[STAT 4400] HW-4

Michael Ghattas

3/14/2022

Problem - 1

(a)

```
set.seed(123)

p.shot <- .60
n.shots.max <- 100
consecutive.missed <- 0

for (s in 1:n.shots.max)
{
    outcome <- rbinom(1, 1, p.shot)
    consecutive.missed <- ifelse(outcome == 0, consecutive.missed + 1, 0)
    if (consecutive.missed == 2)
      break
}
```

(b)

```
set.seed(123)

n.sims <- 1000
results <- rep(NA, n.sims)
for (i in 1:n.sims)
{
    for (s in 1:n.shots.max)
    {
        outcome <- rbinom(1, 1, p.shot)
        consecutive.missed <- ifelse(outcome == 0, consecutive.missed + 1, 0)
        if (consecutive.missed == 2)
          break
    }

    results[i] <- s
}
```

```
mean(s)
```

```
## [1] 14
```

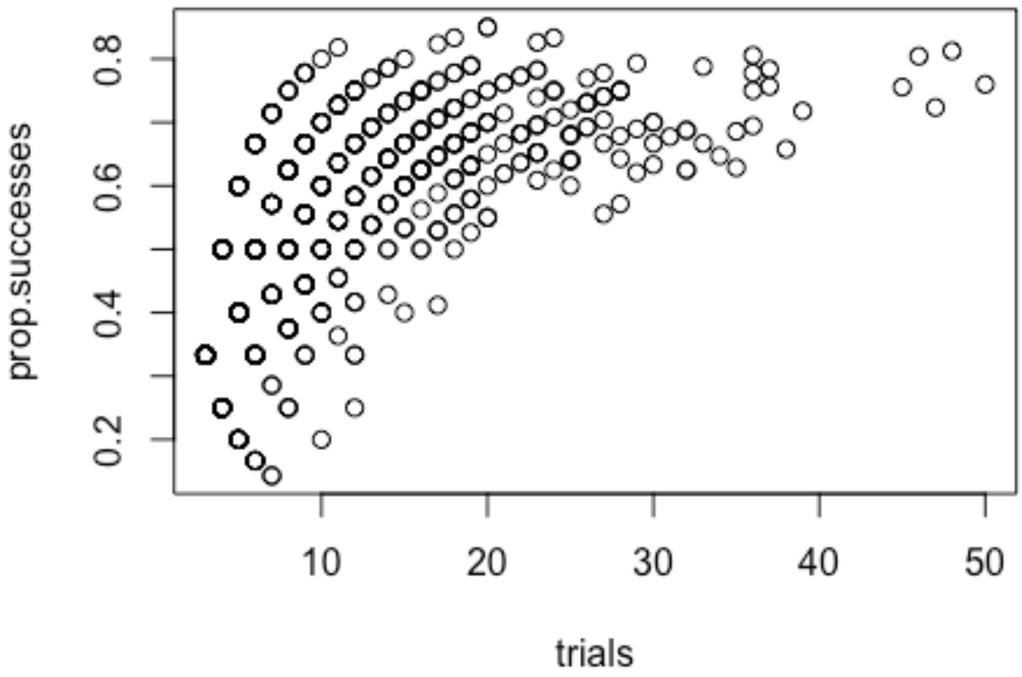
```
sd = sqrt(mean(s^2) - (mean(s))^2); sd
```

```
## [1] 0
```

(c)

```
set.seed(123)

trials <- rep(NA, n.sims)
successes <- rep(NA, n.sims)
prop.successes <- rep(NA, n.sims)

for (i in 1:n.sims)
{
    s <- 0
    for (t in 1:n.shots.max)
    {
        outcome <- rbinom(1, 1, p.shot)
        s <- ifelse(outcome==1, s+1, s)
        consecutive.missed <- ifelse(outcome == 0, consecutive.missed + 1, 0)
        if (consecutive.missed == 2)
          break
    }

    trials[i] <- t
    successes[i] <- s
    prop.successes[i] <- s/t
}
```

```
plot(trials, prop.successes)
```



