504124

Figure 4. Table comparing the 95th Percentile Analytical and Bootstrapped Confidence Intervals

(b) 1 histogram (properly labeled) showing your bootstrap-sample results. How you do this one is up to you.

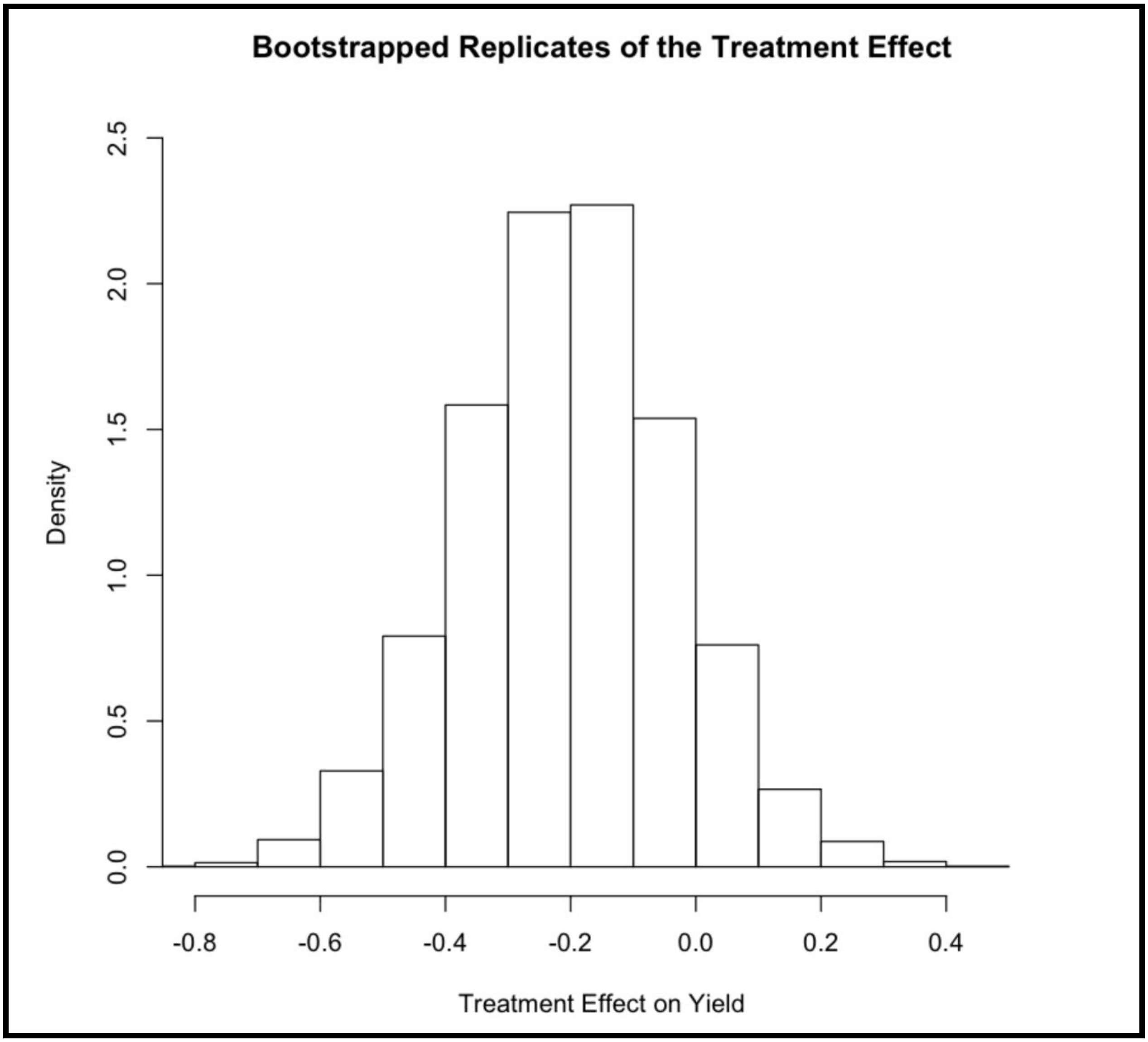


Figure 5. Histogram for the bootstrap values *weight* coefficient in the PlantGrowth dataset

(c) No more than 3 sentences summarizing the results and drawing any conclusions you find relevant and interesting.

From the table in (a), it's observed that the bootstrapped confidence interval values are similar to the analytical values. Therefore, the bootstrap allows us to get CIs on parameters without having to make unrealistic assumptions about the distribution where the data is obtained; as is the case with the analytical methods.

