Regression Inferences

Philippe Rast

PSC 204B
UC Davis, Winter 2018

Topics

Single-level Regression:

Week 1 Linear Regression (G&H: 3,4)
Week 2 Multiple Regression
Week 3 Violation of Assumptions
Week 4 Logistic Regression and GLM (G&H: 5, 6)
Week 5 Over-fitting, Information Criteria and Model comparison (McE: 6)
Week 6 Regression inference via simulations (G&H: 7–10)

Multilevel Regression:

Week 7 Multilevel Linear Models (G&H: 11–13)
Week 8 Multilevel Generalized Models (G&H: 14, 15)
Week 9 Bayesian Inference (G&H: 18 / McE: 1, 2, 3)
Week 10 Fitting Models in Stan and brms (G&H: 16, 17 / McE: 11)

Overview

1. Simulation of probability models and statistical inferences
2. Summarizing Linear Regressions using Simulation
   - Predictive Uncertainty
3. Bootstrapping
4. Causal inference
5. Problem of Causal Inference
6. Observational studies
   - Propensity Score Matching

Simulation of Probability Models

- Data reduction:
  - Representing inferences for a parameter using a point estimate and standard error
- Works if
  - Estimate is normally distributed
  - Summary discards no information because the normal distribution is completely defined by its mean and variance.

But: It can be useful to *represent the uncertainty in the parameter estimation* by a set of random simulations

- ▷ Represent possible values of the parameter vector
- Nature of probability works for us: More likely values are more likely to appear in the simulation
- ▷ Summarizing inferences by random numbers rather than by point estimates and standard errors.

Simulation of Probability Models

Example

- Example: Discrete predictive simulation
- The probability that a baby is a girl or boy is 48.8% or 51.2%, respectively.
- Suppose that 400 babies are born in a hospital in a given year. How many will be girls?
- Simulate the 400 births using the binomial distribution:

```
n.girls <- rbinom (1, 400, .488)
print (n.girls)
```

- Shows what could happen in 400 births

Simulation of Probability Models

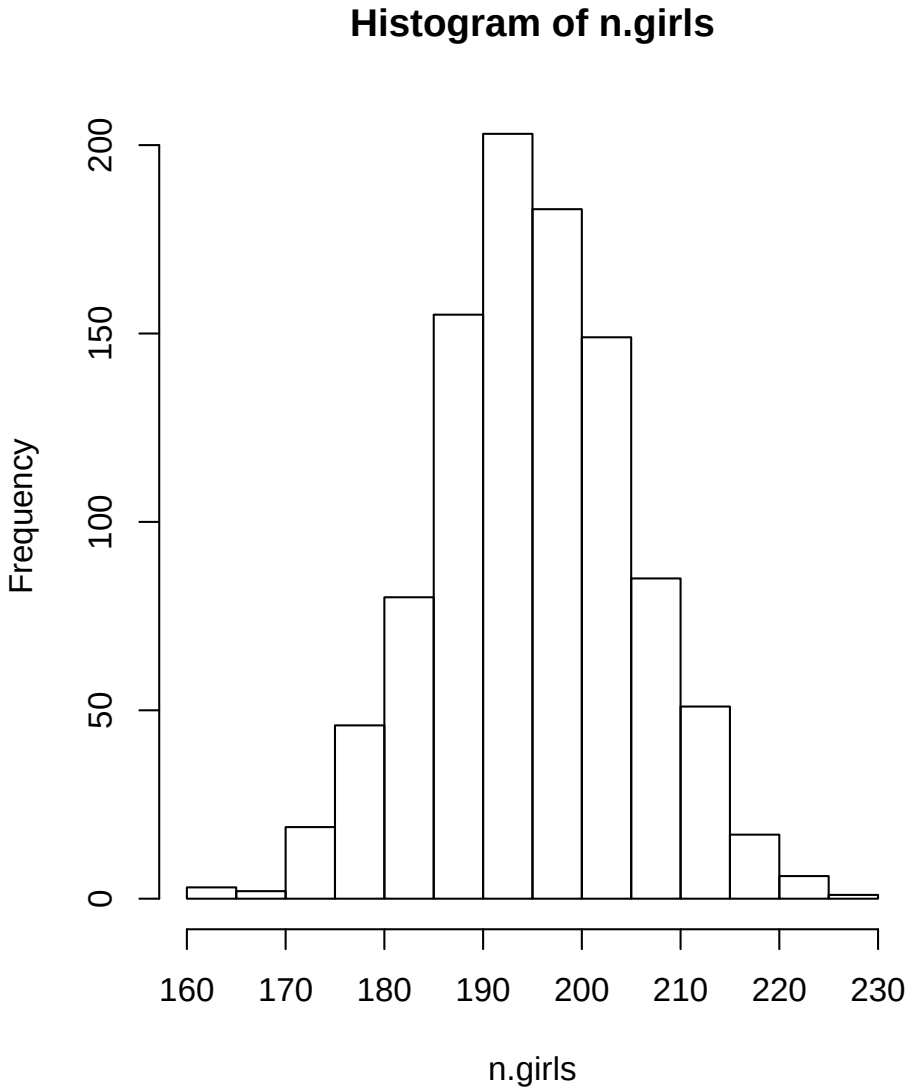
Example

- To get a sense of the *distribution*
- Simulate the process 1000 times

```
n.sims <- 1000
n.girls <- rep (NA, n.sims)
for (s in 1:n.sims){
  n.girls[s] <-
       rbinom (1, 400, .488)}
hist(n.girls)
```

- Or equivalently:

```
hist(rbinom(1000, 400, .488))
```



Simulation of Probability Models

Example: Accounting for Twins

- We can complicate the model in various ways
- There is a $1/125$ chance that a birth event results in fraternal twins
  - of which each has an approximate 49.5% chance of being a girl,
  - and a $1/300$ chance of identical twins
  - which have an approximate 49.5% chance of being girls.
- We can simulate 400 birth events

```
birth.type <- sample (c("fraternal twin","identical twin","single birth"),
  size=400, replace=TRUE, prob=c(1/125, 1/300, 1 - 1/125 - 1/300))
girls <- rep (NA, 400)
for (i in 1:400){
  if (birth.type[i]=="single birth"){
    girls[i] <- rbinom (1, 1, .488)}
  else if (birth.type[i]=="identical twin"){
    girls[i] <- 2*rbinom (1, 1, .495)}
  else if (birth.type[i]=="fraternal twin"){
    girls[i] <- rbinom (1, 2, .495)}
}
n.girls <- sum (girls)
```

- Girls is a vector of length 400, of 0's, 1's, and 2's (mostly 0's and 1's) representing the number of girls in each birth event
- To approximate the *distribution* of the number of girls in 400 births, we put the simulation in a loop and repeat it 1000 times:

```
n.girls <- rep (NA, n.sims)
for (s in 1:n.sims){
  birth.type <-sample(c("fraternal twin","identical twin","single birth"),
    size=400, replace=TRUE, prob=c(1/125, 1/300, 1 - 1/125 - 1/300))
    girls <- rep (NA, 400)
    for (i in 1:400){
      if (birth.type[i]=="single birth"){
        girls[i] <- rbinom (1, 1, .488)}
      else if (birth.type[i]=="identical twin"){
        girls[i] <- 2*rbinom (1, 1, .495)}
      else if (birth.type[i]=="fraternal twin"){
        girls[i] <- rbinom (1, 2, .495)}
      }
  n.girls[s] <- sum (girls)
}
```

Continuous Predictive Simulation

Example: Height of US Adults

- So far: Discrete simulation
- Simulate continuous random variables
- ▷ Example: 52% of adults in the US are women, 48% are men.
- The heights of the men are approximately normally distributed with
  - $M = 69.1$ inches and $SD = 2.9$ inches for men
  - $M = 63.7$ and $SD = 2.7$ for women.
- Suppose we select 10 adults at random.

```
sex <- rbinom (10, 1, .52)
height <- ifelse (sex==0, rnorm (10, 69.1, 2.9),
                          rnorm (10, 64.5, 2.7))
avg.height <- mean (height)
> print (avg.height)
[1] 67.88213
```

Continuous Predictive Simulation

Example: Height of US Adults

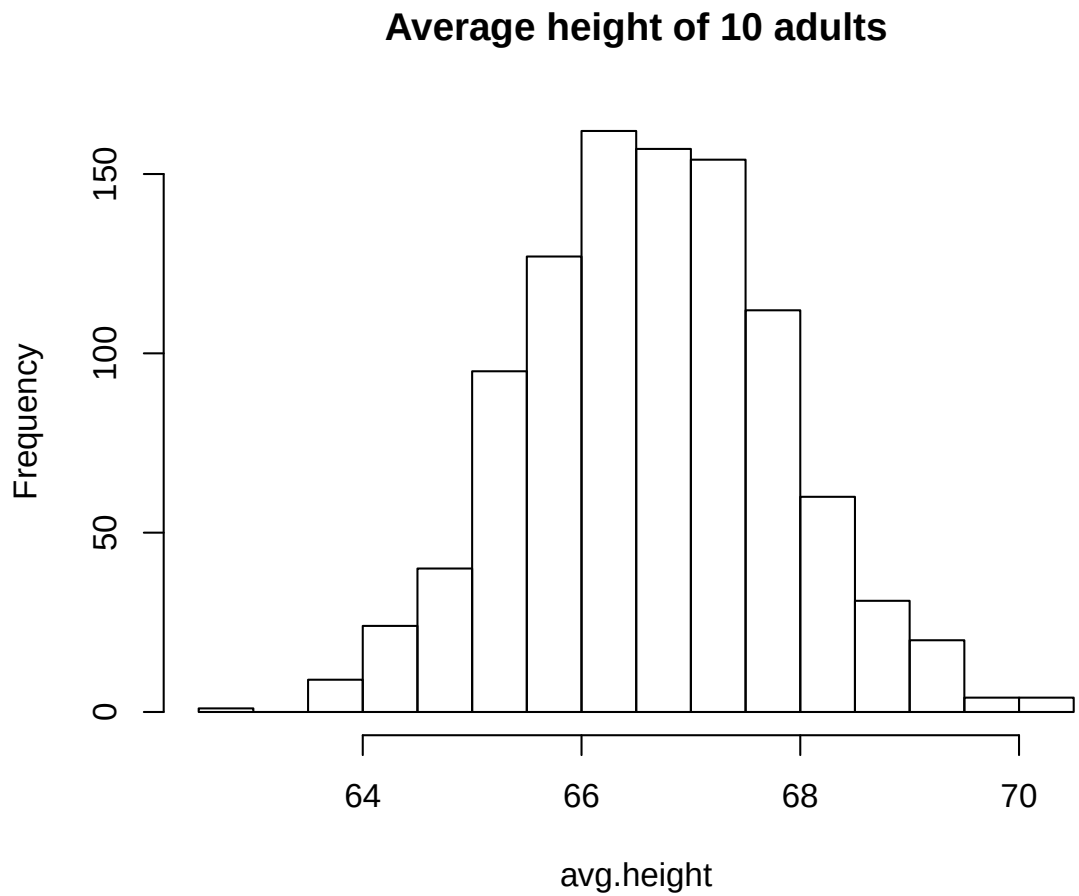
- Simulate the distribution of `avg.height`
  - Loop the simulation 1000 times:

```
n.sims <- 1000
avg.height <- rep (NA, n.sims)
for (s in 1:n.sims){
  sex <- rbinom (10, 1, .52)
  height <- ifelse (sex==0, rnorm (10, 69.1, 2.9),
                            rnorm (10, 64.5, 2.7))
avg.height[s] <- mean (height)
}
```

Continuous Predictive Simulation

Example: Height of US Adults

```
hist (avg.height, main="Average height of 10 adults")
```



Continuous Predictive Simulation

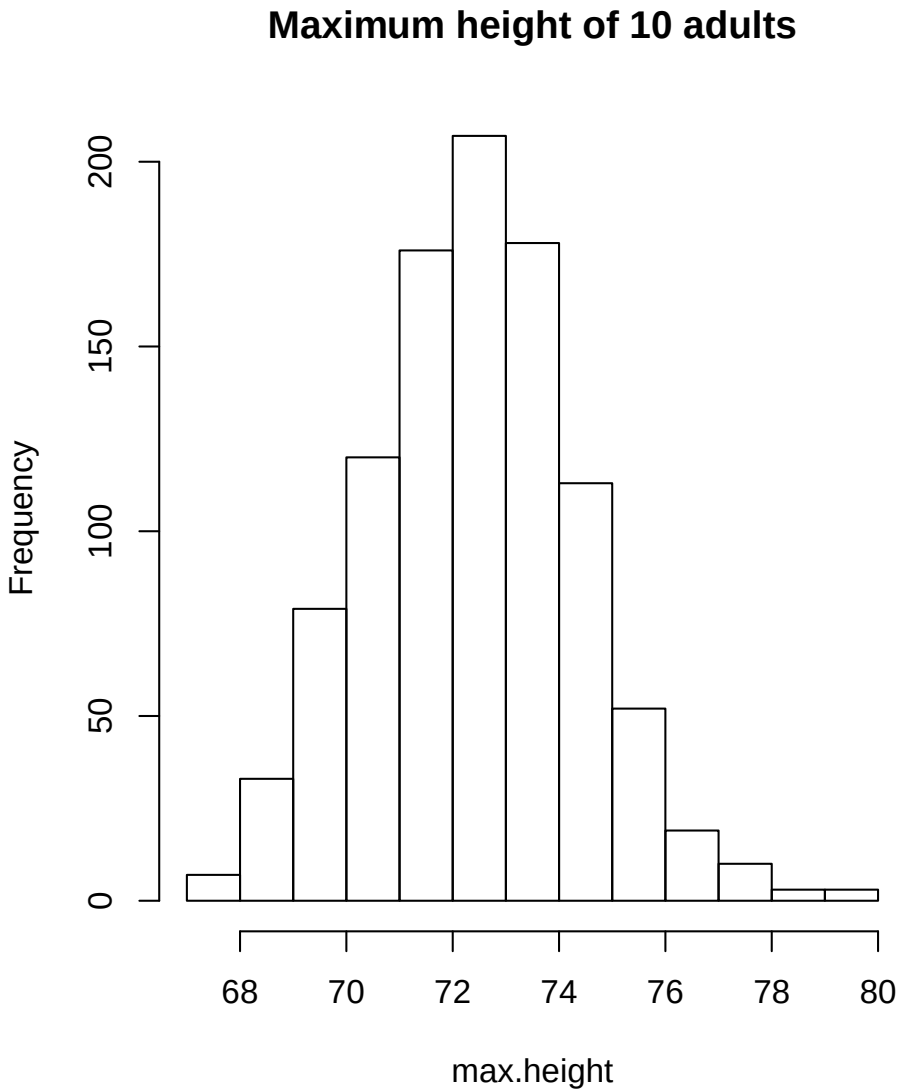
- Now the we have a simulated sample, we can investigate other properties

e.g. Maximum height

- We can add two lines to the code:

```
max.height <- rep (NA, n.sims)
max.height[s] <- max (height)
```

- And `hist(max.height)`



Simulation in R using custom-made functions

- Coding for simulations becomes cleaner if we express the steps for a single simulation as a function in R.

e.g. Simulation of average heights.

```
Height.sim <- function (n.adults, m.m, sd.m, m.f, sd.f){
  sex <- rbinom (n.adults, 1, .52)
  height <- ifelse (sex==0, rnorm (10, m.m, sd.m),
                            rnorm (10, m.f, sd.f))
  return (mean(height))
}
```

- Use the `replicate()` function to call `Height.sim()` 1000 times:

```
avg.height <- replicate(1000,
              Height.sim(10, 69.5, 2.9, 64.5, 2.7))
```

Summarizing Linear Regressions using Simulation

Informal Bayesian Approach

- In a regression setting we can use simulation to capture both
  - Predictive uncertainty: The error term in the regression model
  - Inferential uncertainty: Standard errors of the coefficients and uncertainty about the residual error
- First: Focus on simplest case of simulating prediction errors
- Later: Consider inferential uncertainty and the combination of both sources of variation.