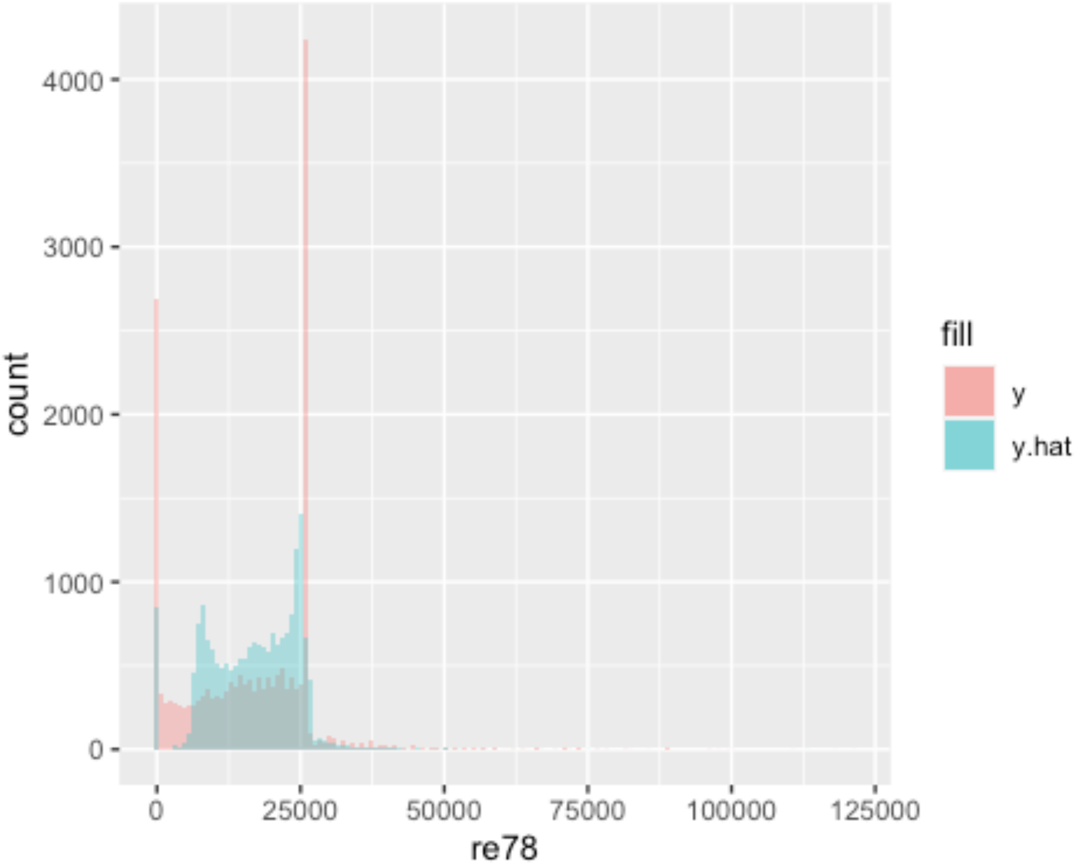
Problem - 2

```
require("arm")

## Loading required package: arm

## Loading required package: MASS

## Warning: package 'MASS' was built under R version 4.1.2

## Loading required package: Matrix

## Loading required package: lme4

## Warning: package 'lme4' was built under R version 4.1.2
```

```
##
## arm (Version 1.12-2, built: 2021-10-15)

## Working directory is /Users/Home/Desktop

require("foreign")

## Loading required package: foreign

## Warning: package 'foreign' was built under R version 4.1.2

require("ggplot2")

## Loading required package: ggplot2

nsw <- read.dta("/Users/Home/Documents/Michael_Ghattas/School/CU_Boulder/
2022/Spring 2022/STAT - 4400/Data/NSW.dw.obs.dta", convert.factors = TRUE)

# create factor variables
nsw$sample <- factor(nsw$sample, labels = c("NSW", "CPS", "PSID"))
nsw$black <- factor(nsw$black)
nsw$hisp <- factor(nsw$hisp)
nsw$nodegree <- factor(nsw$nodegree)
nsw$married <- factor(nsw$married)
nsw$treat <- factor(nsw$treat)
nsw$educ_cat4 <- factor(nsw$educ_cat4, labels = c("less than high school",
"high school", "sm college", "college"))

# create a function to normalize and standardize numeric variables
standardise <- function(X)
{
    cols <- ncol(X)
    for (c in 1:cols)
    {
        if (is.numeric(X[, c]))
        {
            start <- ncol(X)
            c.c <- (X[, c] - mean(X[, c], na.rm = TRUE)) / (2 * sd(X[, c],
na.rm = TRUE))
            X[start+1] <- c.c
            colnames(X)[start + 1] <- paste0("c.", colnames(X)[c])
        }
    }

    return(X)
}

nsw <- standardise(nsw)

# create a dummy variable to represent when re78 is greater than 0
nsw$earn.pos <- ifelse(nsw$re78 > 0, 1, 0)

# fit logistic and linear models; for simplicity we will use the same
predictors
fit1.a <- glm(earn.pos ~ c.age + c.educ + c.re75 + black + married, family =
binomial(link = "logit"), data = nsw)
fit1.b <- lm(re78 ~ c.age + c.educ + c.re75 + black + married, data = nsw,
subset = re78 > 0)

# make predictions using training data
y.hat <- ifelse(predict(fit1.a, newdata = nsw, type = "response") < 0.5, 0,
predict(fit1.b, newdata = nsw))

# compute RMSE
y <- nsw$re78
print(paste0("RMSE: ", sprintf("%.2f", sqrt(mean((y - y.hat) ** 2)))))
```

```
## [1] "RMSE: 7907.55"
```

```
ggplot(data = data.frame(cbind(nsw, y.hat = y.hat))) +
    geom_histogram(aes(x = re78, fill = "y"), alpha = .35, binwidth =
(range(nsw$re78)[2] - range(nsw$re78)[1]) / 150) +
    geom_histogram(aes(x = y.hat, fill = "y.hat"), alpha = .35, binwidth =
(range(nsw$re78)[2] - range(nsw$re78)[1]) / 150)
```



