3301HW5

Joshua O’Neill

2024-04-03

R Markdown

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

###1. You will analyze a reduced version of a dataset from Karagas et al. (1996). There are n = 21 subjects. The response is arsenic.toenail, which is the level of arsenic in the subject’s toenail. There are three explanatory variables:***

###• arsenic.water, the level of arsenic in the subject’s household water supply;

###• gender, the gender of the subject;

###• age, the age of the subject in years.

###The dataset is in the dataframe object arsenic, which is stored in the arsenic.RDS file available on canvas. To read the data in R, use

```
arsenic <- readRDS("arsenic.RDS")
head(arsenic)
```

```
##   arsenic.toenail arsenic.water age gender
## 1           0.119       0.00087  44 Female
## 2           0.118       0.00021  45 Female
## 3           0.099       0.00000  44   Male
## 4           0.118       0.00115  66 Female
## 5           0.277       0.00000  37   Male
## 6           0.358       0.00000  45 Female
```

###For the following parts of the question, you may not use the lm or drop1 functions to complete the problem. Of course, you are allowed to use them to check your work, but the code you submit must have all computations done “by-hand”.

(a) (2pts) Fit a linear regression model to these data, where the response is the natural logarithm of arsenic.toenail, and the explanatory variables are those listed above with the addition of an interaction between gender and arsenic.water. Report estimates of the regression coefficients and the error standard deviation.

```
arsenic.X <- cbind(1,arsenic$age,arsenic$arsenic.water, 1*(arsenic$gender=='Female'),
arsenic$arsenic.water*(arsenic$gender=='Female'))
arsenic.y <- log(arsenic$arsenic.toenail)
arsenic.beta <- qr.solve(crossprod(arsenic.X),crossprod(arsenic.X,arsenic.y))
arsenic.n.minus.p <- length(arsenic.y)-length(arsenic.beta)
error <- arsenic.y - (arsenic.X %*% arsenic.beta)
sE <- sqrt((sum(error^2))/(arsenic.n.minus.p))
print(arsenic.beta)
```

```
##             [,1]
## [1,] -0.76923387
## [2,] -0.02293539
## [3,] 16.03106323
## [4,]  0.03483843
## [5,] 10.45649872
```

```
print(sE)
```

```
## [1] 0.4864371
```

Intercept: -0.76923387

Coefficient of age: -0.02293539

Coefficient of arsenic.water: 16.03106323

Coefficient of gender: 0.03483843

Coefficient of arsenic.water interaction with gender: 10.45649872

Standard deviation of error: 0.4864371

(b) (2pts) Compute a 99% confidence interval for the regression coefficient corresponding to the explanatory variable age. Does this interval indicate that age is significant in this model at the 1% significance level? Explain.

```
arsenic.XinvX <- qr.solve(crossprod(arsenic.X))
MOE <- (sE * sqrt(arsenic.XinvX[2,2])) * qt(1-0.01/2,arsenic.n.minus.p)
c(arsenic.beta[2]-MOE,arsenic.beta[2]+MOE)
```

```
## [1] -0.044551773 -0.001319013
```

Because 0 is not present within this confidence interval, this interval does indicate that age is significant in this model at the 1% significance level.

(c) (4pts) Perform an F-test to check whether the interaction between gender and arsenic.water is significant at the α = 0.01 significance level. To complete this part of the problem (i) state the full model (Mf), (ii) state the null and alternative hypotheses in terms of parameters of the full model, (iii) compute the F-test statistics, (iv) compute the p-value, (v) and state your conclusion in the context of the problem.

