Homework 06

Simulation

Jinfei XUE

Oct 31, 2018

Discrete probability simulation:

suppose that a basketball player has a 60% chance of making a shot, and he keeps taking shots until he misses two in a row. Also assume his shots are independent (so that each shot has 60% probability of success, no matter what happened before).

1. Write an R function to simulate this process.

```
sim_shots<-function(){
    ee <- TRUE
    shots<- rbinom(1, 1, 0.6)
    i=1
    while( ee ) {
         i = i + 1
        ashot<- rbinom(1, 1, 0.6)
        if(shots[i-1]==0 && ashot==0){
            ee=FALSE
        }
        shots <- c(shots,ashot)
    }
    return(shots)
}
```

2. Put the R function in a loop to simulate the process 1000 times. Use the simulation to estimate the mean, standard deviation, and distribution of the total number of shots that the player will take.

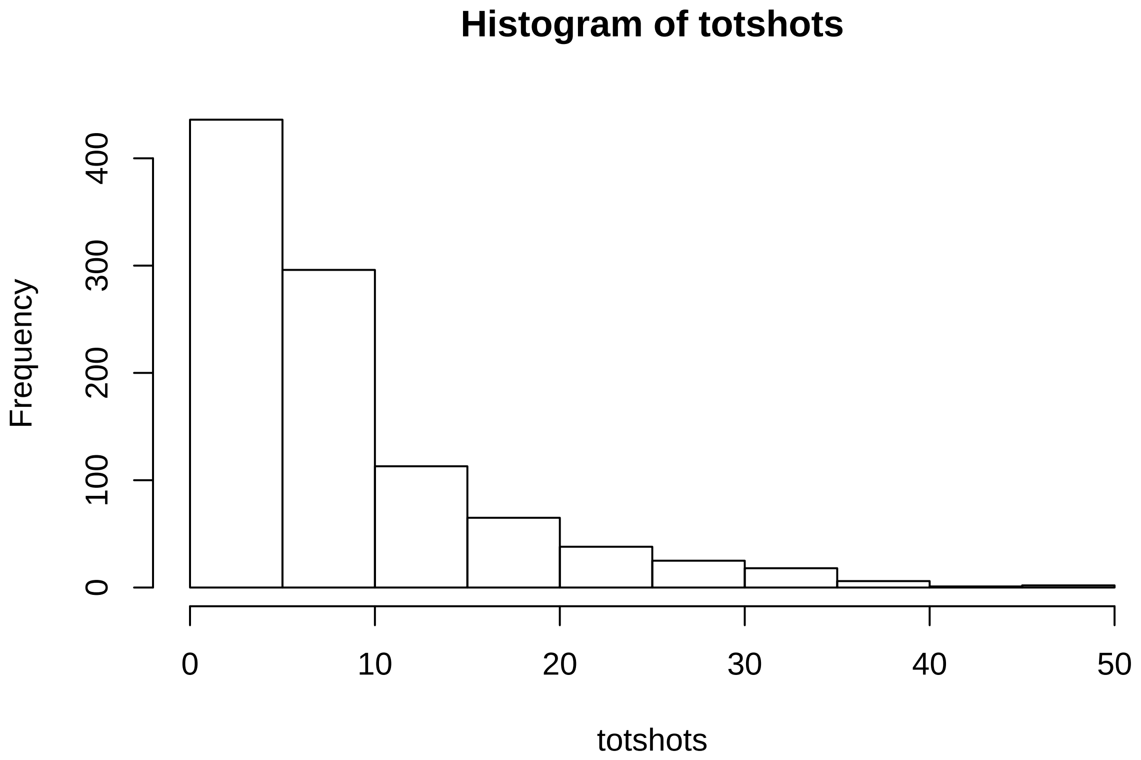
```
n_samp   <- 1000
totshots <- rep(NA,n_samp)
propshots<- rep(NA,n_samp)
for(i in 1:n_samp){
    simshots    <- sim_shots()
    totshots[i] <- length(simshots)
    propshots[i]<- mean(simshots)
}
mean(totshots)
```

```
## [1] 8.776
```

```
sd(totshots)
```

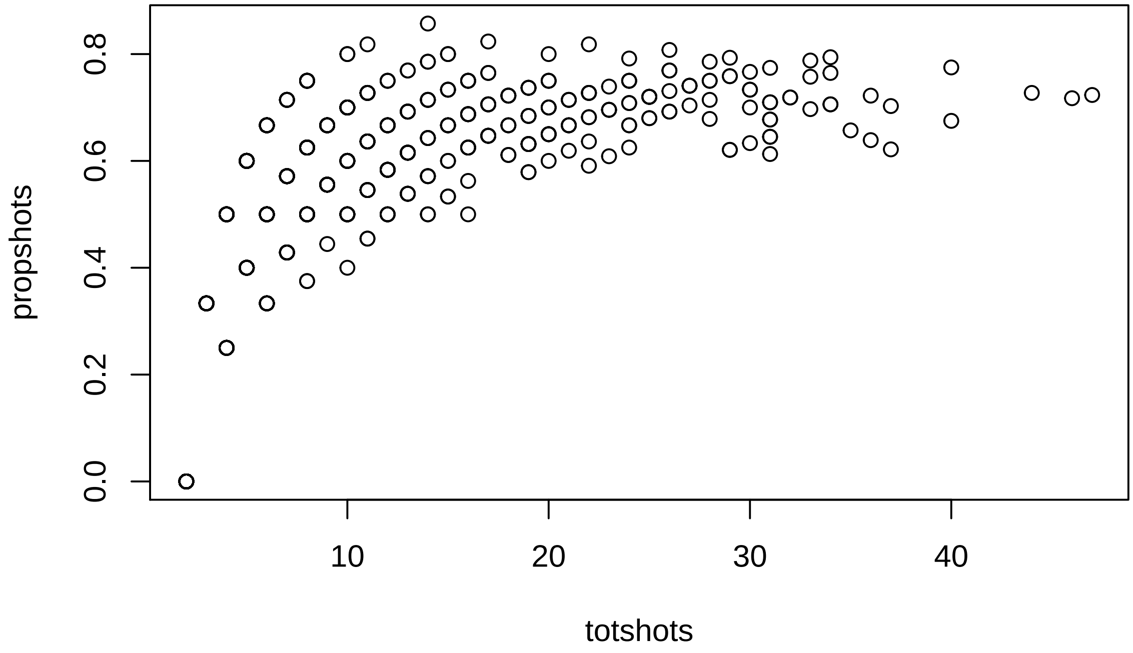
```
## [1] 7.651168
```

```
hist(totshots)
```