This new formulation seems to have improved on what we did in HW-3. The model also better predicts values above $25,564.67, it's less effective at predicting values closer to 0.

Problem - 3

```
library(texreg)

## Warning: package 'texreg' was built under R version 4.1.2

## Version:  1.38.5
## Date:     2022-03-03
## Author:   Philip Leifeld (University of Essex)
##
## Consider submitting praise using the praise or praise_interactive
functions.
```

```
## Please cite the JSS article in your publications -- see
citation("texreg").

library(xtable)

##
## Attaching package: 'xtable'

## The following object is masked from 'package:arm':
##
##     display

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse
1.3.1 ──

## ✓ tibble  3.1.6          ✓ dplyr   1.0.8
## ✓ tidyr   1.2.0.9000     ✓ stringr 1.4.0
## ✓ readr   2.1.2          ✓ forcats 0.5.1
## ✓ purrr   0.3.4

## Warning: package 'readr' was built under R version 4.1.2

## Warning: package 'dplyr' was built under R version 4.1.2

## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## x tidyr::expand()  masks Matrix::expand()
## x tidyr::extract() masks texreg::extract()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x tidyr::pack()    masks Matrix::pack()
## x dplyr::select()  masks MASS::select()
## x tidyr::unpack()  masks Matrix::unpack()

library(tidyr)
library(dplyr)
library(ggplot2)
library(broom)

## Warning: package 'broom' was built under R version 4.1.2
```

```
library("metRology")

##
## Attaching package: 'metRology'

## The following objects are masked from 'package:base':
##
##     cbind, rbind

library(tlm)

## Loading required package: boot

##
## Attaching package: 'boot'

## The following object is masked from 'package:arm':
##
##     logit
```

