Observational Studies Simulation Homework

Jennifer Hill, Ray Lu & Zarni Htet

```
library(ggplot2)
```

Objective

The goal of this exercise is to learn how to simulate a few different types of observational causal structures and evaluate the properties of different approaches to estimating the treatment effect through linear regression.

Problem Statement

You should be familiar with the assumptions of linear regression (both **structural** and **parametric**) for causal effect estimation. Suppose we want to simulate a simple causal data set from the joint distribution of the covariates, treatment, and potential outcomes.

The data generating process (DGP) is:p(X, Z, Y0,Y1)=p(X)p(Z|X)p(Y1,Y0|Z, X). (As per usual, X is the pretest variable, Z is the treatment variable and Y0 and Y1 are the potential outcomes.)

Part A: Linear Parametric form

Question 1: Simulate the data

(a) Start with the marginal distribution of X. Simulate as X~N(0,1) with sample size of 1000. Set the seed to be.

```
# Setting seed to 1234, given R version 4.0.2
set.seed(1234)
# Set sample size
N = 1000
# Generate 1000 simulated data points with normal distribution
X = rnorm(N,0,1)
```

(b) Look at the DGP. What role does X play? X is the pretest variable. The outcome variables Y0 or Y1 will be dependant on the value of X, as well as the binary value of Z. This effect on the binary of Z will represent the selection bias.

(c) The distribution of binary Z depends on the value of X. Therefore, the next step is to simulate Z from p(Z|X) = Binomial(p), where the vector of probabilities, p, can vary across observations. Come up with a strategy for generating the vector Z conditional on X that forces you to be explicit about how these probabilities are conditional on X (an inverse logit function would be one strategy but there are others). Make sure that X is significantly associated with Z and that the vector of probabilities used to draw Z doesn't vary below .05 or above .95.

```
# Use inverse logit to turn values of X into values of p between 0 and 1
p = 1/(1+exp(-X))
# Creating ceiling for values above 0.95
p[p>0.95] = 0.95
# Creating floor for values below 0.05
p[p<0.05] = 0.05
```

```r
# Use these probability values to generate binary Z
Z = rbinom(N,1,p)
# Showing significance with log regression
summary(glm(Z~X,family = binomial))
```

```
##
## Call:
## glm(formula = Z ~ X, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2469  -0.9866   0.3596   1.0133   2.3785
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.05473    0.07011   0.781    0.435
## X            1.06058    0.08683  12.215   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1386.2  on 999  degrees of freedom
## Residual deviance: 1185.8  on 998  degrees of freedom
## AIC: 1189.8
##
## Number of Fisher Scoring iterations: 3
```

(d) The last step is to simulate Y from p(Y0,Y1|Z,X). Come up with a strategy for simulating each potential outcome with appropriate conditioning on Z and X with the following stipulations.

- (i) Make sure that E[Y(1)|X] - E[Y(0)|X] = 5.
- (ii) Make sure that X has a linear and statistically significant relationship with the outcome.
- (iii) Finally, set your error term to have a standard deviation of 1 and allow the residual standard error to be different for the same person across potential outcomes.
- (iv) Create a data frame containing X,Y,Y0,Y1 and Z.

```r
# Create Y0's
Y0 = X + rnorm(N,0,1)
# Create Y1's where E[Y(1)|X] - E[Y(0)|X] = 5
Y1 = X + rnorm(N,0,1) + 5
# Logic to determine outcome Y based on treatment assignment
Y = ifelse(Z==1,Y1,Y0)
# Combine everything into a data frame
dat = data.frame(pretest=X,treatment=Z,y=Y,y0=Y0,y1=Y1)
# Checking linear and statistically significant relationship
summary(lm(data = dat,Y~pretest))
```

```
##
## Call:
## lm(formula = Y ~ pretest, data = dat)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -5.7591 -2.0075  0.0314  2.1385  6.3793
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.54317    0.07916   32.13   <2e-16 ***
## pretest      2.07182    0.07939   26.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.502 on 998 degrees of freedom
## Multiple R-squared:  0.4056, Adjusted R-squared:  0.405
## F-statistic: 681.1 on 1 and 998 DF,  p-value: < 2.2e-16
```

