LALONDE 3 WAYS

STATISTICAL ANALYSIS REPORT

YOAV RABINOVICH – OCT 1979

EXECUTIVE SUMMARY

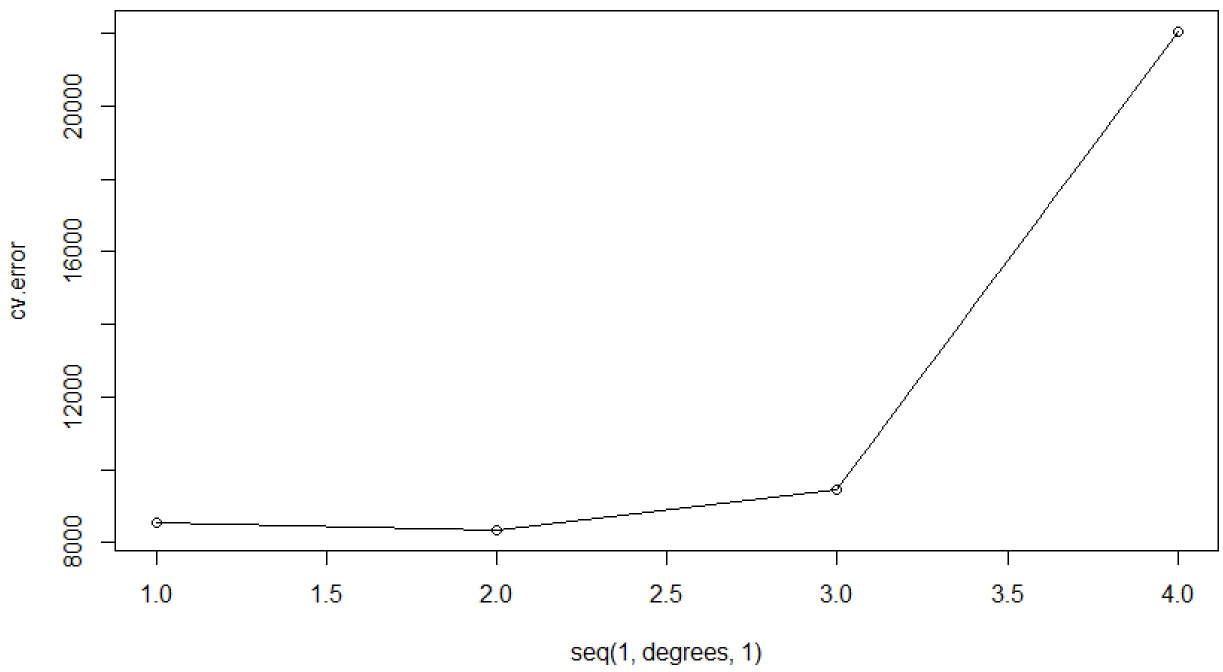
The “Lalonde” intervention in question should be targeted at subjects with high school diplomas. An analysis of our data showed no significant signs of effect of the intervention on subjects who haven’t graduated high school, and a positive treatment effect on subjects who have.

DETAILED REPORT

First, we examined our data using regression methods. Data obtained on subjects after the intervention (1975) was discarded, since there will be no analogous data available when deciding to use the intervention a second time. We left our 20% of the data, chosen randomly, as a “test set” to be used later. The rest of the data was labeled as “training set”, and only that portion can be used as information to inform our construction of a predictive model.

To assess the difference in effect between high school graduates (from no on referred to as ‘high’s) and non-graduates (‘low’s) we separated our data accordingly, and tried to fit a model that could predict a subject’s earnings in ’78 given a set of all other input variables. A linear model predicts the dependent variable as a linear combination of all independent variables, and we used a gradient descent algorithm to find the coefficient for each variable that would minimize the RMSE (root mean squared error) for prediction of each of our observations

