36-463/663: Multilevel & Hierarchical Models
Fall 2016
HW05 – SOLUTIONS

Announcements

1. Exercises from Ch's 7 & 8:

   (a) G&H Chapter 7, #10, parts a, b, c:

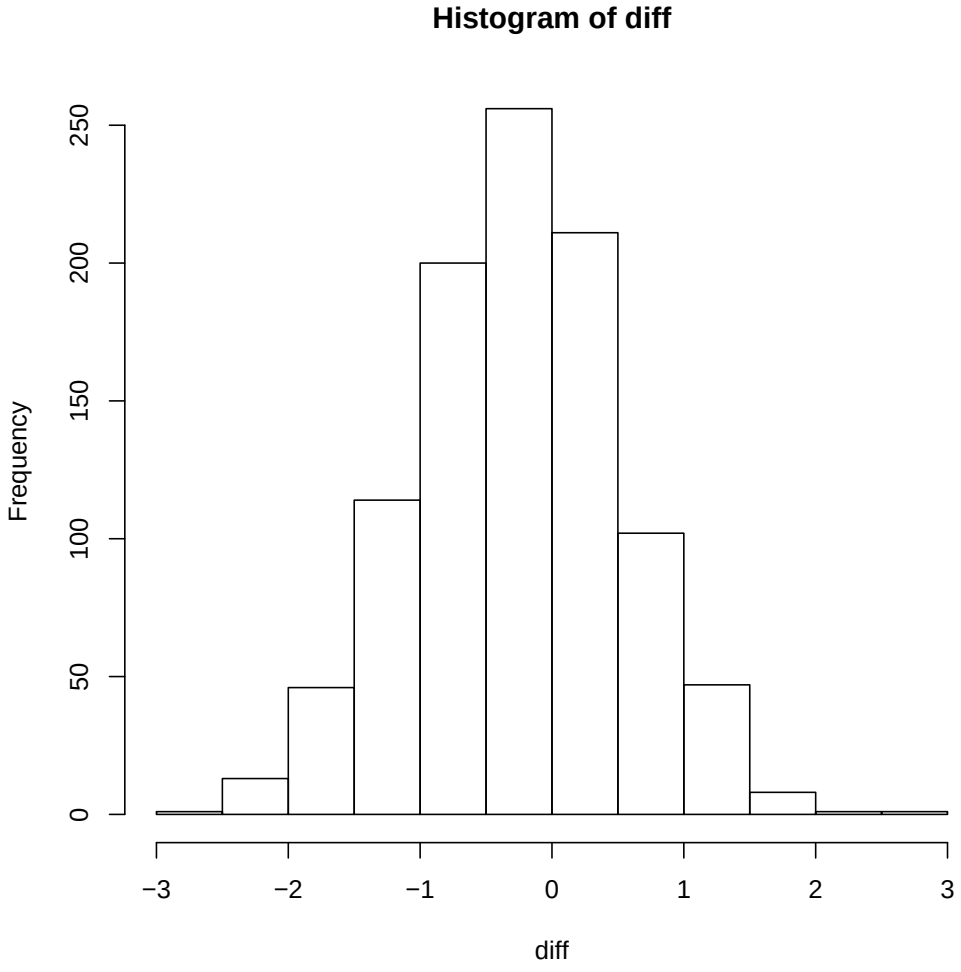
```
library(arm)
data <- read.csv("beauty/ProfEvaltnsBeautyPublic.csv")
object <- lm(courseevaluation ~ btystdave,data)
n<-10000
sim.1 <- sim(object,n)
display(object)

c(mean(sim.1@coef[,1]),sd(sim.1@coef[,1]))
[1] 4.01002335 0.02566414
c(mean(sim.1@coef[,2]),sd(sim.1@coef[,2]))
[1] 0.13284816 0.03232940
display(lm(formula = courseevaluation ~ btystdave, data = data))
            coef.est coef.se
(Intercept) 4.01     0.03
btystdave   0.13     0.03
---
n = 463, k = 2
residual sd = 0.55, R-Squared = 0.04
```

   We can see that after 10000 simulations, the mean and standard deviations of the coefficient estimates are very close to the output from display. The simulation variability is going to increase when we have smaller iterations. There is no fixed rule for how many simulations that are needed to give a good approximation. But given the computational power we have these days, people usually try @n ¿ 1000@.