M_f = β_1 + β_2*age + β_3*arsenic.water + β_4*gender + β_5*(Interaction of gender and arsenic.water)

M_0 = β_1 + β_2*age + β_3*arsenic.water + β_4*gender

H_0: β_1 + β_2*age + β_3*arsenic.water + β_4*gender + β_5*(Interaction of gender and arsenic.water) + ε̝ = arsenic.toenails

H_a: β_1 + β_2*age + β_3*arsenic.water + β_4*gender + ε̝ = arsenic.toenails

```
null.X <- cbind(1,arsenic$age,arsenic$arsenic.water, 1*(arsenic$gender=='Female'))
null.beta <- qr.solve(crossprod(null.X),crossprod(null.X,arsenic.y))
null.error <- arsenic.y - (null.X %*% null.beta)
arsenic.rssf <- sum(error^2)
arsenic.rss0 <- sum(null.error^2)
#d = 1, so dividing the numerator by d would have no effect.
arsenic.f <- (arsenic.rss0-arsenic.rssf)/(arsenic.rssf/arsenic.n.minus.p)
print(arsenic.f)
```

```
## [1] 1.670073
```

```
arsenic.p.value <- 1 - pf(arsenic.f,1,arsenic.n.minus.p)
print(arsenic.p.value)
```

```
## [1] 0.214602
```

The F-test statistic is 1.670073, and the p-value is 0.214602

At the α=0.01 level of significance, we fail to reject the null hypothesis in favor of the alternative hypotheses. In the context of the problem, we fail to reject the M_0 model is favor of the M_f model, meaning that the interaction between gender and arsenic.water is not significant at the α=0.01 significance level

(d) (3pts) Compute a 99% prediction interval for the yet-to-be observed value of arsenic.toenail for a 45-year-old male subject with 0 for the value of arsenic.water. You should first make this interval for the logarithm of arsenic.toenail and then evaluate the exponential function at the left and right endpoints. Explain the meaning of this interval.

```
arsenic.x_new <- c(1,45,0,0,0)
arsenic.y_new <- sum(arsenic.x_new * arsenic.beta)
arsenic.XinvX <- qr.solve(crossprod(arsenic.X))
MOE <- qt(1-0.01/2,arsenic.n.minus.p) * sE * sqrt(1 + t(arsenic.x_new) %*% arsenic.XinvX %*% arsenic.x_new)
c(arsenic.y_new - MOE, arsenic.y_new + MOE)
```

```
## [1] -3.3377719 -0.2648812
```

```
c(exp(arsenic.y_new-MOE),exp(arsenic.y_new+MOE))
```

```
## [1] 0.0355160 0.7672971
```

$$\text{So we have:}$$
$$-3.3377719 < \log(\text{arsenic.toenail}) < -0.2648812$$
$$e^{-3.3377719} < e^{\log(\text{arsenic.toenail})} < e^{-0.2648812}$$
$$0.0355160 < \text{arsenic.toenails} < 0.7672971$$

We are 99% confident that a 45 year old male with no arsenic in his household water-supply would have between 0.0355160 and 0.7672971 level of arsenic in his toenail.

(e) (3pts) Compute a 99% confidence interval for the expected value of arsenic.toenail for a 45-year-old male subject with 0 for the value of arsenic.water. You should first make this interval for the logarithm of arsenic.toenail and then evaluate the exponential function at the left and right endpoints. Compare this interval to that from the previous part. Specifically, comment on whether the interval is more narrow than in (d); and explain why it is or is not.

```
MOE <- qt(1-0.01/2,arsenic.n.minus.p) * sE * sqrt(t(arsenic.x_new) %*% arsenic.XinvX %*% arsenic.x_new)
c(arsenic.y_new - MOE, arsenic.y_new + MOE)
```

```
## [1] -2.386184 -1.216469
```

```
c(exp(arsenic.y_new-MOE),exp(arsenic.y_new+MOE))
```

```
## [1] 0.09198002 0.29627441
```

$$\text{So we have:}$$
$$-2.386184 < \log(\text{arsenic.toenail}) < -1.216469$$
$$e^{-2.386184} < e^{\log(\text{arsenic.toenail})} < e^{-1.216469}$$
$$0.09198002 < \text{arsenic.toenails} < 0.29627441$$

This interval is more narrow than the prediction interval computed in part d. This makes sense, because in part d, our goal was to predict y=xβ+ε, whereas in part e, our goal was to predict xβ. Thus, in part d, our interval had to account for the variation of ε, necessitating that our prediction covered a wider area, whereas in part e, our interval did not have to account for this extra variation.