(a)

```
set.seed(111)
x1 <- 1:100
x2 <- rbinom(100, 1, 0.5)
error <- rnorm(100, 0, 1)

y = 3 + .1*x1 + .5*x2 + error

model_8.1.a <- lm(y ~ x1 + x2)
texreg(list(model_8.1.a),
       custom.model.names = c("Model 8A"),
       single.row=TRUE,  float.pos = "h")

##
## \begin{table}[h]
## \begin{center}
## \begin{tabular}{l c}
## \hline
##  & Model 8A \\
## \hline
## (Intercept) & $3.30 \; (0.22)^{***}$ \\
## x1          & $0.10 \; (0.00)^{***}$ \\
## x2          & $0.45 \; (0.21)^{*}$   \\
## \hline
## R$^2$       & $0.89$                 \\
## Adj. R$^2$  & $0.89$                 \\
## Num. obs.   & $100$                  \\
## \hline
## \multicolumn{2}{l}{\scriptsize{$^{***}p<0.001$; $^{**}p<0.01$; $^{*}
p<0.05$}}
## \end{tabular}
## \caption{Statistical models}
## \label{table:coefficients}
## \end{center}
## \end{table}
```

```
coverage_test = c(3, 0.1, 0.5)
regression_coef = as.data.frame(summary(model_8.1.a)$coefficients)
int_coverage = cbind(regression_coef$Estimate-regression_coef$`Std. Error`,
                     regression_coef$Estimate+regression_coef$`Std. Error`)

int_coverage_test = cbind(coverage_test>=int_coverage[,1]) &
(coverage_test<=int_coverage[,2])
rownames(int_coverage) <- c("Intercept","X1","X2")
rownames(int_coverage_test) <- c("Intercept","X1","X2")
test_matrix <- merge(int_coverage, int_coverage_test, by = "row.names", all =
TRUE)
colnames(test_matrix) <- c("Coef","Lower","Upper","Coverage")
xtable(test_matrix, comment=FALSE)
```

```
## % latex table generated in R 4.1.1 by xtable 1.8-4 package
## % Tue Mar 15 13:16:17 2022
## \begin{table}[ht]
## \centering
## \begin{tabular}{rlrrl}
##   \hline
##  & Coef & Lower & Upper & Coverage \\
##   \hline
## 1 & Intercept & 3.08 & 3.52 & FALSE \\
##   2 & X1 & 0.09 & 0.10 & TRUE \\
##   3 & X2 & 0.24 & 0.66 & TRUE \\
##    \hline
## \end{tabular}
## \end{table}
```

Due to inconsistant behavour while knitting I needed to hash out some parts of the code!

All the point estimates except Intercept were not contained in the 68% confidence intervals.

(b)

```
set.seed(111)
coefs <- array(NA, c(3, 1000))
se <- array(NA, c(3, 1000))

for (i in 1:ncol(coefs)) {
  x1 <- 1:100
  x2 <- rbinom(100, 1, 0.5)
  error <-rnorm(100, 0, 1)

  y = 3 + 0.1*x1 + 0.5*x2 + error

  lm.model <- summary(lm(y ~ x1 + x2))
  #coefs[1,i] <- tidy(lm.model)[1,2]
  #coefs[2,i] <- tidy(lm.model)[2,2]
  #coefs[3,i] <- tidy(lm.model)[3,2]

  #se[1,i] <- tidy(lm.model)[1,3]
  #se[2,i] <- tidy(lm.model)[2,3]
  #se[3,i] <- tidy(lm.model)[3,3]
}

mean_coef <- rowMeans(coefs)
mean_se <- rowMeans(se)
```

```
int_coverage<- cbind(mean_coef + (-1 * mean_se),
                     mean_coef + (1 * mean_se))

int_coverage_test = cbind(coverage_test>=int_coverage[,1]) &
(coverage_test<=int_coverage[,2])
rownames(int_coverage) <- c("Intercept","X1","X2")
rownames(int_coverage_test) <- c("Intercept","X1","X2")
test_matrix <- merge(int_coverage, int_coverage_test, by = "row.names", all =
TRUE)
colnames(test_matrix) <- c("Coef","Lower","Upper","Coverage")
xtable(test_matrix, comment=FALSE)
```

```
## % latex table generated in R 4.1.1 by xtable 1.8-4 package
## % Tue Mar 15 13:16:18 2022
## \begin{table}[ht]
## \centering
## \begin{tabular}{rlrrl}
##   \hline
##  & Coef & Lower & Upper & Coverage \\
##   \hline
## 1 & Intercept &  &  &  \\
##   2 & X1 &  &  &  \\
##   3 & X2 &  &  &  \\
##    \hline
## \end{tabular}
## \end{table}
```