Predictive Uncertainty

- Example: Problem of predicting the earnings of a 68-inch-tall man, using model
- `lm(log.earn ~ height + male + height:male)`

```
             coef.est coef.se
(Intercept)  8.388    0.844
height       0.017    0.013
male        -0.079    1.258
height:male  0.007    0.019
  n = 1192, k = 4
  residual sd = 0.88, R-Squared = 0.09
```

Predictive Uncertainty

Example: Predicting Earnings

- Obtaining the point and interval predictions automatically.
- Access predictive estimate and confidence interval

```
x.new <- data.frame (height=68, male=1)
pred.interval <- predict (earn.logmodel, x.new,
  interval="prediction", level=.95)
```

- Exponentiate to get the predictions on the original (unlogged) scale of earnings:
- `exp (pred.interval)`

Predictive Uncertainty

Example: Predicting Earnings

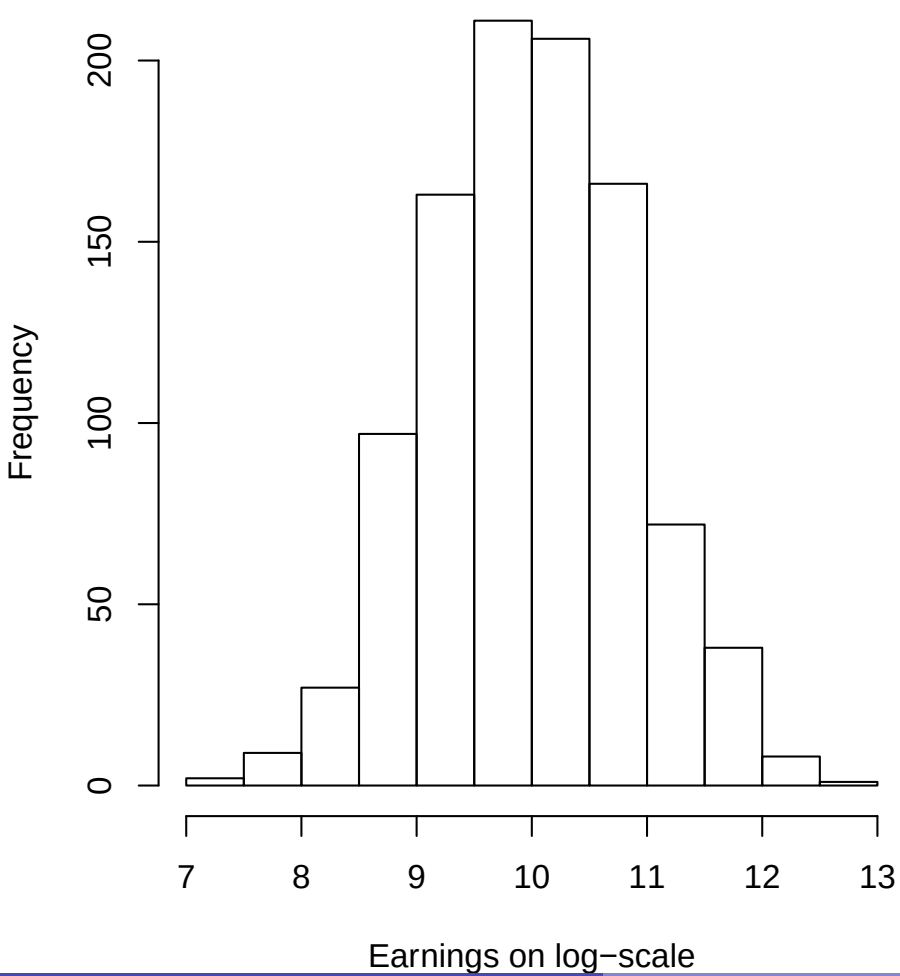
- Construct *predictive interval* using simulation
- ▷ Obtain predictive intervals "manually" via simulations derived from the fitted regression model
  - Point estimate for log earnings is $8.4 + 0.017 \times 68 - 0.079 \times 1 + 0.007 \times 68 \times 1 = 9.95$
  - SD of 0.88
  - Original scale (unlogged): $\exp(9.95) \approx 21,000$ with SD of $\exp(0.88) \approx 2.4$
  - The simulation prediction is a set of random numbers whose logarithms have mean 9.95 and standard deviation 0.88.
- We can summarize the predictive distribution using the following command:

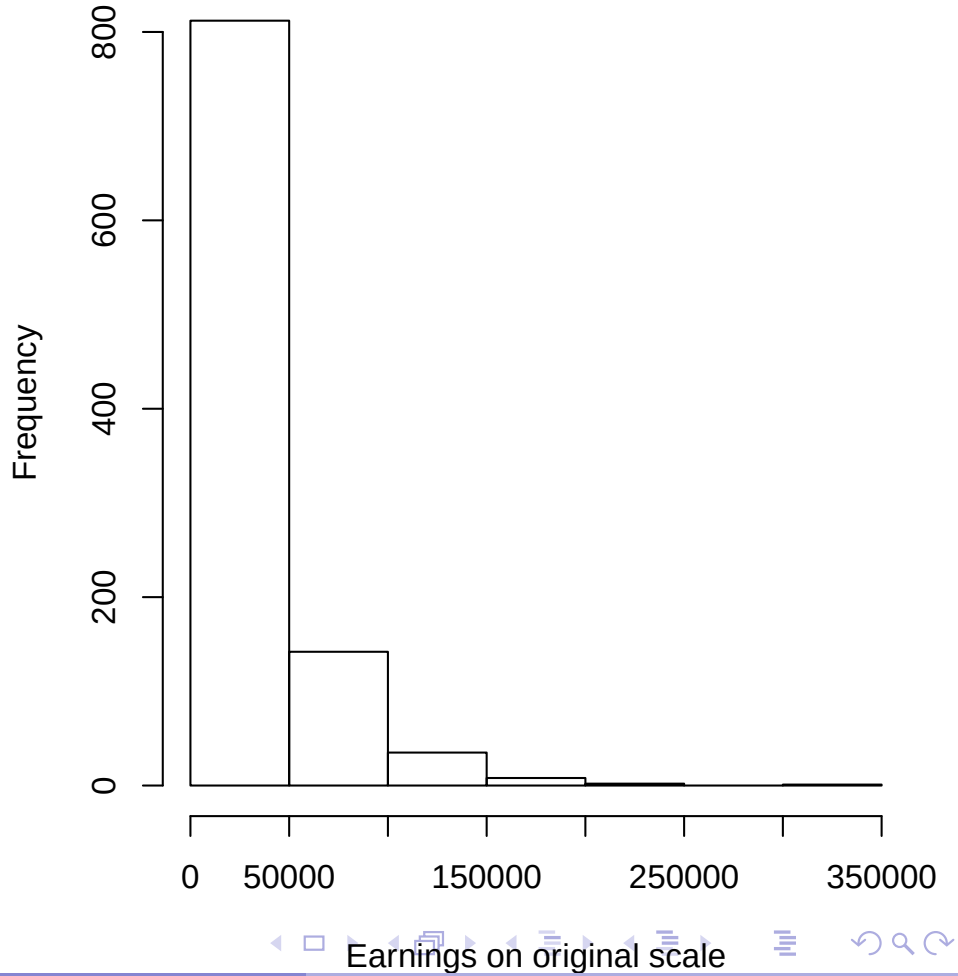
R:
```
pred <- exp(rnorm(1000, 9.95, .88))
```

Predictive Uncertainty

Example: Predicting Earnings

- Draw 1000 random numbers from the predictive distribution with mean 9.95 and SD 0.88, and then exponentiate these values.





Predictive Uncertainty

- We can also compute various numerical summaries, for example
- Mean:

```
> mean(pred)
[1] 31518.17
```

- Median:

```
> median(pred)
[1] 21072.4
```

- 50% interval:

```
> quantile(pred,c(.25,.75))
25%      75%
11954.28 40447.70
```

- 95% interval:

```
> quantile(pred,c(.05,.95))
5%       95%
5439.761 92185.550
```

Why do we need simulation for predictive inferences?

- For many purposes, point estimates, standard errors, and the intervals obtained are sufficient because the Central Limit Theorem ensures that for all but the smallest sample sizes and for reasonably well-behaved error distributions, coefficient estimates are approximately normally distributed
- t-distribution with n-k degrees of freedom accounts for the uncertainty in the standard-error estimates

- Not always this easy
- For more general predictions, however, the easiest and most reliable way to compute uncertainties is by simulation.

Why do we need simulation for predictive inferences?

- Example: We have a 68-inch-tall woman and a 68-inch-tall man, and we would like to use the previous model to predict the difference of their earnings.
- As a point estimate, we can use the difference of the point predictions: $\exp(8.4 + 0.017 \times 68 - 0.079 \times 1 + 0.007 \times 68 \times 1) - \exp(8.4 + 0.017 \times 68 - 0.079 \times 0 + 0.007 \times 68 \times 0) = 6900.$

```
pred.man <- exp(rnorm(1000,8.4+.017*68-.079*1+.007*68*1, .88))
pred.woman<-exp(rnorm(1000,8.4+.017*68-.079*0+.007*68*0, .88))
pred.diff <- pred.man - pred.woman
pred.ratio <- pred.man/pred.woman
```

Why do we need simulation for predictive inferences?

- We can summarize the distribution just like we did before
- `mean(pred.diff)` or `quantile(pred.ratio,c(.25,.75))`
- Simulation is valuable because it can be used to summarize *any* function of estimated and predicted values.
- Mean:

```
> mean(pred.diff)
[1] 10227.84
```

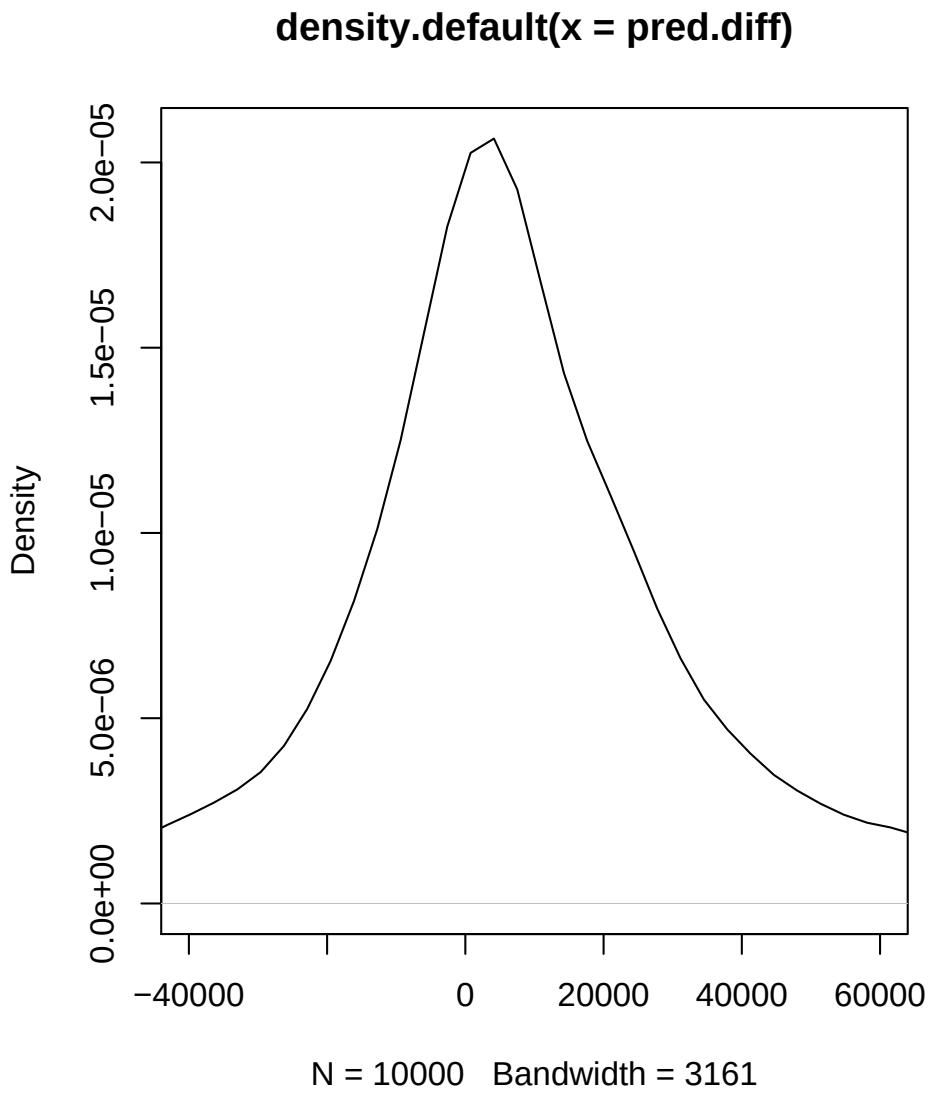
- Average Ratio:

```
> mean(pred.ratio)
[1] 3.247445
```

- Note that mean is quite far off of point estimate (6,900)

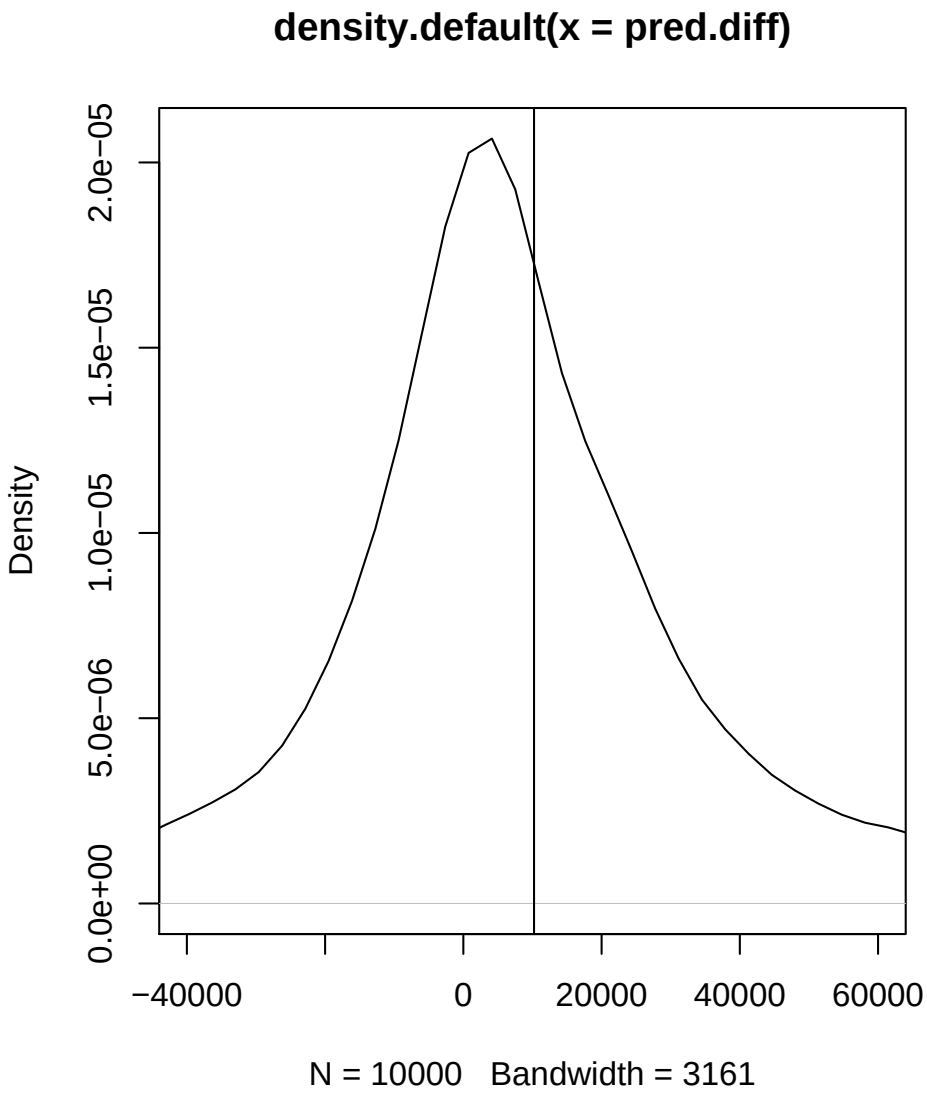
Why do we need simulation for predictive inferences?