(e) Turn all of the above steps into a function.

```
dgp1 = function(N,meanX,intercept,coef,seed){
  # Set seed
  set.seed(seed)
  # Generate 1000 simulated data points with normal distribution
  X = rnorm(N,meanX,1)
  # Use inverse logit to turn values of X into values of p between 0 and 1
  p = 1/(1+exp(-X))
  # Creating ceiling for values above 0.95
  p[p>0.95] = 0.95
  # Creating floor for values below 0.05
  p[p<0.05] = 0.05
  # Use these probability values to generate binary Z
  Z = as.factor(rbinom(N,1,p))
  # Create Y0's
  Y0 = intercept + coef*X + rnorm(N,0,1)
  # Create Y1's where E[Y(1)|X] - E[Y(0)|X] = 5
  Y1 = intercept + coef*X + rnorm(N,0,1) + 5
  # Logic to determine outcome Y based on treatment assignment
  Y = ifelse(Z==1,Y1,Y0)
  # Combine everything into a data frame
  dat = data.frame(pretest=X,treatment=Z,y=Y,y0=Y0,y1=Y1)
return(dat)
}
# Check linear relationship and statistically significant relationship
summary(lm(data=dgp1(1000,0,0,1,1234),y~pretest+treatment))
```

```
##
## Call:
## lm(formula = y ~ pretest + treatment, data = dgp1(1000, 0, 0,
##     1, 1234))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0973 -0.7132  0.0314  0.6859  3.1924
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.02349    0.04944  -0.475    0.635
## pretest      1.00206    0.03633  27.583   <2e-16 ***
```

```
## treatment1   5.02614    0.07243  69.392   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.037 on 997 degrees of freedom
## Multiple R-squared:  0.898,  Adjusted R-squared:  0.8978
## F-statistic:  4391 on 2 and 997 DF,  p-value: < 2.2e-16
```

(f) Set your seed to 1234 and generate a dataset of size 1000 from this function. Save it for later.

```
# Since it isn't specified, we're going to generate X with mean zero
# and coefficient of 1, and Y with intercept zero
data1 = dgp1(1000,0,0,1,1234)
```

(g) Think about the difference between the DGP used in this homework and the first DGP from previous homework (completely randomized experiment). How is the difference in the study design encoded?

In the previous assignment, treatment is randomly assigned, independent of the pretest values. In this assignment, the bias in who is selected for treatment is determined by their pretest scores. The values of X are encoded into the generation of the vector Z. This results in differences between the groups other than the assignment of the treatment variable, Z.

(h) Calculate the SATE from (g) (save it for use later).

```
# Calculate sample average treatment effect
sate1 = mean(data1$y1 - data1$y0)
sate1
```

```
## [1] 5.002736
```

Question 2: Playing the role of the researcher Now switch to the role of the researcher for a moment. Pretend someone handed you a dataset generated as specified above and asked you to estimate a treatment effect – for this you will use the dataset generated in 1f above. You will try two approaches: difference in means and regression.

(a) Estimate the treatment effect using a difference in mean outcomes across treatment groups (save it for use later).

```
# Renaming dataset to keep track better
data2 = data1
# Calculate difference of means
difmeans2 = mean(data2$y[data2$treatment==1]) - mean(data2$y[data2$treatment==0])
difmeans2
```

```
## [1] 5.873948
```

(b) Estimate the treatment effect using a regression of the outcome on the treatment indicator and covariate (save it for use later).

```
# Use lm function for linear regression, call coefficients
summary(lm(y ~ pretest + treatment, data2))$coefficient
```