3. Using your simulations, make a scatterplot of the number of shots the player will take and the proportion of shots that are successes.

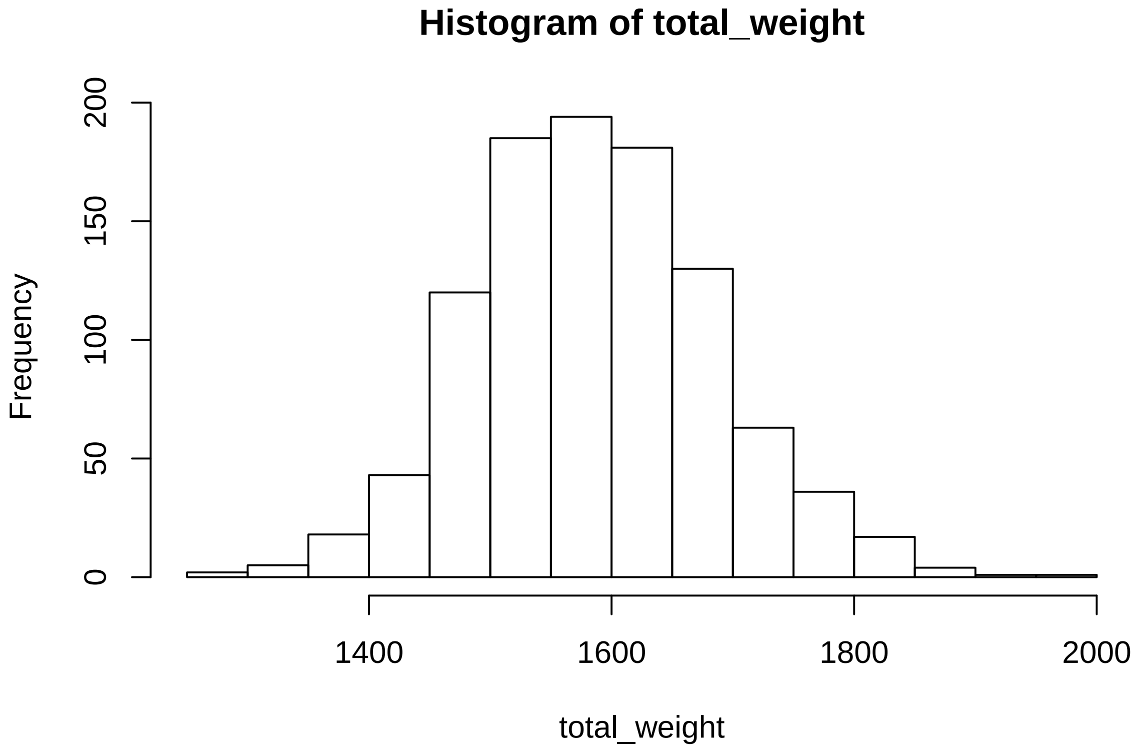
```
plot(totshots,propshots)
```



Continuous probability simulation:

the logarithms of weights (in pounds) of men in the United States are approximately normally distributed with mean 5.13 and standard deviation 0.17; women with mean 4.96 and standard deviation 0.20. Suppose 10 adults selected at random step on an elevator with a capacity of 1750 pounds. What is the probability that the elevator cable breaks?

```
n_sim<-1000
total_weight<- rep(NA,n_sim)
for(i in 1:n_sim){
  #Assume the ratio between men and woman in US is about 0.49:0.51
  male<- rbinom(10,1,0.49)
  male_weight<-rnorm(sum(male),5.13,0.17)
  nfem <- 10-sum(male);
  if(nfem>0){
    female_weight<-rnorm(nfem,4.96,0.2)}
  else {
    female_weight<-0
    }
  total_weight[i]<-sum(c(exp(male_weight),exp(female_weight)))
  }
hist(total_weight)
```



```
# the probability that the elevator cable breaks
mean(total_weight>1750)
```

```
## [1] 0.059
```

Predictive simulation for linear regression:

take one of the models from previous excessive that predicts course evaluations from beauty and other input variables. You will do some simulations.

```
prof <- read.csv("http://www.stat.columbia.edu/~gelman/arm/examples/beauty/ProfEvaltnsBeautyPublic.csv")

# convert into factors
prof$profnumber <- as.factor(prof$profnumber)
prof$female <- as.factor(prof$female)

# convert dummy `class*` variables into a factor
dummies <- prof[, 18:47]
prof$class <- factor(apply(dummies, FUN=function(r) r %*% 1:30, MARGIN=1))

# remove dummy variables
prof <- prof[-c(18:47)]

# normalise and centre professor evaluation (all other predictors are binary)
prof$c.profevaluation <- prof$profevaluation - mean(prof$profevaluation) / (2 * sd(prof$profevaluation))
```

1. Instructor A is a 50-year-old woman who is a native English speaker and has a beauty score of 1. Instructor B is a 60-year-old man who is a native English speaker and has a beauty score of - .5. Simulate 1000 random draws of the course evaluation rating of these two instructors. In your simulation, account for the uncertainty in the regression parameters (that is, use the `sim()` function) as well as the predictive uncertainty.