- Investigate source of discrepancy
- Plot distribution



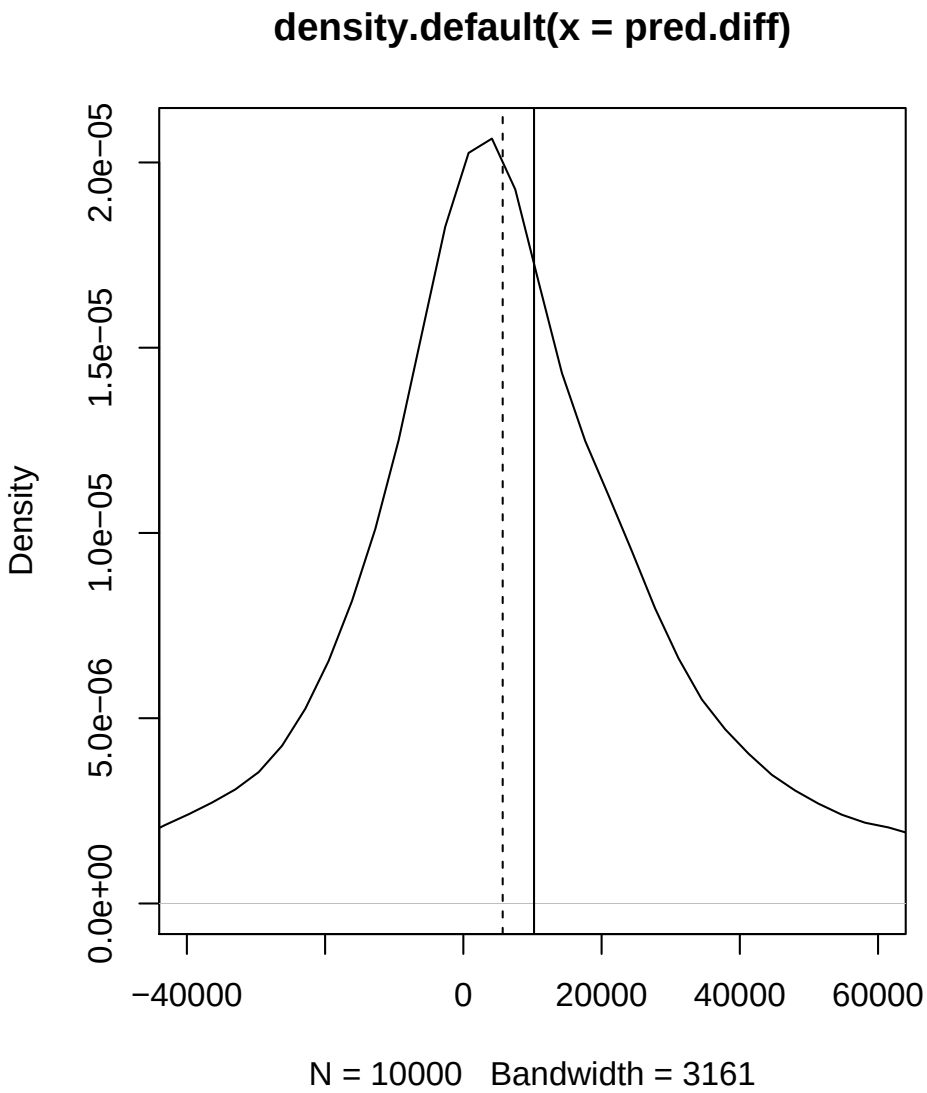
Why do we need simulation for predictive inferences?

- Investigate source of discrepancy
- Plot distribution
- Mean: 10,227



Why do we need simulation for predictive inferences?

- Investigate source of discrepancy
- Plot distribution
- Mean: 10,227
- Median: 5,693



Simulation to Represent Uncertainty in Regression Coefficients

- Typically: Standard errors represent uncertainty
- helpful to summarize inferences by simulation, which gives us complete flexibility in propagating uncertainty about combinations of parameters and predictions.

Using the sim() Function

- The `arm` library has a function sim() that will use simulation to take samples of probable model parameters, given the observed data.
- It takes uncertainty in error variance and parameter estimates into account.
- The result of `sim()` is a list with two components:
  - `$beta` is a matrix of regression coefficients corresponding to the model matrix.
  - `$sigma` is an estimate of the standard deviation of the normal error.
- For most purposes, a simulation of about 1000 realizations of the model parameters is sufficient

Informal Bayesian inference

- Bayesian inference refers to statistical procedures that model unknown parameters as random variables.
- Bayesian inference starts with a *prior distribution* on the unknown parameters and updates this with the *likelihood* of the data
  - ▷ yielding a *posterior distribution* which is used for inferences and predictions.
- The simulations presented here correspond to noninformative prior distributions

Simulation for checking statistical procedures and model fits

- Ways in which probabilistic simulation can be used to better understand statistical procedures
- fake-data simulation
- Controlled experiments in which the parameters of a statistical model are set to fixed "true" values
- Simulations are used to study the properties of statistical methods
- Estimated parameters are compared to true parameters, to check that a statistical method performs as advertised

Fake-data simulation

- validate statistical algorithms and to check the properties of estimation procedures
- Illustrated with a simple regression model
- simulate fake data from the model, $y = \alpha + \beta x + \epsilon$
- Refit the model to the simulated data
- Check coverage of 68% and 95% intervals for β

Fake-data simulation

- set up the true values of the parameters
  - $\alpha = 1.4$
  - $\beta = 2.3$
  - $\sigma = 0.9$
  - Predictors $x = [1, 2, 3, 4, 5]$

```
a <- 1.4
b <- 2.3
sigma <- 0.9
x <- 1:5
n <- length(x)
```

Fake-data simulation

Simulation

- Simulate a vector y of fake data
- Fit a regression model to these data
- We fit a normal model, as if we didn't know the population values

```
y <- a + b*x + rnorm (n, 0, sigma)
lm.1 <- lm (y ~ x)
display (lm.1)
```

with the Output

```
lm(formula = y ~ x)
            coef.est coef.se
(Intercept) 1.43     1.59
x           2.25     0.48
  n = 5, k = 2
  residual sd = 1.52, R-Squared = 0.88
```

Fake-data simulation

Simulation

- Fit seems reasonable enough
- ▷ Estimates are not exact but are within the margin of error
- "Automate" by extracting from the regression object the estimate and standard error of β

```
b.hat <- coef (lm.1)[2]   # "b" is 2nd coef in the model
b.se <- se.coef (lm.1)[2] # "b" is 2nd coef in the model
```

- Check whether true β falls within the estimated 68% and 95% CI

```
cover.68 <- abs (b - b.hat) < b.se   # this will be TRUE or FALSE
cover.95 <- abs (b - b.hat) < 2*b.se # this will be TRUE or FALSE
> cat (paste ("68% coverage: ",cover.68, "\n"))
68% coverage:  TRUE
> cat (paste ("95% coverage: ",cover.95, "\n"))
95% coverage:  TRUE
```

Fake-data simulation

Simulation

- This was for one sample
- What about correct coverage probabilities?

```
n.fake <- 1000
cover.68 <- rep (NA, n.fake)
cover.95 <- rep (NA, n.fake)
for (s in 1:n.fake){
  y <- a + b*x + rnorm (n, 0, sigma)
  lm.1 <- lm (y ~ x)
  b.hat <- coef (lm.1)[2]
  b.se <- se.coef (lm.1)[2]
  cover.68[s] <- abs (b - b.hat) < b.se
  cover.95[s] <- abs (b - b.hat) < 2*b.se
}
> cat (paste ("68% coverage: ", mean(cover.68), "\n"))
68% coverage:  0.584
> cat (paste ("95% coverage: ", mean(cover.95), "\n"))
95% coverage:  0.842
```

Fake-data simulation

Simulation

- 68% `coverage:  0.584`
- 95% `coverage:  0.842`
- Does not seem right:
  - Only 58% of the 68% intervals and
  - Only 84% of the 95% intervals covered the true parameter values
- What's going on?
- ± 1 and ± 2 standard-error intervals are appropriate for the normal distribution
- Our sample was just $n = 5$!
- Redo with t_3-distribution with $df = 3$

Fake-data simulation

Simulation

```
n.fake <- 1000
cover.68 <- rep (NA, n.fake)
cover.95 <- rep (NA, n.fake)
t.68 <- qt (.84, n-2)
t.95 <- qt (.975, n-2)
for (s in 1:n.fake){
  y <- a + b*x + rnorm (n, 0, sigma)
  lm.1 <- lm (y ~ x)
  b.hat <- coef (lm.1)[2]
  b.se <- se.coef (lm.1)[2]
  cover.68[s] <- abs (b - b.hat) < t.68*b.se
  cover.95[s] <- abs (b - b.hat) < t.95*b.se
}
> cat (paste ("68% coverage ", mean(cover.68), "\n"))
68% coverage  0.667
> cat (paste ("95% coverage: ", mean(cover.95), "\n"))
95% coverage:  0.956
```

Fake-data simulation

Simulation

- Now we obtain coverages of 67% and 96%, as predicted
  - 68% `coverage:  0.667`
  - 95% `coverage:  0.956`
- Almost perfect!

Fake-data simulation to understand residual plots

- Simulate from a regression model to get insight into residual plots
- Understand why we plot residuals versus *fitted* values rather than versus observed values
- Illustration: Predicting final exam scores from midterms in an introductory statistics class:

```
> lm.1 <- lm (final ~ midterm)
> display (lm.1)
lm(formula = final ~ midterm)
            coef.est  coef.se
(Intercept) 64.50     16.98
midterm      0.70      0.21
  n = 52, k = 2
  residual sd = 14.75, R-Squared = 0.18
```

Fake-data simulation to understand residual plots

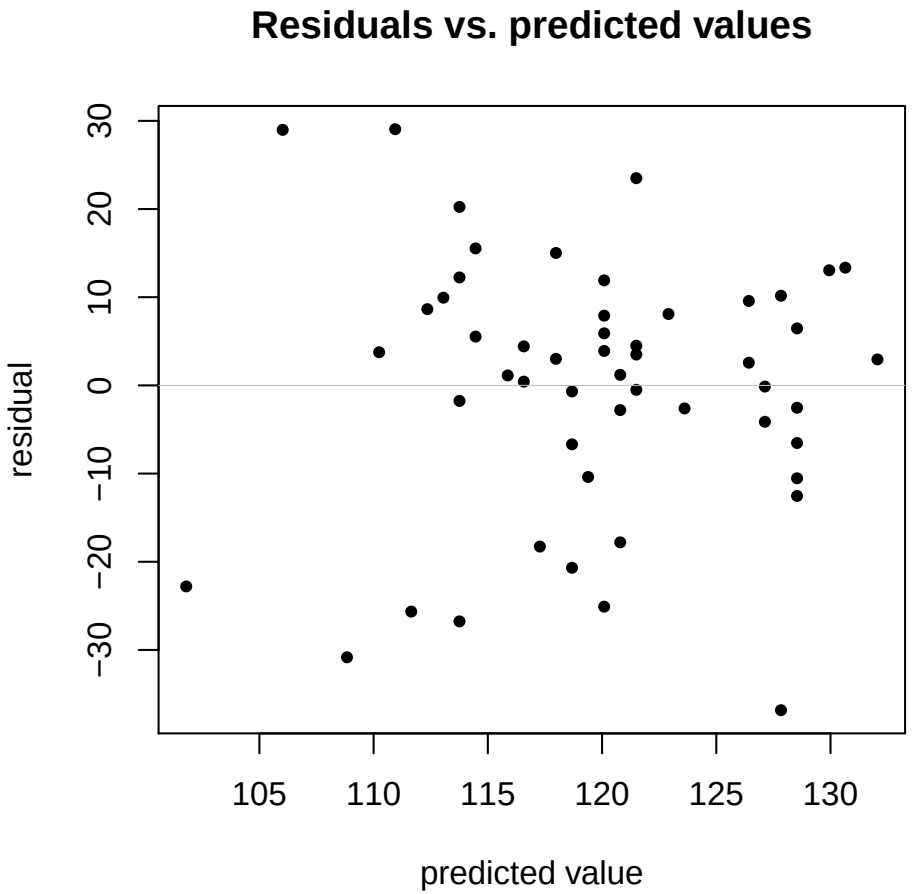
Midterm Example

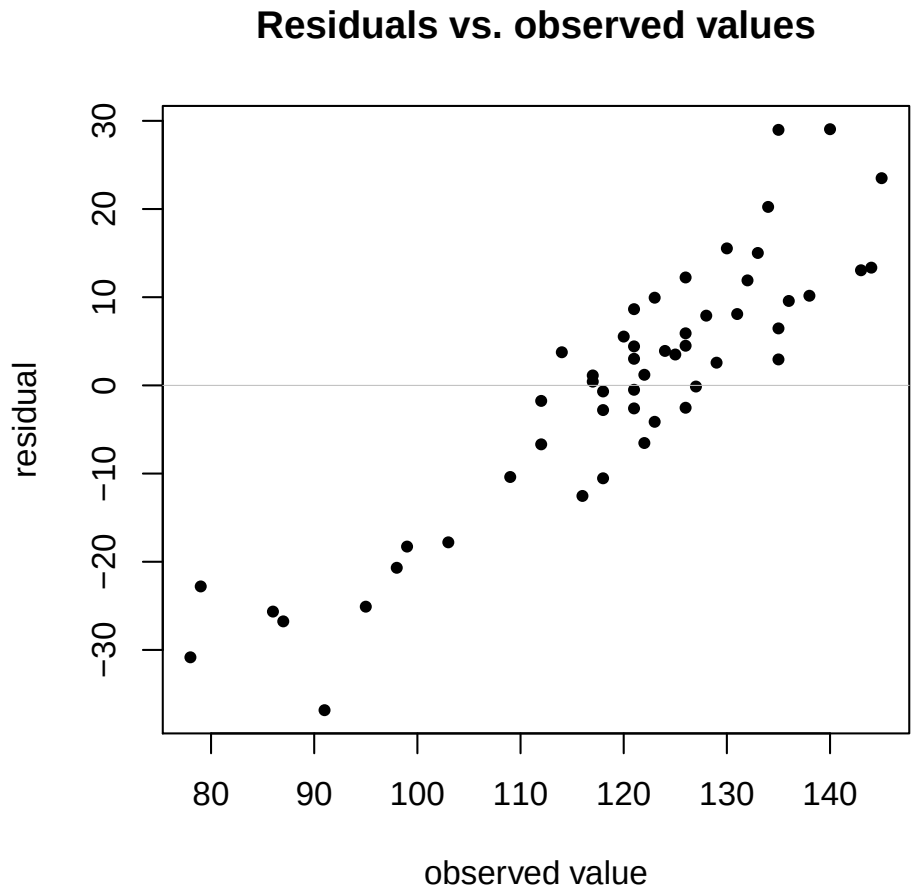
- Construct fitted values $\hat{y} = X\hat{\beta}$ and residuals $y - X\hat{\beta}$

```
n <- length(final)
X <- cbind (rep(1,n), midterm)
predicted <- X %*% coef (lm.1)
resid <- lm.1$residuals
```

- Generate plots

Fake-data simulation to understand residual plots





- Residuals versus observed does not necessarily inform us about issues
- Regression model requires that the errors ϵ are independent of the predictors x, not the data y

Fake-data simulation to understand residual plots

Generated Data

- While we don't know the actual relation in real data, we do in fake-data
- Demonstrate fitted vs. residual and observed vs. residual
- Use similar values as in the Midterm example