(f) (4pts) Perform a simulation study to make simulation-based inference for the coverage probability of the random prediction interval for a future realization of the response. Use the design matrix from 1a. Set β ∊ R5 and σ to be the estimates computed in 1a (this is called the fitted model). Make the regression errors independent copies of Z = σ√12(U − 0.5), where U ~ Unif(0, 1). Using 10,000 replication, make a 99% simulation-based score approximate confidence interval for the coverage probability of the 95% random prediction interval for the yet-to-be observed value of arsenic.toenail for a Male of age 45 with arsenic.water = 0.0. In each replication, this prediction inter- val should be computed for the logarithm of arsenic.toenail and then the exponential function should be evaluated at the endpoints. Is there simulation-based evidence that the coverage probability of this random prediction interval is not equal to 95%? Explain.

```
reps <- 1e4
prediction.true <- numeric(reps)
# sd.error <- sd(error)
# mean.guess <- t(null.beta) %*% arsenic.x_new
mean.guess <- sum(arsenic.beta*arsenic.x_new)
# null.XinvX <- qr.solve(crossprod(null.X))

for (i in 1:reps) {
  Z = sE*sqrt(12)*(runif(length(arsenic.y),-0.5,0.5))
  arsenic.y.guess <- (mean.guess * c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1) + Z)
  new_beta <- qr.solve(crossprod(arsenic.X), crossprod(arsenic.X,arsenic.y.guess))
  residuals <- arsenic.y.guess - arsenic.X %*% new_beta
  sE <- sqrt(sum(residuals^2)/(arsenic.n.minus.p))
  MOE <- qt(1-0.05/2,arsenic.n.minus.p) * sE * sqrt(1 + t(arsenic.x_new) %*% arsenic.
XinvX %*% arsenic.x_new)
  prediction <- t(arsenic.x_new) %*% new_beta + sE*sqrt(12)*(runif(1,-0.5,0.5))
  prediction.true[i] <- 1*(exp(mean.guess) > exp(prediction - MOE)) * (exp(mean.guess
) < exp((prediction) + MOE))
}
prop.test(x=sum(prediction.true), n=10000, conf.level=0.99, correct=FALSE)$conf.int[1
:2]
```

```
## [1] 0.9814219 0.9877351
```

It appears that the coverage probability of this random prediction interval is significantly greater than, and therefore not equal to 95%. The likely reason for this, is that the new produced data assumes that the chosen mean of beta dotted with x_new is correct, and then produces data based on that assumption. This leads to produced data to being concentrated x_new dotted with beta, when real life data might not actually be.

2. Recall the resting heart rate data from Homework 4, where researchers recruited 35 subjects who had never performed any type of cardiovascular exericse and assigned them to perform either 0.5, 1.5, or 2.5 hours of low-intensity cardiovascular exercise per week. Researchers then measured how their resting heart rate changed after three months of the protocol. The data from the study can be downloaded using

```
rhrdat <- readRDS("RHR.RDS")
head(rhrdat)
```

```
##        RHRdec exercise age
## 1 -0.07485900      0.5  24
## 2 -0.22325231      1.5  26
## 3 -0.20845607      2.5  30
## 4 -0.06535626      0.5  37
## 5 -0.14499982      1.5  22
## 6 -0.13974863      2.5  37
```

###The data contain three variables

###• RHRdec: the decrease in resting heart rate after three months of the protocol, computed as

$$\text{RHRdec} = \frac{\text{Month 3 RHR} - \text{Initial RHR}}{\text{Initial RHR}}$$

###so that a value of -0.1 can be interpreted as a 10% decrease in RHR.