   (b) G&H Chapter 7, #4, a & b:

```
library(arm)
data <- read.csv("beauty/ProfEvaltnsBeautyPublic.csv")
object <- lm(courseevaluation ~ btystdave + female + nonenglish +age,data = data)

n.sims <- 1000
sim.1000 <- sim(object,n.sims)
n.tilde <- 1
x.tilde <- cbind(rep(1,n.tilde),rep(-1,n.tilde),rep(1,n.tilde),
rep(0,n.tilde),rep(50,n.tilde))

x.tildeB <- cbind(rep(1,n.tilde),rep(-0.5,n.tilde),rep(0,n.tilde),
rep(0,n.tilde),rep(60,n.tilde))
```

```
y.tilde <- array(NA, c(n.sims,n.tilde))
y.tildeB <- array(NA, c(n.sims,n.tilde))
for (s in 1:n.sims){
y.tilde[s,] = rnorm(n.tilde,x.tilde %*% sim.1000@coef[s,],
sim.1000@sigma[s])

y.tildeB[s,] = rnorm(n.tilde,x.tildeB %*% sim.1000@coef[s,],
sim.1000@sigma[s])

}

diff <- y.tilde - y.tildeB
sum(diff>0)/length(diff)
0.37
hist(diff)
```



The prob-

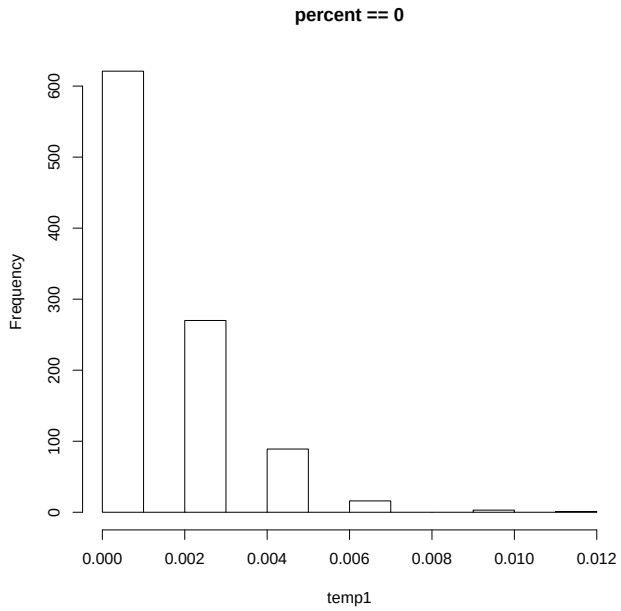
ability that A will have a higher course evaluation is 0.37.

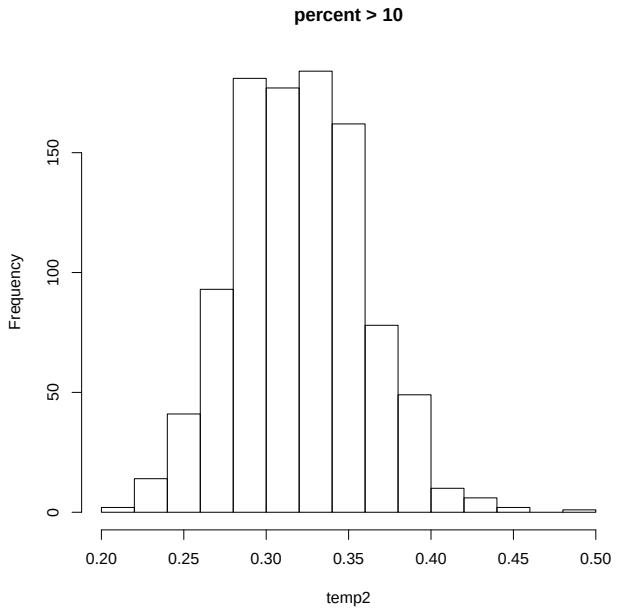
(c) G&H, Chapter 8, #4, a, b, c:

```
risky <- read.dta("risky.business/risky_behaviors.dta",T)
object <- glm(round(fupacts) ~ factor(bs_hiv) ,data = risky,family = poisson)
display(object)
n <- length(risky[,1])

n.sims <- 1000
y.tilde <- array(NA, c(n.sims,n))
X <- cbind(rep(1,n),risky$bs_hiv)

sim.1000 <- sim(object, n.sims)
for (s in 1:n.sims){

y.tilde[s,]= exp(sim.1000@coef[s,] %*% t(X))
y.tilde[s,] = rpois(n,y.tilde[s,])
}

count_percentile_0 = function(a){
return(sum(a ==0)/length(a))
}

count_percentile_10 = function(b){
return(sum(b >10)/length(b))
}

temp1<- apply(y.tilde, 1,count_percentile_0)
temp2<- apply(y.tilde, 1,count_percentile_10)

hist(temp1,main = "percent == 0")
hist(temp2,main = "percent > 10")

sum(risky$fupacts ==0)/length(risky$fupacts)
[1] 0.2926267
sum(risky$fupacts >10)/length(risky$fupacts)
[1] 0.3640553

part(b)

object <- glm(round(fupacts) ~ factor(bs_hiv) ,data = risky,
  family = quasipoisson)
display(object)
n <- length(risky[,1])

n.sims <- 1000
y.tilde <- array(NA, c(n.sims,n))
X <- cbind(rep(1,n),risky$bs_hiv)
```