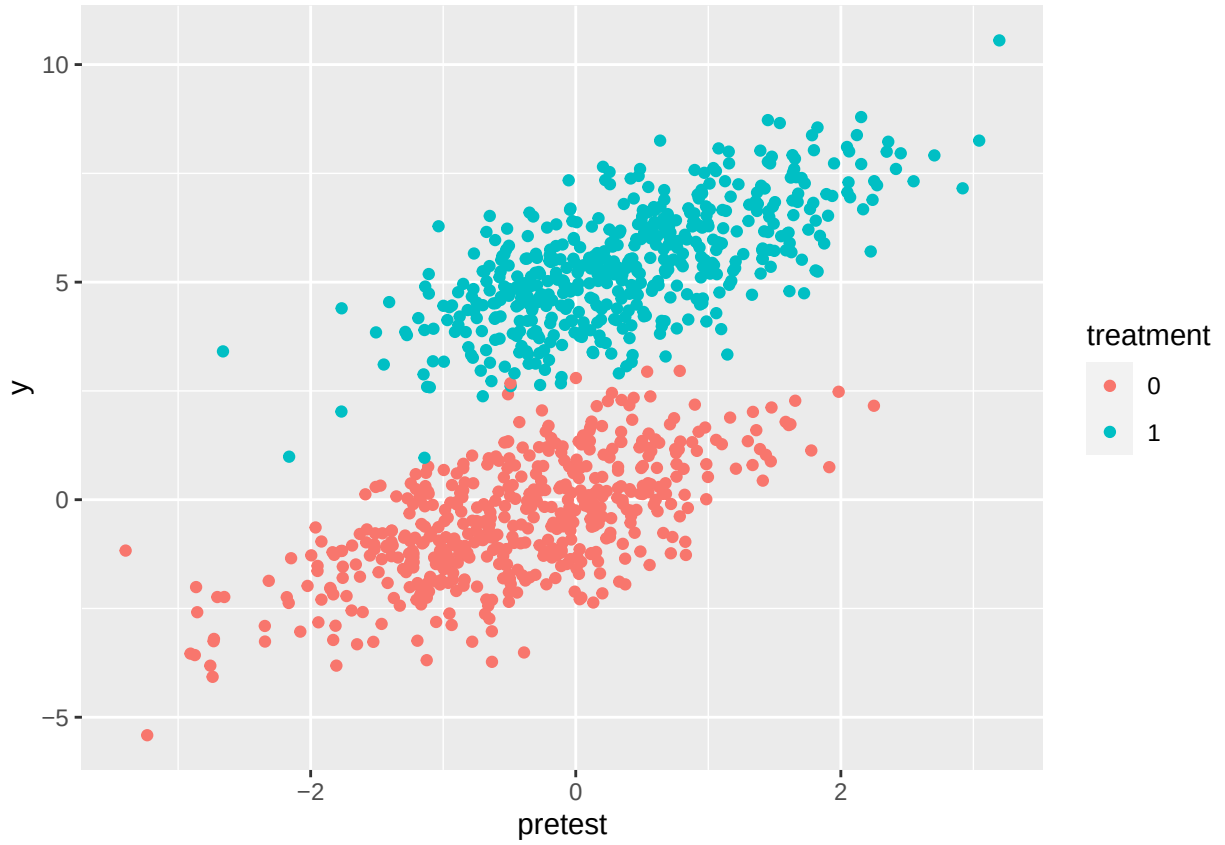
```
##                Estimate Std. Error    t value      Pr(>|t|)
## (Intercept) -0.02348768 0.04943814 -0.4750924  6.348252e-01
## pretest      1.00205515 0.03632855 27.5831308 6.492862e-125
## treatment1   5.02613881 0.07243105 69.3920486  0.000000e+00
```

```
# Call value for estimated treatment effect
linreg2 = summary(lm(y ~ pretest + treatment, data2))$coefficient[3]
linreg2
```

```
## [1] 5.026139
```

(c) Create a scatter plot of X versus the observed outcome with different colors for treatment and control observations (suggested: red for treated and blue for control). If you were the researcher would you be comfortable using linear regression in this setting?

```
# Call ggplot and color by name
ggplot(data2,aes(x=pretest,y=y,color=treatment)) +
  geom_point()
```



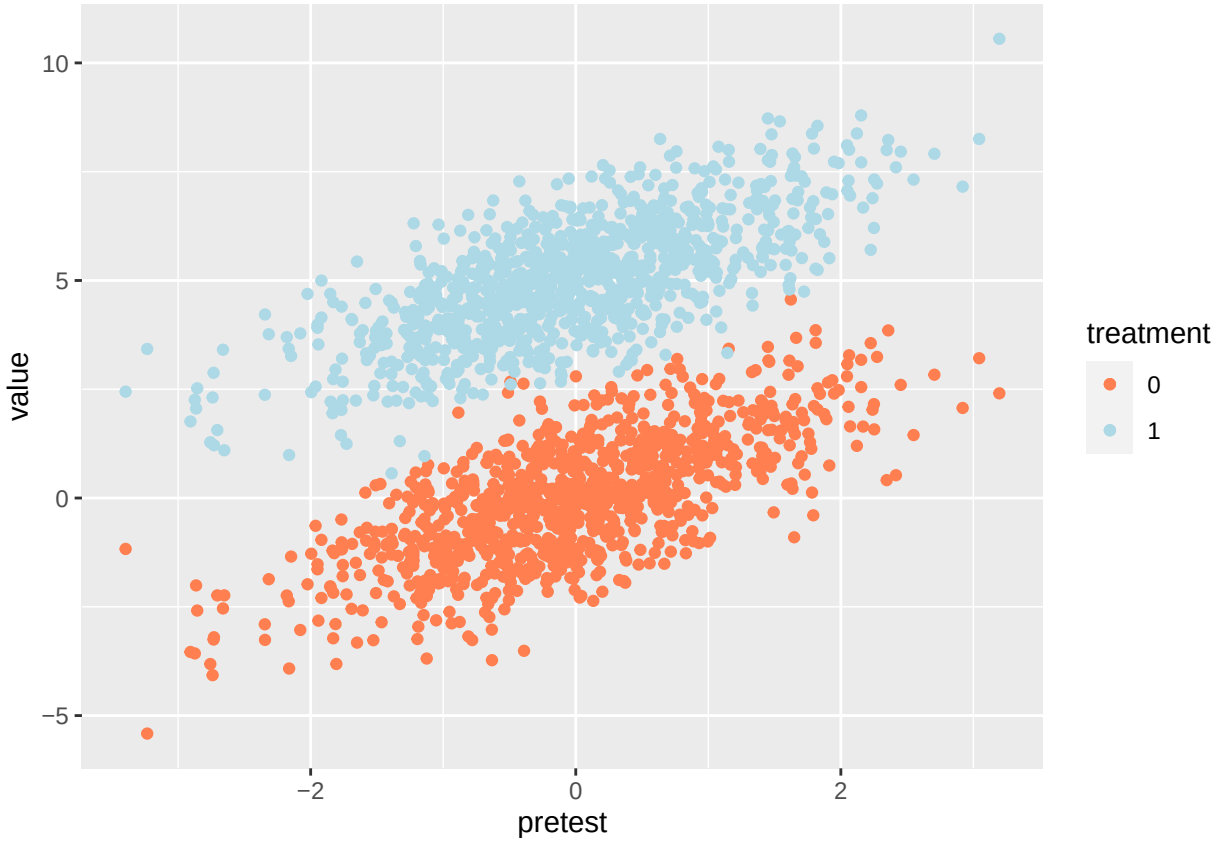
Linear regression here is not advised. Regression would be appropriate if all covariates are measured, SUTVA is satisfied, and linearity is satisfied, but in this case, and in most real world cases, this would be a difficult argument to make.