###• exercise: the number of hours of exercise per week during the three month study

###• age: the subject’s age at the start of the study

(a) (1pt) By hand (i.e., not using lm), compute the R-squared to the linear regression model with RHRdec as the response and exercise and age as predictors.

```
rhr.X <- cbind(1,rhrdat$exercise,rhrdat$age)
rhr.y <- rhrdat$RHRdec
rhr.beta.cap <- qr.solve(crossprod(rhr.X),crossprod(rhr.X,rhr.y))
rhr.error.squared <- sum((rhr.y - rhr.X %*% rhr.beta.cap)^2)/(length(rhr.y)-1)
rhr.R.squared <- (var(rhr.y)-rhr.error.squared)/var(rhr.y)
c(rhr.R.squared)
```

```
## [1] 0.3632317
```

It is reasonable to treat exercise as a categorical variable. To do so, you will need to create four “dummy” variables, like so

```
exer05 <- 1*(rhrdat$exercise==0.5)
exer15 <- 1*(rhrdat$exercise==1.5)
exer25 <- 1*(rhrdat$exercise==2.5)
```

(b) (2pt) The model that treats exercise as numeric is nested in the model which treats exercise as categorical. Explain why. Note here that the predictor can only take the values 0.5, 1.5, and 2.5.

A nested model is defined by Portland university as “A nested model is a model that uses the same variables (and cases!) as another model but specifies at least one additional parameter to be estimated”. This is true, because the exact same variable-based information is fed to both models, and by treating exercise as being three categorical traits as opposed to one numeric trait, two paramters are being added. Thus, our new model is a nested in the original model.

Additionally, we can think about this mathematically: M1: y=b1+b2x_age+b3x_ex15+b4x_ex25, then when b3=b4-b3, M1 will degenerate to M0. This means that M0 is essentially a less flexible version of M1, and so M1 can be minimized in more ways than M0, and therefore can be minimized more than M1 can.

(c) (3pts) By hand, fit the linear regression model with age and the categorical version of exercise as predictors. (1pt) Print the coefficient estimates. (1pt) Interpret the coefficient estimate corresponding to exer15. (1pt) Based on these coefficient estimates, what is the expected value of RHRdec for someone who was in the group assigned 0.5 hours of exercise?

```
nested.X <- cbind(1,rhrdat$age,exer15,exer25)
nested.beta <- qr.solve(crossprod(nested.X),crossprod(nested.X,rhr.y))
print(nested.beta)
```

```
##                [,1]
##        -0.226091786
##         0.004505763
## exer15 -0.055394034
## exer25 -0.077380144
```

```
# print(nested.beta[3]-nested.beta[4])
```

If it is true that someone is part of the exer15 group, then they would have a, 0.077380144-0.021986110=0.05539403 greater rhr reduction than someone in the exer05 group, and they would have a 0.021986110 lesser rhr reduction than someone in the exer25 group.

```
print(mean(rhrdat$age))
```

```
## [1] 28.68571
```

```
print(nested.beta[1] + mean(rhrdat$age) * (nested.beta[2]) + nested.beta[3])
```

```
## [1] -0.1522348
```

-0.09684074 Given that the average age in the data is 28.68571, we can identify that the average person in the study, irrespective of exercise, should experience a -0.303471930 + 28.68571(0.004505763) decrease in rhr. Someone being in the exer05 group then means that value should be offset by 0.077380144. Thus, -0.303471930 + 28.68571(0.004505763) + 0.077380144 = -0.09684074 represents the expected value of rhr decrease for someone in the exer05 group.