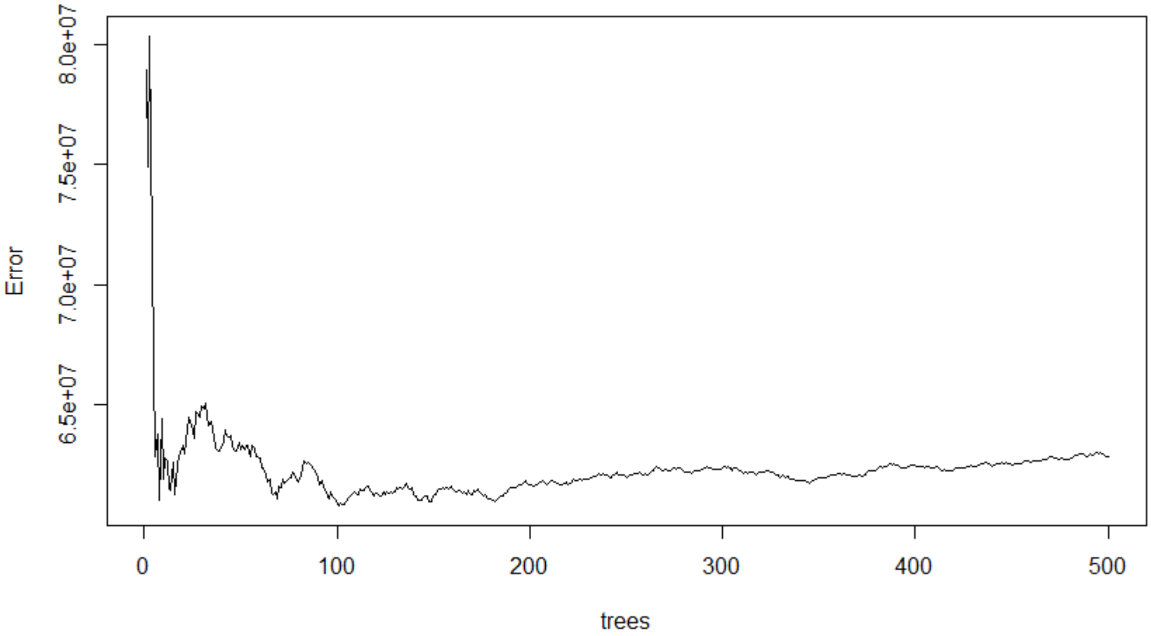
Two of our variables, age and earnings in ’74, that had sufficient unique values to make sense to consider adding to the model some nonlinear combination of them, so we decided to try and see if this modification could significantly reduce error. Through a process of k-fold cross-validation, we iterated over different models fitted using differing degrees of polynomial features based on age and earnings in ’74. For each degree, 10 models were fitted using 10 samples from the original training data as training data, and the rest as test data. We averaged the error for each degree of models to find the optimal degree. (For simplicity, the degree of polynomial used of each of the variables and their interaction variable was kept constant for each model). We found that non-linear interpretations generally increase error for the data, but a small improvement was observed with fitting curves of the second degree.



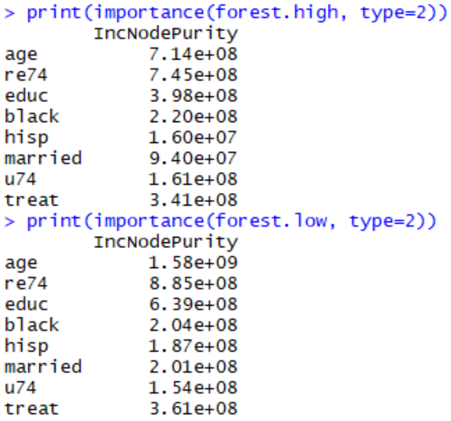
With this information, we could determine that the best fit will be obtained by including the square terms for age, earnings in ’74 and their interaction term. We tested our model against the test set, and found an RMSE value of 6104, corresponding to the same amount in USD. This did not bode well for our model, so we tried to simplify it by getting rid of features that might be introducing noise. Using backward selection, we were able to identify that the terms that contribute significantly to the value of earnings in ’78 were the polynomial features of the ’74 earnings and the presence of treatment. Fitting a model using only these values only managed to reduce our RMSE to 5684. The average treatment effect in regression is constant for all units and it’s represented by the coefficient our model fitted to the treatment variable. We found an effect of 4109, with a confidence interval of 95% between 189 and 6375.

We repeated the process for ‘low’s, this time observing a polynomial degree of 3 as best for our model structure, and found an RMSE of 5132. Our backward selection defined only the status of treatment and the African decent of the subjects significant to predict their earnings, and using only them we found an RMSE of 5049. We estimated average treatment effect as 1287 with a confidence interval of 95% between -262 and 2837. The existence of a constant coefficient for the treatment variable assures us that our regression models assume a constant treatment effect for all units.