Question 3: Exploring the properties of estimators Now we're back to the role of god of Statistics.

(a) Create a scatter plot of X versus each potential outcome with different colors for treatment and control observations (suggested: red for Y(1) and blue for Y(0)). Is linear regression a reasonable model to estimate causal effects for the observed data set? Why or why not?

```
# Renaming dataset to keep track better
data3 = data2
# Call ggplot and color by name
ggplot(data3, aes(x=pretest, y = value, color = treatment)) +
    geom_point(aes(y = y0, col = "coral")) +
    geom_point(aes(y = y1, col = "light blue")) +
  scale_color_identity(name = "treatment",
                       breaks = c("coral", "light blue"),
```

```
                    labels = c("0", "1"),
                    guide = "legend")
```



Linear regression, in God mode, is appropriate for estimating causal effects. Because we can see all counterfactuals, we have all confounders related to the regression and the data overlaps more than on the observed, non-God mode outcomes.

(b) Calculate the difference between SATE and each of the estimates calculated by the researcher in Question 2.

```
# Subtract God-mode SATE from difference in means and linear regression estimates
difmeans2-sate1
```

```
## [1] 0.8712121
```

```
linreg2-sate1
```

```
## [1] 0.02340298
```

(c) Think harder about the practical significance of the bias by dividing this estimate by the standard deviation of the observed outcome Y.

Dividing by the standard deviation gives the observed outcome a standard unit by which to compare with. This removes the effect of variance between the two estimators to return greater bias.

(d) Create a randomization distribution for each estimator. [Hint: The randomization distribution here needs to respect the original DGP. So make sure to generate the random treatment assignment based on the p you created in 1c (that is, repeat what you did in 1c each time you need a draw of a vector of treatment assignments).] Use these to calculate the standardized bias for each of the **estimators**. That is, for each estimator, calculate its bias relative to SATE and divide by the sd of the outcome variable.

```r
# Randomly drawing 1000 datasets with DGP function.
# Each dataset has 1000 observations.
ITER = 1000
# Initialize vectors for estimators
rdmeandif = rep(NA,ITER)
rdlinreg = rep(NA,ITER)

# Begin loop for difference in means
for (i in 1:ITER) {
  # Draw new dataset, seed comes from loop iteration
  dat = dgp1(1000,0,0,1,i)
  # Calculate difference in means, subtract SATE to get bias, divide by SD
  rdmeandif[i] = ((mean(dat$y[dat$treatment==1]) - mean(dat$y[dat$treatment==0]))-mean(dat$y1-dat$y0))/s
  # Calculate difference in means, subtract SATE to get bias, divide by SD
  rdlinreg[i] = (summary(lm(y ~ pretest + treatment, dat))$coefficient[3]-mean(dat$y1-dat$y0))/sd(dat$y)
}

# Find average value for standardized bias of each estimator
mean(rdmeandif)
```

```
## [1] 0.2570123
```

```r
mean(rdlinreg)
```

```
## [1] 0.0003452189
```

(e) What assumption is violated by the difference in means estimator?

Difference in means here violates the assumption of overlap.

