Final Paper of ADA

BI Norwegian Business School — December 7, 2022

Assignment 1

A

I

Variable Name	Num of Missing Values
PIN	0
CIN	0
continent	0
country	0
city	166
countrycode	641
partnum	20
partcode	15
sample	7795
sex	0
age	0
religious	64
religion	497
relstat	171
relstat2	171
rellength	7072
ideal_intelligence	36
ideal_kindness	45
ideal_health	67
ideal_physatt	351
ideal_resources	401
mate_age	5479
popsize	0
country_religion	0

lattitude	0
gem1995	3302
gdi1995	1842
gii	281
gdi2015	251
gggi	0
gdp_percap	0
infect_death	0
infect_yll	0
cmc_yll	0
gb_path	9453

```r
data <- read.csv("ReplicationProcessedfinaldata04202018.csv")
miss <- data.frame(columns=colnames(data),missing.num=NA)
for (column in miss$columns){
  miss[miss$columns==column,2] <- sum(is.na(data[,column]))
}
rows.before <- nrow(data)
miss
# here we didn't use the commands given in the hints because we
   think directly creating a dataframe would be more convenient.
```

II

We have removed 40.1764% of the total observations.

```r
data <- data[-c(which((data$mate_age<12)|(is.na(data$mate_age))))
   ,]
rows.after <- nrow(data)
proportion.removed <- (rows.before-rows.after)/rows.before
proportion.removed #0.401764
```

III

Country Name	Num of Observations
Hungary	839
Pakistan	474
Poland	380
Slovenia	476

Turkey	564

```
countries <- data.frame(country.name=unique(data$country), nCountries=NA)
for (name in countries$country.name) {
  countries[countries$country.name==name,2]=sum(data$country==name)
}
countries <- countries[-c(which(countries$nCountries<350)),]
```

B

I

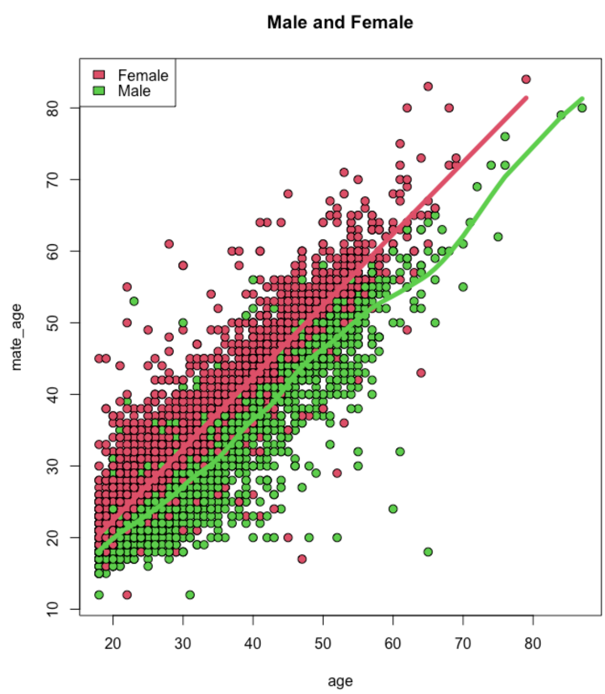
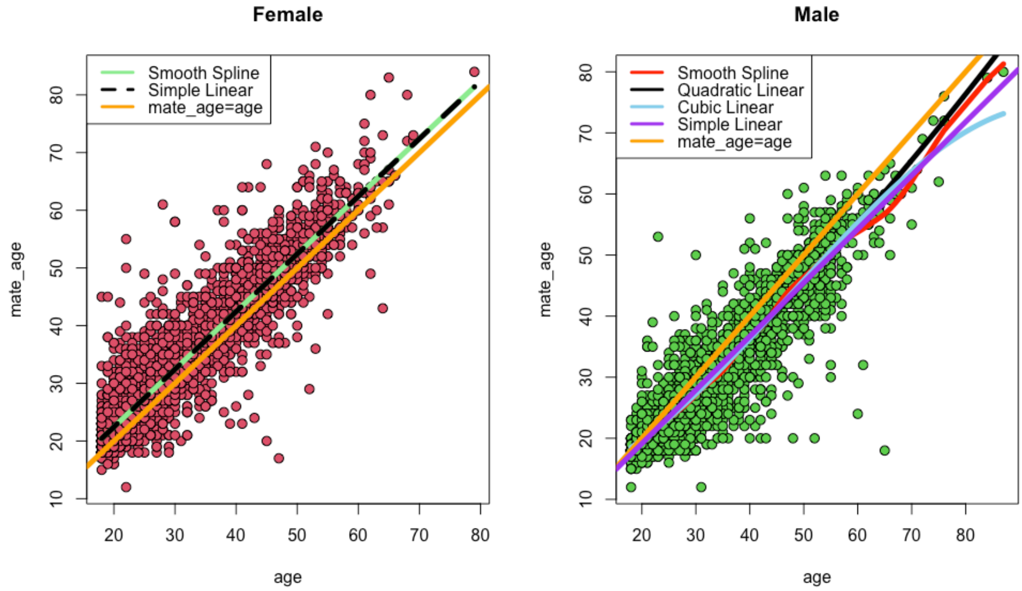


Figure 1: Scatter Plot of Male and Female

After testing the fitness of simple, quadratic and cubic linear models for both male and female groups. We identify the model as shown in equation 1 below which implements simple linear for female and quadratic linear for male. Averagely speaking, the male tend to find a mate around 3 years younger and females' mates tend to be approximate 2.5 years older. Looking at the plot for females, the regression line shoots above the line of mate_age=age. It indicates that females tend to choose older mates than their age. In contrast, for males, the opposite happens, which indicates that males tend to prefer younger partners. All the p-values are statistically significant, meaning that we should include them all in our model.