4. Write your own function (5 lines max) that takes Ys and predicted Ys as inputs, and outputs R^2. Copy/paste an example using the *PlantGrowth* data (from #3 above) that shows it working.

```
r_squared <- function(actual_y, predicted_y)
    {
  RSS <- sum((actual_y - predicted_y)**2)
  TSS <- sum((actual_y - mean(actual_y))**2)
  return(cor(actual_y, predicted_y)**2)
    }
lm_plantgrowth <- lm(group~weight, data = plantgrowth_new)
pred_reg <- predict(lm_plantgrowth)

rsquared_1 <- r_squared(plantgrowth_new$group,pred_reg)
rsquared_2 <- summary(lm_plantgrowth)$r.squared

rsquared_1
rsquared_2
```

```
0.0730775989903854
0.0730775989903855
```

Figure 6. A function that outputs values R^2 and compares it to the r.squared R function to prove its result

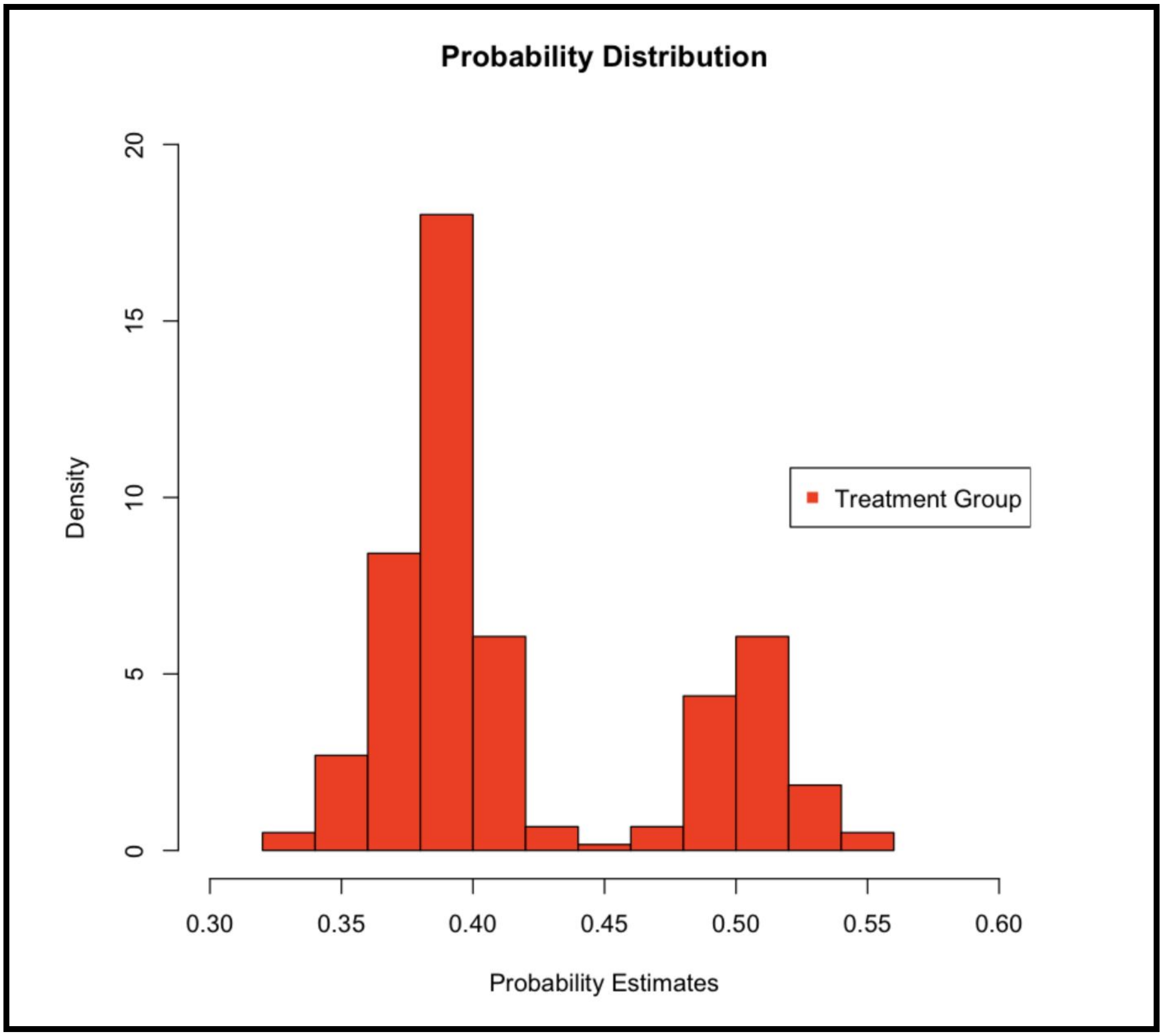
5. Obtain the *nsw.dta* dataset from http://users.nber.org/~rdehejia/data/nswdata2.html.