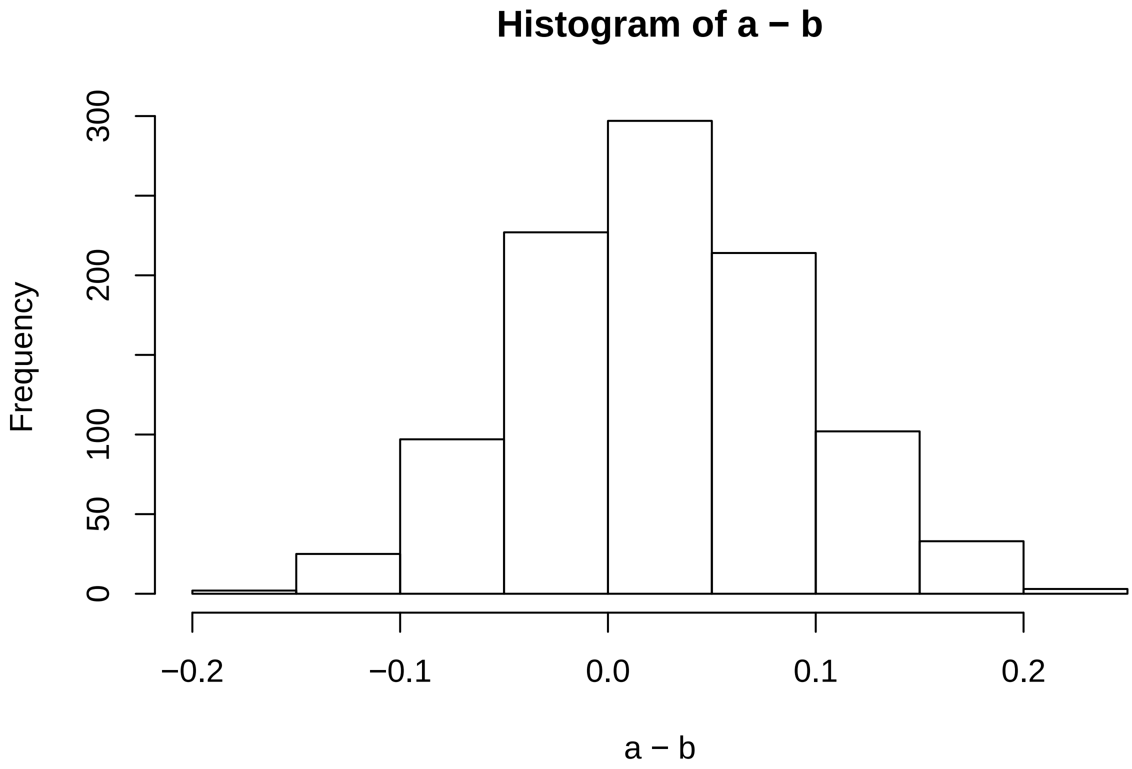
```
r_prof <- lm(courseevaluation ~ btystdave + age + female + nonenglish, data=prof)
simfit <- sim(r_prof, n.sims = 1000)
coef <- simfit@coef

# Instructor A:  btystdave=1, age=50, female1=1, nonenglish=0,
a <- simfit@coef[,1]*1 + simfit@coef[,2]*1 + simfit@coef[,3]*50 +
  simfit@coef[,4]*1 + simfit@coef[,5]*0

# Instructor B: btystdave=-0.5, age=60, female1=0, nonenglish=0
b <- simfit@coef[,1]*1 + simfit@coef[,2]*(-0.5) + simfit@coef[,3]*60 +
  simfit@coef[,4]*0 + simfit@coef[,5]*0
```

2. Make a histogram of the difference between the course evaluations for A and B. What is the probability that A will have a higher evaluation?

```
# Histogram of the difference between the course evaluations for A and B
hist(a-b)
```



```
# The probability that A will have a higher evaluation
mean(a>b)
```

```
## [1] 0.649
```

How many simulation draws are needed:

take the model from previous exercise that predicts course evaluations from beauty and other input variables. Use display() to summarize the model fit. Focus on the estimate and standard error for the coefficient of beauty.

```
beauty <- read.csv("http://www.stat.columbia.edu/~gelman/arm/examples/beauty/ProfEvaltnsBeautyPublic.cs
```

1. Use sim() with n.sims = 10000. Compute the mean and standard deviations of the 1000 simulations of the coefficient of beauty, and check that these are close to the output from display.

```
simfit2 <- sim(r_prof, n.sims = 10000)
mean(simfit2@coef[,2])
```

```
## [1] 0.1414542
```

```
sd(simfit2@coef[,2])
```

```
## [1] 0.03264143
```

```
display(r_prof)
```

```
## lm(formula = courseevaluation ~ btystdave + age + female + nonenglish,
##     data = prof)
##             coef.est coef.se
## (Intercept)  4.24     0.14
## btystdave    0.14     0.03
## age          0.00     0.00
## female1     -0.21     0.05
## nonenglish  -0.33     0.10
## ---
## n = 463, k = 5
## residual sd = 0.53, R-Squared = 0.09
```

Yes, they are close to the output from disaplay.