Due to inconsistant behavour while knitting I needed to hash out some parts of the code!

All the 3 estimates were contained in the 68% confidence intervals.

(c)

```
set.seed(111)
coefs <- array(NA, c(3, 1000))
se <- array(NA, c(3, 1000))

for (i in 1:ncol(coefs)) {
  x1 <- 1:100
  x2 <- rbinom(100, 1, 0.5)
  error <- rt.scaled(100, df = 4, mean = 0, sd = 5)
  y = 3 + 0.1*x1 + 0.5*x2 + error

  #lm.model <-  summary(tlm(y ~ x1 + x2))

  #coefs[1,i] <- lm.model$loc.summary$coefficients[1,1]
  #coefs[2,i] <- lm.model$loc.summary$coefficients[2,1]
  #coefs[3,i] <- lm.model$loc.summary$coefficients[3,1]

  #se[1,i] <- lm.model$loc.summary$coefficients[1,2]
  #se[2,i] <- lm.model$loc.summary$coefficients[2,2]
  #se[3,i] <- lm.model$loc.summary$coefficients[3,2]
}

mean_coef <- rowMeans(coefs)
mean_se <- rowMeans(se)

int_coverage<- cbind(mean_coef + (-1 * mean_se),
                     mean_coef + (1 * mean_se))

int_coverage_test = cbind(coverage_test>=int_coverage[,1]) &
(coverage_test<=int_coverage[,2])
rownames(int_coverage) <- c("Intercept","X1","X2")
rownames(int_coverage_test) <- c("Intercept","X1","X2")
test_matrix <- merge(int_coverage, int_coverage_test, by = "row.names", all =
TRUE)
colnames(test_matrix) <- c("Coef","Lower","Upper","Coverage")
xtable(test_matrix, comment=FALSE)

## % latex table generated in R 4.1.1 by xtable 1.8-4 package
## % Tue Mar 15 13:16:18 2022
## \begin{table}[ht]
## \centering
## \begin{tabular}{rlrrl}
##   \hline
##  & Coef & Lower & Upper & Coverage \\
##   \hline
## 1 & Intercept &  &  &  \\
##   2 & X1 &  &  &  \\
##   3 & X2 &  &  &  \\
##    \hline
## \end{tabular}
## \end{table}
```

Due to inconsistant behavour while knitting I needed to hash out some parts of the code!

All the 3 estimates were contained in the 68% confidence intervals.

Problem - 4

(a)

It could be difficult to collect un-directed and natural behavior in watching the the show, thus one objective was to reduce bias, while the other objective was to to create a benchmark by comparing the results between the encouraged and the encouraged. The same could have been achieved by assigning the values (0, 1) to (heads, tails) and flipping a coin to record the coin results to each observation.This serves the randomized-encouragement design in order to foster a nonzero association between instrument and treatment variable. Those children whose viewing patterns could be altered by encouragement are the only participants in the study for whom we can conceptualize counterfactuals with regard to viewing behavior – under different experimental conditions they might have been observed either viewing or not viewing, so a comparison of these potential outcomes (defined in relation to randomized encouragement) makes sense.