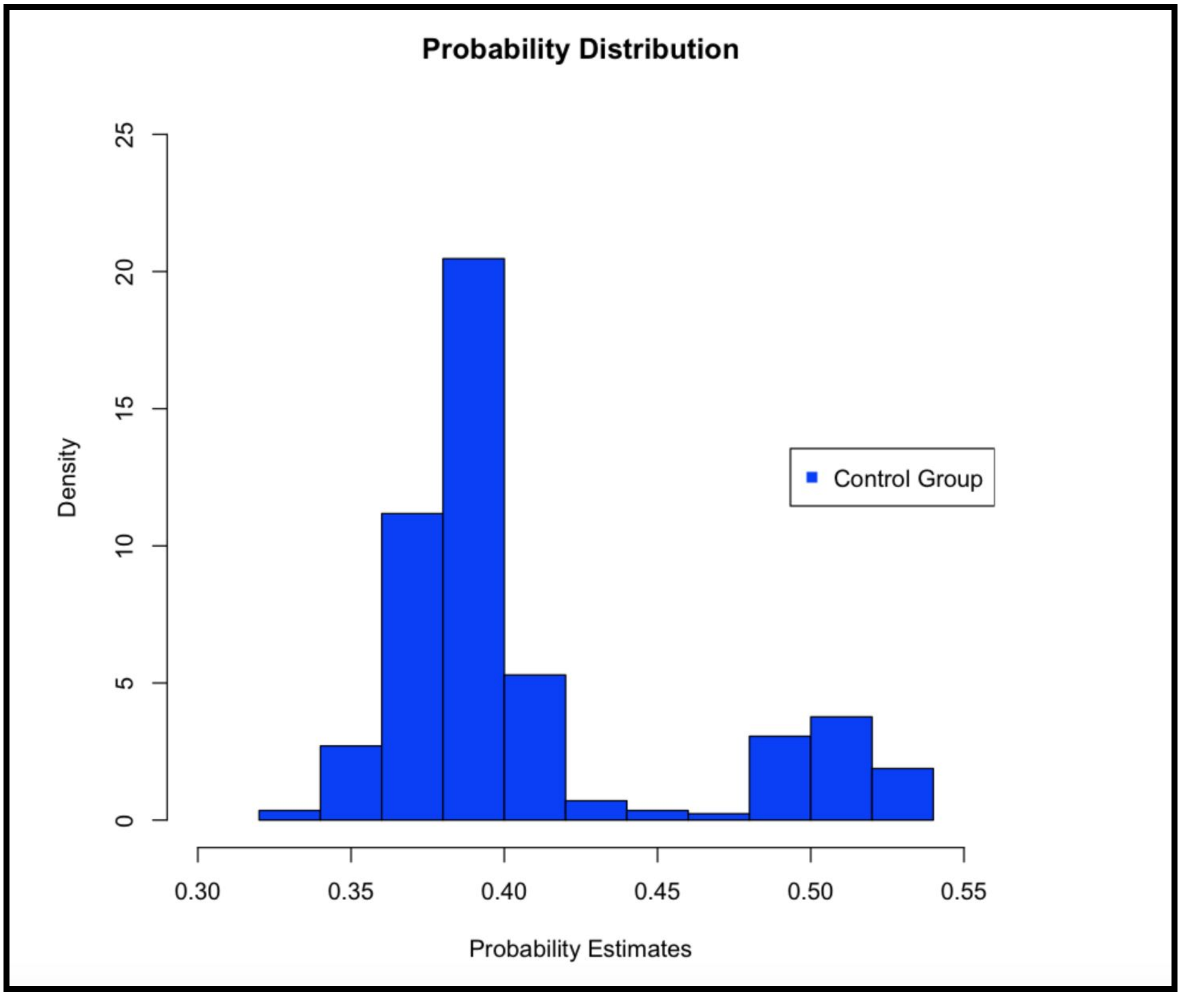
   Read the description of this data set provided on the page. If you proceed with this work in R (recommended) use the *foreign* library to open it (so you can use *read.dta*).

   Use this *nsw.dta* dataset to estimate the probability of being assigned to the treatment group (vs. the control group) for every observation in the data set. Your logistic regression model should be a linear additive function of all predictors available to you -- no interaction terms needed. NOTE: re78 is not a predictor because it postdates the treatment. (In other words, it's an outcome.)

   Your answer to this question should consist of the following:

   (a) Two properly labeled histograms: one in red (showing the distribution of the treatment group's estimated probabilities) and one in blue (showing the distribution of the control group's estimated probabilities). Extra credit for a legend in the plot.