2. Repeat with n.sims = 1000, n.sims = 100, and n.sims = 10. Do each of these a few times in order to get a sense of the simulation variability.

```
# n.sim=1000
n <- 10
mean_1000 <- rep(NA,n)
sd_1000 <- rep(NA,n)
for(i in 1:n){
  mean_1000[i] <- mean(simfit@coef[,2])
  sd_1000[i] <- sd(simfit@coef[,2])
}
sd(mean_1000)
```

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

```
sd(sd_1000)
```

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

```
# n.sim=100
simfit3 <- sim(r_prof, n.sims=100)
n <- 10
mean_100 <- rep(NA,n)
sd_100 <- rep(NA,n)
for(i in 1:n){
  mean_100[i] <- mean(simfit3@coef[,2])
  sd_100[i] <- sd(simfit3@coef[,2])
}
sd(mean_100)
```

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

```
sd(sd_100)
```

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

```
# n.sim=10
simfit4 <- sim(r_prof, n.sims=10)
n <- 10
mean_10 <- rep(NA,n)
sd_10 <- rep(NA,n)
for(i in 1:n){
  mean_10[i] <- mean(simfit4@coef[,2])
  sd_10[i] <- sd(simfit4@coef[,2])
}
sd(mean_10)
```

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

```r
sd(sd_10)
```

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

```r
# Display
mean(simfit@coef[,2])
```

```
## [1] 0.1412211
```

```r
sd(simfit@coef[,2])
```

```
## [1] 0.0341474
```

```r
mean(simfit3@coef[,2])
```

```
## [1] 0.137709
```

```r
sd(simfit3@coef[,2])
```

```
## [1] 0.03180932
```

```r
mean(simfit4@coef[,2])
```

```
## [1] 0.1254016
```

```r
sd(simfit4@coef[,2])
```

```
## [1] 0.02895868
```

```r
display(r_prof)
```

```
## lm(formula = courseevaluation ~ btystdave + age + female + nonenglish,
##     data = prof)
##             coef.est coef.se
## (Intercept)  4.24     0.14
## btystdave    0.14     0.03
## age          0.00     0.00
## female1     -0.21     0.05
## nonenglish  -0.33     0.10
## ---
## n = 463, k = 5
## residual sd = 0.53, R-Squared = 0.09
```

We can see that the simulation variability is almost zero

3. How many simulations were needed to give a good approximation to the mean and standard error for the coefficient of beauty?

From the previous simulations with different n.sims, 1000 simulations are needed to give a good approximation because both mean and sd of simulations are very close to the outputs from display.

Predictive simulation for linear regression:

using data of interest to you, fit a linear regression model. Use the output from this model to simulate a predictive distribution for observations with a particular combination of levels of all the predictors in the regression.

```r
# Study of teenage gambling in Britain
data(teengamb)
#?teengamb
```

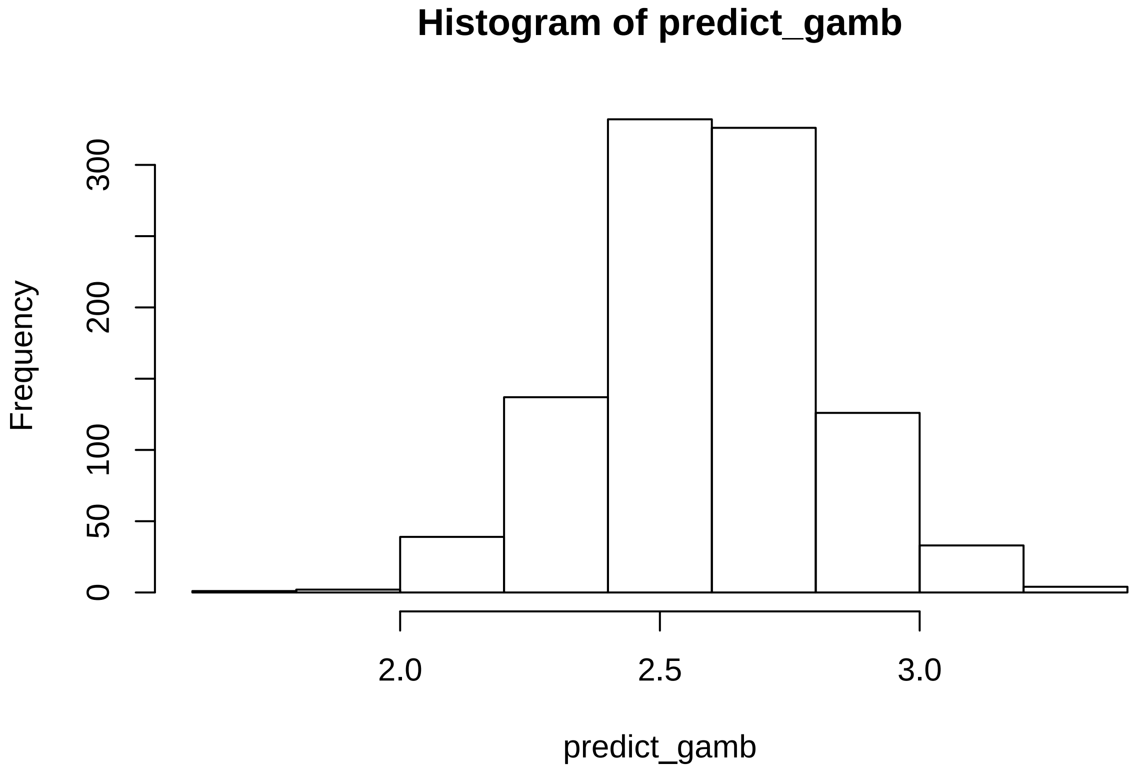
```r
# Linear model
gamble_log<-log(teengamb$gamble+1)
status_zscore<-(teengamb$status-mean(teengamb$status))/sd(teengamb$status)
r_gamb<-lm(gamble_log~sex+status_zscore+income+verbal, data = teengamb)
#summary(r_gamb)

# Simulation
simfit_gamb <- sim(r_gamb,n.sims=1000)

# prediction: male(sex=0), status_zscore=0, income=5, verbal=6
head(simfit_gamb@coef)
```

```
##      (Intercept)        sex status_zscore    income      verbal
## [1,]    3.856946 -1.1332845     0.6524050 0.2013088 -0.39439520
## [2,]    1.728204 -0.6990111     0.5892128 0.1809120 -0.06133728
## [3,]    3.512086 -1.4050636     0.4642746 0.1856127 -0.28553809
## [4,]    5.084791 -1.2877661     0.5879231 0.1696807 -0.53042419
## [5,]    3.052168 -0.9782535     0.5236641 0.2316617 -0.25706125
## [6,]    3.772145 -1.2183877     0.4980737 0.1613667 -0.30791088
```

```r
predict_gamb <- simfit_gamb@coef[,1]+simfit_gamb@coef[,2]*0+
  simfit_gamb@coef[,3]*0+simfit_gamb@coef[,4]*5++simfit_gamb@coef[,5]*6
hist(predict_gamb)
```