(a) Female Group (b) Male Group

Figure 2: Two Groups

$$mate\_age = \beta_0 + \beta_1 age + \beta_2 sex + \beta_3 age \times sex + \beta_4 sex \times age^2 \quad (1)$$

Table 3: Regression Outcomes

	Dependent variable:					
	Mate_Age		Mate_Age			Mate_Age
	Female		*Male*			*Both*
	lmFe	lmFeQu	lmMe	lmMeQu	lmMeCu	lmJoin
age	0.999^{***}	1.027^{***}	0.878^{***}	0.678^{***}	0.017	0.999^{***}
	(0.005)	(0.036)	(0.005)	(0.032)	(0.110)	(0.005)
age^2		−0.0004		0.003^{***}	0.020^{***}	
		(0.001)		(0.0004)	(0.003)	
age^3					-0.0001^{***}	
					(0.00002)	
sex						2.389^{***}
						(0.603)
age×sex						-0.321^{***}

						(0.034)
sex×age^2						0.003***
						(0.0005)
Constant	2.449***	2.018***	1.576***	4.839***	12.527***	2.449***
	(0.175)	(0.584)	(0.181)	(0.553)	(1.336)	(0.169)
Observations	4,752	4,752	3,862	3,862	3,862	8,614
R^2	0.876	0.876	0.872	0.873	0.875	0.878
Adjusted R^2	0.876	0.876	0.872	0.873	0.874	0.878
Residual SE	4.160	4.161	3.856	3.837	3.818	4.019

Note: *p<0.1; **p<0.05; ***p<0.01

II

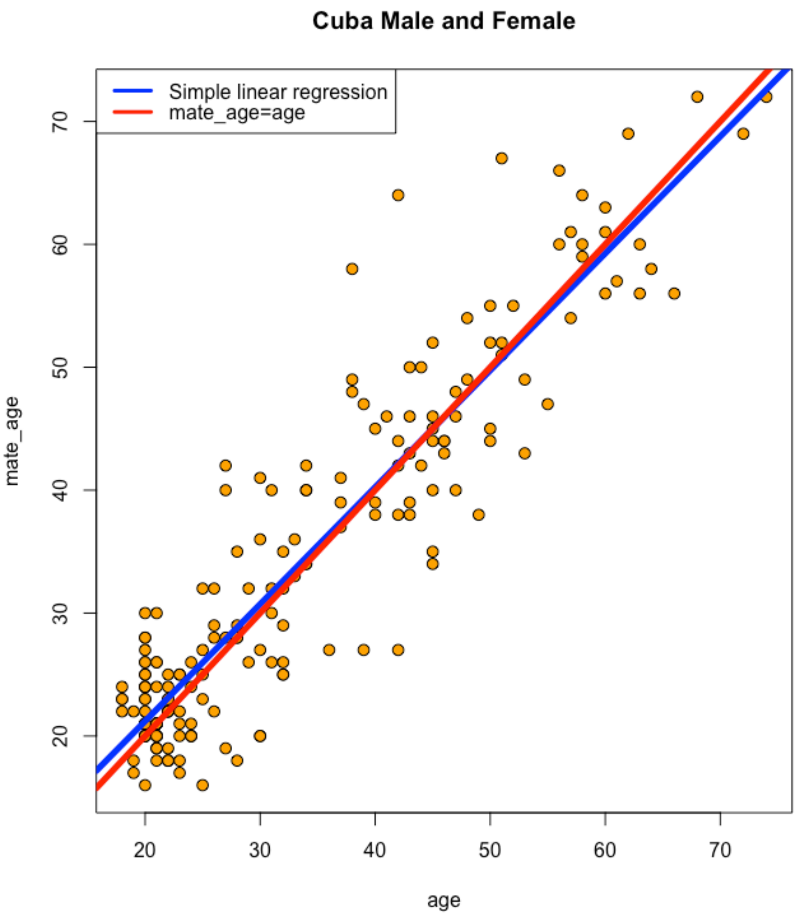


Figure 3: Scatter Plot of Cuba Male and Female

The F-test suggests for the Cuba dataset, without considering the term gender, we have no evidence to reject the null hypothesis which means there is no mate age preference for each gender.

```
Analysis of Variance Table

Model 1: mate_age ~ age
Model 2: mate_age ~ offset(1 * age) - 1
  Res.Df  RSS Df Sum of Sq     F Pr(>F)
1    186 5850
2    188 5992 -2   -142.03 2.258 0.1074
```