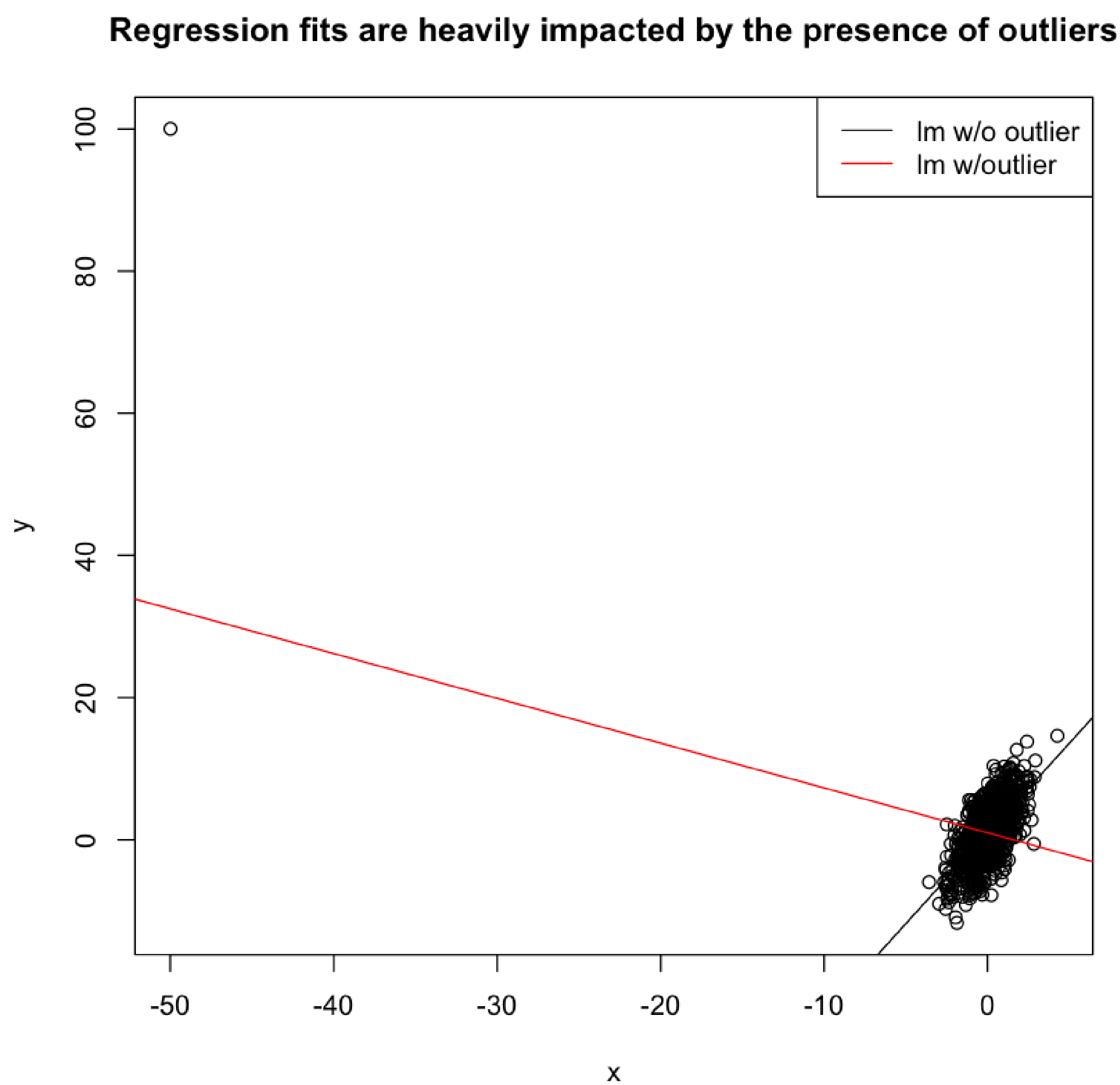


Figures 7 & 8. Histograms for the distribution of probability estimates of the treatment (red) and control groups respectively in the NSW dataset

(b) No more than 3 sentences summarizing the differences between the two distributions of estimated probabilities, and whether/not your results are surprising and/or intuitive.

The histograms are similar which is surprising. The treatment and control data sets are unevenly split (295 vs. 425), so I would expect to have different probability distributions. This implies that the logistic regression model is inaccurate.

In [8]:
```
plot(data2, xlab = "x", ylab = "y", main = "Regression fits are heavily
 impacted by the presence of outliers")
abline(lm_data)
abline(lm_data2, col="red")
legend("topright", legend=c("lm w/o outlier" ,"lm w/outlier"),
       col=c("black", "red"), lty=1, cex=1)
```



Question 2

In [49]:
```
library(Matching)
library(arm)
library(dplyr)
data(lalonde)
```

In [50]:
```
new_lalonde <- lalonde%>%filter(treat!=1)
lm.lalonde <- lm(re78 ~ age + educ + re74 + re75 + educ * re74 + educ *
re75 + age *
       re74 + age * re75 + re74 * re75,
     data = new_lalonde)
summary(lm.lalonde)
```

```
Call:
lm(formula = re78 ~ age + educ + re74 + re75 + educ * re74 +
    educ * re75 + age * re74 + age * re75 + re74 * re75, data = new_lal
onde)

Residuals:
   Min     1Q Median     3Q    Max
 -7264  -4148  -1590   3014  33846

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.686e+03  2.630e+03   1.401    0.162
age          2.216e+00  5.206e+01   0.043    0.966
educ         3.907e+01  2.292e+02   0.170    0.865
re74        -1.552e-02  4.790e-01  -0.032    0.974
re75         7.845e-01  1.878e+00   0.418    0.677
educ:re74    3.441e-02  6.671e-02   0.516    0.606
educ:re75   -7.204e-02  1.341e-01  -0.537    0.592
age:re74    -5.705e-03  2.783e-02  -0.205    0.838
age:re75     9.309e-03  4.212e-02   0.221    0.825
re74:re75   -2.294e-05  1.413e-05  -1.623    0.106

Residual standard error: 5511 on 250 degrees of freedom
Multiple R-squared:  0.02514,	Adjusted R-squared:  -0.009957
F-statistic: 0.7163 on 9 and 250 DF,  p-value: 0.6938
```

In [51]:
```
simulation_lalonde <- sim(lm.lalonde, n.sims = 10000) #to run simulation
```

Part A

In [76]:
```
simulated.ys_median.1 <- matrix(NA, nrow = 10000, ncol = 39)
```

In [78]:
```
re74_median.1 <- median(new_lalonde$re74)
re75_median.1 <- median(new_lalonde$re75)
educ_median.1 <- median(new_lalonde$educ)
for (age in (17:55)) {
  Xs.1 <- c(1, age, educ_median.1, re74_median.1, re75_median.1, educ_me
dian.1*re74_median.1,
          educ_median.1*re75_median.1, age*re74_median.1, age*re75_media
n.1, re74_median.1*re75_median.1)
  for (i in 1:10000) {
    simulated.ys_median.1[i, age + 1 - min(new_lalonde$age)] <- sum(Xs.1
*simulation_lalonde@coef[i,])
  }
}

storage_1 <- 0
for (age in 1:39) {
    storage_1[age] <- median(simulated.ys_median.1[age, ])
    }

median_confint.1 <- apply(simulated.ys_median.1, MARGIN = 2, quantile, p
robs = c(0.025, 0.975))
median_table.1 <- data.frame( "Age"= 17:55, "Median"=storage_1, "Lower Q
uartile_Median" = median_confint.1[1, ], "Upper Quartile_Median" = media
n_confint.1[2, ])
median_table.1
```