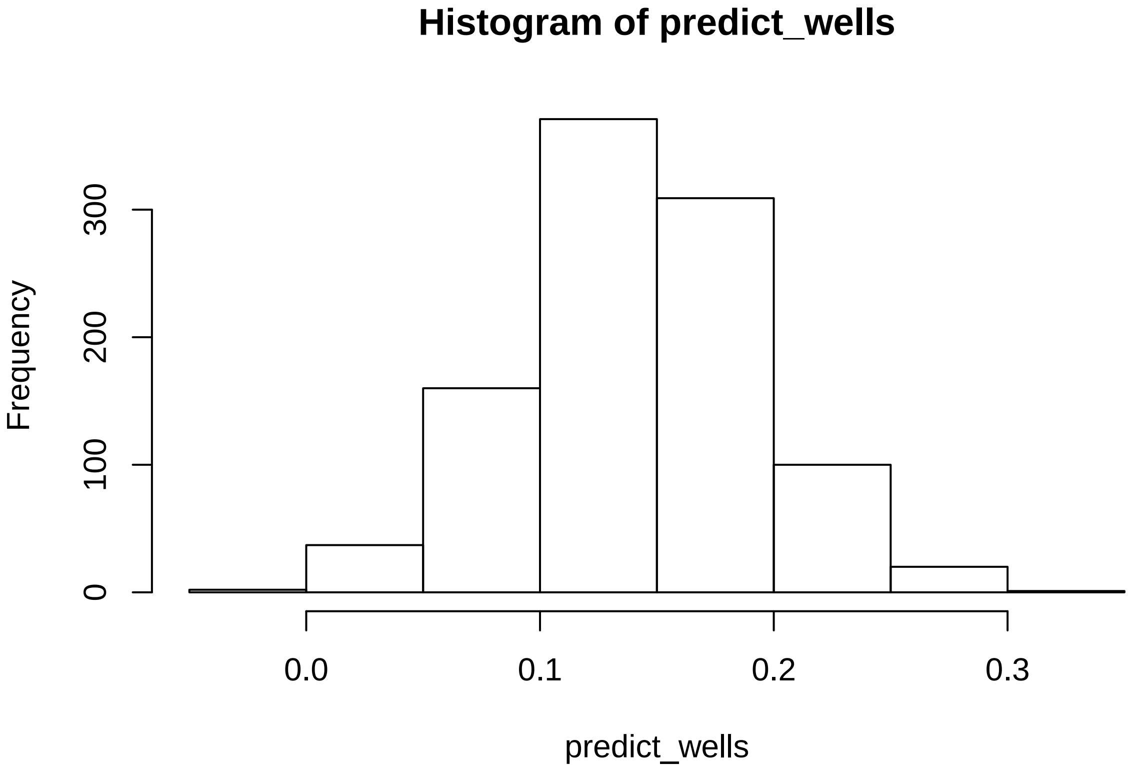
Repeat the previous exercise using a logistic regression example.

```
# the well-switching data
wells = read.table("http://www.stat.columbia.edu/~gelman/arm/examples/arsenic/wells.dat")
wells_dt <- data.table(wells)

# Logistic model
r_wells <- glm(switch ~ log(dist), family=binomial(link="logit"), data = wells_dt)

#simulation
simfit_wells <- sim(r_wells,n.sim=1000)

#prediction: log(disc)=log(80)
predict_wells <- simfit_wells@coef[,1]+simfit_wells@coef[,2]*log(80)
hist(predict_wells)
```



Repeat the previous exercise using a Poisson regression example.