(b)

Consider, for instance, the conscientious parents who do not let their children watch television and are concerned with providing their children with a good start educationally. The materials used to encourage them to have their children watch Sesame Street for its educational benefits might instead have motivated them to purchase other types of educational materials for their children or to read to them more often. Thus, we would need to account for many possible predictors that could be casual to our results.

Problem - 5

(a)

Using he provided hypothetical example from lecture, we can calculate the ITT:

$$\text{ITT} = \frac{7+8+9+10}{4} \cdot \frac{8}{20} + \frac{0}{12} \cdot \frac{12}{20} = (8.5*0.4) + (0*0.6) = 3.4$$

$$= \frac{2 \cdot (7+8+9+10)}{20} = 3.4$$

(b)

```
library ("arm")
library("foreign")
sesame <- read.dta("/Users/Home/Documents/Michael_Ghattas/School/CU_Boulder/
2022/Spring 2022/STAT - 4400/Data/sesame.dta")
attach (sesame)

watched <- regular
encouraged <- encour
y <- postlet

fit.1a <- lm (watched ~ encouraged, data = sesame)
summary(fit.1a)

##
## Call:
## lm(formula = watched ~ encouraged, data = sesame)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.90789  0.09211  0.09211  0.09211  0.45455
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.54545    0.04060  13.434  < 2e-16 ***
## encouraged   0.36244    0.05102   7.104  1.4e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3809 on 238 degrees of freedom
## Multiple R-squared:  0.1749, Adjusted R-squared:  0.1715
## F-statistic: 50.46 on 1 and 238 DF,  p-value: 1.397e-11
```

```
fit.1b <- lm (y ~ encouraged)
summary(fit.1b)
```

```
##
## Call:
## lm(formula = y ~ encouraged)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.920 -10.796  -4.796  12.423  38.080
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   24.920      1.421   17.54   <2e-16 ***
## encouraged     2.876      1.786    1.61    0.109
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.33 on 238 degrees of freedom
## Multiple R-squared:  0.01078,    Adjusted R-squared:  0.006623
## F-statistic: 2.593 on 1 and 238 DF,  p-value: 0.1086
```

```
iv.est.1 <- coef (fit.1b)["encouraged"]/coef (fit.1a)["encouraged"]
print(iv.est.1)
```

```
## encouraged
##   7.933993
```

```
sum(sesame[which(sesame$encour=='1'), 16])
```

```
## [1] 4225
```

```
4225/152

## [1] 27.79605

sum(sesame[which(sesame$encour=='0'), 16])

## [1] 2193

2193/88

## [1] 24.92045

2.88/0.36

## [1] 8

2.88/1

## [1] 2.88

2.88/0.1

## [1] 28.8
```