Listing 1: Anova Analysis of Both Gender Dataset

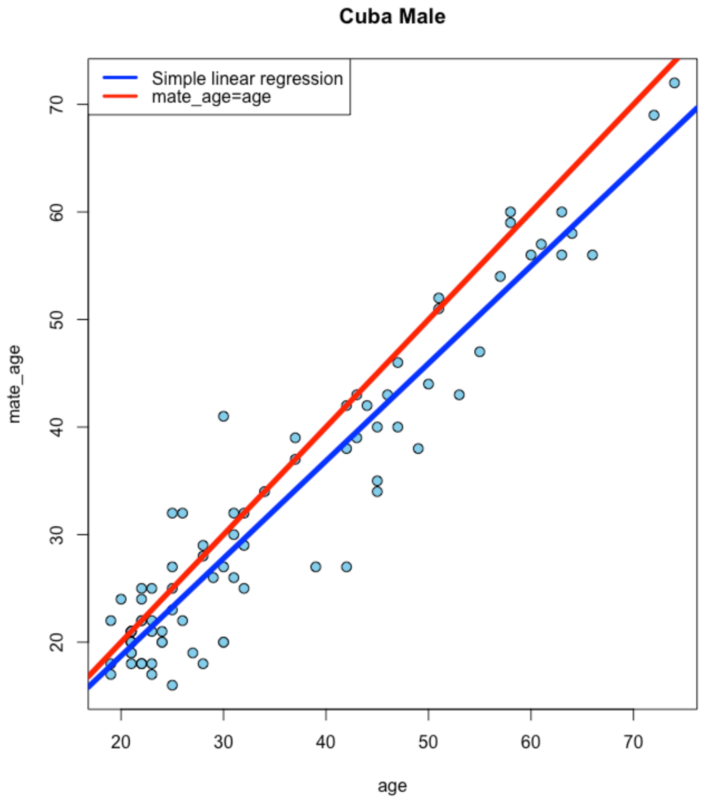


Figure 4: Scatter Plot of Cuba Male

The F-test is significant for the male data group. Combined with Figure 4 we have the evidence to conclude that males tend to prefer younger mates on average.

```
Analysis of Variance Table

Model 1: mate_age ~ age
Model 2: mate_age ~ offset(1 * age) - 1
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     86 1524.8
2     88 2292.0 -2   -767.15 21.633 2.452e-08 ***
```

Listing 2: Anova Analysis of Male Dataset

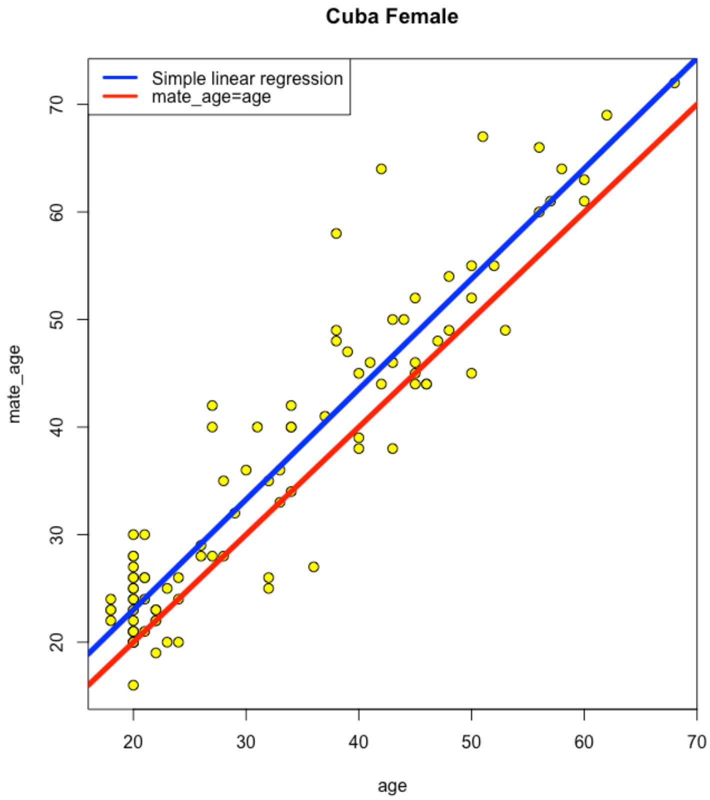


Figure 5: Scatter Plot of Cuba Female

The F-test for female group is also significant, meaning on average, females tend to prefer older mates.

```
Analysis of Variance Table

Model 1: mate_age ~ age
Model 2: mate_age ~ offset(1 * age) - 1
  Res.Df    RSS Df Sum of Sq     F    Pr(>F)
1     98 2573.1
2    100 3700.0 -2   -1126.9 21.46 1.863e-08 ***
```

Listing 3: Anova Analysis of Female Dataset

The fact that male and female have opposite age preference with regard to choosing mate could justify the insignificance in Listing 1 since the two opposite age preference could be offset when male and female merge as one large group