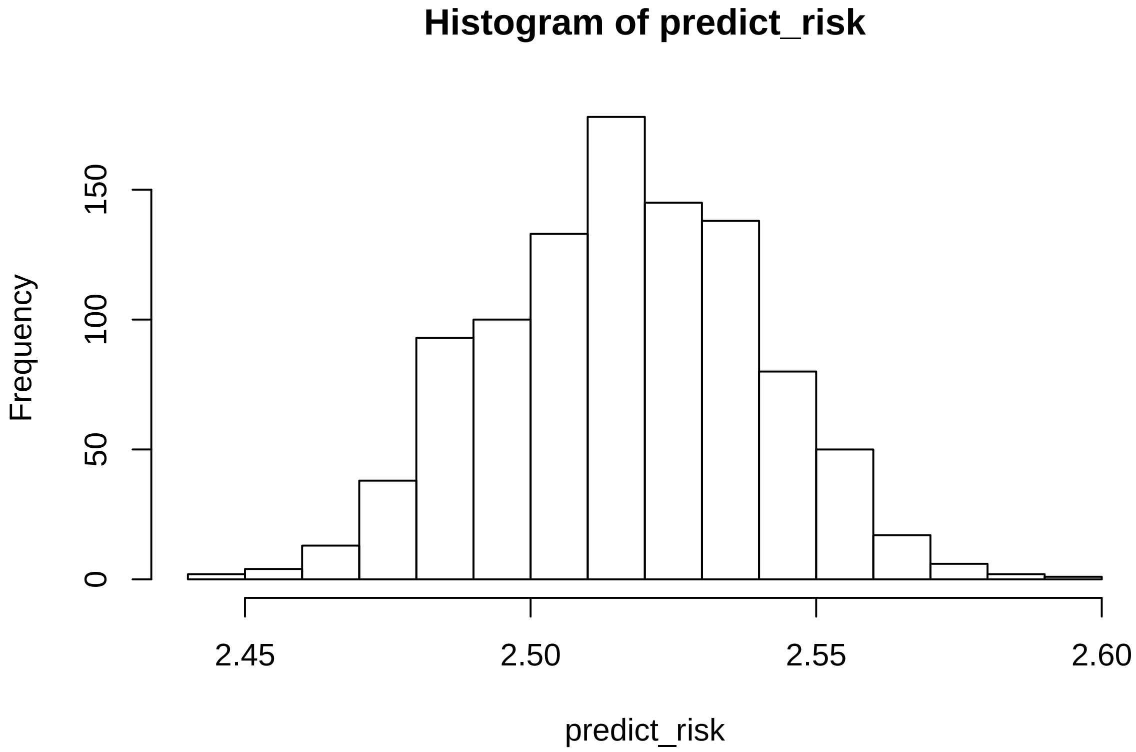
```
# Risky behaviors data
risky_behaviors<-read.dta("http://www.stat.columbia.edu/~gelman/arm/examples/risky.behavior/risky_behav

# Poisson regression model
risky_behaviors$fupacts = round(risky_behaviors$fupacts)
r_risk <- glm(fupacts~couples+women_alone, data = risky_behaviors, family = poisson())

# simulation
```

```
simfit_risk <- sim(r_risk, n.sims=1000)

#prediciton: couples=0, women_alone=1
predict_risk <- simfit_risk@coef[,1]+simfit_risk@coef[,2]*0+simfit_risk@coef[,3]*1
hist(predict_risk)
```



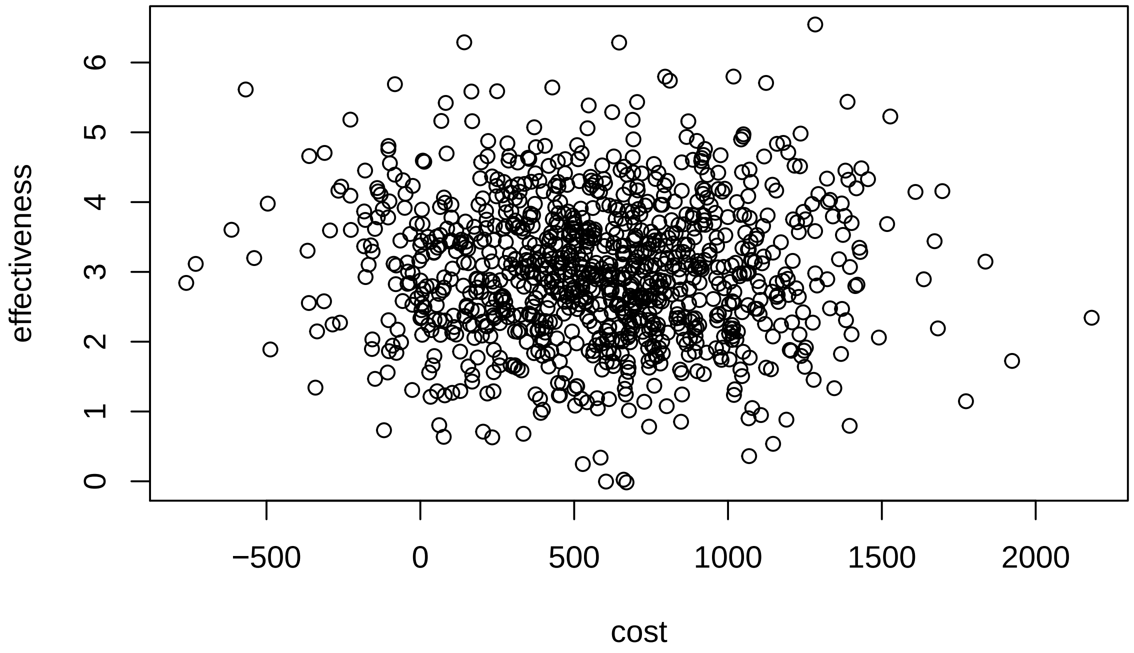
Inference for the ratio of parameters:

a (hypothetical) study compares the costs and effectiveness of two different medical treatments. - In the first part of the study, the difference in costs between treatments A and B is estimated at $600 per patient, with a standard error of $400, based on a regression with 50 degrees of freedom. - In the second part of the study, the difference in effectiveness is estimated at 3.0 (on some relevant measure), with a standard error of 1.0, based on a regression with 100 degrees of freedom. - For simplicity, assume that the data from the two parts of the study were collected independently.