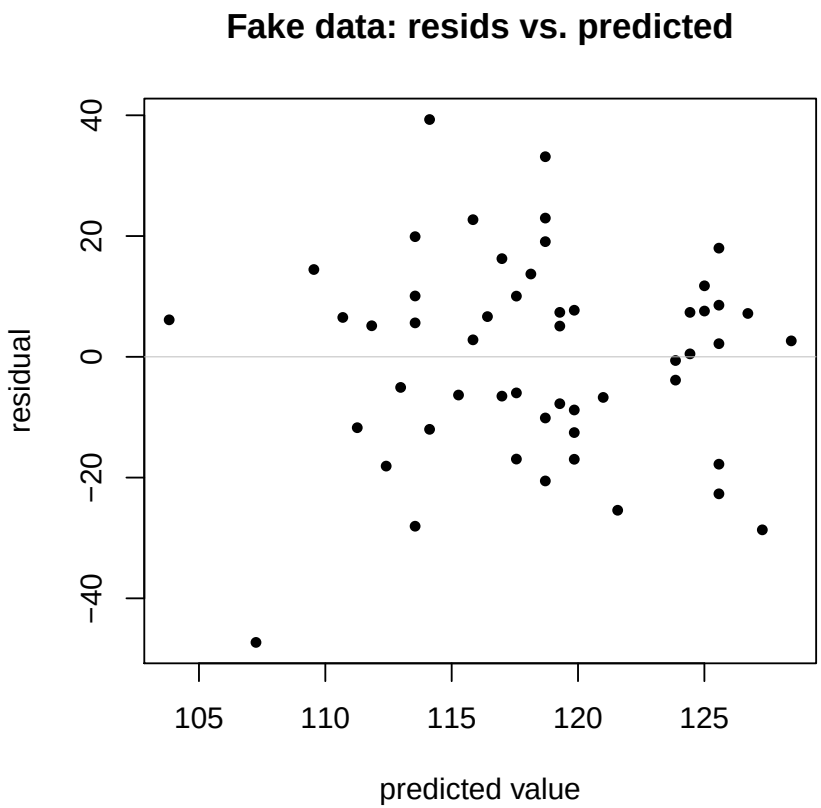
```
a <- 65
b <- 0.7
sigma <- 15
y.fake <- a + b*midterm + rnorm(n, 0, 15)
```

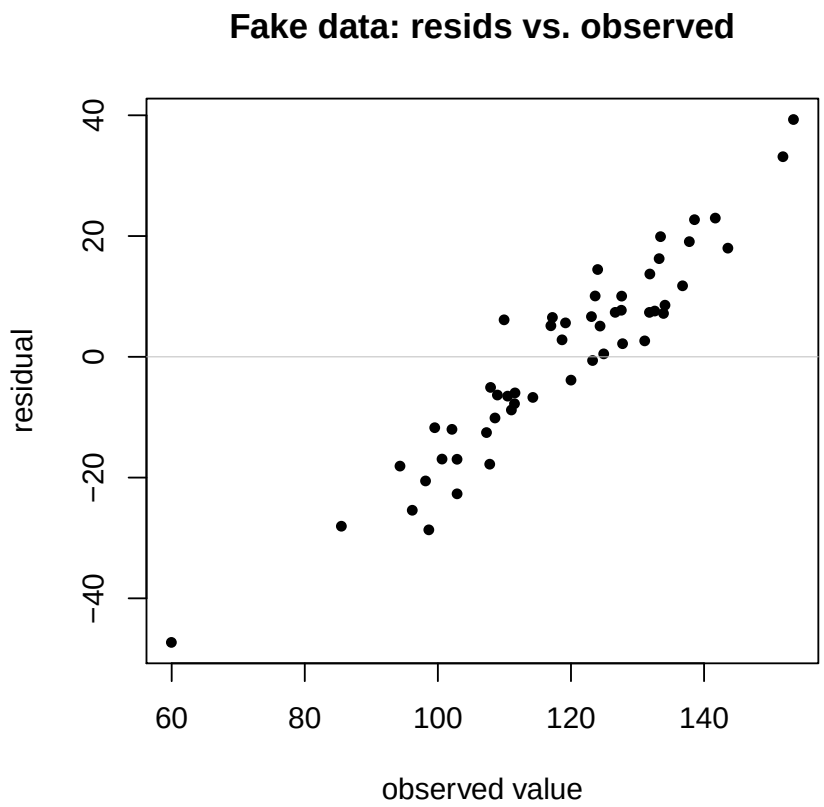
- Fit and obtain fitted values:

```
lm.fake <- lm(y.fake ~ midterm)
predicted.fake <- X %*% coef(lm.fake) # or fitted(lm.fake)
resid.fake <- y.fake - predicted.fake
```

Fake-data simulation to understand residual plots

Generated Data





- Again, pattern is very clear:
  - fitted vs. observed does not tell us about distribution of residuals

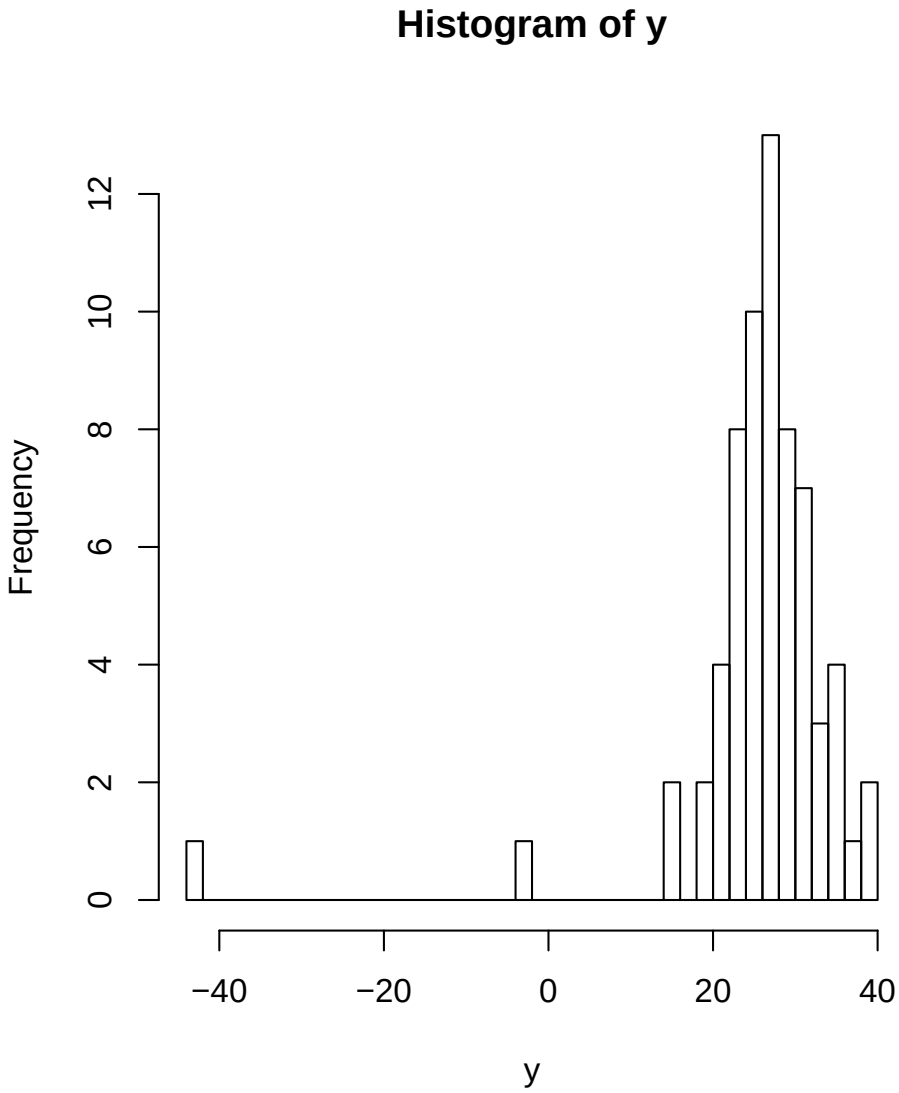
Simulating from the fitted model and comparing to data

- Simulate replicated data under the fitted model
- Compare to the observed data
- Fundamental way to check model fit:
  - Display replicated datasets
  - Compare them to the actual data
- Illustration with dataset that does not fit the normal distribution

Goal: Demonstrate how the lack of fit can be seen using predictive replications

Example: Simon Newcomb's Speed of Light

- Measurements taken by Simon Newcomb in 1882 as part of an experiment to estimate the speed of light.
- data represent the amount of time required for light to travel a distance of 7442 meters
- Recorded as deviations from 24,800 nanoseconds



Example: Simon Newcomb's Speed of Light

- Let's fit (inappropriately) a normal distribution to these data with no predictors
- `light <- lm (y ~ 1)`
- Simulate 1000 replications from the parameters in the fitted model

```
n.sims <- 1000
sim.light <- sim (light, n.sims)
```

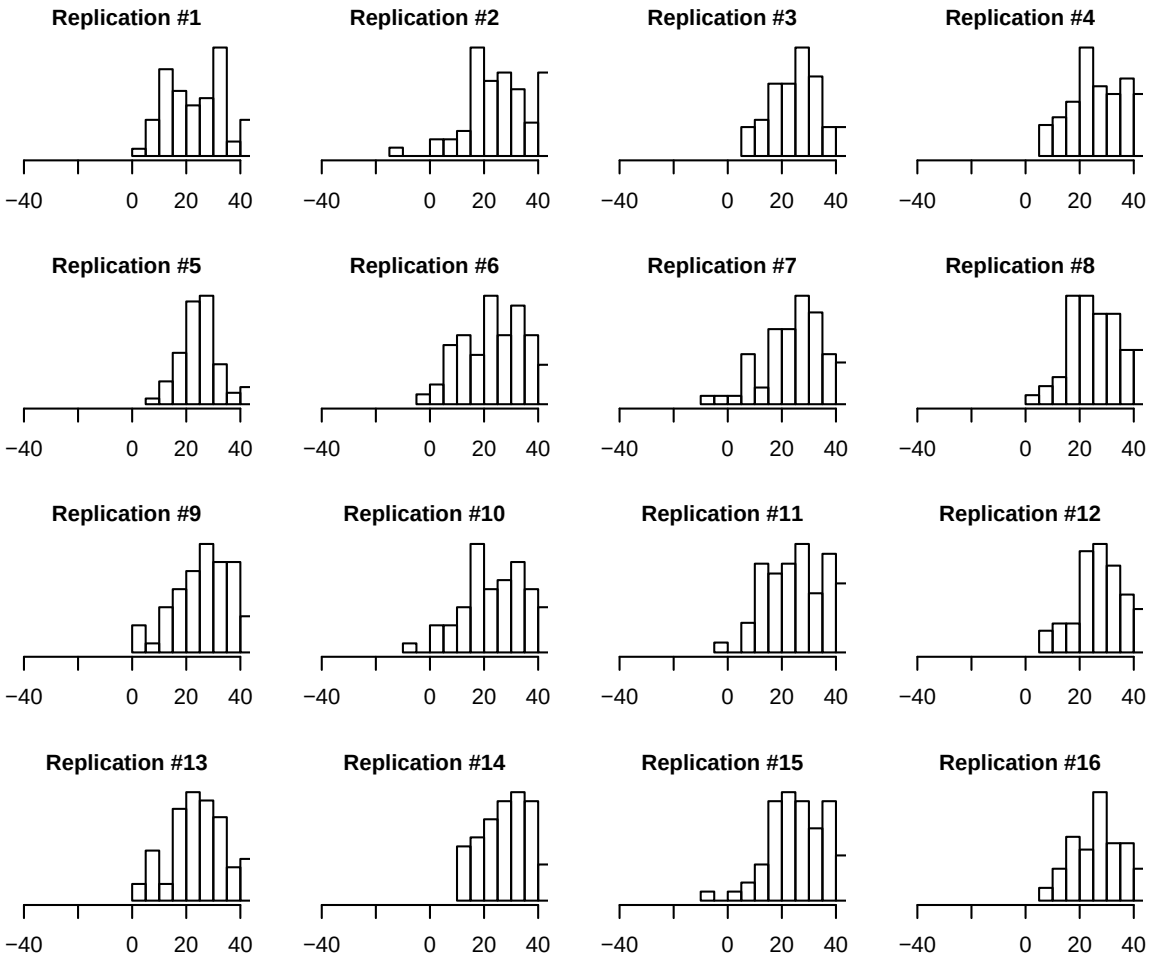
- Use these simulations to create 1000 fake datasets of 66 observations

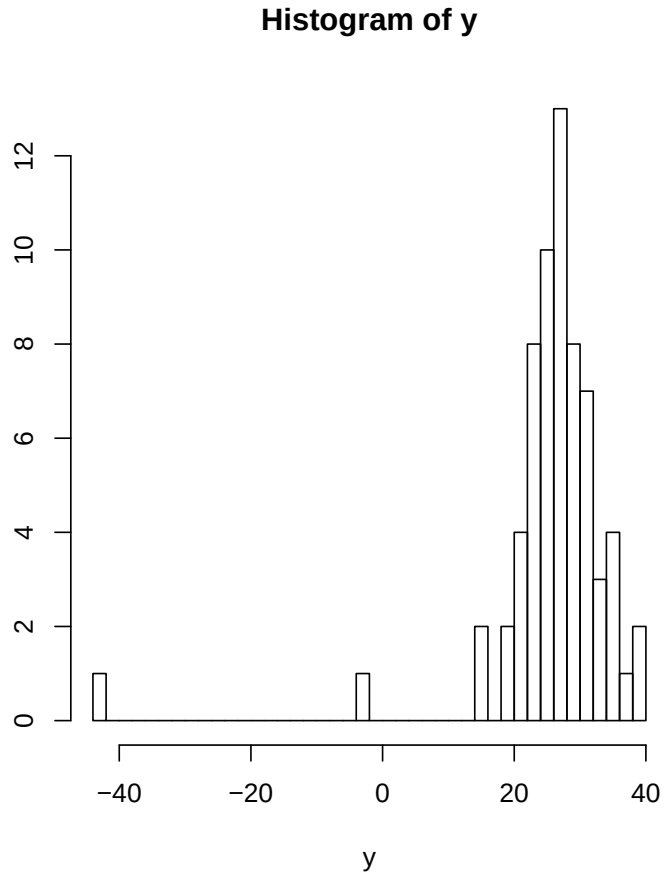
```
n <- length (y)   # n = 66
y.rep <- array (NA, c(n.sims, n))
for (s in 1:n.sims){
  y.rep[s,] <- rnorm (n,sim.light@coef[s], sim.light@sigma[s])
}
```