III

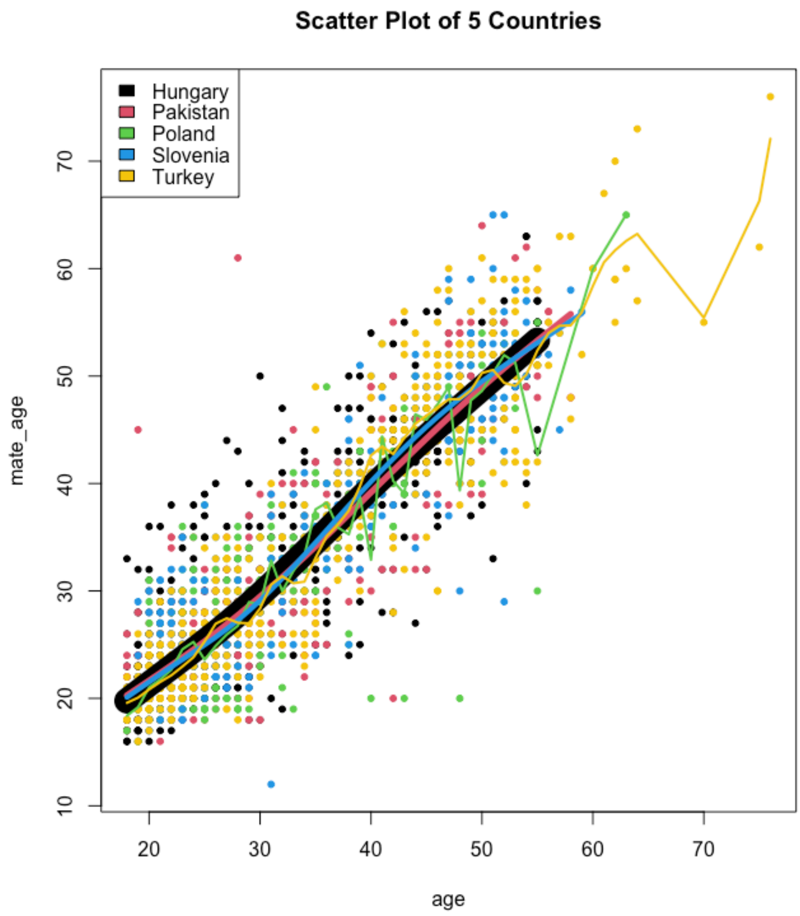
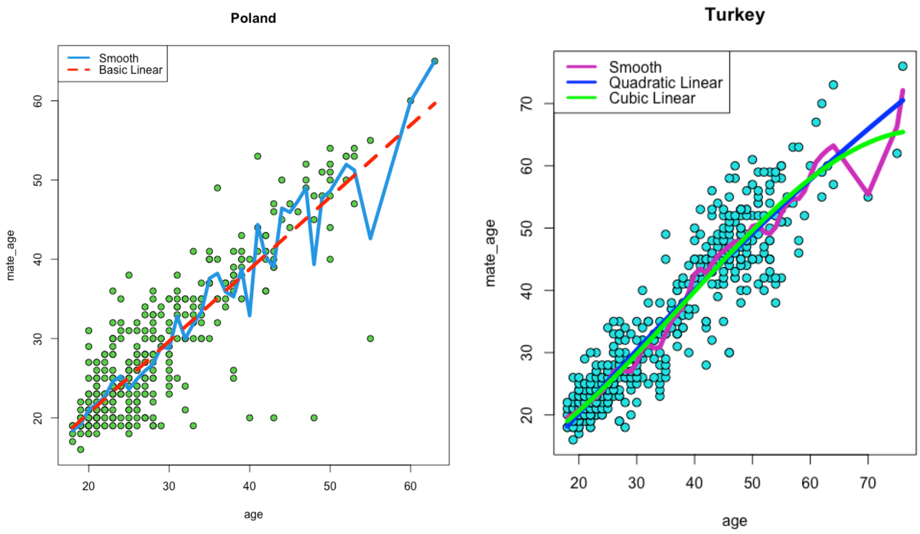


Figure 6: Scatter Plot and Smooth Splines

The scatterplot suggests a highly overlapped common trend for Hungary, Pakistan, and Slovenia. As a result, we treat these countries as base case. For Turkey and Poland, we introduce two indicator variables. From our analysis, one can draw that the best fit models for base case and Poland are both simple linear models, whereas a cubic linear model for Turkey. The term *Poland* would be no longer significant after introducing its interaction term with *age*. The result of F-test between models before and after introducing interaction term is insignificant indicating that the interaction term is unnecessary. Hence we remove the interaction term of *Poland*, and we end up with the final model as shown in equation 2 and regression result shown in table 5 (lmJoint2) which is statistically more plausible.



(a) Poland

(b) Turkey

Figure 7: Scatter plot seperately

Table 4

	Dependent variable:					
	Mate_Age	Mate_Age		Mate_Age		
	ThreeCountry	*Poland*		*Turkey*		
	lmThree	lmPo	lmPoQu	lmTu	lmTuQu	lmTuCu
age	0.933***	0.909***	0.822***	0.952***	1.138***	0.200
	(0.008)	(0.026)	(0.170)	(0.015)	(0.093)	(0.367)
age^2			0.001		−0.003**	0.022**
			(0.002)		(0.001)	(0.009)
age^3						−0.0002***
						(0.0001)
Constant	2.170***	2.405***	3.744	1.539***	−1.485	9.519**
	(0.254)	(0.793)	(2.726)	(0.517)	(1.568)	(4.452)
Observations	2,733	380	380	564	564	564
R^2	0.839	0.758	0.758	0.883	0.884	0.886
Adjusted R^2	0.839	0.757	0.756	0.883	0.884	0.885
Residual SE	4.52	4.59	4.60	4.40	4.39	4.36