Inference is desired for the incremental cost-effectiveness ratio: the difference between the average costs of the two treatments, divided by the difference between their average effectiveness. (This problem is discussed further by Heitjan, Moskowitz, and Whang, 1999.)

1. Create 1000 simulation draws of the cost difference and the effectiveness difference, and make a scatterplot of these draws.

```
#cost difference mean=600,sd=400
cost <- rnorm(1000, 600, 400)
#effectiveness difference mean=3,sd=1
effectiveness <- rnorm(1000,3,1)
# Scatterplot
plot(cost,effectiveness)
```



2. Use simulation to come up with an estimate, 50% interval, and 95% interval for the incremental cost-effectiveness ratio.

```
ratio <- cost/effectiveness
# 50% interval
quantile(ratio,c(0.25,0.75))
```

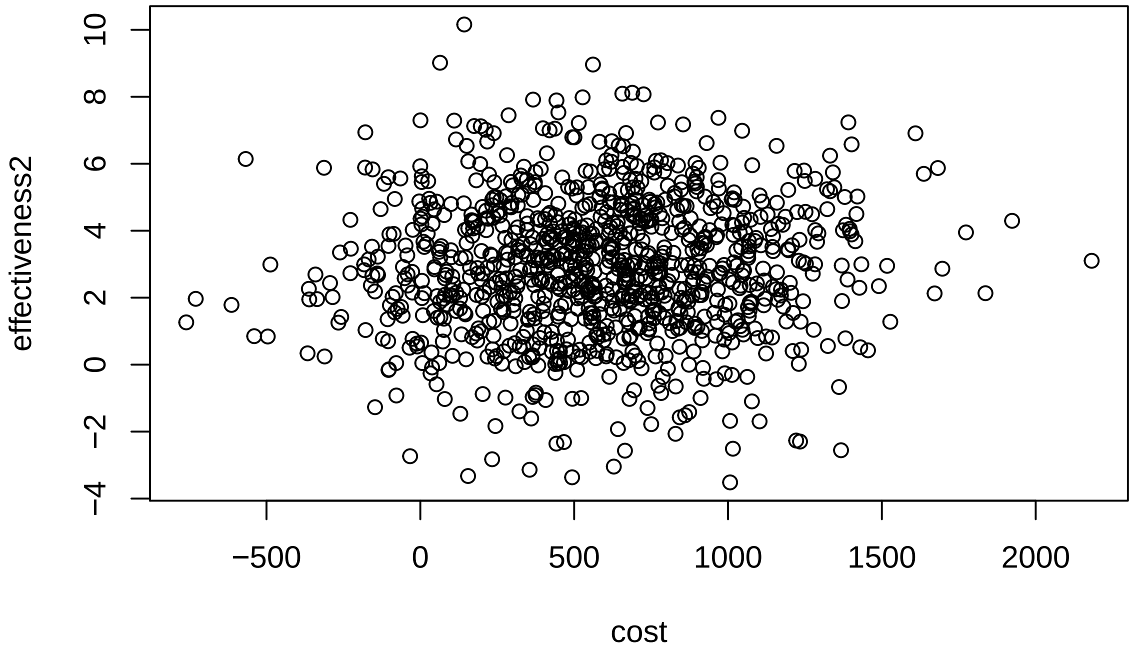
```
##       25%       75%
##  96.96087 308.56161
```

```
# 95% interval
quantile(ratio,c(0.025,0.975))
```

```
##      2.5%     97.5%
## -62.73158 694.37471
```

3. Repeat this problem, changing the standard error on the difference in effectiveness to 2.0.

```
#effectiveness difference mean=3,sd=2
effectiveness2 <- rnorm(1000,3,2)
plot(cost,effectiveness2)
```



```
ratio2 <- cost/effectiveness2
# 50% interval
quantile(ratio2,c(0.25,0.75))
```

```
##       25%       75%
##  64.83139 332.99475
```