Example: Simon Newcomb's Speed of Light

Visual comparison of actual and replicated datasets

- Obvious difference among y^{rep} and y

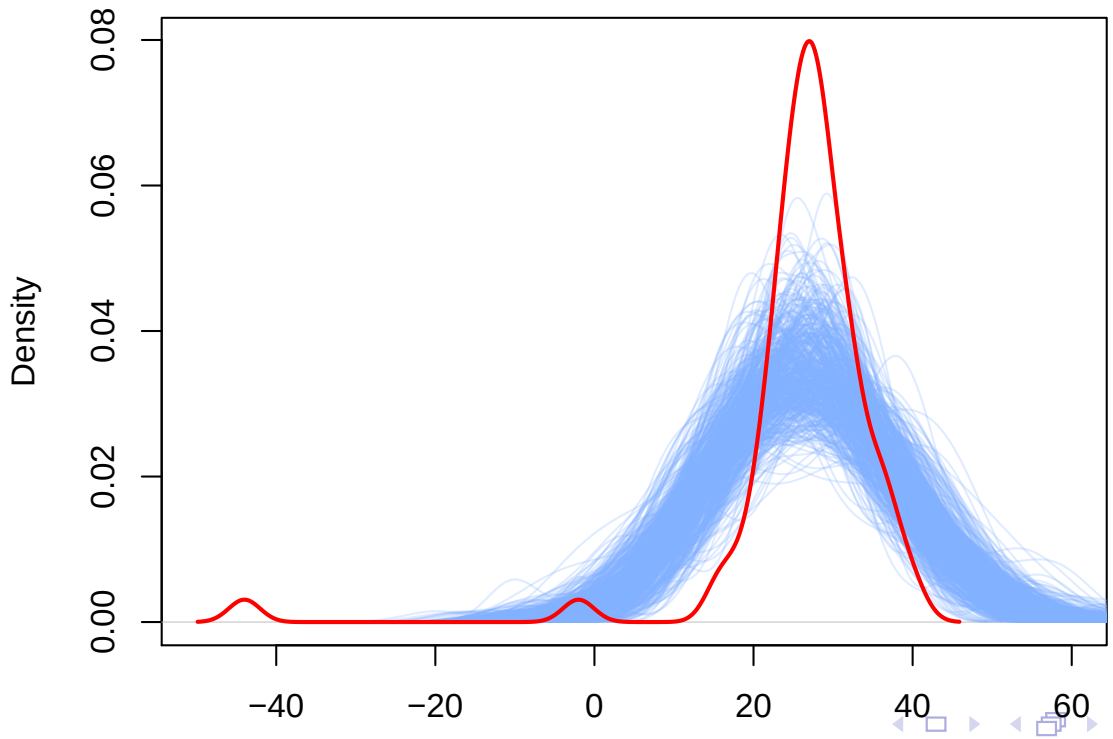




Example: Simon Newcomb's Speed of Light

Visual comparison of actual and replicated datasets

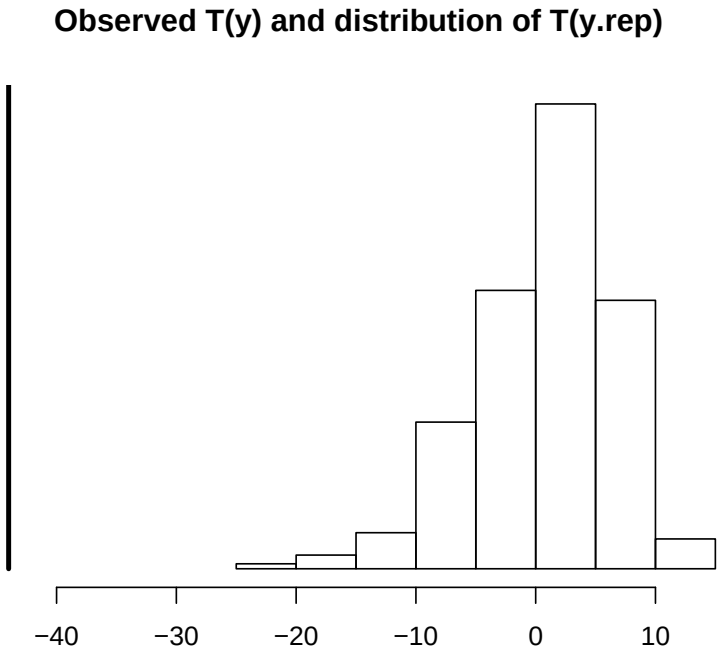
```
plot(density(y), xlim = c(-50, 60), type = 'n', main = "y.rep vs. y")
for(s in 1:500){
  lines(density(y.rep[s,]), col = "#82b1ff50")
}
lines(density(y), col = "red", lwd = 2)
```



Checking model fit using a numerical data summary

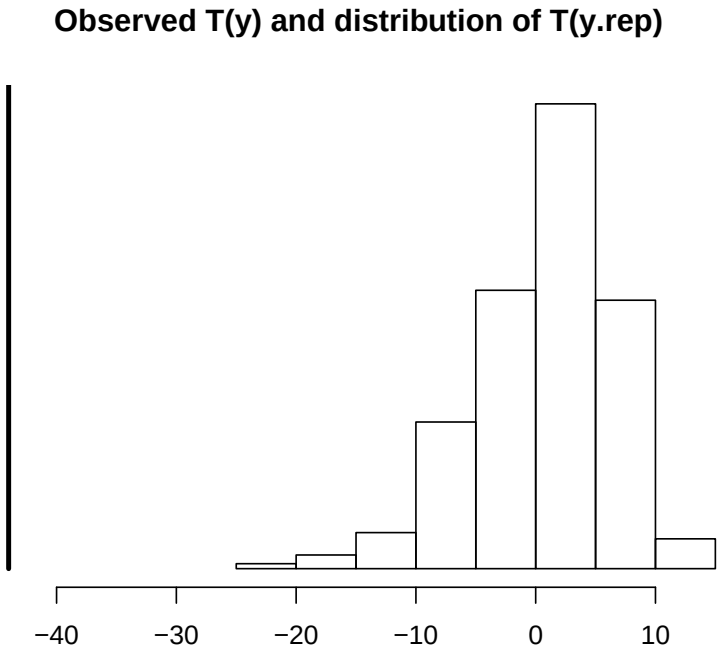
- Graphical checks show that the data have some extremely low values that do not appear in the replications
- Formalize this check by defining a test statistic
  - $T(y) = min(y)$, minimum value of the data
- Calculate $T(y^{\text{rep}})$ for each of the replicated datasets

```
Test <- function (y){
  min (y)
}
test.rep <- rep (NA, n.sims)
for (s in 1:n.sims){
  test.rep[s] <- Test (y.rep[s,])
}
hist(test.rep)
abline(v=Test(y), lwd =3)
```



Checking model fit using a numerical data summary

- Smallest observations in each of the hypothetical replications (y^{rep}) are all much larger than Newcomb's smallest observation
- $min(y)$ is indicated by a vertical line on the graph
- Normal model clearly does not capture the variation that Newcomb observed.
- A revised model might use an asymmetric distribution



Simulations of Probability Model

Summary

- Simulate from our obtained solution
- Summarize generated data to obtain insight that otherwise would be difficult to obtain
- Model checking
- Visualize and check assumptions
- Visualize and characterize uncertainty in parameters
- By “reversing” regression models we also assume that observed data is sampled from a normal distribution or some other well-known theoretical distribution

Bootstrapping

- Simulations after fitting a model share same assumptions as the model

e.g Normality or any other known distribution

- There will be cases in when we are not willing to assume a distribution
- Our parameters and tests will be biased

- One way out of this problem is *bootstrapping*

Bootstrapping

- Goal of bootstrapping
- Generate an empirical distribution of a test statistic or set of test statistics
- Approach: Repeated random sampling with replacement from the *original sample*

i.e. rather than generating samples *after* having fit our model, we sample from the original data

- As with simulations, we obtain a large number of re-sampled datasets
  - Obtain features of data
  - ▷ Test statistics

Bootstrapping

Example

Logic of bootstrapping:

- Goal: Obtain 95% CI for a sample mean
- Sample with $N = 10$, $M = 40$, and a sample $SD = 5$
- For normally distributed data: $\bar{X} - t\frac{s}{\sqrt{n}} < \mu < \bar{X} - t\frac{s}{\sqrt{n}}$
- With $\alpha = 0.05$, the critical value for $1 - \alpha/2$ is $t = 2.262$

R: `qt(.975, 9)`

- We would expect:
  $40 - 2.262(5/3.163) < \mu < 40 - 2.262(5/3.163)$
  $36.424 < \mu < 43.577$

- What if you aren't willing to assume that the sampling distribution of the mean is normally distributed?
  - ▷ Bootstrap!

Bootstrapping

Example

Obtaining CI:

1. Randomly select 10 observations from the sample, with replacement after each selection.
   - ▷ Some observations may be selected more than once, and some may not be selected at all.
2. Calculate and record the sample mean.
3. Repeat the first two steps many times. Eg. 1,000 times
4. Order the 1,000 sample means from smallest to largest.
5. Find the sample means representing the 2.5th and 97.5th percentiles. In this case, it's the 25th number from the bottom and top. These are your 95% confidence limits.

Bootstrapping

Bootstrapping in R

In general, bootstrapping involves three main steps:

1. Write a function that returns the statistic or statistics of interest. If there is a single statistic (for example, a median), the function should return a number. If there is a set of statistics (for example, a set of regression coefficients), the function should return a vector.
2. Process this function through the `boot()` function in order to generate R bootstrap replications of the statistic(s).
3. Use the `boot.ci()` function to obtain confidence intervals for the statistic(s) generated in step 2.

Bootstrapping

Bootstrapping in R

Example `mtcars`

- 32 automobiles reported in the 1974 Motor Trend magazine
- Multiple regression to predict miles per gallon from a car's weight (lb/1,000) and engine displacement (cu. in.).
- In addition to the standard regression statistics, we'd like to obtain a 95% confidence interval for the R-squared value
- First: Write a function to obtain R^2

```
rsq <- function(formula, data, indices) {
   d <- data[indices,]   ## Required for boot()
   fit <- lm(formula, data=d)
   return(summary(fit)$r.square)
}
```

- Function returns the R-squared value from a regression

Bootstrapping

Bootstrapping in R

Example `mtcars`

- Next: Draw large number of bootstraps - e.g 1,000

```
library(boot)
set.seed(1234)
results <- boot(data=mtcars, statistic=rsq,
                R=1000, formula=mpg~wt+disp)
```

- The original R^2 is

```
> summary(lm(mpg~wt+disp, data = mtcars))$r.square
[1] 0.7809306
```

Bootstrapping

Bootstrapping in R

Example `mtcars`

- R^2 statistics from bootstrap

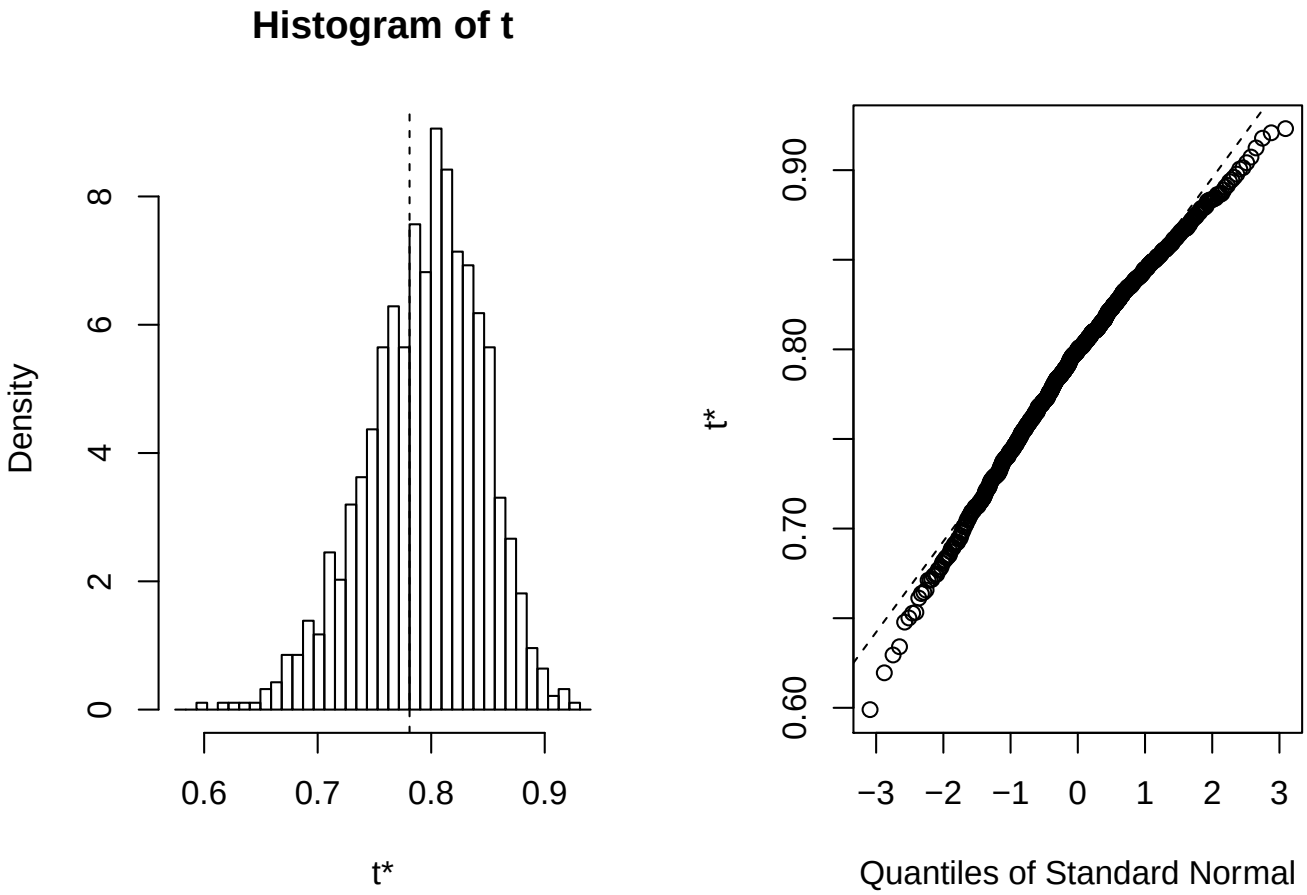
```
> print(results)
ORDINARY NONPARAMETRIC BOOTSTRAP

Call:
boot(data = mtcars, statistic = rsq, R = 1000,
     formula = mpg ~ wt + disp)

Bootstrap Statistics :
     original    bias    std. error
t1* 0.7809306 0.0133367  0.05068926
```

Bootstrapping

Bootstrapping in R

Example `mtcars`

- Visualize distribution on bootstrapped R^2



Bootstrapping

Bootstrapping in R

Example `mtcars`

- Obtain 95% CI

```
> boot.ci(results, type=c("perc", "bca"))
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = c("perc", "bca"))

Intervals :
Level     Percentile            BCa
95%   ( 0.6838,  0.8833 )   ( 0.6344,  0.8549 )
Calculations and Intervals on Original Scale
Some BCa intervals may be unstable
```

Bootsrapping

- Test hypotheses and form confidence intervals without reference to a known theoretical distribution.
- Very valuable when your data comes from unknown population distributions
  - when there are serious outliers
  - sample sizes are small
  - no existing parametric methods to answer the hypotheses of interest
- If your original samples are not representative of the population of interest then these techniques won't help!

Causal Inference

Three basic concepts are used to define causal effects (Rubin, 2007)

- **Unit**: physical object, for example, a patient, at a particular place and point of time, say time t.
- **Treatment**: action or intervention that can be initiated or withheld from that unit at t
  - e.g., a drug or a behavioral intervention etc.
  - If the active treatment is withheld, we will say that the unit has been exposed to the control treatment.
- Associated with that unit are two **potential outcomes** at a future point in time, $t^* > t$:
  - Value of some outcome measurements Y if the *treatment* is given at t
  - the value of Y at the same future point for the *control* at t.
- **Causal effect** of treatment: comparison of the treatment and control potential outcomes at t^*

Causal Inference

Gelman and Hill (2007) compare causal inference to predictive inference.

- Causal inference: What would happen to an outcome y as a result of a hypothesized "treatment" or intervention.
  - In a regression framework, the treatment can be written as a variable:

$$T = \begin{cases} 1 & \text{if unit } i \text{ receives the treatment} \\ 0 & \text{if unit } i \text{ receives the control} \end{cases}$$

- In regression context:
  - "**Predictive inference** relates to comparisons *between* units, whereas causal inference addresses comparisons of different treatments if applied to the *same* units. More generally, causal inference can be viewed as a special case of prediction in which the goal is to predict what would have happened under different treatment options." (p. 167, Gelman and Hill, 2007)

Zero Causal Effect but Positive Predictive Comparison

Hypothetical Example

- 100 patients receive the treatment and 100 receive the control condition
- Causal effect: Comparison between what would have happened to a given patient had she received the treatment vs. control
- Scenario: Treatment has no effect
  - ▷ Causal effect of the treatment is zero.
- Treatment and control groups systematically differ
  - Healthier patients receiving the treatment
  - Sicker patients receiving control

Zero Causal Effect but Positive Predictive Comparison

Hypothetical Example

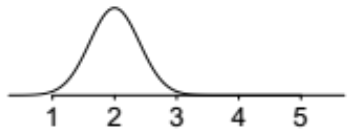
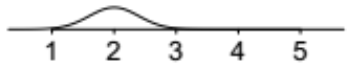
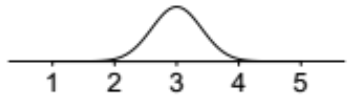
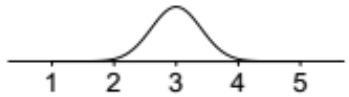
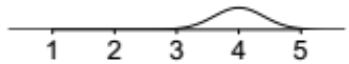
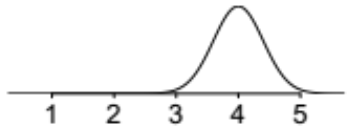
Previous health status	Distribution of measurements after experiment if assigned the control condition	if assigned the treatment
Poor		
Fair		
Good		

Figure 9.1 (Gelman & Hill, 2007)

- Distributions of potential outcomes are *identical* under control and treatment.
- Overall mean difference!