(d) (3pts) By hand, compute the R-squared for the model treating exercise as categorical. Is the R-squared higher than the model treating exercise as numeric? If so explain, why. If not, explain why not.

```
nested.error.squared <- sum((rhr.y - (nested.X %*% nested.beta))^2)/(length(rhr.y)-1)
nested.R.squared <- (var(rhr.y) - nested.error.squared) / var(rhr.y)
print(nested.R.squared)
```

```
## [1] 0.3793446
```

The R-squared is higher than the model treating exercise as numeric. This is likely the case, because there are more dimensions being analyzed when exercise is treated as categorical, meaning that more nuances from within the data are being detected. It also allows for exer05, exer15, and exer25 to be analyzed separately from one another, allowing for the naunces of how the differences between exer05 and exer15, and exer15 and exer25 may vary from one another to be detected, which wouldn't be if a linear relationship was assumed as it was in the numeric model.

(e) (3pts) Using an F-test, test for an interaction between the categorical version of exercise and age. To do so, (i) state the full model you are testing, (ii) state the null and alternative hypotheses in terms of parameters of the full model, (iii) compute the F-test statistics, (iv) compute the p-value, (v) and state your conclusion in the context of the problem.

H_0: β_1 + β_2*age + β_3*exer05 + β_4*exer15

H_a: β_1 + β_2*age + β_3*exer05 + β_4*exer15 + β_5*exer05*age + β_6*exer15*age

```
interact.X <- cbind(1, rhrdat$age, exer05, exer15, rhrdat$age*exer05, rhrdat$age*exer15)
interact.beta <- qr.solve(crossprod(interact.X),crossprod(interact.X,rhr.y))
interact.residuals <- rhr.y - interact.X %*% interact.beta
nested.residuals <- rhr.y - nested.X %*% nested.beta
rhr.rssf <- sum(interact.residuals^2)
rhr.rss0 <- sum(nested.residuals^2)
n.minus.p <- length(rhr.y) - length(interact.beta)
d <- length(interact.beta) - length(nested.beta)
rhr.f.stat <- ((rhr.rss0-rhr.rssf)/(d))/((rhr.rssf)/(n.minus.p))
rhr.p.value <- 1 - pf(rhr.f.stat,d,n.minus.p)
c(rhr.f.stat,rhr.p.value)
```

```
## [1] 0.5120037 0.6046101
```

f-statistic: 0.5120037, p-value: 0.6046101

We fail to reject the null hypothesis, the model without any interaction between age and exercise, in favor of the alternative hypothesis, the model with interaction between age and exercise, under any practical level of significance. This means that, the interaction between age and exercise does not appear to be statistically significant in its contribution towards resting heart rate decrease.

(f) (3pts) Using a (roughly) 50:50 split of the data (e.g., allocating 18 individuals to the training set), compute an estimate of the out-of-sample prediction error for the two models that

• Treats exercise as numeric, i.e.,

. y ~ age + exercise

• Treats exercise as categorical, i.e.,

. y ~ age + exerc15 + exer25

```
train.indices <- 1:18
X.num.tr <- rhr.X[train.indices,]
X.nest.tr <- nested.X[train.indices,]
y.tr <- rhr.y[train.indices]
X.num.va <- rhr.X[-train.indices,]
X.nest.va <- nested.X[-train.indices,]
y.va <- rhr.y[-train.indices]
num.tr.beta <- qr.solve(crossprod(X.num.tr),crossprod(X.num.tr,y.tr))
nest.tr.beta <- qr.solve(crossprod(X.nest.tr),crossprod(X.nest.tr,y.tr))
num.error <- mean((y.va - X.num.va %*% num.tr.beta)^2)
nest.error <- mean((y.va - (X.nest.va %*% nest.tr.beta))^2)
c(num.error,nest.error)
```

```
## [1] 0.003983784 0.003888860
```

The out of sample prediction error for our numeric model is 0.003983784, and the out of sample prediction error for our categorical model is 0.003888860

###3. In this problem you will perform simulation studies for the linear regression model with iid Normal errors. This model assumes that the measured response for the n subjects y = (y1, . . . , yn)⊤ is a realization of the random vector

$$Y = X\beta + \epsilon$$

###where X is the known design matrix with n rows and p columns; β = (β1, . . . , βp)⊤ is the vector of unknown regression coefficients; and ε = (ε1, . . . , εn)⊤ has ε1, . . . , εn iid N (0, σ2).