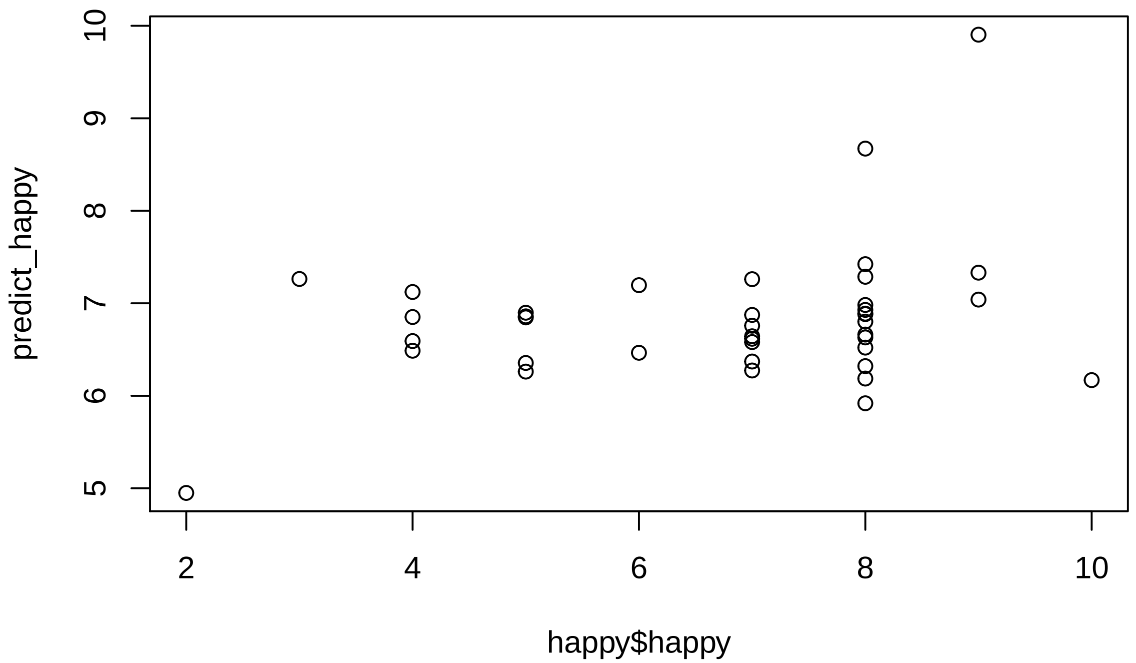
```
# 95% interval
quantile(ratio2,c(0.025,0.975))
```

```
##      2.5%     97.5%
## -670.3604 2367.3557
```

Predictive checks:

using data of interest to you, fit a model of interest. 1. Simulate replicated datasets and visually compare to the actual data.

```
data(happy)
r_happy <- lm(happy ~ money, data = happy)
simfit_happy <- sim(r_happy, length(happy$money))
predict_happy <- simfit_happy@coef[,2]*happy$money + simfit_happy@coef[,1]
print(plot(happy$happy, predict_happy))
```



```
## NULL
```

2. Summarize the data by a numerical test statistic, and compare to the values of the test statistic in the replicated datasets.

```
summary(happy$happy)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   5.000   7.000   6.744   8.000  10.000
```

```
summary(predict_happy)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.950   6.476   6.802   6.822   7.011   9.904
```

(optional) Propagation of uncertainty:

we use a highly idealized setting to illustrate the use of simulations in combining uncertainties. Suppose a company changes its technology for widget production, and a study estimates the cost savings at $5 per unit, but with a standard error of $4. Furthermore, a forecast estimates the size of the market (that is, the number of widgets that will be sold) at 40,000, with a standard error of 10,000. Assuming these two sources of uncertainty are independent, use simulation to estimate the total amount of money saved by the new product (that is, savings per unit, multiplied by size of the market).

(optional) Fitting the wrong model:

suppose you have 100 data points that arose from the following model: $y = 3 + 0.1x_1 + 0.5x_2 + error$, with errors having a t distribution with mean 0, scale 5, and 4 degrees of freedom. We shall explore the implications of fitting a standard linear regression to these data.

1. Simulate data from this model. For simplicity, suppose the values of `x_1` are simply the integers from 1 to 100, and that the values of `x_2` are random and equally likely to be 0 or 1. In R, you can define

x_1 <- 1:100, simulate x_2 using rbinom(), then create the linear predictor, and finally simulate the random errors in y using the rt() function. Fit a linear regression (with normal errors) to these data and see if the 68% confidence intervals for the regression coefficients (for each, the estimates ±1 standard error) cover the true values.

2. Put the above step in a loop and repeat 1000 times. Calculate the confidence coverage for the 68% intervals for each of the three coefficients in the model.
3. Repeat this simulation, but instead fit the model using t errors (use hett::tlm).

(optional) Using simulation to check the fit of a time-series model:

find time-series data and fit a first-order autoregression model to it. Then use predictive simulation to check the fit of this model as in GH Section 8.4.

(optional) Model checking for count data:

the folder risky.behavior contains data from a study of behavior of couples at risk for HIV;

"sex" is a factor variable with labels "woman" and "man". This is the member of the couple that reporting sex acts to the researcher

The variables "couple" and "women_alone" code the intervention:

couple women_alone 0 0 control - no conselling 1 0 the couple was counselled together 0 1 only the woman was counselled

"bs_hiv" indicates whether the member reporting sex acts was HIV-positive at "baseline", that is, at the beginning of the study.

"bupacts" - number of unprotected sex acts reportied at "baseline", that is, at the beginning of the study

"fupacts" - number of unprotected sex acts reported at the end of the study (final report).

1. Fit a Poisson regression model predicting number of unprotected sex acts from baseline HIV status. Perform predictive simulation to generate 1000 datasets and record both the percent of observations that are equal to 0 and the percent that are greater than 10 (the third quartile in the observed data) for each. Compare these values to the observed value in the original data.
2. Repeat (1) using an overdispersed Poisson regression model.

```
# afunction to geneate from quasi poisson
rqpois = function(n, lambda, phi) {
  mu = lambda
  k = mu/phi/(1-1/phi)
  return(rnbinom(n, mu = mu, size = k))
}
# https://www.r-bloggers.com/generating-a-quasi-poisson-distribution-version-2/
```

3. Repeat (2), also including gender and baseline number of unprotected sex acts as input variables.