The large error values and wide confidence intervals did not allow us to use the models comfortable for prediction, so we decided to try another method. We trained a group of 500 decision trees on the data, each considering a random subset of our features of each of its branches. When trying to use them to predict values, the entire forest voted on an option and the mean of their predictions was used. This method wasn’t much better, giving us RMSE values of 5904 and 5248 for the ‘high’ and ‘low’ sets respectively. By plotting the error over the number of trees consulted, we see that less trees would’ve produced less error, but not enough for us to value their prediction.

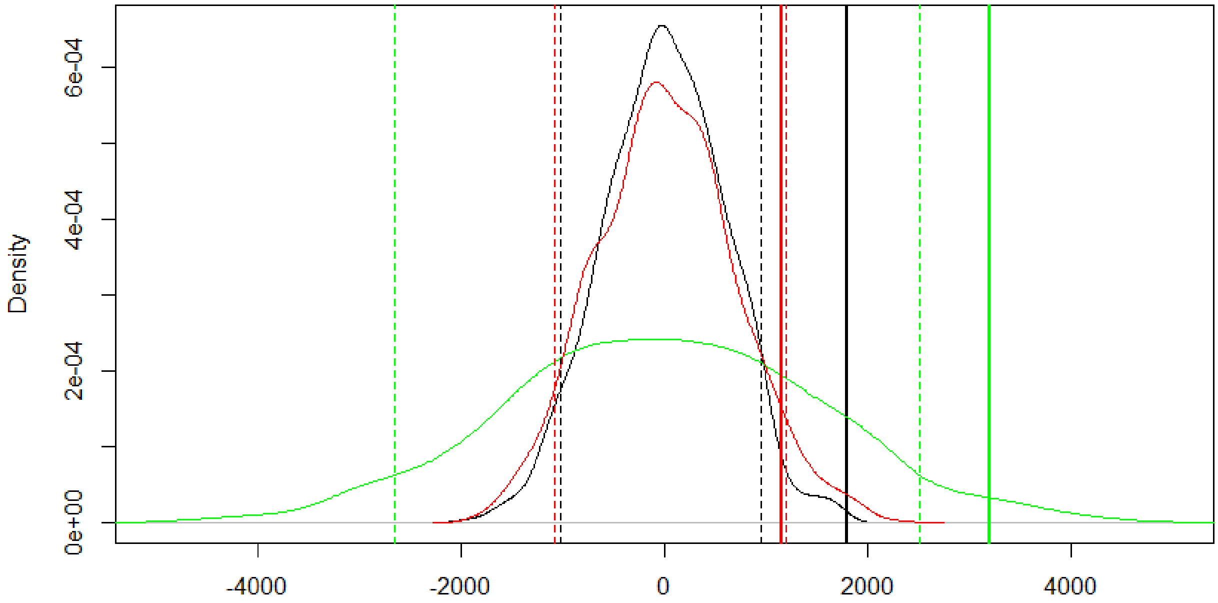


The ‘high’ forest ranked the earnings from ’74 as the most important predictor, in accordance with the regression model, but the ‘low’ forest ranked age as more important, completely in contrast to the regression model which chose treatment and African descent. The most striking observation was that the forests predicted a treatment effect that is bigger for the ‘high’s, contradictory to the regression model. We believe that this might stem from the fact that the forests don’t assume a constant treatment effect for all units, and so outliers (such as subjects who became unemployed after treatment) make the forest predict loss of earnings for similar inputs. If time would’ve allowed, we would’ve tried to clean the dataset through eliminating outliers



```
> print(importance(forest.high, type=2))
        IncNodePurity
age          7.14e+08
re74         7.45e+08
educ         3.98e+08
black        2.20e+08
hisp         1.60e+07
married      9.40e+07
u74          1.61e+08
treat        3.41e+08
> print(importance(forest.low, type=2))
        IncNodePurity
age          1.58e+09
re74         8.85e+08
educ         6.39e+08
black        2.04e+08
hisp         1.87e+08
married      2.01e+08
u74          1.54e+08
treat        3.61e+08
```

Since our predictor models failed us, we turned to a Fisher’s Exact Test to estimate whether we can conclude that the treatment led to any statistically significant effect at all. First, we declared a null hypothesis of zero treatment effect. Next, we simulated a complete randomization of the assignment for treatment for all subjects repeatedly, and measured the difference in mean earnings in ’78 between the new treatment and control if our null hypothesis were true and each subject’s value would’ve remained the same regardless of treatment. We collected a distribution of means, and compared our observed mean to it. We repeated this process for ‘high’s (green) and ‘low’s (red). We found that the real observations fall outside of our 95% percentile, meaning that we can reject the null hypothesis for the entire population and for the ‘high’s, but not for the ‘low’. One would be correct to mention that the ‘high’s group is significantly smaller, and that according to the law of small numbers, we can expect this group to be more likely to represent outliers out of the population, but the fact that the population itself seemed to show a significant treatment effect, we can be more certain of our observation.