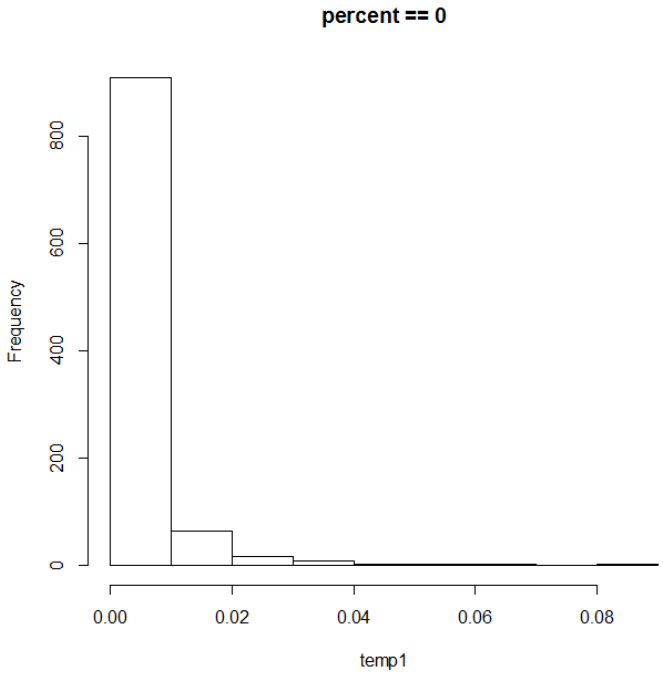
```
sim.1000 <- sim(object, n.sims)
for (s in 1:n.sims){

y.tilde[s,]= exp(sim.1000@coef[s,] %*% t(X))
y.tilde[s,] = rpois(n,y.tilde[s,])
}

count_percentile_0 = function(a){
return(sum(a ==0)/length(a))
}

count_percentile_10 = function(b){
return(sum(b >10)/length(b))
}

temp1<- apply(y.tilde, 1,count_percentile_0)
temp2<- apply(y.tilde, 1,count_percentile_10)

hist(temp1,main = "percent == 0")
hist(temp2,main = "percent > 10")


part(c) -- ethnicity wasn't available, so I used sex of the partner
  reporting the sex acts instead...
# risky <- read.dta("risky_behaviors.dta",T)
object <- glm(round(fupacts) ~ factor(bs_hiv) + sex + bupacts,
   data = risky,family = quasipoisson)

display(object)
n <- length(risky[,1])

n.sims <- 1000
y.tilde <- array(NA, c(n.sims,n))
X <- cbind(rep(1,n),risky$bs_hiv,risky$sex,risky$bupacts)

sim.1000 <- sim(object, n.sims)
for (s in 1:n.sims){

y.tilde[s,]<- exp(sim.1000@coef[s,] %*% t(X))
y.tilde[s,] <- rpois(n,y.tilde[s,])
}

count_percentile_0 = function(a){
return(sum(a ==0)/length(a))
}

count_percentile_10 = function(b){
```

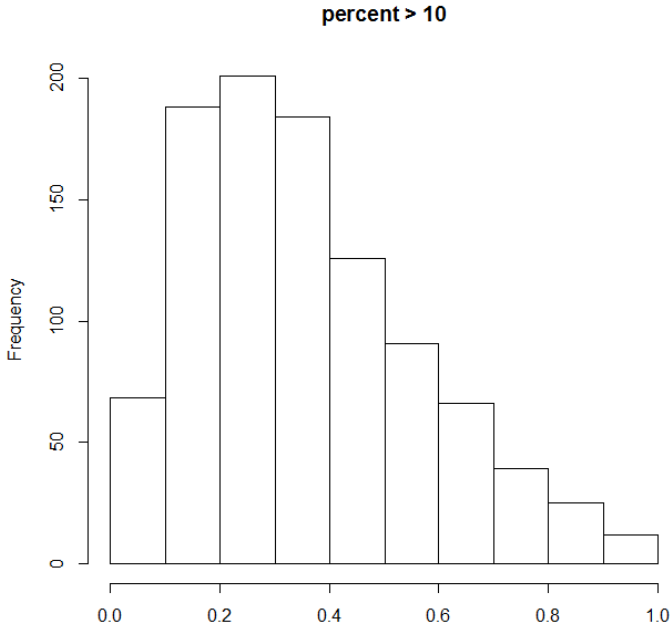
```
return(sum(b >10)/length(b))
}

temp1<- apply(y.tilde, 1,count_percentile_0)
temp2<- apply(y.tilde, 1,count_percentile_10)

hist(temp1,main = "percent == 0")
hist(temp2,main = "percent > 10")
```