(a) (1pt) Suppose that there are n subjects and two numerical explanatory variables. Write an R function called gen.design.matrix that randomly generates a design matrix X with n rows and 3 columns. In the ith row, set xi1 = 1 and generate (xi2, xi3) as a realization of the random variables (Xi2, Xi3) defined by

$$X_{i2} = \mu + A_i + Z_{i,2}$$
$$X_{i3} = \mu + A_i + Z_{i,3}, i = 1, \ldots, n$$

where μ ∈ R, A1, . . . , An are iid N(0, ρσ2, X) with ρ ∈ [0, 1); and Z1,2, . . . , Zn,2, Z1,3, . . . , Zn,3 are iid N (0, (1 − ρ)σ2 X). All the Ai's, Zi's, and Zi,3's are independent.

This function should have four arguments:

• n, the number of subjects,

• mu, the value of μ defined above,

• sigma.X, the value of σX defined above,

• rho, the value of ρ defined above.

This function returns the generated design matrix, which has n rows and 3 columns.

```
gen.design.matrix <- function(n, mu, sigma, rho) {
  A <- rnorm(n,0,sqrt(rho*sigma^2))
  Z.2 <- rnorm(n,0,sqrt((1-rho)*sigma^2))
  Z.3 <- rnorm(n,0,sqrt((1-rho)*sigma^2))
  X.2 <- mu + A + Z.2
  X.3 <- mu + A + Z.3
  design.matrix <- cbind(1,X.2,X.3)
  return(design.matrix)
}
```

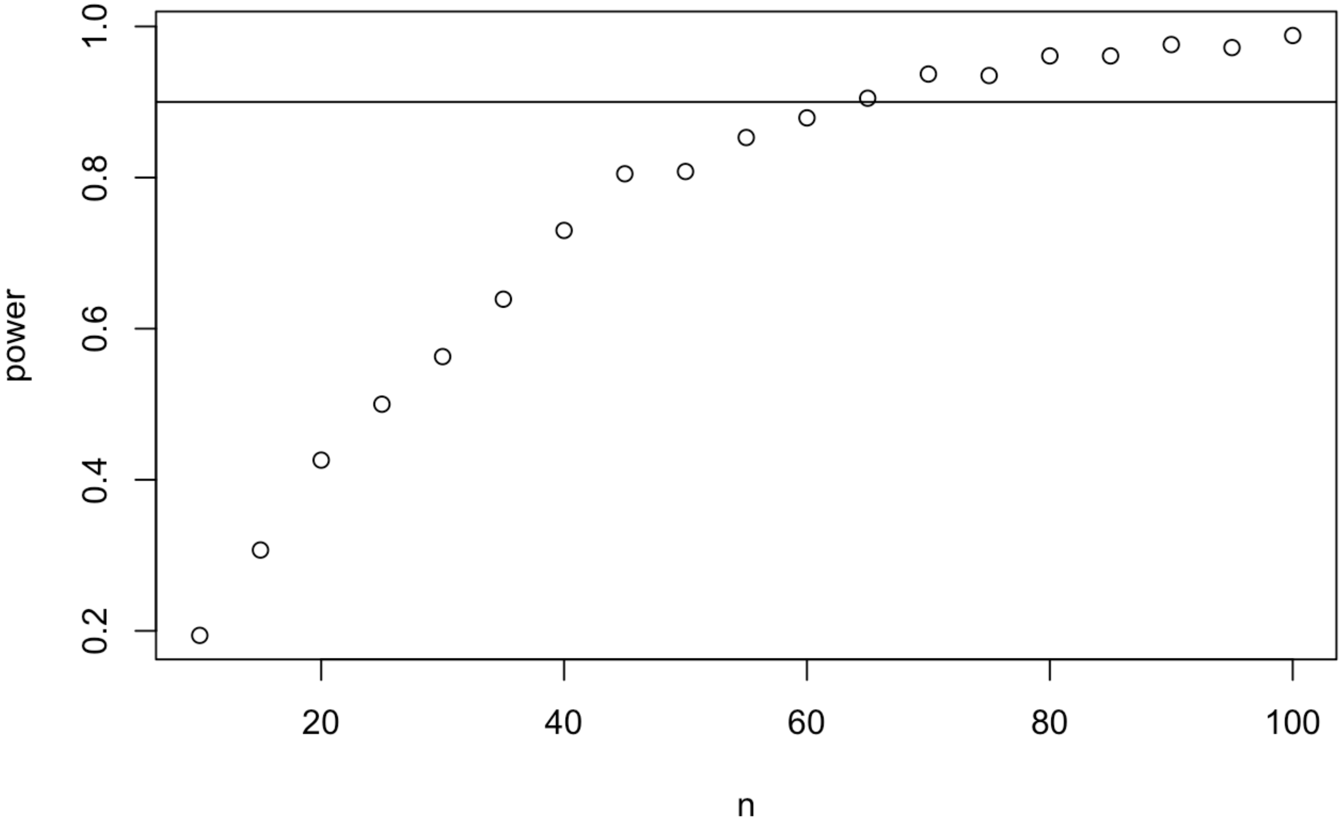
(b) (2pts) Suppose that the model in (1) is correct, p = 3, β = (1, 0, −1)⊤, and σ = 4. Use simulation to find the number of subjects n so that the power of the test of