Note: *p<0.1; **p<0.05; ***p<0.01

$$
\begin{aligned}
mate\_age = &\beta_0 + \beta_1 age + \beta_2 Poland + \beta_3 Turkey \\
&+ \beta_4 Turkey \times age^2 + \beta_5 Turkey \times age^3
\end{aligned}
\tag{2}
$$

```
Analysis of Variance Table

Model 1: mate_age ~ age + Poland + Poland * age + Turkey + Turkey
    * I(age^2) +
     Turkey * I(age^3) - I(age^3) - I(age^2)
Model 2: mate_age ~ age + Poland + Turkey + Turkey * I(age^2) +
    Turkey *
     I(age^3) - I(age^3) - I(age^2)
  Res.Df   RSS Df Sum of Sq      F Pr(>F)
1   2726 55520
2   2727 55528 -1   -8.5642 0.4205 0.5167
```

Listing 4: Anova Analysis of Models with and without Interaction Term

Table 5

	Dependent variable:	
	mate_age	
	lmJoint1	lmJoint2
age	0.927***	0.925***
	(0.010)	(0.009)
Poland	−0.058	−0.577**
	(0.841)	(0.255)
Turkey	−1.692***	−1.734***
	(0.588)	(0.584)
age×Poland	−0.018	
	(0.028)	
Turkey×age^2	0.004***	0.004***
	(0.001)	(0.001)
Turkey×age^3	−0.0001***	−0.0001***
	(0.00002)	(0.00002)

Constant	2.462***	2.531***
	(0.318)	(0.299)
Observations	2,733	2,733
R^2	0.840	0.840
Adjusted R^2	0.840	0.840
Residual Std. Error	4.513	4.512
Note:	*p<0.1; **p<0.05; ***p<0.01	

Assignment 2

A

We already know that:

$$x = a + by \tag{3}$$
$$y = c + dx \tag{4}$$

Now we use equation 4 to substitute the y in equation 3, so we can get the following equation:

$$\begin{aligned} x &= a + b(c + dx) \\ x &= a + bc + bdx \\ (1 - bd)x &= a + bc \end{aligned}$$

Since $bd \neq 1$, we can perform the following transformation:

$$x = \frac{a + bc}{1 - bd} \tag{5}$$

In the same way, we can compute the y as follows:

$$\begin{aligned} y &= c + d(a + by) \\ y &= c + ad + bdy \\ (1 - bd)y &= c + ad \\ y &= \frac{c + ad}{1 - bd} \\ y &= \frac{c + ad + bdc - bdc}{1 - bd} \\ y &= \frac{c(1 - bd) + d(a + bc)}{1 - bd} \\ y &= c + d\frac{a + bc}{1 - bd} \qquad (6) \end{aligned}$$

B

From the text we know that if $i = 2j$ for an integer j which means i is an even integer, then:

$$\mathcal{I} = \{i - 1\}$$

And if $i = 2j - 1$ which means i is an odd integer, then:

$$\mathcal{I} = \{i + 1\}$$

Thus, if we want to prove $\mathcal{I} = \{i - (-1)^i\}$, we just need to prove $\mathcal{I} = \{i - (-1)^{2j}\}$ for i being even and $\mathcal{I} = \{i - (-1)^{2j-1}\}$ for i being odd. Since we know that $(-1)^2 = 1$ and $(-1)^{2j} = [(-1)^2]^j$ and $(-1)^{2j-1} = [(-1)^2]^j/(-1)$, we can conclude that:

$$\begin{aligned} \mathcal{I} &= \{i - (-1)^{2j}\} \\ &= \{i - 1\} \end{aligned} \tag{7}$$

$$\begin{aligned} \mathcal{I} &= \{i - (-1)^{2j-1}\} \\ &= \{i + 1\} \end{aligned} \tag{8}$$

Equation 7 and 8 confirms the correctness of $\mathcal{I} = \{i - (-1)^i\}$.

C

According to the network model and the equation we proved in problem B, we can draw the conclusion that $\mid \mathcal{I}(i) = 1 \mid$ always stands. And by using $i - (-1)^i$ to substitute the k, we can draw the following equation:

$$\begin{aligned} y_i =& \beta_0 + \beta_1 (\frac{1}{\mid \mathcal{I}(i) \mid} \sum_{k \in \mathcal{I}(i)} y_k) + \mu_i \\ =& \beta_0 + \beta_1 y_{i-(-1)^i} + \mu_i \end{aligned} \tag{9}$$

To be able to use equation 5 and 6, we should use $i - (-1)^i$ to substitute i to create a new equation as follows:

$$y_{i-(-1)^i} = \beta_0 + \beta_1 y_{i-(-1)^i-(-1)^{i-(-1)^i}} + \mu_{i-(-1)^i} \tag{10}$$

Since the following two scenarios:

1. If i an odd integer, then $(-1)^i = -1$ and $i - (-1)^i$ is even which leads to the result that $(-1)^{i-(-1)^i}$ being 1 and $i - (-1)^i - (-1)^{i-(-1)^i}$ being i because the second term and third term are offset with each other.