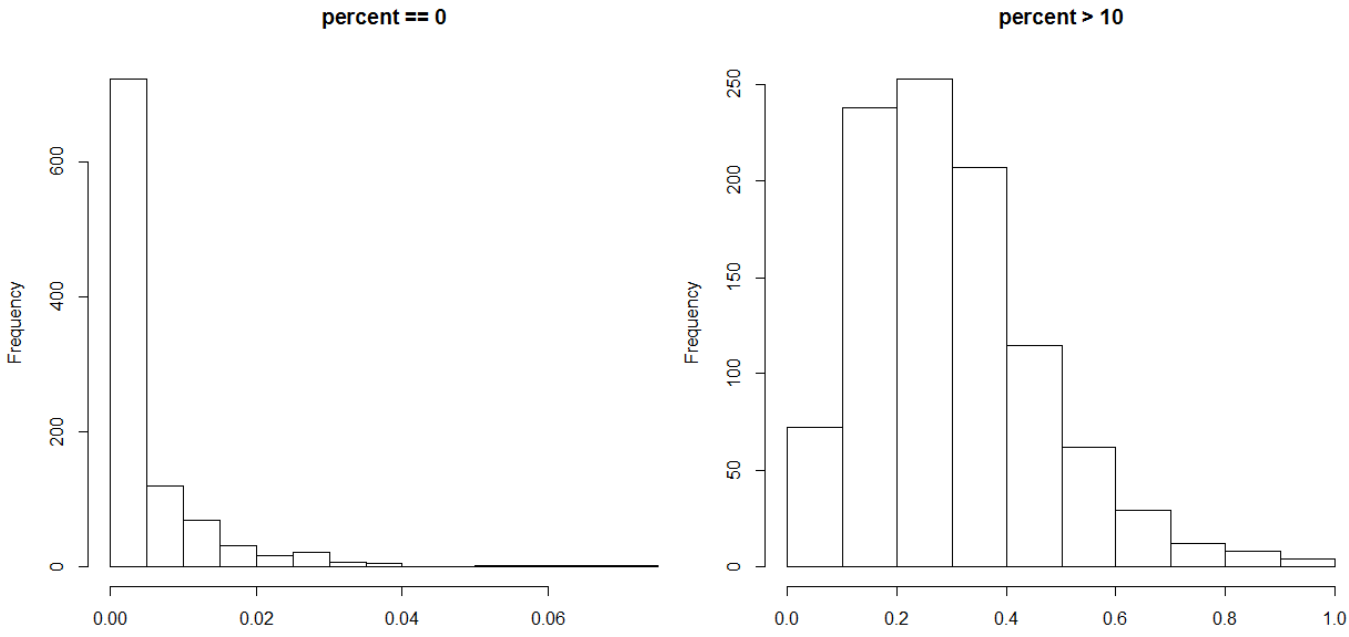


from the histograms we can see that the percentile of the case when $fupacts > 10$ falls in the simulation histogram fairly well but not the case for when $fupacts == 0$.





after fitting the overdispersed model, we can see that the case when $fupacts > 10$ improved quite a bit. however, they still seem not good enough.



after including the sex and bupacts variables as an input, from the histogram the fit seems more or less the same as part b.

2. G&H, Ch 9 & 10 exercises:

   (a) G&H, Chapter 9, #3

   (Examples on page 187-188 in the book are really helpful; see also lecture 10, week 05.)

   Dr. Smith,

   Thanks for sharing with me your proposal to study the effects of teacher quality on student scores.

   In a typical randomized experiment there are two or more groups (often a "treatment" and "control" group but it can be two treatments as well), and students would be randomly assigned to each group. In this case, if we have a way of identifying high and low quality teachers, we might consider having a "high quality" group and a "low quality" group, assign students at random to each group, and look at their test scores after som period of instruction. While this is a clean design that could tell us the effects of high vs low quality instruction, there are undoubtedly ethical—and perhaps legal—problems with deliberately assigning students to poor teachers for the purpose of seeing what happens.

   Instead we are probably forced to take whatever assignment of students to teachers has been made by the school or school district, as a matter of allocating teaching resources, and try to infer

from that what the effect of teacher quality on student outcomes is. The problem with this is that we are not in control of the reasons that students get assigned to different teachers. For example, perhaps high-achieving students have parents who demand the better teachers for their children. Then we would not know if better student outcomes were due to starting with better students, or due to better teachers.

If we are able to collect data on this and all other confounding variables, we can still make an inference about the effect of teacher quality on student outcomes from a regression equation like this

$$y_i = \beta_0 + \beta_1 T_i + \beta_2 X_{2i} + \cdots \beta_k X_{ki} + \epsilon_i$$

where y_i is student outcome (score), T_i indicates whether the student had a good or poor teacher, and X_{2i} through X_{ki} are confounding variables. The quality and correctness of our inferences, though, will be dependent on how good we are at finding all the relevant confounding variables.

Sincerely yours,

Brian Junker

(b) G&H, Chapter 9, #8, a, b, c:

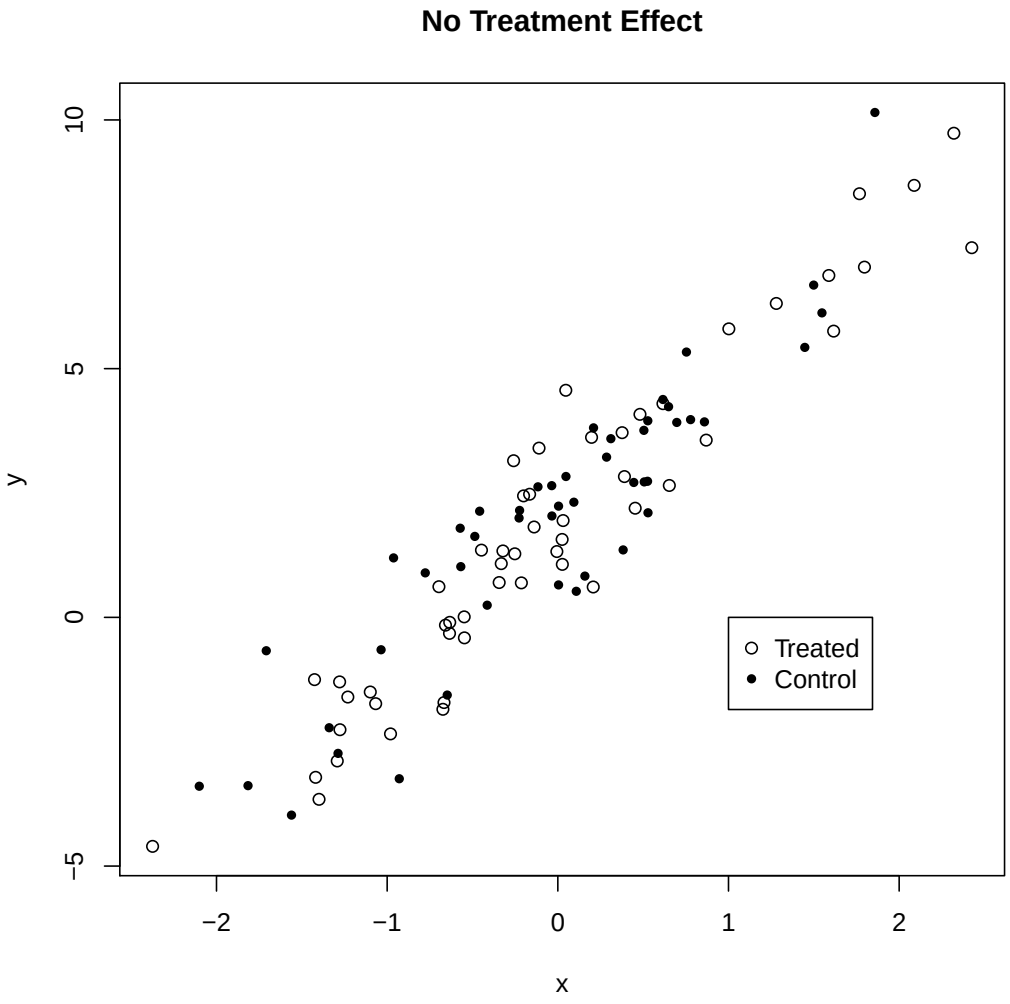
(explanations on page 169-170 are very helpful in answering this question.)

No Treatment Effect.

If you look at equation (9.1)

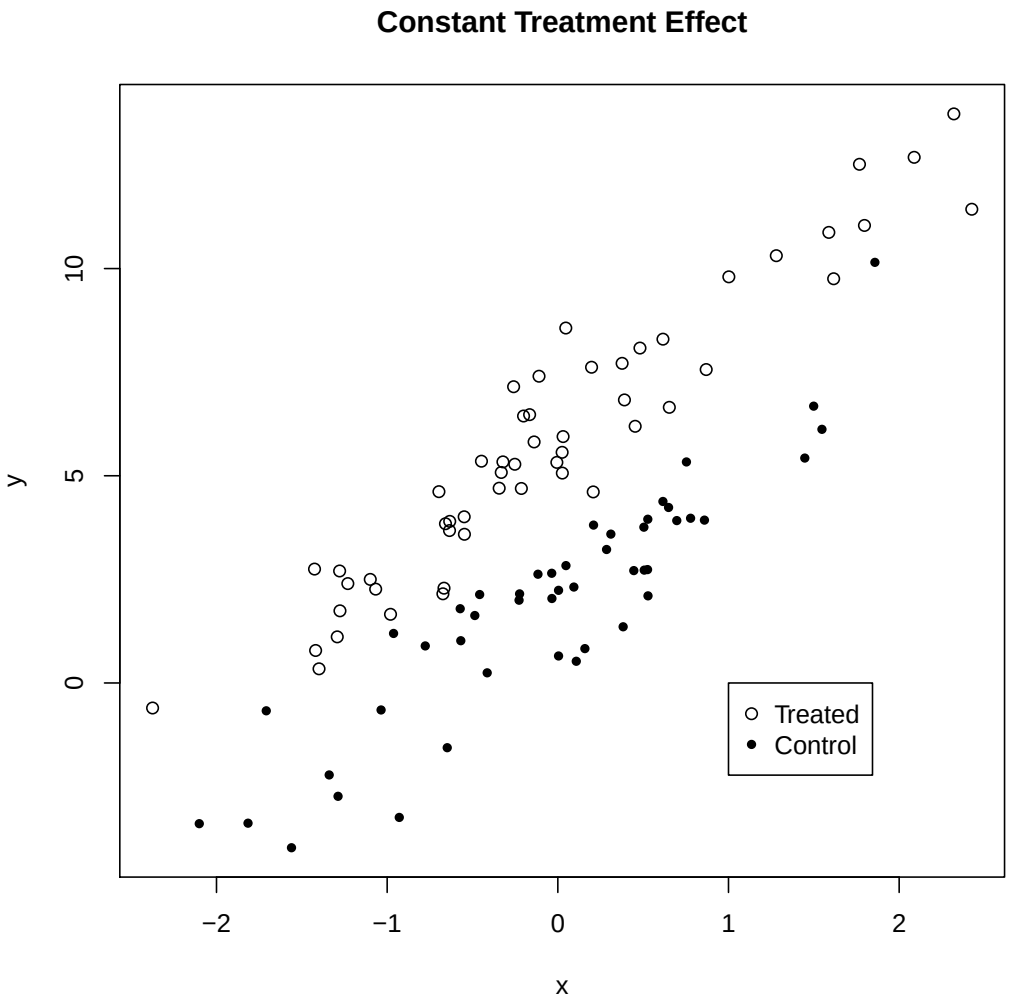
$$y_i = \beta_0 + \beta_1 T_i + \beta_2 x_i + \epsilon_i$$