Part B: Non-Linear Parametric form

Now we'll explore what happens if we fit the wrong model in an observational study.

Question 1: Simulate the data

(a) Create function sim.nlin with the following DGP.

- (i) X should be drawn from a uniform distribution between 0 and 2.
- (ii) Treatment assignment should be drawn from a Binomial distribution with the following properities (make sure you save the p vector for use later).

$$\mathrm{E}[Z \mid X] = p = \text{logit}^{-1}(-2 + X^2)Z \sim \text{Binom}(N, p)$$

- (iii) The response surface (model for Y(0) and Y(1)) should be drawn from the following distributions:

$$Y(0) = \mathbf{2}X + \epsilon_{\mathbf{0}}$$
$$Y(1) = \mathbf{2}X + \mathbf{3}X^2 + \epsilon_{\mathbf{1}}$$

where both error terms are normally distributed with mean 0 and standard deviation of 1.

- (iv) Make sure the returned dataset has a column for the probability of treatment assignment as well.

```r
sim.nlin = function(N,seed){
  # Set seed
  set.seed(seed)
  # Generate 1000 simulated data points with uniform distribution
```

```
  X = runif(N,0,2)
  # Use inverse logit to turn values of X into values of p between 0 and 1
  p = 1/(1+exp(-(-2+X^2)))
  # Use these probability values to generate binary Z
  Z = as.factor(rbinom(N,1,p))
  # Create Y0's
  Y0 = 2*X + rnorm(N,0,1)
  # Create Y1's
  Y1 = 2*X + 3*(X^2) + rnorm(N,0,1)
  # Logic to determine outcome Y based on treatment assignment
  Y = ifelse(Z==1,Y1,Y0)
  # Combine everything into a data frame
  dat = data.frame(pretest=X,probability=p,treatment=Z,y=Y,y0=Y0,y1=Y1)
return(dat)
}
```

(b) Simulate a data set called data.nlin with sample size 1000.

```
data.nlin = sim.nlin(1000,1234)
head(data.nlin)
```

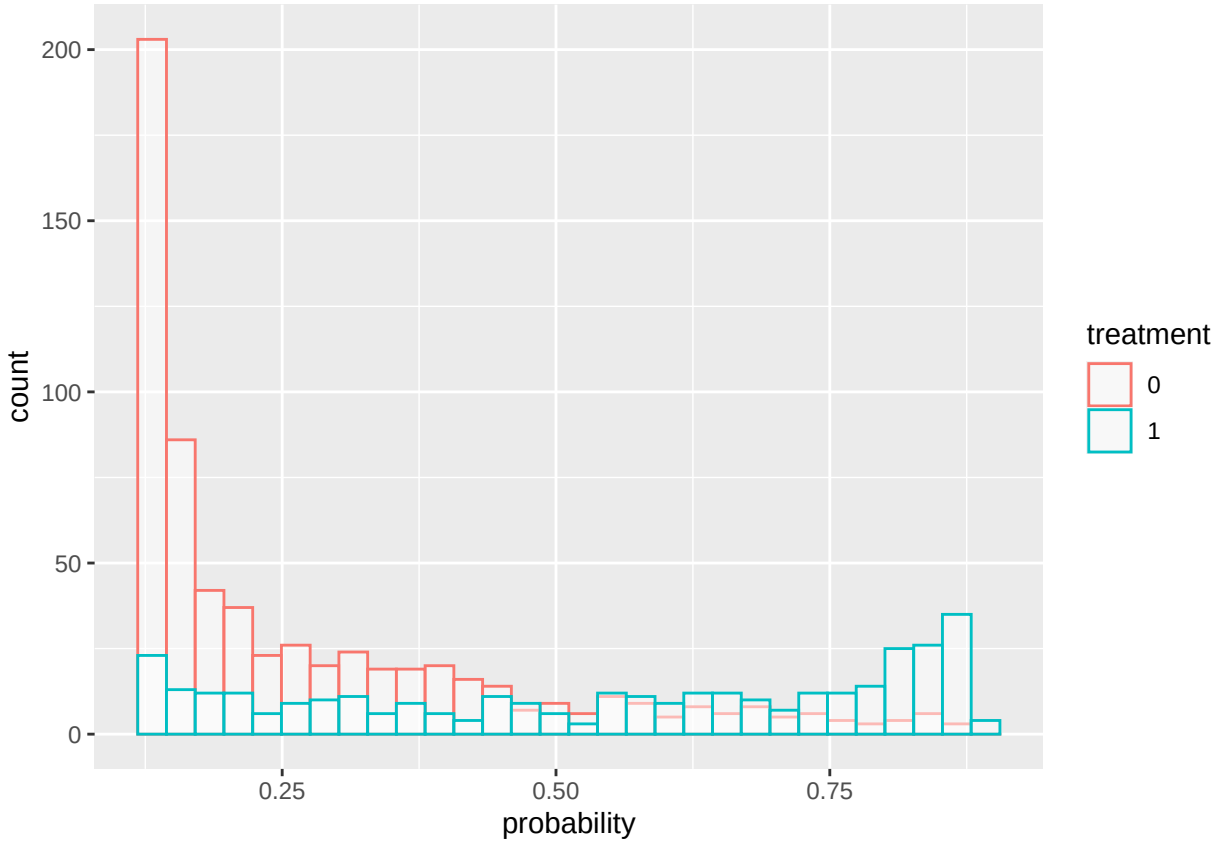
```
##     pretest probability treatment          y         y0         y1
## 1 0.2274068   0.1247404         0 -0.7505198 -0.7505198 -0.3638634
## 2 1.2445988   0.3891293         0  2.7906644  2.7906644  7.0366450
## 3 1.2185495   0.3739900         0  0.8979537  0.8979537  6.7809523
## 4 1.2467589   0.3904092         0  3.1288885  3.1288885  8.3489355
## 5 1.7218308   0.7240621         0  4.1466133  4.1466133 10.6818792
## 6 1.2806212   0.4109573         0  0.6553596  0.6553596  6.4355712
```