2. If i an even integer, then $(-1)^i = 1$ and $i - (-1)^i$ is odd which leads to the result that $(-1)^{i-(-1)^i}$ being -1 and $i - (-1)^i - (-1)^{i-(-1)^i}$ being i because the second term and third term are offset with each other.

equation 10 can also be written as:

$$y_{i-(-1)^i} = \beta_0 + \beta_1 y_i + \mu_{i-(-1)^i} \tag{11}$$

Use equation 11 to substitute $y_{i-(-1)^i}$ in equation 9 we can get the following Equation:

$$\begin{aligned}
y_i =& \beta_0 + \beta_1 y_{i-(-1)^i} + \mu_i \\
=& \beta_0 + \beta_1(\beta_0 + \beta_1 y_i + \mu_{i-(-1)^i}) + \mu_i \\
=& \beta_0 + \beta_0\beta_1 + \beta_1^2 y_i + \beta_1\mu_{i-(-1)^i} + \mu_i \\
(1-\beta_1^2)y_i =& \beta_0 + \beta_0\beta_1 + \beta_1\mu_{i-(-1)^i} + \mu_i \\
(1+\beta_1)(1-\beta_1)y_i =& \beta_0(1+\beta_1) + \beta_1\mu_{i-(-1)^i} + \mu_i \\
y_i =& \frac{\beta_0}{1-\beta_1} + \frac{1}{1-\beta_1^2}(\mu_i + \beta_1\mu_{i-(-1)^i})
\end{aligned} \tag{12}$$

D

I

Using equation $y_i = \beta_0 + \beta_1 y_{i-(-1)^i} + \mu_i$ to simulate $y_1, y_2 ... y_n$ is nearly impossible since if we want to calculate y_1, we need to know y_2. If we would like to know y_2 then we need to know y_1, which is a dead loop.

II

- Line 1-6: Set the basic parameters for the afterward simulation.
- Line 7: Build the index vector so we can access both μ_i and $\mu_{i-(-1)^i}$ in each calculation.
- Line 8: Simulate 200 random normal numbers as the error terms.
- Line 9: "beta0/(1-beta1)" corresponds to the calculation of $\frac{\beta_0}{1-\beta_1}$. Using the iSeq as index we can assess both μ_i and $\mu_{i-(-1)^i}$ and "u[iSeq]+beta1*u[iSeq-(-1)^iSeq]" corresponds to $\mu_i + \beta_1\mu_{i-(-1)^i}$. "(1-beta1^2)" corresponds to $\frac{1}{1-\beta_1^2}$.

III

```
check <- NULL
y.calculated <- NULL
for (i in (1:n)) {
  check[i] <- (beta0+beta1*y[i-(-1)^i]+u[i]-y[i])<1e-10
  y_calculated[i] <- beta0+beta1*y[i-(-1)^i]+u[i]
}
sum(check)==n #True
plot(y.calculated,y,cex=1.5,pch=21,main = "Simulated y and calculated y",xlab="Y Calculated",ylab="Y Simulated")
abline(a=0,b=1,col="red",lwd=4)
# which verifies that every y satisfies the equation
```

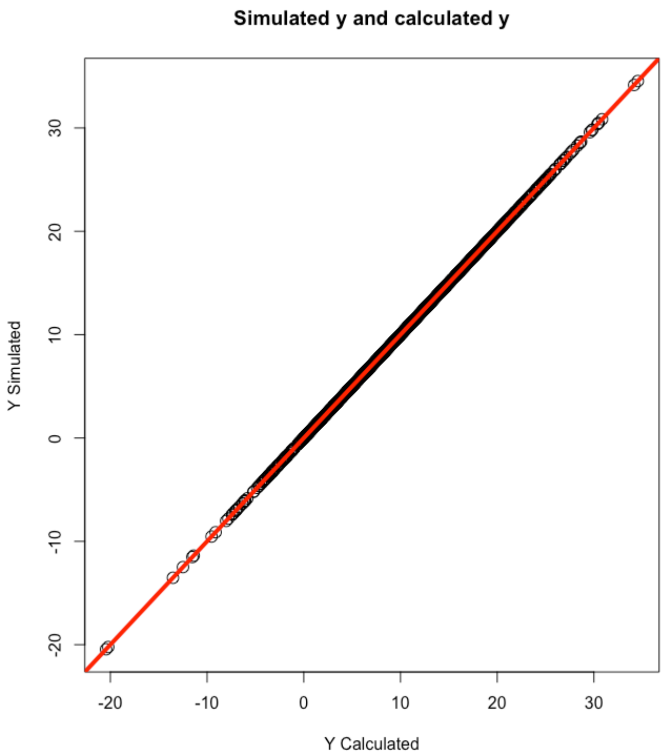


Figure 8: Scatter Plot of Cuba Male

E

I

Simulating 1000 times the sample mean of y and comparing it with μ shows that the sample mean of y is exactly the same as the population mean mu.

```
numRep=1000
m <- 100
n <- 2*m
beta0 = 1
beta1 = 0.9
sigma.u = 1
mu = beta0/(1-beta1)
mean.y.collect <- NULL
for (i in (1:numRep)) {
  iSeq <- (1:n)
  u <- rnorm(n, mean=0, sd=sigma.u)
  y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-
    beta1^2)
  mean.y.collect[i] <- mean(y)}
hist(mean.y.collect,main = "Histogram of y means",col = "white")
abline(v=mu,col="red",lwd=4)
```