no treatment effect translates to that the coefficient β_1 does not differ from 0 significantly. Now because we are plotting y vs x, the linear fit lines for the treatment and control groups have the same intercept β_0 and slope β_2:


No Treatment Effect
y
x
10
5
0
−5
−2
−1
0
1
2
Treated
Control

Constant Treatment Effect.

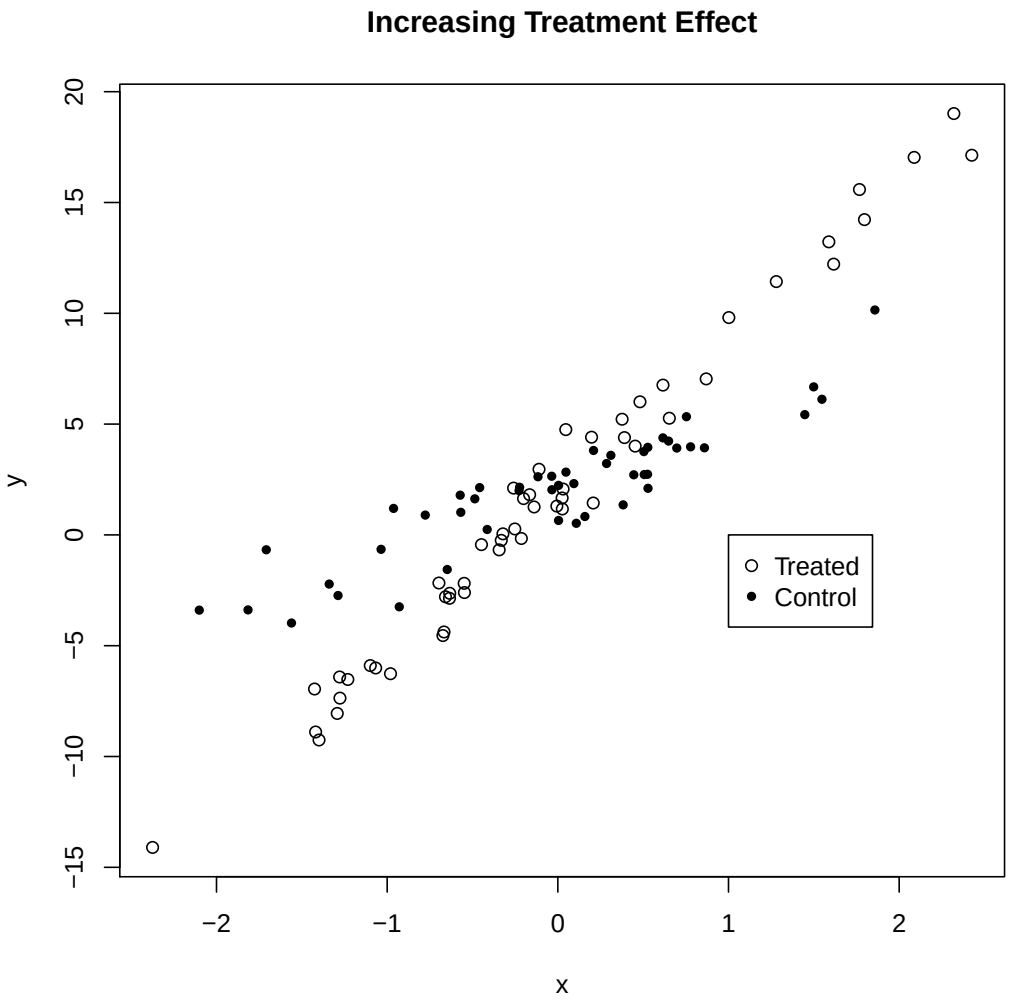
Now constant treatment effect translates to that β_1 now is significantly different from zero so the two linear fit lines differ in intercept: β_0 and $\beta_0 + \beta_1$:



Increasing Treatment Effect.
For increasing treatment effect as a function of x, equation (9.1) changes:

$$y_i = \beta_0 + \beta_1 T_i + \beta_2 x_i + \beta_3 T_i x_i + \epsilon_i$$

with $\beta_3 > 0$. So this contributes to both the intercept and the slope of x. so in this case, the two fit lines differ in both intercept and slopes:



3. *In class we considered the* `sesame` *data (available in the* `week05` *area of the class website). This is an example of an "encouragement design": subjects were randomly assigned to be encouraged (or not) to watch the Sesame Street TV show, and then they were tested to see if their letter skills had improved.*

   (a) Why do you suppose kids were randomly assigned to "encouraged" or "not encouraged", rather than "watch Sesame Street" or "don't watch Sesame street"?