(c) Make the following plots.

- (i) Create overlaid histograms of the probability of assignment.

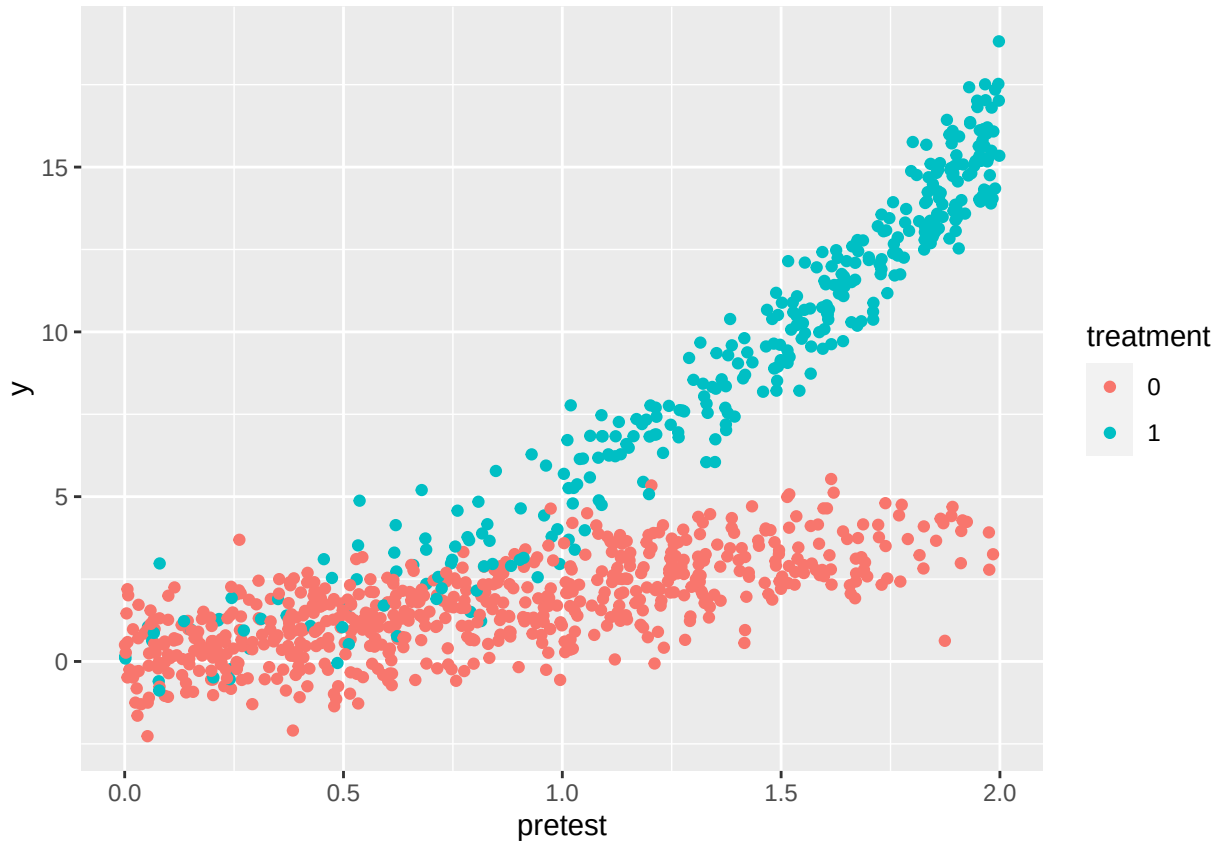
```
ggplot(data.nlin, aes(x=probability, color=treatment)) +
    geom_histogram(fill="white",alpha=0.5, position="identity")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



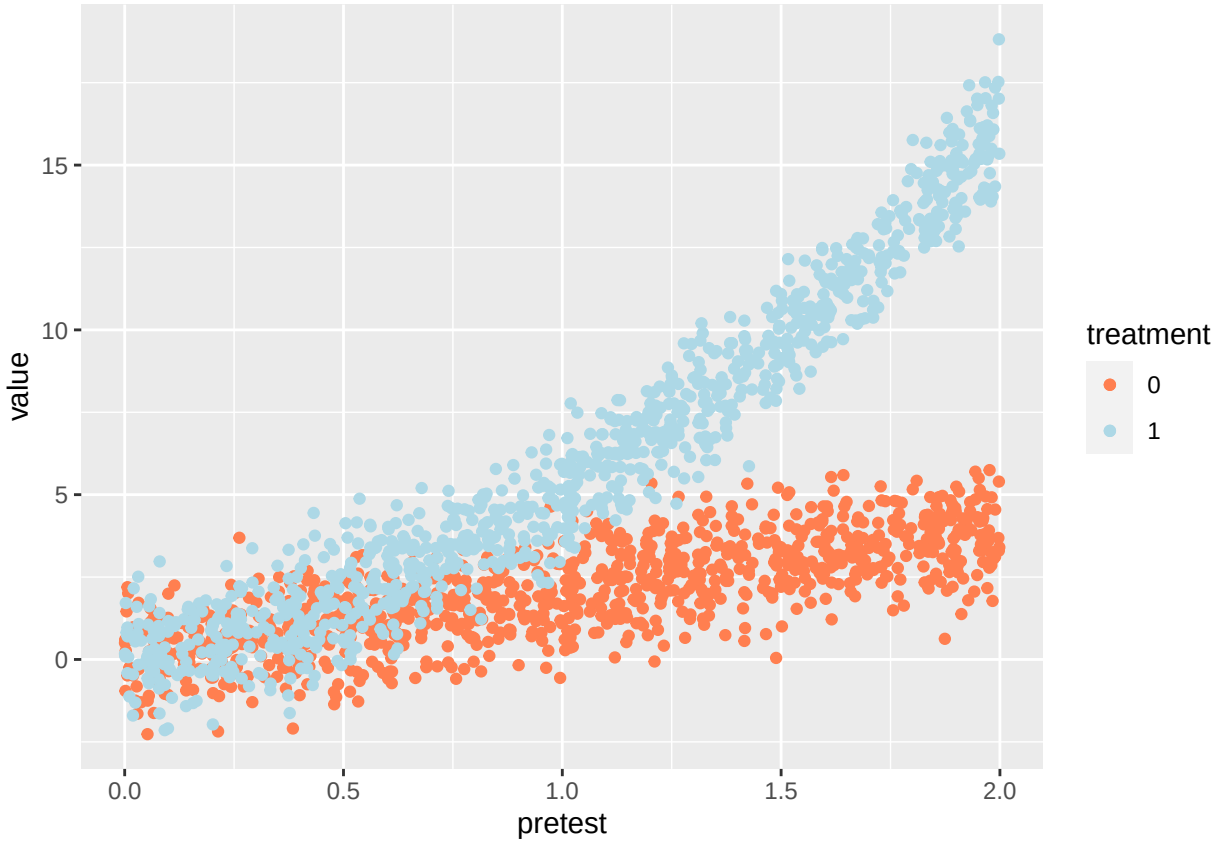
- (ii) Make a scatter plot of X versus the observed outcomes versus X with different colors for each treatment group.

```
ggplot(data.nlin,aes(x=pretest,y=y,color=treatment)) +
  geom_point()
```



- (iii) Create a scatter plot of X versus each potential outcome with different colors for treatment and control observations (suggested: red for Y(1) and blue for Y(0)). Does linear regression of Y ond X seem like a good model for this response surface?

```
# Call ggplot and color by name
ggplot(data.nlin, aes(x=pretest, y = value, color = treatment)) +
    geom_point(aes(y = y0, col = "coral")) +
    geom_point(aes(y = y1, col = "light blue")) +
  scale_color_identity(name = "treatment",
                       breaks = c("coral", "light blue"),
                       labels = c("0", "1"),
                       guide = "legend")
```



Linear regression is not advisable here. Regression trees may be a better option to estimate causal effects.

(d) Create randomization distributions to investigate the properties of each of 3 estimators with respect to SATE: (1) difference in means, (2) linear regression of the outcome on the treatment indicator and X, (3) linear regression of the outcome on the treatment indicator, X, and X^2.