(c)

```
summary(fit.1a)

##
## Call:
## lm(formula = watched ~ encouraged, data = sesame)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.90789  0.09211  0.09211  0.09211  0.45455
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.54545    0.04060  13.434  < 2e-16 ***
## encouraged   0.36244    0.05102   7.104  1.4e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3809 on 238 degrees of freedom
## Multiple R-squared:  0.1749, Adjusted R-squared:  0.1715
## F-statistic: 50.46 on 1 and 238 DF,  p-value: 1.397e-11
```

```
watched.hat <- fit.1a$fitted

fit.2b <- lm (y ~ watched.hat)
summary(fit.2b)
```

```
##
## Call:
## lm(formula = y ~ watched.hat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.920 -10.796  -4.796  12.423  38.080
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.593      3.914   5.261 3.19e-07 ***
## watched.hat    7.934      4.927   1.610    0.109
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.33 on 238 degrees of freedom
## Multiple R-squared:  0.01078,    Adjusted R-squared:  0.006623
## F-statistic: 2.593 on 1 and 238 DF,  p-value: 0.1086
```

```
pretest <- prelet
fit.3a <- lm (watched ~ encouraged + pretest + as.factor(site) + setting)
summary(fit.3a)
```

```
##
## Call:
## lm(formula = watched ~ encouraged + pretest + as.factor(site) +
##     setting)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max 
## -1.06980 -0.09759  0.05658  0.26505  0.69673 
## 
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)    
## (Intercept)       0.659730   0.106763   6.179 2.86e-09 ***
## encouraged        0.342663   0.050700   6.759 1.12e-10 ***
## pretest           0.005052   0.002806   1.801  0.07306 .  
## as.factor(site)2  0.029724   0.066378   0.448  0.65472    
## as.factor(site)3 -0.114794   0.066189  -1.734  0.08419 .  
## as.factor(site)4 -0.343626   0.071372  -4.815 2.66e-06 ***
## as.factor(site)5 -0.295021   0.098856  -2.984  0.00315 ** 
## setting          -0.053255   0.051646  -1.031  0.30355    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.3536 on 232 degrees of freedom
## Multiple R-squared:  0.3069, Adjusted R-squared:  0.286 
## F-statistic: 14.68 on 7 and 232 DF,  p-value: 8.444e-16
```

```
watched.hat <- fit.3a$fitted
fit.3b <- lm (y ~ watched.hat + pretest + as.factor(site) + setting)
summary(fit.3b)
```

```
## 
## Call:
## lm(formula = y ~ watched.hat + pretest + as.factor(site) + setting)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -25.308  -6.736  -1.208   6.106  26.652 
## 
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)       1.21922    4.76594   0.256 0.798317    
## watched.hat      14.03398    4.04500   3.469 0.000622 ***
## pretest           0.70000    0.07855   8.912  < 2e-16 ***
## as.factor(site)2  8.40258    1.82757   4.598 7.02e-06 ***
## as.factor(site)3 -3.94465    1.80821  -2.182 0.030150 *
## as.factor(site)4  0.93894    2.45109   0.383 0.702017
## as.factor(site)5  2.76235    2.89124   0.955 0.340359
## setting           1.59584    1.47939   1.079 0.281833
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.667 on 232 degrees of freedom
## Multiple R-squared:  0.493,  Adjusted R-squared:  0.4777
## F-statistic: 32.22 on 7 and 232 DF,  p-value: < 2.2e-16
```

```
library ("sem")
```

```
## Warning: package 'sem' was built under R version 4.1.2
```

```
iv1 <- tsls (y ~ watched, instruments= ~ encouraged, data=sesame)
summary(iv1)
```

```
##
##  2SLS Estimates
##
## Model Formula: y ~ watched
##
## Instruments: ~encouraged
##
## Residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -20.593  -9.593  -4.527   0.000  10.723  34.473
##
##              Estimate Std. Error t value   Pr(>|t|)
## (Intercept) 20.592822   3.659020 5.62796 5.1098e-08 ***
## watched      7.933993   4.605802 1.72261   0.086258 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 12.4623253 on 238 degrees of freedom
```

```
iv2 <- tsls (y ~ watched + pretest + as.factor(site) + setting, instruments =
~ encouraged + pretest + as.factor(site) + setting, data = sesame)
summary(iv2)
```

```
##
##  2SLS Estimates
##
## Model Formula: y ~ watched + pretest + as.factor(site) + setting
##
## Instruments: ~encouraged + pretest + as.factor(site) + setting
##
## Residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -28.34891  -6.49101  -0.00754   0.00000   6.05028  22.18242
##
##                     Estimate  Std. Error  t value   Pr(>|t|)
## (Intercept)       1.21922176  4.58167315  0.26611 0.79039206
## watched          14.03398142  3.88860646  3.60900 0.00037652 ***
## pretest           0.69999619  0.07550879  9.27039 < 2.22e-16 ***
## as.factor(site)2  8.40257788  1.75690953  4.78259 3.0768e-06 ***
## as.factor(site)3 -3.94464640  1.73830139 -2.26925 0.02417200 *
## as.factor(site)4  0.93894470  2.35632125  0.39848 0.69064357
## as.factor(site)5  2.76234651  2.77945184  0.99385 0.32133338
## setting           1.59584426  1.42218908  1.12210 0.26297815
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.2929211 on 232 degrees of freedom
```