Age	Median	Lower.Quartile_Median	Upper.Quartile_Median
17	4290.431	2966.562	5257.639
18	4000.063	3044.739	5194.304
19	4366.493	3124.796	5126.138
20	4619.554	3191.345	5068.367
21	5693.097	3252.990	5019.055
22	4834.486	3307.211	4976.599
23	4420.778	3347.395	4941.726
24	4748.828	3375.942	4913.544
25	4032.068	3381.386	4903.216
26	4651.852	3380.916	4904.685
27	3963.994	3370.365	4930.197
28	5121.214	3350.040	4966.752
29	4205.702	3312.129	5015.956
30	4101.119	3256.959	5078.984
31	3812.408	3198.211	5128.291
32	3771.764	3131.199	5203.623
33	4174.852	3065.206	5275.938
34	5404.141	3000.027	5353.794
35	3674.768	2924.230	5437.169
36	4889.218	2848.157	5523.019
37	4564.086	2762.973	5614.824
38	4603.096	2675.239	5703.142
39	4977.287	2592.640	5792.376
40	4547.461	2507.067	5885.968
41	5245.697	2414.787	5976.600
42	3034.004	2324.540	6070.298
43	5481.220	2242.611	6165.811
44	3830.312	2158.964	6264.888
45	4441.433	2068.990	6366.105
46	5289.993	1979.221	6464.371
47	4187.247	1895.047	6557.294
48	4374.583	1805.216	6658.738
49	3649.424	1714.124	6757.450
50	3853.746	1619.524	6856.655
51	3246.312	1529.142	6956.380
52	3462.897	1429.877	7061.584

Age	Median	Lower.Quartile_Median	Upper.Quartile_Median
53	4959.290	1331.945	7171.502
54	4868.574	1241.284	7277.612
55	3100.057	1131.360	7385.369

Part B

In [65]:
```
simulated.ys_q75.1 <- matrix(NA, nrow = 10000, ncol = 39)
```

In [66]:
```
educ_q75.1 <- quantile(new_lalonde$educ, 0.75)
re74_q75.1 <- quantile(new_lalonde$re74, 0.75)
re75_q75.1 <- quantile(new_lalonde$re75, 0.75)

for (age in (17:55)) {
  Xs_q75.1 <- c(1, age, educ_q75.1, re74_q75.1, re75_q75.1, educ_q75.1*r
e74_q75.1,
          educ_q75.1*re75_q75.1, age*re74_q75.1, age*re75_q75.1, re74_q7
5.1*re75_q75.1)
  for (i in 1:10000) {
    simulated.ys_q75.1[i, age + 1 - min(new_lalonde$age)] <- sum(Xs_q75.
1*simulation_lalonde@coef[i,])
  }
}

storage_2 <- 0
for (age in 1:39) {
    storage_2[age] <- median(simulated.ys_q75.1[age, ])
    }

coinfint_q75.1 <- apply(simulated.ys_q75.1, MARGIN = 2, quantile, probs
= c(0.025, 0.975))
table_q75.1 <- data.frame("Age"= 17:55, "Median" = storage_2, "Lower Qua
rtile" = coinfint_q75.1[1, ], "Upper Quartile" = coinfint_q75.1[2, ])
table_q75.1
```

Age	Median	Lower.Quartile	Upper.Quartile
17	4306.130	3060.866	5494.397
18	4477.172	3149.328	5431.870
19	5044.870	3223.439	5374.617
20	4489.810	3294.456	5315.026
21	5605.834	3364.422	5270.547
22	4493.371	3413.328	5236.383
23	4671.647	3462.284	5205.506
24	5284.061	3495.979	5184.702
25	4152.613	3517.304	5176.210
26	4868.906	3527.305	5185.006
27	4627.675	3517.625	5204.854
28	4727.872	3509.698	5244.595
29	4054.349	3491.291	5292.456
30	4471.509	3448.110	5350.510
31	3459.143	3401.832	5407.688
32	4091.913	3345.489	5478.064
33	4088.888	3288.362	5561.072
34	5652.559	3229.834	5639.064
35	3580.431	3163.490	5725.100
36	4958.771	3092.219	5814.703
37	5406.855	3022.617	5899.417
38	4750.869	2936.655	6001.543
39	4973.155	2860.138	6103.351
40	4545.609	2783.856	6207.449
41	5005.449	2693.400	6300.758
42	3553.983	2609.674	6390.888
43	5741.144	2534.978	6495.670
44	4390.154	2451.446	6597.701
45	4809.063	2360.626	6706.761
46	5006.667	2274.166	6807.883
47	4475.209	2179.870	6917.335
48	4606.451	2099.565	7022.743
49	4546.101	2000.681	7134.698
50	3908.726	1908.736	7240.611
51	3434.963	1821.861	7345.849
52	3779.818	1735.157	7448.281