We can estimate the effect size of the difference in treatment effect between the two groups by using Cohen’s d, which gives us d=0.25,8, corresponding to a small effect size. We could also try to estimate the treatment effects more accurately, through iterating over different null hypotheses and finding which possible treatment effects fall within the alpha confidence margins we chose for each group. Our team has tried to do that and failed, but theoretically this would give us a set of plausible values for each effect size that we can compare.

In conclusion, we have found no reason to believe that the treatment had any effect on the uneducated, but we believe that it has had some effect on the educated. Therefore, they should be the target group for our next intervention.

APPENDIX

CODE (READY FOR PASTING INTO R-STUDIO

```
library(Matching)
data(lalonde)


#data cleaning
lalonde <- lalonde[sample(nrow(lalonde)),]


#For high school graduates:
lalonde.high <- subset(lalonde, lalonde$nodegr == 0)[,-6][,-7][,-9]
lalonde.low <- subset(lalonde, lalonde$nodegr == 1)[,-6][,-7][,-9]


#split train and test sets
indexes.high = sample(1:nrow(lalonde.high), size=0.2*nrow(lalonde.high))
lalonde.high.train = lalonde.high[-indexes.high,]
lalonde.high.test = lalonde.high[indexes.high,]


#regression
#model selection:
library("DAAG")
modelselection <- function(data, degrees, k){
  cv.error = integer(degrees)
  for(i in 1:degrees){
    #fit regression model
    fit.high <- lm(re78 ~ poly(age,i) +poly(re74,i) +poly(I(re74*age),i)+ educ + black + hisp + married + u74 + treat,data)


    #assess error by cross validation
    cv <- cv.lm(data=lalonde.high, fit.high, m=k)
    cv.error[i] = sqrt(attr(cv,"ms"))
  }
  plot(seq(1,degrees,1), cv.error)
  lines(seq(1,degrees,1),cv.error)
  best <- which.min(cv.error)
  return(best)
}

polynomial = modelselection(lalonde.high.train, 4, 10)
fit.high.best <- lm(re78 ~ poly(age,polynomial) +poly(re74,polynomial) + educ + black + hisp + married + u74 + treat,lalonde.high.train)
predictions <- predict(fit.high.best,lalonde.high.test)
rmse = sqrt(mean((predictions - lalonde.high.test$re78)^2))

#backward selection and refitting

step <- step(fit.high.best, direction = "backward")
fit.high.best <- lm(re78 ~ poly(re74,polynomial)+ treat,lalonde.high.train)
predictions <- predict(fit.high.best,lalonde.high.test)
rmse = sqrt(mean((predictions - lalonde.high.test$re78)^2))

#evaluate treatment effect

averageeffect.high <- coef(fit.high.best)["treat"]
averageeffect.high.ci <- confint(fit.high.best,"treat")

#repeat for non graduates
#split train and test sets
indexes.low = sample(1:nrow(lalonde.low), size=0.2*nrow(lalonde.low))
lalonde.low.train = lalonde.low[-indexes.low,]
lalonde.low.test = lalonde.low[indexes.low,]

#model selection
polynomial = modelselection(lalonde.low.train, 5, 10)
fit.low.best <- lm(re78 ~ poly(age,polynomial) +poly(re74,polynomial) + educ + black + hisp + married + u74 + treat,lalonde.low.train)
predictions <- predict(fit.low.best,lalonde.low.test)
rmse2 = sqrt(mean((predictions - lalonde.low.test$re78)^2))

#backward selection and refitting
```

```
step <- step(fit.low.best, direction = "backward")
fit.low.best <- lm(re78 ~ black + treat,lalonde.low.train)
predictions <- predict(fit.low.best,lalonde.low.test)
rmse2 = sqrt(mean((predictions - lalonde.low.test$re78)^2))

#evaluate treatment effect