Zero Causal Effect but Positive Predictive Comparison

Hypothetical Example

- Distributions are centered at same place for both treatment and control
- Overall mean difference represents self-selection into conditions at different rates
- Heights of each distribution reflect the differential proportions
- Positive *predictive comparison*
- ▷ Causal effect is zero

Positive Causal Effect but Zero Predictive Comparison

Hypothetical Example

- Opposite scenario
- 100 patients receive the treatment and 100 receive the control condition
- Scenario: Treatment has a positive effect
- ▷ Causal effect of the treatment is *non*zero.
- Predictive effect is zero
- Treatment and control groups systematically differ
  - Healthier patients receiving the control
  - Sicker patients receiving treatment

Positive Causal Effect but Zero Predictive Comparison

Hypothetical Example

Previous health status	Distribution of measurements after experiment if assigned the control condition	if assigned the treatment
Poor		
Fair		
Good		

Figure 9.2 (Gelman & Hill, 2007)

- Treatment group are centered one point to the right of the corresponding distributions in the control group.
- Overall *no* mean difference!

Positive Causal Effect but Zero Predictive Comparison

Hypothetical Example

- Distributions are shifted by one unit for control and treatment
- Still: Self-selection into conditions at different rates
- No *predictive comparison*
  - Causal effect is positive

- Previous health status plays an important role
  - Related both to treatment assignment and future health status
- Confounding covariate: Effect of treatment is "confounded" with effect of previous health status

Causal Effect and Predictive Comparison

- In this simple example:
- Simple solution to confounding varible
  - Compare treated and control units *conditional* on previous health status.
  - ▷ Compare only within groups
  - ▷ Regression approach: Include treatment indicator *and* previous health status
- Similar problem in Sympson's paradox
- In this easy case, accurate estimation comes down to proper modeling
- If the confounding covariates are not observed
  - "omitted" or "lurking" variables

Omitted Variable Bias

- Quantify the bias incurred by excluding a confounding covariate
- True model:

  $y_i = \beta_0 + \beta_1 T_i + \beta_2 x_i + \epsilon_i$
- T_i is the treatment and x_i is the covariate for unit i
- If confounding covariate, x_i, is ignored:

  $y_i = \beta_0^* + \beta_1^* T_i + \epsilon_i^*$
- What is the relation between these models?
- ▷ Define a third regression:

  $x_i = y_0 + y_1 T_i + \nu_i$
- Substituting x into the original equation, and rearranging:

  $y_i = \beta_0 + \beta_2 y_0 + (\beta_1 + \beta_2 \gamma_1) T_i + \epsilon_i + \beta_2 \nu_i$
- Equating the coefficients of T yields

  $\beta_1^* = \beta_1 + \beta_2^* \gamma_1$

Omitted Variable Bias

$$\beta_1^* = \beta_1 + \beta_2^* \gamma_1$$

- If there is no association between the *treatment* and confounder
  - $\gamma_1 = 0$
- If there is no association between the *outcome* and confounder
  - $\beta_2 = 0$
- Then the variable is not a confounder because there will be no bias
  - $\beta_2^* \gamma_1 = 0$ and $\beta_1^* = \beta_1$

The Fundamental Problem of Causal Inference

- Problem of estimating the causal effect
- Formally: Causal effect of a treatment T on an outcome y for an observational or experimental unit i can be defined by comparisons between the outcomes that *would have occurred* under each of the different treatment possibilities

- Binary treatment T taking on the value 0 (control) or 1 (treatment)
- Outcome y_i^0 for control
- Outcome y_i^1 for treatment

The Fundamental Problem of Causal Inference

The problem:

- For someone assigned to the treatment condition (that is, $T_i = 1$)
  - y_i^1 is observed and y_i^0 is the unobserved *counterfactual* outcome
  - ▷ *what would have happened* to the individual if assigned to control
- A simple treatment effect for unit i can be defined as

  treatment effect for unit $i = y_i^1 - y_i^0$

The Fundamental Problem of Causal Inference

Example

- Hypothetical data for an experiment with 100 units (200 potential outcomes).
- The table shows what the data that is required to determine causal effects for each person in the dataset
  - ▷ it includes both potential outcomes for each person

Units	X_i	T_i	y_i^0	y_i^1	$y_i^0 - y_i^1$
1	2	0	**69**	75	-6
2	6	1	80	**76**	4
3	3	1	71	**69**	1
⋮					
100	9	0	**81**	78	3

The Fundamental Problem of Causal Inference

Example

Units	X_i	T_i	y_i^0	y_i^1	$y_i^0 - y_i^1$
1	2	0	**69**	75	-6
2	6	1	80	**76**	4
3	3	1	71	**69**	1
⋮					
100	9	0	**81**	78	3

The observed data is actually

Units	X_i	T_i	y_i^0	y_i^1	$y_i^0 - y_i^1$
1	2	0	**69**	?	?
2	6	1	?	**76**	?
3	3	1	?	**69**	?
⋮					
100	9	0	**81**	?	?

The Fundamental Problem of Causal Inference

The observed data is actually

Units	X_i	T_i	y_i^0	y_i^1	$y_i^0 - y_i^1$
1	2	0	**69**	?	?
2	6	1	?	**76**	?
3	3	1	?	**69**	?
⋮					
100	9	0	**81**	?	?

- The Fundamental Problem of Causal Inference:
  - Only one of these two potential outcomes, y_i^0 and y_i^1, can be observed for each unit i.
  - The y_i^1 values are "missing" for those in the control group
  - and the y_i^0 values are "missing" for those in the treatment group.

Ways of Getting Around the Problem

- We cannot observe both – we can never measure a causal effect directly.
- Essentially, we can think of causal inference as a prediction of what would happen to unit i if $T_i = 0$ or $T_i = 1$.
- ▷ Predictive inference in the potential-outcome framework
- Estimating causal effects requires one or some combination of the following:
  - *Close substitutes* for the potential outcomes (e.g. Propensity score matching)
  - Randomization
  - Statistical adjustment

Close substitutes

- Create situations where one can measure both y_i^0 and y_i^1 on the same unit
- In the natural and physical sciences
  - e.g Imagine dividing a piece of plastic into two parts and then exposing each piece to a corrosive chemical
    - ▷ Hidden assumption: Pieces are identical in how they would respond with and without treatment, that is, $y_1^0 = y_2^0$ and $y_1^1 = y_2^1$.
- Pre-Post
  - ▷ Hidden assumption: Pre-treatment measure accounts for all confounding sources
- Propensity Score Matching (PSM)

Randomization

- We cannot compare treatment and control outcomes for the same units
  - Try to compare them on similar units.
- Similarity can be attained by using *randomization* when assigning to treatment ot control
- Cleanest scenario
- Randomization enables unbiased estimation of treatment effects; for each covariate, randomization implies that treatment-groups will be balanced on average

Average Causal Effects and Randomized Experiments

- In most practical situations it's impossible to estimate individual-level causal effects
- Design studies to estimate the population average treatment effect:

  average treatment effect $= \bar{y}^1 - \bar{y}^0$,

  where $\bar{y}^1 = \sum_{i=1}^{n_1} y_i^1 / n_1$ and $\bar{y}^0 = \sum_{i=1}^{n_0} y_i^0 / n_0$

- More broadly:
- Think of the control group as a group of units that could just as well have ended up in the treatment group
- On average, their outcomes represent what would have happened
- ▷ Control group plays an essential role in causal analysis

Average Causal Effects and Randomized Experiments

- If n_0 treatment units are selected at random from the population
- And n_1 control units are selected at random from the population
  - Then: Observed sample averages can be used to estimate the corresponding population quantities, $\bar{y}^0$ and $\bar{y}^1$ with standard error $\sqrt{s_0^2/n_0 + s_1^2/n_1}$
- Control group mean can act as a counterfactual for the treatment group (and vice versa).
- What if the $n_0 + n_1$ units are selected nonrandomly from the population but then treatment is assigned at random within this sample?
- Common practice in human subjects research
  - In psychology, experiments are often conducted on university students taking introductory psychology courses.
  - Causal inferences are still justified, but inferences no longer generalize to the entire population.

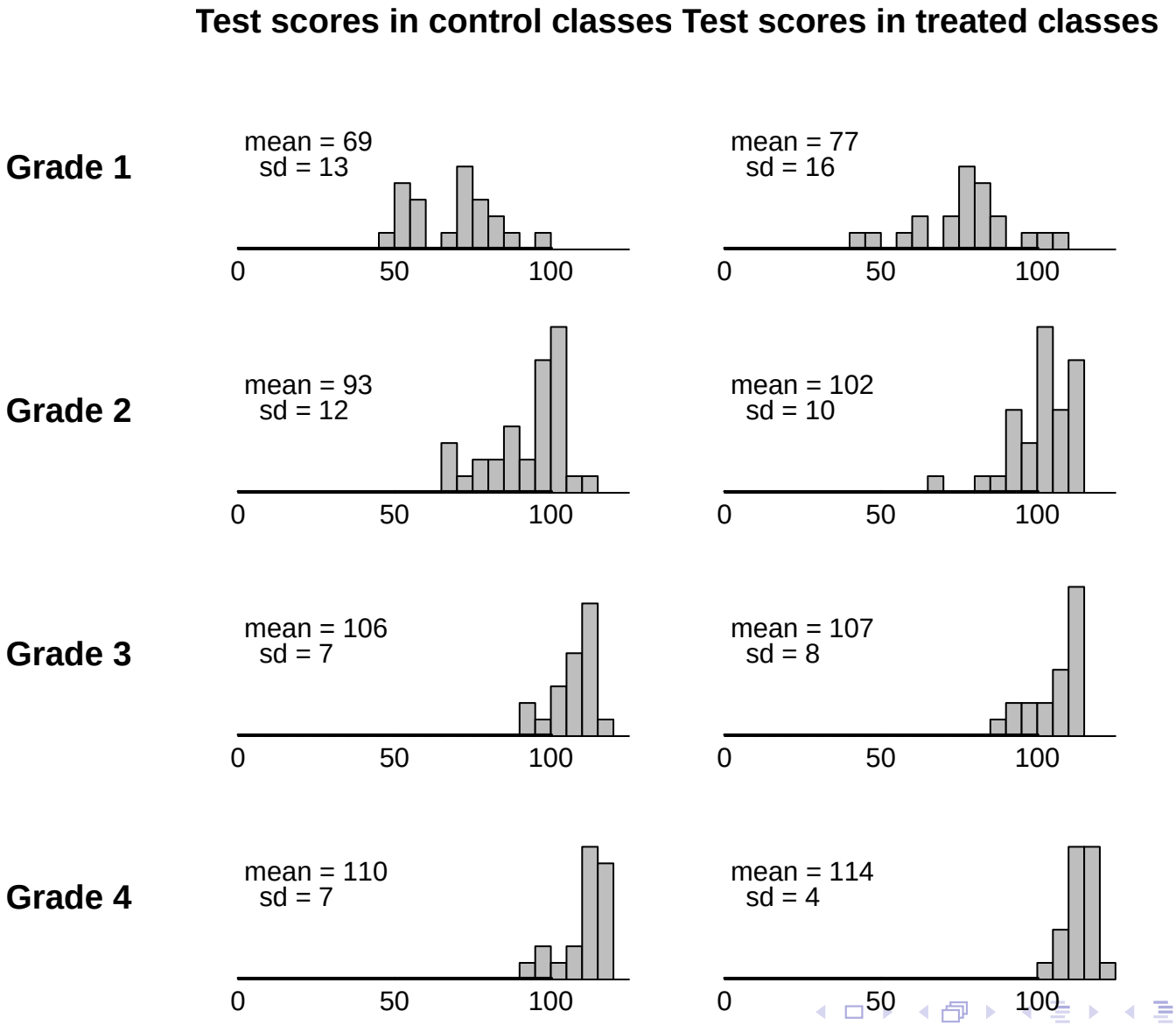
Average Causal Effects and Randomized Experiments

Example: Educational Television Show

- Educational experiment performed around 1970 on a set of elementary school classes
- The experiment was performed in two cities (Fresno and Youngstown)
- Treatment: Exposure to a new educational television show called The Electric Company
- Within each school
  - Selected the two poorest reading classes of that grade
  - For each pair, one of these classes was randomly assigned to continue with its regular reading course and the other was assigned to view the TV program
- Four grades with each treatment and control
- At the end of the school year: Reading test
- Average reading score for each school class

Average Causal Effects and Randomized Experiments

Example: Educational Television Show



Basic analysis of a completely randomized experiment

Example: Educational Television Show

- If treatments are assigned completely at random:
- Different treatment groups may be seen as a set of random samples from a common population.
  - ▷ Population average under each treatment is $\bar{y}^0 - \bar{y}^1$

i.e. Average causal effect of the treatment corresponds to the coefficient θ in the regression, $y_i = \alpha + \theta T_i + \epsilon_i$

```
for (k in 1:4) {
display (lm (post.test ~ treatment, subset=(grade==k)))
}
```

Basic analysis of a completely randomized experiment

Example: Educational Television Show

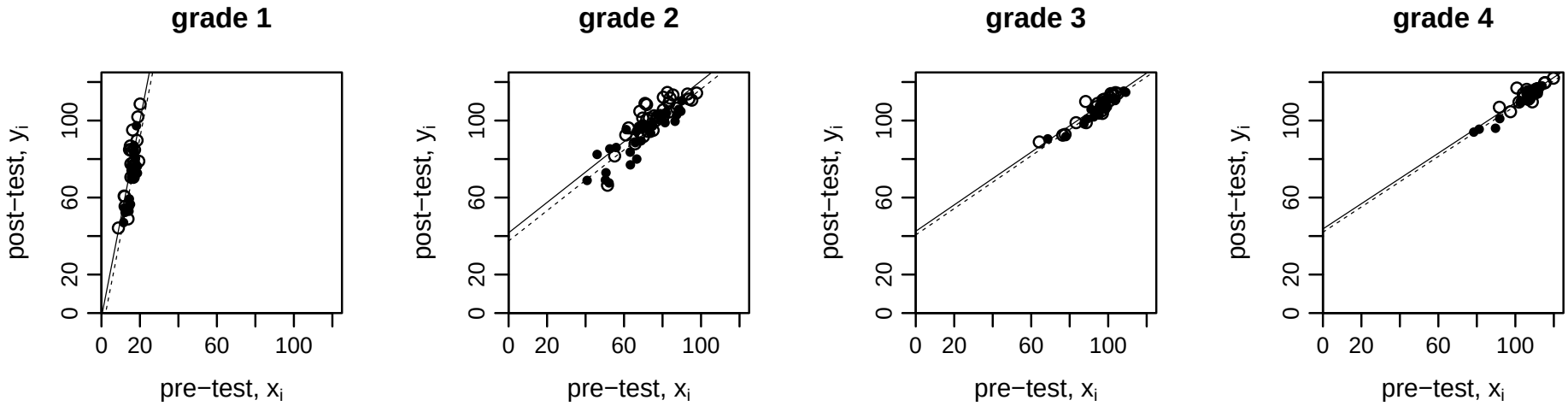
```
(grade == 1)
            coef.est  coef.se
(Intercept) 68.79      3.27
treatment    8.30      4.62
---
n = 42, k = 2
residual sd = 14.98, R-Sq. = 0.07

(grade == 2))
            coef.est  coef.se
(Intercept) 93.21      1.91
treatment    8.36      2.70
---
n = 68, k = 2
residual sd = 11.12, R-Sq. = 0.13
```

```
(grade == 3))
            coef.est  coef.se
(Intercept) 106.17     1.66
treatment     0.33     2.35
---
n = 40, k = 2
residual sd = 7.44, R-Sq. = 0.00

(grade == 4))
            coef.est  coef.se
(Intercept) 110.36     1.30
treatment     3.71     1.84
---
n = 42, k = 2
residual sd = 5.95, R-Sq. = 0.09
>
```