Age	Median	Lower.Quartile	Upper.Quartile
53	5601.628	1642.972	7552.439
54	4950.664	1548.406	7657.231
55	3580.461	1468.539	7758.785

Part C

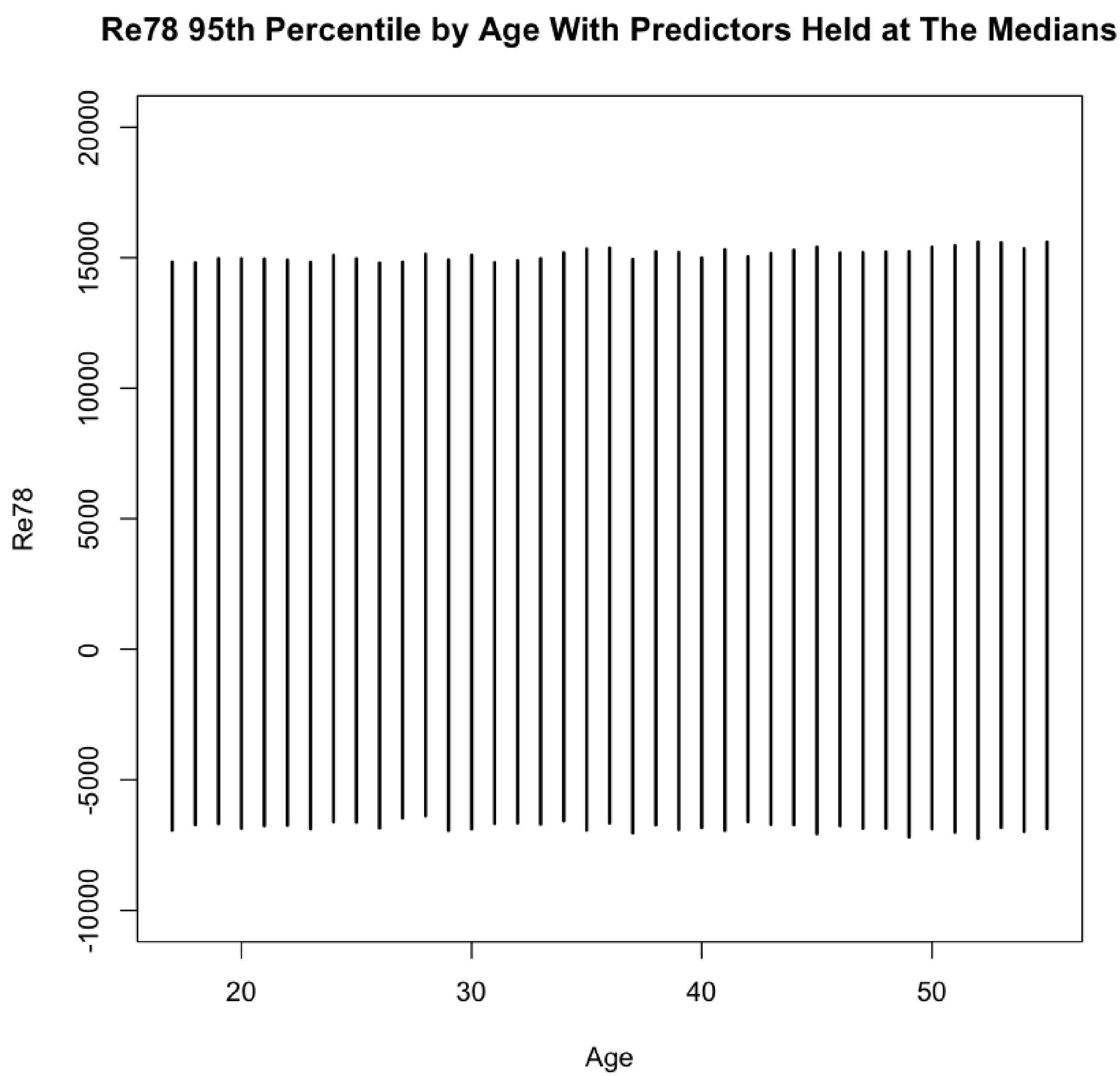
In [69]:
```
simulated.ys_median <- matrix(NA, nrow = 10000, ncol = 39)
```

In [70]:
```
re74_median <- median(new_lalonde$re74)
re75_median <- median(new_lalonde$re75)
educ_median <- median(new_lalonde$educ)
for (age in (17:55)) {
  Xs <- c(1, age, educ_median, re74_median, re75_median, educ_median*re7
4_median,
          educ_median*re75_median, age*re74_median, age*re75_median, re7
4_median*re75_median)
  for (i in 1:10000) {
    simulated.ys_median[i, age + 1 - min(new_lalonde$age)] <- sum(Xs*sim
ulation_lalonde@coef[i,]) +
            rnorm(1, 0, simulation_lalonde@sigma[i])
  }
}

storage_3 <- 0
for (age in 1:39) {
    storage_3[age] <- median(simulated.ys_median[age, ])
    }

median_confint <- apply(simulated.ys_median, MARGIN = 2, quantile, probs
= c(0.025, 0.975))
median_table <- data.frame( "Age"= 17:55, "Median"=storage_3, "Lower Qua
rtile_Median" = median_confint[1, ], "Upper Quartile_Median" = median_co
nfint[2, ])
median_table
```