   You can't actually control whether or not kids will watch Sesame Street, but you can control whether or not you encourage them to. Since we can control "encouraged", that is what was randomly assigned.

(b) In class we obtained Wald and two-stage least-squares (TSLS) IV estimates of the effect of watching Sesame street. Install package `sem` *on your computer and use the* `tsls()` *function to estimate the same effect.*

```
sesame <- read.dta("sesame.dta",T)
watched <- sesame$regular
encouraged <- sesame$encour
y <- sesame$postlet
pretest <- sesame$prelet

library(sem)

summary(est.1 <- tsls(y ~ watched, ~ encouraged))

 2SLS Estimates

Model Formula: y ~ watched

Instruments: ~encouraged

Residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -20.60   -9.59   -4.53    0.00   10.70   34.50

            Estimate Std. Error t value Pr(>|t|)
(Intercept)    20.59       3.66    5.63  5.1e-08 ***
watched         7.93       4.61    1.72    0.086 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 12.4623 on 238 degrees of freedom
```

So, the IV estimate of the effect of "watched" is 7.93 (additional points on the post-test), with an SE of 4.61. Not quite significant at the 0.05 level!

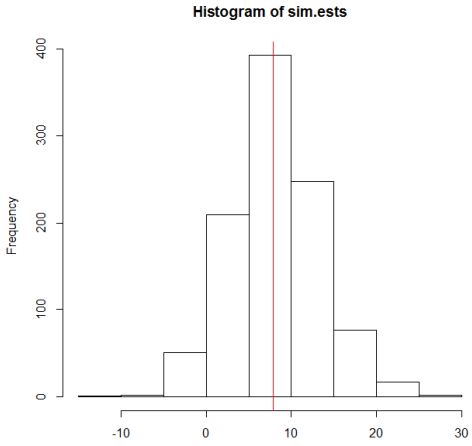
(c) Use the `sim()` *function from* `library(arm)` *to generate 1000 simulated values of the Wald estimate, from fake data sets similar to the* `sesame` *data* (HINT: to get 1000 Wald estimates, you will have to run `sim()` 2000 times.) *Use these to generate a standard error (SE) for the Wald estimate, and compare this to the SE generated by the* `tsls` *function.*

```
library(arm) # if needed!

# First we calculate the Wald estimate from the data

reg1 <- lm(y ~ encouraged)
reg2 <- lm(watched ~ encouraged)
(wald.est <- coef(reg1)[2]/coef(reg2)[2])

# get 7.933993 , agrees with tsls() as expected...
```

```
# now set up the simulation stuff

n <- 1000

sim1 <- sim(reg1, n)
sim2 <- sim(reg2, n)

sim.ests <- sim1@coef[,2]/sim2@coef[,2]
summary(sim.ests)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-10.670   4.777   7.944   8.161  11.470  27.930

# looks reasonable, if a bit right-tailed...

hist(sim.ests)
abline(v=wald.est,col="red")
sd(sim.ests)

[1] 5.24278
```

We can see from the histogram that the real-data Wald estimate is right in the middle of the simulated Wald estimates, where it should be. The simulation SE for the estimate is simply the SD of the simulated values, 5.24278.

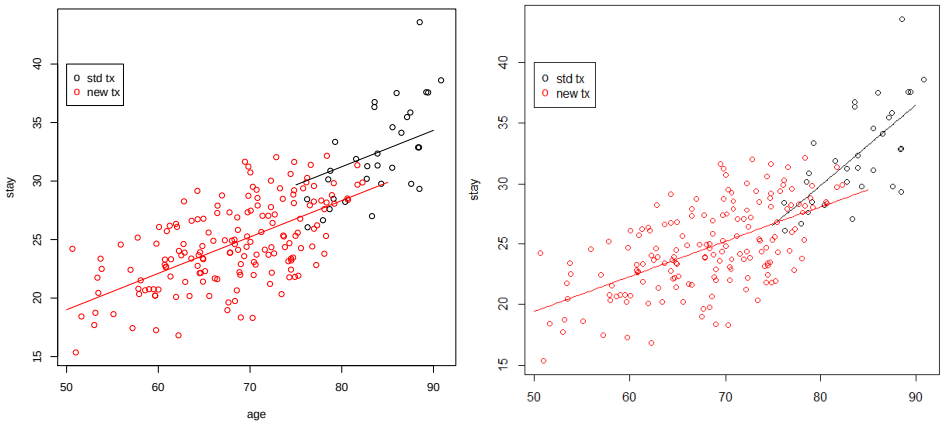
This value is somewhat larger than the one from the tsls() function, 4.61, mainly because of the right tail in the distribution of simulated Wald estimates (this is probably more realistic than the normal assumptions that underlie the tsls() function).