$$H_0 : \beta_3 = 0,\ H_a : \beta_3 \neq 0,$$

is roughly 90% when a 5% significance level is used in the following two cases:

i. The design matrix is generated from the procedure described in part 2a with μ = 68, σX = 2, and ρ = 0.5. In this case, cor(Xi2, Xi3) = 0.5.

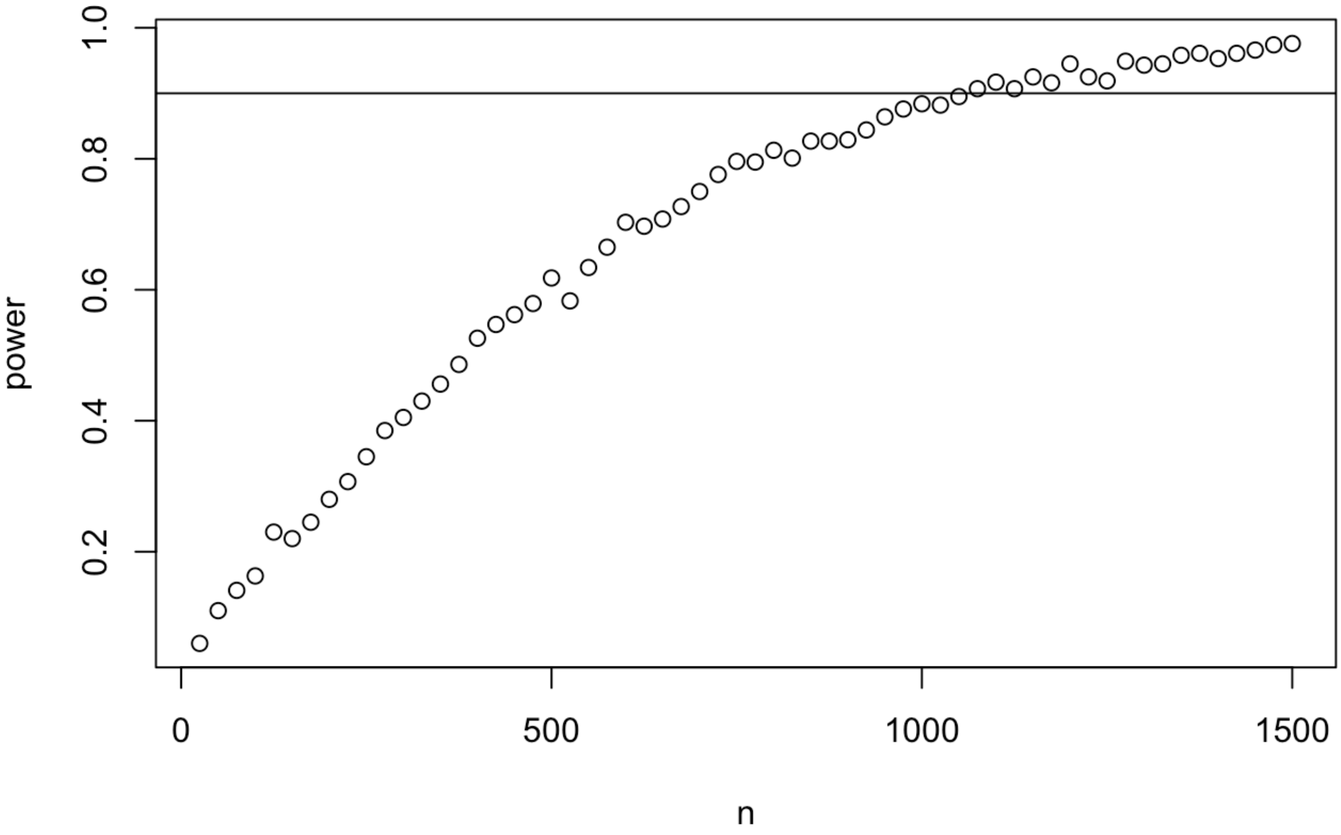
```
nseq <- seq(10,100,5)
design.beta <- c(1,0,-1)
power <- numeric(length(nseq))
for (i in 1:length(nseq)) {
  rejected <- numeric(1000)
  n <- nseq[i]
  for (j in 1:1000) {
    design.matrix <- gen.design.matrix(n,68,2,0.5)
    design.y <- (design.matrix %*% design.beta) + rnorm(n,0,4)
    design.beta.hat <- qr.solve(crossprod(design.matrix),crossprod(design.matrix,desi
gn.y))
    design.XinvX.3 <- sqrt((qr.solve(crossprod(design.matrix)))[3,3])
    design.sE <- sqrt(sum((design.y - (design.matrix %*% design.beta.hat))^2)/(n-3))
    design.T <- (design.beta.hat[3])/(design.XinvX.3 * design.sE)
    rejected[j] <- 2*pt(-1*abs(design.T),n-3) < 0.05
  }
  power[i] <- mean(rejected)
}

plot(nseq,power,xlab='n',ylab='power')