Age	Median	Lower.Quartile_Median	Upper.Quartile_Median
17	4899.815	-6745.662	15187.62
18	3768.843	-7090.567	15134.13
19	5097.896	-6698.972	14877.65
20	3698.584	-6803.416	15174.24
21	5237.257	-6963.978	15093.18
22	6781.144	-6684.871	15027.81
23	4797.538	-6710.497	15014.63
24	6929.153	-6710.690	15107.79
25	2763.786	-6809.680	15152.37
26	3809.201	-6833.194	15116.02
27	6219.749	-6804.264	15292.19
28	4346.282	-6653.645	15280.51
29	4843.362	-6849.438	15125.48
30	3160.253	-6787.692	14702.21
31	2765.534	-6506.881	15010.41
32	2189.880	-6476.118	14995.35
33	3266.244	-6555.400	15078.43
34	5582.858	-6720.205	14849.41
35	3504.347	-6726.741	14625.59
36	5473.365	-6940.577	15206.59
37	3918.539	-6976.385	15006.55
38	4101.432	-6759.639	15122.83
39	6564.875	-6620.264	15215.79
40	4153.701	-6964.137	15360.09
41	5126.685	-6646.102	15155.68
42	3368.543	-7001.681	15485.18
43	5178.914	-6523.869	14933.72
44	3670.236	-6988.745	15267.04
45	4837.514	-6862.703	15298.75
46	5965.777	-6872.074	15171.58
47	5986.703	-6787.022	15275.61
48	4419.723	-6851.436	15044.77
49	2662.408	-6873.521	15338.27
50	4867.297	-7000.695	15405.17
51	1663.550	-6752.837	15318.42
52	2827.949	-7090.636	15359.69

Age	Median	Lower.Quartile_Median	Upper.Quartile_Median
53	7492.595	-7146.760	15575.95
54	4542.733	-7010.574	15396.87
55	3135.975	-6734.899	15620.39

In [58]:
```
plot(x = c(1:100), y = c(1:100), type = "n",
     xlim = c(17,55),
     ylim = c(-10000,20000),
     main = "Re78 95th Percentile by Age With Predictors Held at The Med
ians", xlab = "Age",
     ylab = "Re78")

for (age in min(new_lalonde$age):max(new_lalonde$age)) {
  segments(
    x0 = age,
    y0 = median_confint[1, age - min(new_lalonde$age) + 1],
    x1 = age,
    y1 = median_confint[2, age - min(new_lalonde$age) + 1],
    lwd = 2)
}
```



Part D

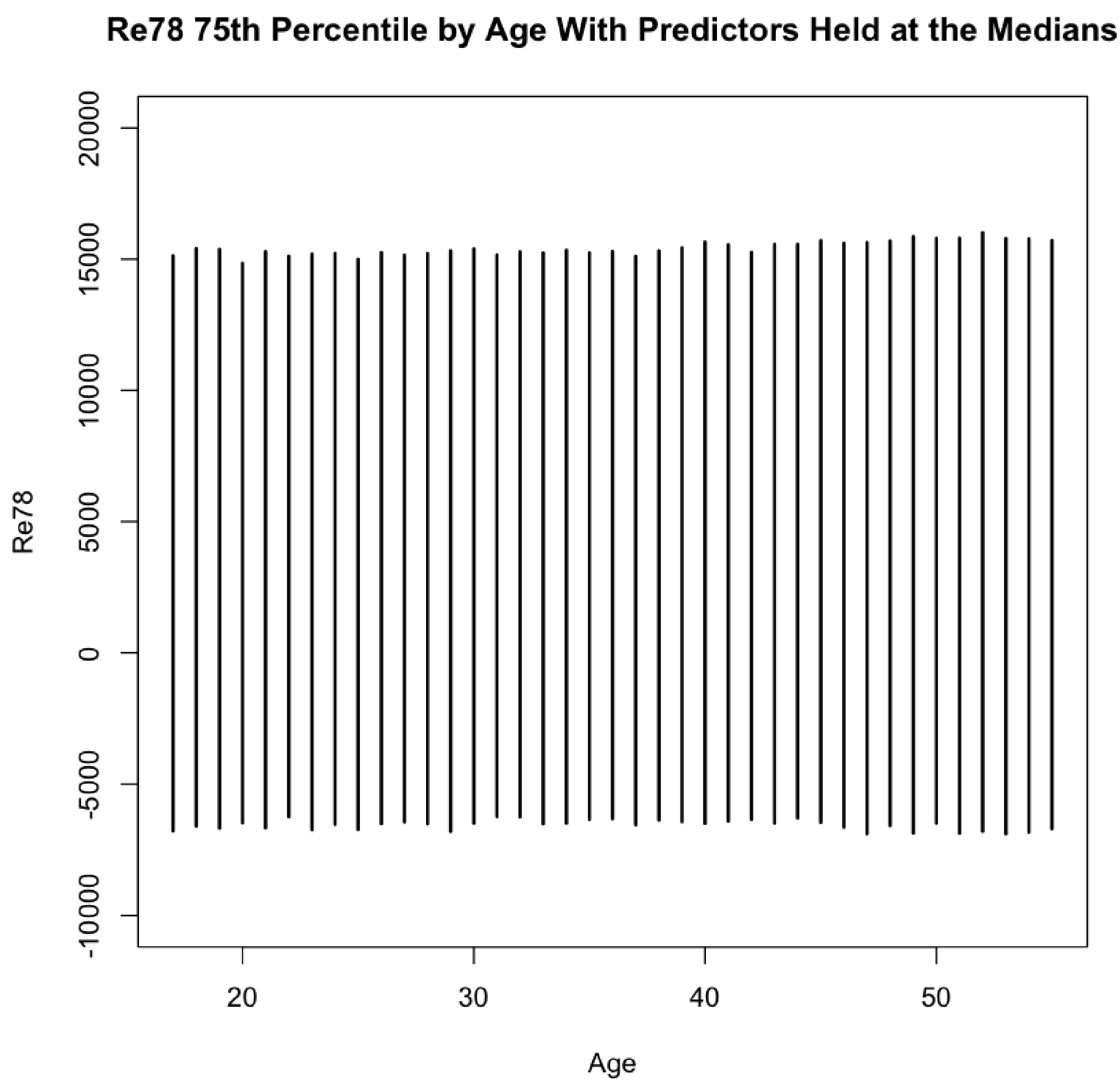
In [59]:
```
simulated.ys_q75 <- matrix(NA, nrow = 10000, ncol = 39)
```