Controlling for pre-treatment predictors

Example: Educational Television Show

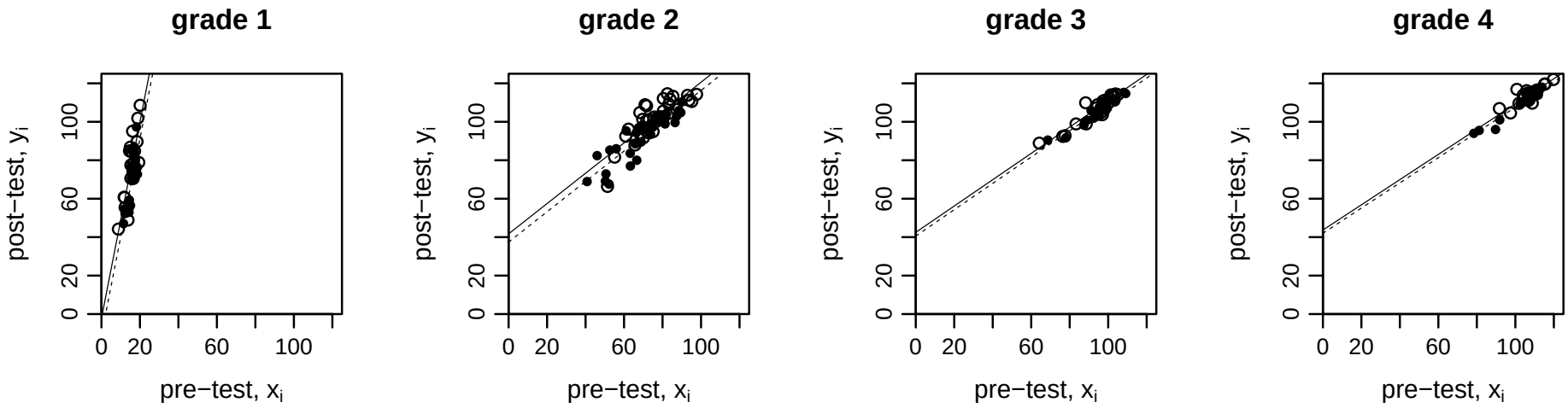
- All kids were administered a pre-test x_i at the beginning of school year
- Treatment effect can be estimated using a regression model: $y_i = \alpha + \theta T_i + \beta x_i + \epsilon_i$ on the pre-treatment predictor x



- Difference between regression lines for both groups represents the treatment effect as a function of pre-test score

Controlling for pre-treatment predictors

Example: Educational Television Show



- For grades 2-4, the pre-test was the same as the post-test, and so it is no surprise that all the classes improved whether treated or not.
- For grade 1, the pre-test was a subset of the longer test, which is probably reason why the pre-test scores for grade 1 are so low

But Causal effects are in the difference between treatment and control conditions

Controlling for pre-treatment predictors

Example: Educational Television Show

- Note that we controlled fro pre-test condition x with

  $y_i = \alpha + \theta T_i + \beta x_i + \epsilon_i$
- Doing so can improve efficiency of estimate.
- We can control for many pre-treatment predictors X

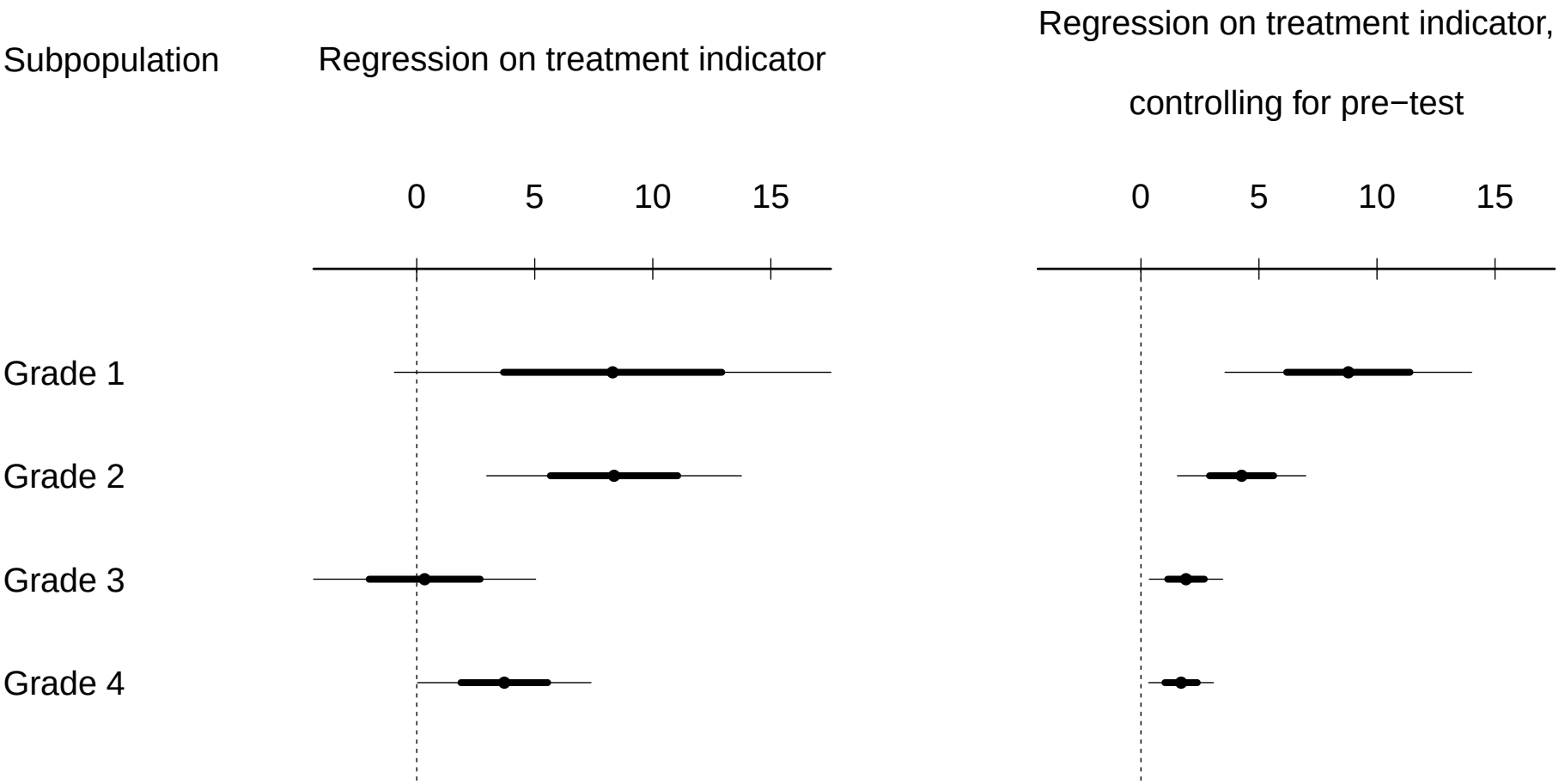
  $y_i = \alpha + \theta T_i + \beta X_i + \epsilon_i$
- We can rerun our analyses

```
for (k in 1:4) {
display (lm (post.test ~ treatment + pre.test, subset=(grade==k)))
}
```

Controlling for pre-treatment predictors

Example: Educational Television Show

- Instead of Tabulating we can plot results
- Point estimate with 50% and 95% intervals: Treatment is effective



Observational studies

- In practice, however, we often work with observational data
- Treatments are observed rather than assigned
- There can be systematic differences between groups of units that receive different treatments
  - Differences that are outside the control of the experimenter – and they can affect the outcome, y.
- Several strategies
  - Controlling for confounding covariates through linear regression
  - Propensity Score Matching (PSM)

Propensity Score Matching (PSM)

- Paul Rosenbaum and Donald Rubin in 1983
- Observational studies: Assignment of treatments to research subjects is typically not random.
- Attempts to estimate the effect of a treatment by accounting for the covariates that predict receiving the treatment
- Matching attempts to mimic randomization
  - ▷ Identify a sample of units that received the treatment that is comparable on all observed covariates to a sample of units that did not receive the treatment.
- Reduce the bias due to confounding variables when comparing $y_i^1 - y_i^0$

Propensity Score Matching (PSM)

Normal matching:

- Match on single characteristics that distinguish treatment and control groups
  - ▷ if the two groups do not have substantial overlap, then substantial error may be introduced

Propensity Score Matching

- In non-experimental settings in which:
  - few units in the non-treatment comparison group are comparable to the treatment units
  - and selecting a subset of comparison units similar to the treatment unit is difficult because units must be compared across a high-dimensional set of pretreatment characteristics.
- PSM employs a predicted probability of group membership to create a counterfactual group

Propensity Score Matching (PSM)

- Definition of Propensity Score: Probability of a unit (e.g., person, classroom, school) being assigned to a particular treatment given a set of observed covariates.
- Propensity scores are used to reduce selection bias by equating groups based on these covariates
- Let Y_0 and Y_1 denote the potential outcomes under control and treatment
- Then treatment assignment is (conditionally) unconfounded if potential outcomes are independent of treatment conditional on background variables X
  - $Y_0, Y_1 \perp T \,|\, X$
- If unconfoundedness holds, then
  - $Y_0, Y_1 \perp (T \,|\, p(X))$

Propensity Score Matching (PSM)

General Procedure

1. Logistic regression is natural candidate as it produces probabilities of group membership
   - Dependent variable: $Y = 1$, if participate; $Y = 0$, otherwise.
   - Select appropriate confounders (variables hypothesized to be associated with both treatment and outcome)
   - Obtain propensity score: predicted probability (p) or $\log[p/(1-p)]$.
2. Check distribution of propensity score
   - Score should be balanced across groups
   - Check that covariates are balanced across groups
3. Match each participant to one or more nonparticipants on propensity score:
   - Different techniques: Nearest neighbor, Caliper matching, Mahalnobis metric, stratification, exact matching, difference-in-difference matching...
4. Verify that covariates are balanced across groups
5. Multivariate analysis based on new sample

Propensity Score Matching (PSM)

Example

- "lalonde" data from the `MatchIt` library
- Data from National Supported Work Demonstration and PSID, as analyzed by Dehejia and Wahba 1999

Causal Effects in Nonexperimental Studies: Reevaluating the Evaluation of Training Programs

Rajeev H. DEHEJIA and Sadek WAHBA

This article uses propensity score methods to estimate the treatment impact of the National Supported Work (NSW) Demonstration, a labor training program, on postintervention earnings. We use data from Lalonde's evaluation of nonexperimental methods that combine the treated units from a randomized evaluation of the NSW with nonexperimental comparison units drawn from survey datasets. We apply propensity score methods to this composite dataset and demonstrate that, relative to the estimators that Lalonde evaluates, propensity score estimates of the treatment impact are much closer to the experimental benchmark estimate. Propensity score methods assume that the variables associated with assignment to treatment are observed (referred to as ignorable treatment assignment, or selection on observables). Even under this assumption, it is difficult to control for differences between the treatment and comparison groups when they are dissimilar and when there are many preintervention variables. The estimated propensity score

Propensity Score Matching (PSM)

Example

- Data peak

```
> head(lalonde)
  treat age educ black hispan married nodegree re74 re75     re78
NSW1   1  37   11     1      0       1        1    0    0  9930.04
NSW2   1  22    9     0      1       0        1    0    0  3595.89
NSW3   1  30   12     1      0       0        0    0    0 24909.45
NSW4   1  27   11     1      0       0        1    0    0  7506.14
NSW5   1  33    8     1      0       0        1    0    0   289.78
NSW6   1  22    9     1      0       0        1    0    0  4056.49
```

Propensity Score Matching (PSM)

Example

- Unmatched Sample

```
> mod0 <- lm(re78 ~ treat, data = lalonde)
> summary(mod0)

Call:
lm(formula = re78 ~ treat, data = lalonde)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6984.2      360.7  19.362   <2e-16 ***
treat         -635.0      657.1  -0.966    0.334
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05

Residual standard error: 7471 on 612 degrees of freedom
Multiple R-squared:  0.001524,Adjusted R-squared:  -0.0001079
F-statistic: 0.9338 on 1 and 612 DF,  p-value: 0.3342
```

Propensity Score Matching (PSM)

Example

- Logistic model

```
pscores.model <- glm(treat ~ re74 + married + black + educ,
                     family = binomial("logit"),
                     data = lalonde)
```

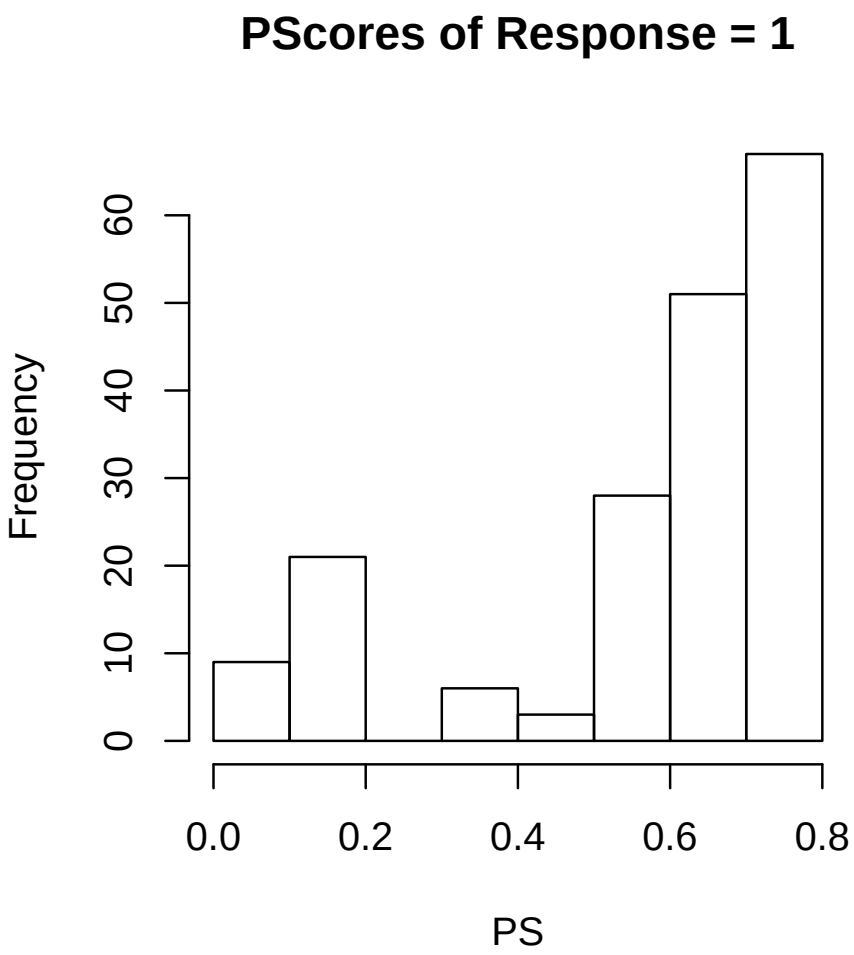
- Propensity scores are the predicted scores from logistic model