abline(0.9,0)
```



The power of the above test becomes about 0.9 when n=60.

ii. The design matrix is generated from the procedure described in part 2a with μ = 68, σX = 2, and ρ = 0.98. In this case, cor(Xi2, Xi3) = 0.98.

```
nseq <- seq(25,1500,25)
design.beta <- c(1,0,-1)
power <- numeric(length(nseq))
for (i in 1:length(nseq)) {
  rejected <- numeric(1000)
  n <- nseq[i]
  for (j in 1:1000) {
    design.matrix <- gen.design.matrix(n,68,2,0.98)
    design.y <- (design.matrix %*% design.beta) + rnorm(n,0,4)
    design.beta.hat <- qr.solve(crossprod(design.matrix),crossprod(design.matrix,desi
gn.y))
    design.XinvX.3 <- sqrt((qr.solve(crossprod(design.matrix)))[3,3])
    design.sE <- sqrt(sum((design.y - (design.matrix %*% design.beta.hat))^2)/(n-3))
    design.T <- (design.beta.hat[3])/(design.XinvX.3 * design.sE)
    rejected[j] <- 2*pt(-1*abs(design.T),n-3) < 0.05
  }
  power[i] <- mean(rejected)
}
plot(nseq,power,xlab='n',ylab='power')
abline(0.9,0)
```



The power of the above test becomes about 0.9 when n=1100