In [67]:
```
educ_q75 <- quantile(new_lalonde$educ, 0.75)
re74_q75 <- quantile(new_lalonde$re74, 0.75)
re75_q75 <- quantile(new_lalonde$re75, 0.75)

for (age in (17:55)) {
  Xs_q75 <- c(1, age, educ_q75, re74_q75, re75_q75, educ_q75*re74_q75,
          educ_q75*re75_q75, age*re74_q75, age*re75_q75, re74_q75*re75_q
75)
  for (i in 1:10000) {
    simulated.ys_q75[i, age + 1 - min(new_lalonde$age)] <- sum(Xs_q75*si
mulation_lalonde@coef[i,]) +
            rnorm(1, 0, simulation_lalonde@sigma[i])
  }
}

storage_4 <- 0
for (age in 1:39) {
    storage_4[age] <- median(simulated.ys_q75[age, ])
    }

coinfint_q75 <- apply(simulated.ys_q75, MARGIN = 2, quantile, probs = c(
0.025, 0.975))
table_q75 <- data.frame("Age"= 17:55, "Median" = storage_4, "Lower Quart
ile" = coinfint_q75[1, ], "Upper Quartile" = coinfint_q75[2, ])
table_q75
```

Age	Median	Lower.Quartile	Upper.Quartile
17	5588.0732	-6776.858	15121.47
18	5191.6143	-6600.940	15400.65
19	3851.1983	-6677.604	15373.97
20	5331.2648	-6478.216	14853.47
21	5729.7570	-6666.583	15292.78
22	3492.7973	-6244.768	15106.49
23	3919.9962	-6731.435	15182.76
24	5288.5170	-6540.712	15217.09
25	3476.7884	-6726.106	14993.56
26	5689.1091	-6512.992	15255.39
27	4090.3057	-6433.525	15147.30
28	6216.7657	-6517.434	15208.22
29	4342.9335	-6794.603	15323.33
30	2471.9528	-6486.586	15389.66
31	2331.5627	-6240.116	15153.84
32	2205.5208	-6251.554	15286.87
33	4278.0438	-6520.206	15238.10
34	6538.6018	-6487.070	15340.34
35	2508.4734	-6345.126	15244.27
36	6335.0596	-6317.367	15298.09
37	5705.5103	-6553.553	15100.91
38	3573.5497	-6366.024	15318.80
39	4024.4775	-6429.013	15435.41
40	3653.7847	-6498.475	15657.09
41	4685.6342	-6404.220	15540.89
42	3871.7325	-6341.474	15264.38
43	3887.2693	-6488.228	15558.58
44	4072.4338	-6287.380	15556.16
45	6410.1398	-6451.261	15707.96
46	6031.2603	-6642.605	15609.48
47	4374.2564	-6899.904	15639.68
48	4110.5627	-6582.251	15695.31
49	2905.5132	-6870.063	15856.53
50	2995.7080	-6485.699	15795.02
51	3931.1267	-6880.214	15798.42
52	3655.9066	-6790.798	16001.24

Age	Median	Lower.Quartile	Upper.Quartile
53	5541.7299	-6900.180	15787.24
54	4482.7215	-6829.033	15779.04
55	486.3455	-6697.542	15710.30

In [68]:
```
plot(x = c(1:100), y = c(1:100), type = "n",
     xlim = c(17,55),
     ylim = c(-10000,20000),
     main = "Re78 75th Percentile by Age With Predictors Held at the Med
ians", xlab = "Age",
     ylab = "Re78")

for (age in min(new_lalonde$age):max(new_lalonde$age)) {
  segments(
    x0 = age,
    y0 = coinfint_q75[1, age - min(new_lalonde$age) + 1],
    x1 = age,
    y1 = coinfint_q75[2, age - min(new_lalonde$age) + 1],
    lwd = 2)
}
```



Question 3