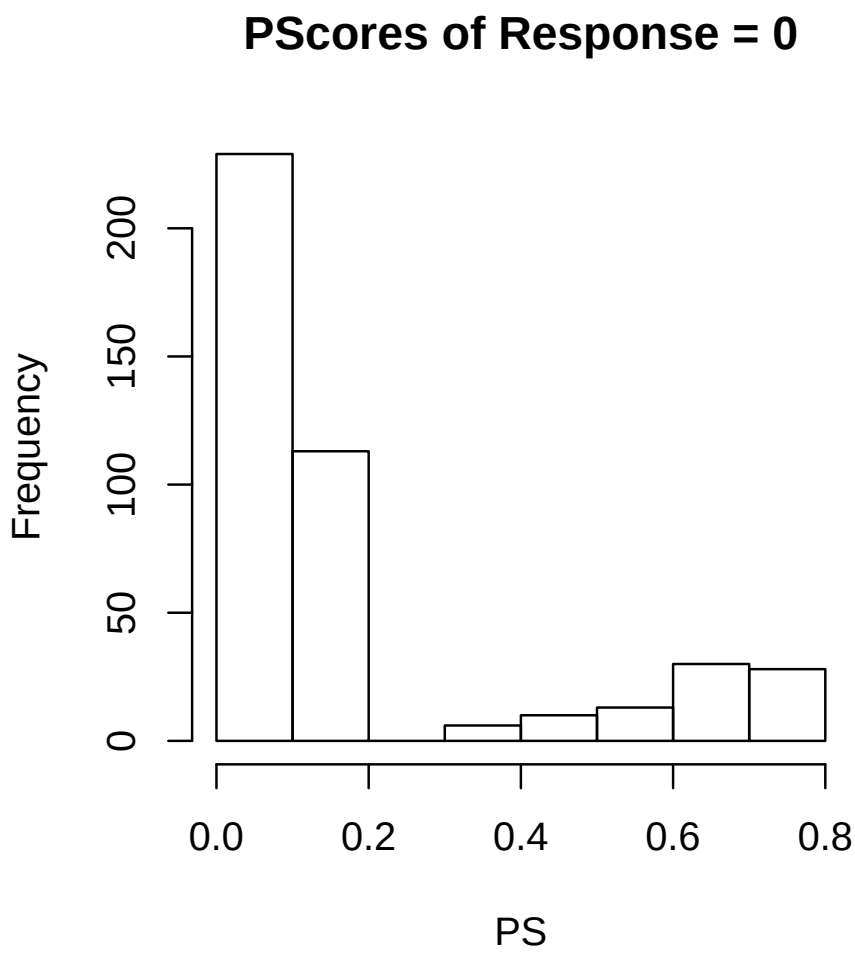
```
lalonde$PScores <- pscores.model$fitted.values
```

Propensity Score Matching (PSM)

Example

- Check distribution of propensity score: Unmatched





Propensity Score Matching (PSM)

Example

- Execute the matching algorithm
- Matche each treated unit with a control unit that has exactly the same values on each covariate.

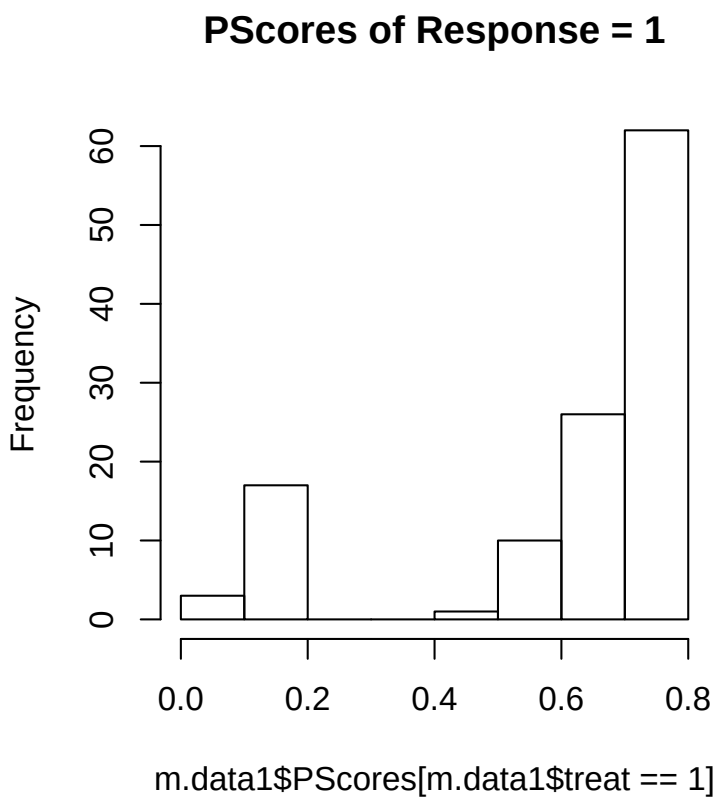
```
> match1 <- matchit(pscores.model, method="exact",data=lalonde)
> match1

Call:
matchit(formula = pscores.model, data = lalonde, method = "exact")

Exact Subclasses: 21

Sample sizes:
Control Treated
All           429     185
Matched        87     119
Unmatched     342      66
```

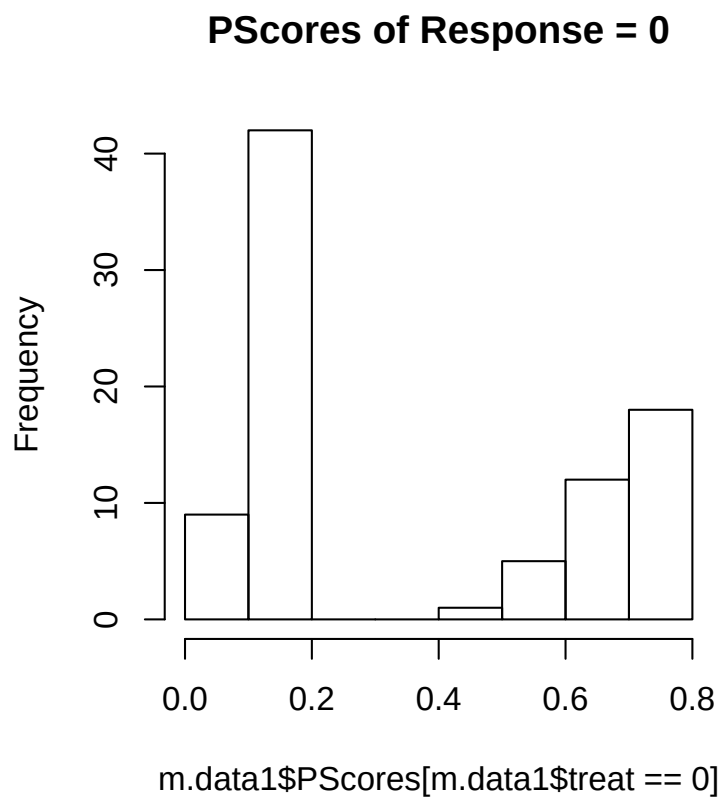
Propensity Score Matching (PSM)

Example

- Obtain matched data for later steps: Matched with "exact" method

```
m.data1 <- match.data(match1)
```





Propensity Score Matching (PSM)

Example

- Results with matched sample

```
> mod_m1<- lm(re78 ~ treat , data = m.data1)
> summary(mod_m1)

Call:
lm(formula = re78 ~ treat, data = m.data1)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4745.5      798.9   5.940 1.21e-08 ***
treat         1898.4     1051.1   1.806   0.0724 .
---

Residual standard error: 7452 on 204 degrees of freedom
Multiple R-squared:  0.01574,Adjusted R-squared:  0.01091
F-statistic: 3.262 on 1 and 204 DF,  p-value: 0.07238
```

Propensity Score Matching (PSM)

Example

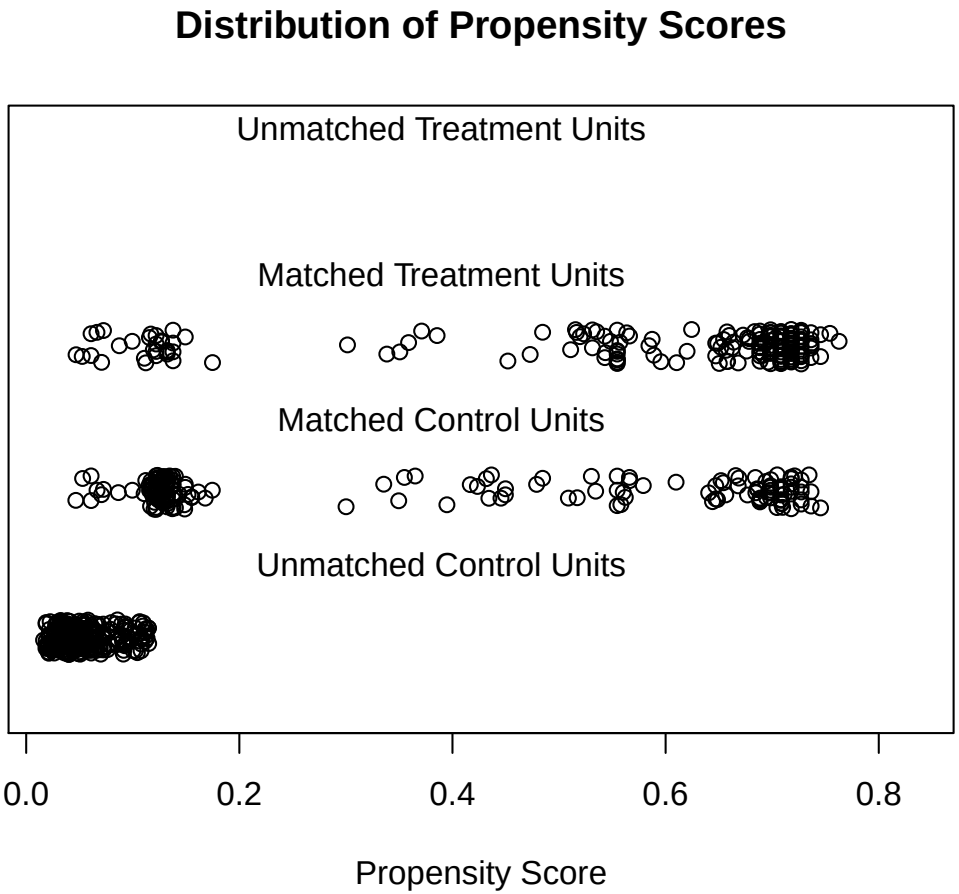
- Matching with different algorithm
- "Nearest" neighbor: his technique matches a treated unit to a control unit(s) that is closest in terms of a distance measure such as a logit

```
> match2 <- matchit(pscores.model, method="nearest",data=lalonde)
> summary(match2)

...[OMITTED OUTPUT]...

Sample sizes:
Control Treated
All             429     185
Matched         185     185
Unmatched       244       0
Discarded         0       0
```

Propensity Score Matching (PSM)

Example

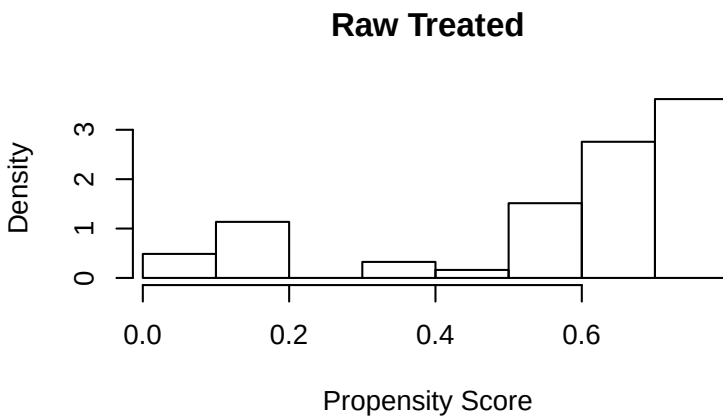
- Visualization of propensity score distribution

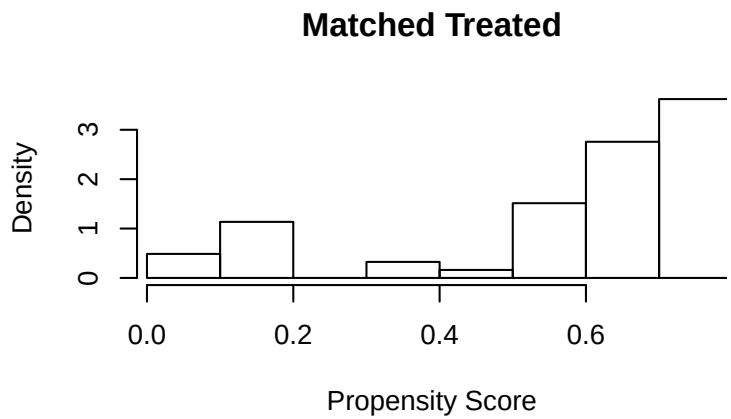


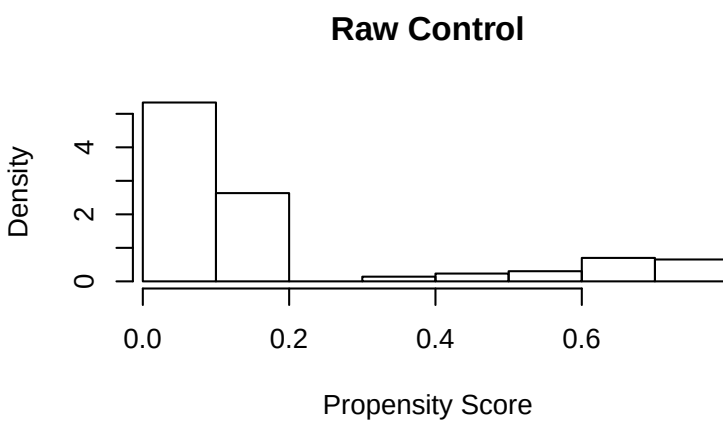
Propensity Score Matching (PSM)

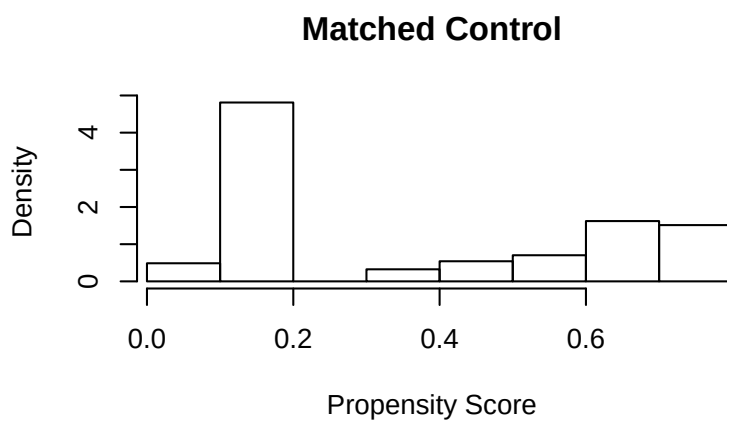
Example

- Visualization of propensity score distribution









Propensity Score Matching (PSM)

Example

- Results with matched sample

```
> match2.data <- match.data(match2)
> mod_m2 <- lm(re78 ~ treat, data = match2.data)
> summary(mod_m2)

Call:
lm(formula = re78 ~ treat, data = match2.data)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5295.9      509.9  10.386   <2e-16 ***
treat         1053.2      721.1   1.461    0.145
---
Residual standard error: 6935 on 368 degrees of freedom
Multiple R-squared:  0.005764,Adjusted R-squared:  0.003062
F-statistic: 2.133 on 1 and 368 DF,  p-value: 0.145
```

Propensity Score Matching (PSM)

- PSM balances treatment and control groups on a large number of covariates without losing a large number of observations
- Goal: Create balanced and random groups *post-hoc*
- PSM only accounts for observed (and observable) covariates
- PSM requires large samples, with substantial overlap between treatment and control groups.

Causal inference

- Distinction between causal inference and predictive inference
- These two types do not necessarily converge
- Under ideal conditions (e.g. perfect randomization) predictive inference converges on average with causal inference
- Causality can not actually be established: Counterfactual approach
- Observational studies do not offer ideal conditions
  - Randomization in a non-random sample
  - Matching: Find subsample with similar conditions to block covariates.
    - Propensity Score Matching
- Causality is in the best case an average effect