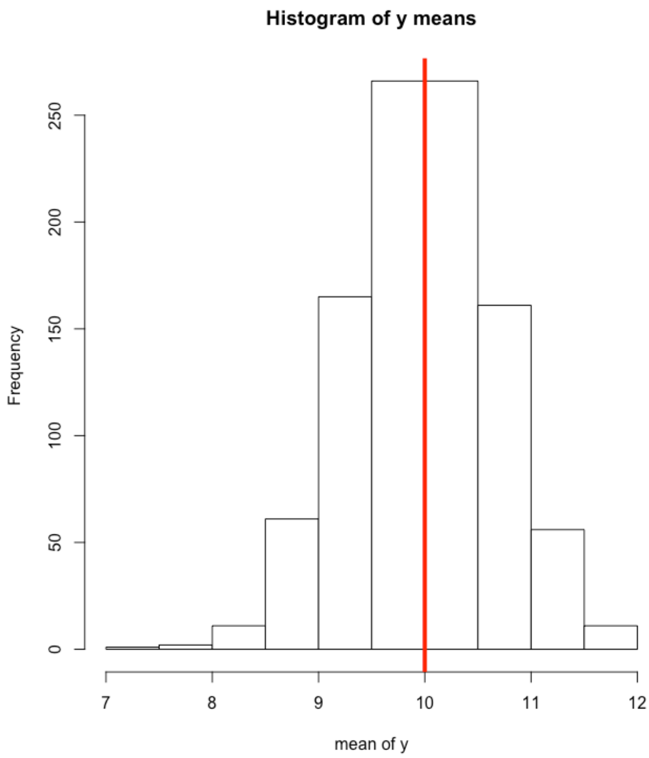


Figure 9: Histogram of y means

II

Simulating with 1000 repetitions, to approximate the coverage rate of the confidence interval, indicates a 83.7% coverage rate, which is indeed not close to 95%.

```
numRep=1000
m <- 100
n <- 2*m
beta0 = 1
beta1 = 0.9
sigma.u = 1
counter <- NULL
for (i in (1:numRep)) {
  iSeq <- (1:n)
  u <- rnorm(n, mean=0, sd=sigma.u)
  y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-
   beta1^2)
  lo.lim <- mean(y) - 1.96*sqrt(var(y))/sqrt(n)
  up.lim <- mean(y) + 1.96*sqrt(var(y))/sqrt(n)
  counter[i] <- (mu>lo.lim) & (mu<up.lim)
}
sum(counter)/length(counter) #0.837
```

Assignment 3

A

I

Let's first compute $\bar{x_n}$ first:

$$
\begin{aligned}
\bar{x_n} =& \frac{1}{n}\sum_{i=1}^{n} x_i \\
=& \frac{1}{n}\sum_{i=1}^{n}(\mu + \frac{1}{1-\beta_1}\mu_i) \\
=& \frac{1}{n}\sum_{i=1}^{n}\mu + \frac{1}{1-\beta_1}\frac{1}{n}\sum_{i=1}^{n}\mu_i \\
=& \mu + \frac{1}{1-\beta_1}\bar{\mu_i}
\end{aligned} \tag{13}
$$

Now, let's look at the $\bar{y_n}$. As the equation of y_i shown in equation 12, so the $\bar{y_n}$ could be formulated as follows:

$$
\begin{aligned}
\bar{y_n} =& \frac{1}{n}\sum_{i=1}^{n}(\frac{\beta_0}{1-\beta_1} + \frac{1}{1-\beta_1^2}(\mu_i + \beta_1\mu_{i-(-1)^i})) \\
=& \frac{1}{n}(\sum_{i=1}^{m}(\frac{\beta_0}{1-\beta_1} + \frac{1}{1-\beta_1^2}(\mu_{2i} + \beta_1\mu_{2i-(-1)^{2i}})) \\
&+ \sum_{i=1}^{m}(\frac{\beta_0}{1-\beta_1} + \frac{1}{1-\beta_1^2}(\mu_{2i-1} + \beta_1\mu_{2i-1-(-1)^{2i-1}}))) \\
=& \frac{1}{n}(\sum_{i=1}^{m}(\frac{\beta_0}{1-\beta_1} + \frac{1}{1-\beta_1^2}(\mu_{2i} + \beta_1\mu_{2i-1})) \\
&+ \sum_{i=1}^{m}(\frac{\beta_0}{1-\beta_1} + \frac{1}{1-\beta_1^2}(\mu_{2i-1} + \beta_1\mu_{2i}))) \\
=& \frac{1}{n}((2m\frac{\beta_0}{1-\beta_1}) + \frac{1}{1-\beta_1^2}\sum_{i=1}^{m}(\mu_{2i} + \beta_1\mu_{2i-1} + \mu_{2i-1} + \beta_1\mu_{2i})) \\
=& \frac{1}{n}((n\frac{\beta_0}{1-\beta_1}) + \frac{1}{1-\beta_1^2}\sum_{i=1}^{m}(\mu_{2i}(1+\beta_1) + \mu_{2i-1}(1+\beta_1))) \\
=& \frac{\beta_0}{1-\beta_1} + \frac{1}{n}\frac{1+\beta_1}{(1-\beta_1)(1+\beta_1)}\sum_{i=1}^{m}(\mu_{2i} + \mu_{2i-1}) \\
=& \frac{\beta_0}{1-\beta_1} + \frac{1}{n}\frac{1}{1-\beta_1}\sum_{i=1}^{n}\mu_i \\
=& \frac{\beta_0}{1-\beta_1} + \frac{1}{n}\frac{n}{1-\beta_1}\frac{\sum_{i=1}^{n}\mu_i}{n} \\
=& \frac{\beta_0}{1-\beta_1} + \frac{1}{1-\beta_1}\bar{\mu_i}
\end{aligned}
$$

Since we already know that $\mu = \frac{\beta_0}{1-\beta_1}$, thus the final equation of $\bar{y_n}$ can be written as:

$$\bar{y_n} = \mu + \frac{1}{1-\beta_1}\bar{\mu_i} \tag{14}$$

By comparing equation 13 and 14 we can draw the conclusion that $\bar{x_n} = \bar{y_n}$.