```
# Randomly drawing 1000 datasets with DGP function.
# Each dataset has 1000 observations.
ITER = 1000
# Initialize vectors for estimators
rdmeandif = rep(NA,ITER)
rdlinreg = rep(NA,ITER)
rdlinreg2 = rep(NA,ITER)

# Begin loop for difference in means
for (i in 1:ITER) {
  # Draw new dataset, seed comes from loop iteration
  dat = sim.nlin(1000,i)
  # Create X squared
  dat$pretest2 = (dat$pretest)^2
  # Calculate difference in means, subtract SATE
  rdmeandif[i] = ((mean(dat$y[dat$treatment==1]) - mean(dat$y[dat$treatment==0]))-mean(dat$y1-dat$y0))
  # Calculate difference in means, subtract SATE
  rdlinreg[i] = (summary(lm(y ~ pretest + treatment, dat))$coefficient[3]-mean(dat$y1-dat$y0))
  # Calculate difference in means, subtract SATE
  rdlinreg2[i] = (summary(lm(y ~ pretest +  pretest2 + treatment, dat))$coefficient[4]-mean(dat$y1-dat$y
}
```

```r
# Find average value for NON-STANDARD bias of each estimator
mean(rdmeandif)
```

```
## [1] 3.554998
```

```r
mean(rdlinreg)
```

```
## [1] 0.9239397
```

```r
mean(rdlinreg2)
```

```
## [1] 0.4876484
```

(e) Calculate the standardized bias (bias divided by the standard deviation of Y) of these estimators relative to SATE. Which are biased?

```r
# Randomly drawing 1000 datasets with DGP function.
# Each dataset has 1000 observations.
ITER = 1000
# Initialize vectors for estimators
rdmeandif = rep(NA,ITER)
rdlinreg = rep(NA,ITER)
rdlinreg2 = rep(NA,ITER)

# Begin loop for difference in means
for (i in 1:ITER) {
  # Draw new dataset, seed comes from loop iteration
  dat = sim.nlin(1000,i)
  # Create X squared
  dat$pretest2 = (dat$pretest)^2
  # Calculate difference in means, subtract SATE
  rdmeandif[i] = ((mean(dat$y[dat$treatment==1]) - mean(dat$y[dat$treatment==0]))-mean(dat$y1-dat$y0))/
  # Calculate difference in means, subtract SATE
  rdlinreg[i] = (summary(lm(y ~ pretest + treatment, dat))$coefficient[3]-mean(dat$y1-dat$y0))/sd(dat$y
  # Calculate difference in means, subtract SATE
  rdlinreg2[i] = (summary(lm(y ~ pretest +  pretest2 + treatment, dat))$coefficient[4]-mean(dat$y1-dat$y
}

# Find average value for STANDARD bias of each estimator
mean(rdmeandif)
```

```
## [1] 0.7491168
```

```r
mean(rdlinreg)
```

```
## [1] 0.1949241
```

```r
mean(rdlinreg2)
```

```
## [1] 0.1030854
```

(f) What assumption is violated by the difference in means estimator? What assumption is violated by the linear regression estimator?

Difference means estimator violates ignorability assumption. In the linear regression estimator, the residuals indicate a non-linear relationship, which violates the parametric assumption.

Part C: Optional Challenge Question

Simulate Linear Causal Structure With Mutiple Covariates

(a). Simulate observational data set from following distribution

P(X1,X2,X3,Y1,Y0,Z)=P(X1)P(X2)P(X3)P(Z|X1,X2,X3)P(Y1,Y0|Z,X1,X2,X3).

Once again make sure that the probability of being treated for each person falls between .05 and .95 and there is a reasonable amount of overlap across the treatment and control groups. Generate the response surface as in the following:

$$Y(0) = X1 + X2 + X3 + \epsilon$$

$$Y(1) = X1 + X2 + X3 + 5 + \epsilon$$

(b) Create randomization distributions for (1) a regression estimator that controls for only one of the 3 covariates and (2) a regression estimator that controls for all 3 covariates. Evaluate the standardized bias of these estimators relative to SATE.

(c) Suppose you instead want to generate from the more general representation of this DGP: P(X1,X2,X3,Y1,Y0,Z)=P(X1,X2,X3)P(Z|X1,X2,X3) P(Y1,Y0|Z,X1,X2,X3).

- (i) What is the key difference between the assumptions in this DGP and the previous one?
- (ii) Provide code to simulate X1,X2 and X3 for this DGP?