averageeffect.low <- coef(fit.low.best)["treat"]
averageeffect.low.ci <- confint(fit.low.best,"treat")
averageeffect.diff = averageeffect.low-averageeffect.high

#decision trees
library("randomForest")
forest.high <- randomForest(re78 ~ age+re7+educ+black+hisp+married+u74+treat,lalonde.high.train)
forest.low <- randomForest(re78 ~ age+re74+educ+black+hisp+married+u74+treat,lalonde.low.train)

averageeffect.forest.high = mean(predict(forest.high,treated.high))-mean(untreated.high$re78)
averageeffect.forest.low = mean(predict(forest.low,treated.low))-mean(untreated.low$re78)
predictions <- predict(forest.high,lalonde.high.test)
rmse3 = sqrt(mean((predictions - lalonde.high.test$re78)^2))
predictions <- predict(forest.low,lalonde.low.test)
rmse3 = sqrt(mean((predictions - lalonde.low.test$re78)^2))
averageeffect.forest.diff = averageeffect.forest.low-averageeffect.forest.high

plot(forest.high)
print(importance(forest.high, type=2))
print(importance(forest.low, type=2))

#bootstrapping
library("boot")

#FET

# Assignment function
assignment <- function(fisher.data) {
  # Shuffle data, establishing random assignment
  fisher.data <- fisher.data[sample(nrow(fisher.data)),]

  treatment.group   <- fisher.data[1:floor(nrow(fisher.data)/2),]
  control.group     <- fisher.data[ceiling(nrow(fisher.data)/2):nrow(fisher.data),]

  return(mean(treatment.group$re78) - mean(control.group$re78))
}

# Iterating the Assignment function
iter.RI <- function(data,effect,iterations = 1000) {
  storage.vector <- NULL
  for (i in 1:iterations)
  {
  storage.vector[i] <- assignment(data)+effect
  }
  return(storage.vector)
}

#checking different hypotheses - can't get it to work:
hypotheses <-function(min,max,data,step = 1,iter = 100){
  quantile.data <- matrix(nrow=3,ncol=((max-min)/step)+1)
  quantile.data[1,] = seq(min,max,step)
  for(i in 1:length(quantile.data[1,])){
    results <- iter.RI(data,i,iterations=iter)
    quantile.data[2,i] = quantile(results, prob = 0.05)
    quantile.data[3,i] = quantile(results, prob = 0.95)
  }
  plot(quantile.data[1,],quantile.data[2,], col="red",ylim=c(-3000,3000),type="l")
  lines(quantile.data[1,],quantile.data[3,], col="green")
  abline(v=mean(data[data$treat==1,]$re78)-mean(data[data$treat==0,]$re78))
  return(quantile.data)
}
```

```
#all
results <- iter.RI(lalonde,0)
alpha <- quantile(results, prob = c(0.95, 0.05))
plot(density(results), xlim=c(-5000, 5000))
yobs = mean(lalonde[lalonde$treat==1,]$re78)-mean(lalonde[lalonde$treat==0,]$re78)
abline(v = yobs, lwd = 2)
abline(v=alpha[1],lty=2)
abline(v=alpha[2],lty=2)
#high
results <- iter.RI(lalonde.high,0)
alpha.high <- quantile(results, prob = c(0.95, 0.05))
lines(density(results), col="green")
yobs.high = mean(lalonde.high[lalonde.high$treat==1,]$re78)-mean(lalonde.high[lalonde.high$treat==0,]$re78)
abline(v = yobs.high, lwd = 2, col = "green")
abline(v=alpha.high[1], col = "green", lty=2)
abline(v=alpha.high[2], col = "green", lty=2)
#low
results <- iter.RI(lalonde.low,0)
alpha.low <- quantile(results, prob = c(0.95, 0.05))
lines(density(results),col="red")
yobs.low = mean(lalonde.low[lalonde.low$treat==1,]$re78)-mean(lalonde.low[lalonde.low$treat==0,]$re78)
abline(v = yobs.low, lwd = 2, col = "red")
abline(v=alpha.low[1], col="red",lty=2)
abline(v=alpha.low[2], col="red",lty=2)

#cohen's d
library("lsr")
cohensD(lalonde.low$re78,lalonde.high$re78)
```