4. G&H, Chapter 10, #2. The folder `bypass` is available in the `hw05` area of the class website.

```
bypass <- read.table("bypass/bypass.data.txt",header=T)
str(bypass)

plot(age,stay,col=new+1)
legend(50,40,pch="o",col=1:2,legend=c("std tx","new tx"))

fit.1 <- lm(stay ~ age + new,data=bypass)

curve(coef(fit.1)[1] + x*coef(fit.1)[2],from=75,to=90,col="black",add=T)
curve(coef(fit.1)[1] + coef(fit.1)[3]+ x*coef(fit.1)[2],
  from=50,to=85,col="red",add=T)


plot(age,stay,col=new+1)
legend(50,40,pch="o",col=1:2,legend=c("std tx","new tx"))

fit.2 <- lm(stay ~ age + new + I(age*new),data=bypass)

curve(coef(fit.2)[1] + x*coef(fit.2)[2],from=75,to=90,col="black",add=T)
curve(coef(fit.2)[1] + coef(fit.2)[3]+ x*(coef(fit.2)[2]+coef(fit.2)[4]),
  from=50,to=85,col="red",add=T)
```

- *Does this seem like an appropriate setting in which to implement a regression discontinuity analysis?*
  As the figures below show, a regression discontinuity analysis with equal slopes does not seem to fit the data well, since the slope of stay on age does not seem constant in the two treatment groups.

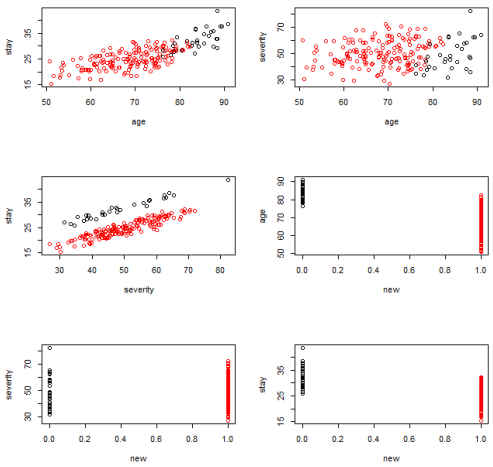


A bigger problem is that there is overlap in the age groups that do and do not receive the new treatment. If this overlap could be considered to be "random", we might still use the "parallel lines" regression discontinuity analysis to estimate the effect of the new treatment (surgery) on patients near the discontinuity age. However, the overlap is not random; it depends on the doctors' assessment of the ability of the patient

to withstand the surgery. So, there is some confounding of a treatment effect with the effect of the patient's robustness or frailty with respect to surgery (most patients above 80 are too frail for the new procedure).

- *The variables are* `age`*,* `stay`*,* `severity`*,* `new`*. Can you find any evidence using these data that the regression discontinuity design is inappropriate?*

  From the age × severity and severity × stay plots below, we can see that (a) age has very little to do with severity; and (b) treatment was not assigned with respect to severity. From the severity × stay plot, it does appear that there may be an effect for the new treatment, but it may not be well-estimated by the regression discontinuity analysis. However, we still do not know if the treatment effect is due to the new surgery or due to the fact that patients who are more frail anyway are not selected for surgery, regardless of age.



- *Estimate the treatment effect using a regression discontinuity estimate (ignoring severity). Estimate the treatment effect in any way you like, taking advantage of the information in severity. Explain the discrepancy between these estimates.*

  The simple parallel-lines RD analysis is illustrated in the first figure above. The estimated effect of the new tx (surgery) is −2.89 days, with an SE of 0.75.

```
> coef(summary(fit.1))
              Estimate Std. Error   t value     Pr(>|t|)
(Intercept)  6.3524305  2.6422622  2.404164 1.713565e-02
age          0.3109125  0.0308759 10.069746 1.665959e-19
new         -2.8919093  0.7508948 -3.851284 1.588309e-04
```

  It's difficult to do a "good" causal analysis here, because we really don't have the variable that would help us to completely understand treatment assignment, that is, the frailty or robustness of the patient to this type of surgery.

However, a simple analysis might simply include all of the other variables in the regression model

```
> fit.3 <- lm(stay ~ age + new + severity,data=bypass)
> coef(summary(fit.3))
              Estimate  Std. Error    t value      Pr(>|t|)
(Intercept)  0.7238960 0.267188667   2.709307  7.339575e-03
age          0.2011527 0.003184931  63.157646 3.031392e-132
new         -4.9034229 0.076431614 -64.154382 1.621653e-133
severity     0.3023016 0.002163839 139.706113 3.243480e-198
```

The effect of "new" now is about −4.90, about twice as large as the effect under the RD analysis (and with a smaller SD as well). The effect is larger, and more precise, because "severity" better matches patients to compare the effect with and without the new treatment (surgery), reducing the influence of variability in severity at particular ages (see again the six scatter plots above).