II

In order to determine whether the means of y and the mean of x are equal. We simulate 1000 times whether the difference between the mean of x and the mean of y is less than 1e-10 (an extremely small number). Comparing the True and False values with our number of simulations gives us positive feedback. The difference between the mean of x and the mean of y is less than 1e-10 at 100% of our simulation. Thus, we can conclude that the mean of x and the mean of y is approximately equal.

```
m <- 1000
n <- 2*m
numRep <- 1000
beta0 <- 1
beta1 <- 0.9
sigma.u <- 1
mu <- beta0/(1-beta1)
xy.collect <- NULL
for (i in (1:numRep)) {
  iSeq <- (1:n)
  u <- rnorm(n, mean=0, sd=sigma.u)
  y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-
    beta1^2)
  x <- mu + 1/(1-beta1)*u
  xy.collect[i] <- (mean(x)-mean(y))<1e-10
}
sum(xy.collect) #1000 which equals to the length of xy.collect, so
    for all the 1000 simulations, mean(x)=mean(y)
```

B

I

Looking at the plot (Figure 10), one can see that the density curve precisely follows the histogram, which justifies that the equation holds.

```
m <- 1000
n <- 2*m
numRep <- 5000
beta0 <- 1
beta1 <- 0.9
sigma.u <- 1
mu <- beta0/(1-beta1)
```

```
mean_collect=NULL
for (i in (1:numRep)) {
  iSeq <- (1:n)
  u <- rnorm(n, mean=0, sd=sigma.u)
  y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-beta1^2)
  mean_collect[i] <- mean(y)}
hist(mean_collect,freq = F,main = "Histogram of the means of y", col = "white",xlab = "mean of y")
x.plot<-seq(mu -3*sqrt((sigma.u/(1-beta1))^2/n),mu + 3*sqrt((sigma.u/(1-beta1))^2/n),length.out = 400)
y.plot<-(1/(sqrt(2*pi)*(sqrt((sigma.u/(1-beta1))^2/n))))*exp(-0.5*(x_plot - mu)^2/((sigma.u/(1-beta1))^2/n))
points(x.plot,y.plot, type="l", col="red",lwd=4)
```

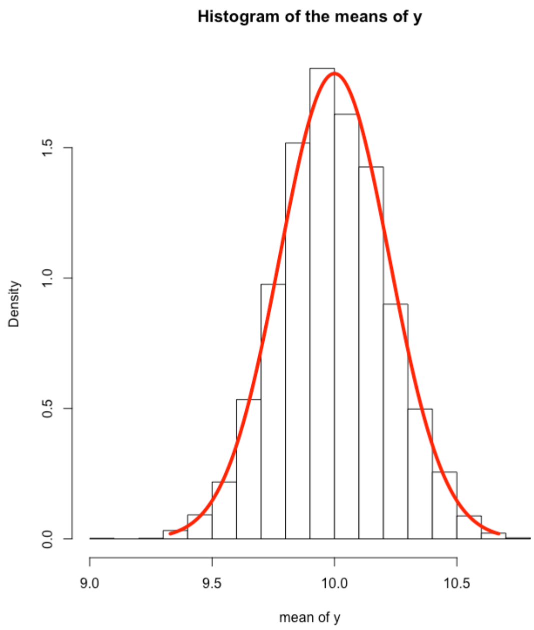


Figure 10: Histogram of y means

II

We have concluded that:

$$\bar{y_n} \sim N(\mu, [\sigma_u/(1-\beta_1)]^2 \frac{1}{n})$$

Since we know that for a normal distribution $X \sim N(\mu, \sigma^2)$, we have that:

$$Pr(\mu - 1.96\sigma \leq X \leq \mu + 1.96\sigma) = 95\%$$

Through some transformation we can get that:

$$Pr(X - 1.96\sigma \leq \mu \leq X + 1.96\sigma) = 95\% \tag{15}$$

In our case, we have that:

$$X = \bar{y_n} \tag{16}$$
$$\sigma = \frac{1}{\sqrt{n}} \frac{\sigma_u}{1-\beta_1} \tag{17}$$

Substitute the X and μ in equation 15 with equation 16 and 17 so we can get that:

$$Pr(\bar{y_n} - 1.96\frac{1}{\sqrt{n}}\frac{\sigma_u}{1-\beta_1} \leq \mu \leq \bar{y_n} + 1.96\frac{1}{\sqrt{n}}\frac{\sigma_u}{1-\beta_1}) = 95\% \tag{18}$$

III

```
m <- 1000
n <- 2*m
numRep <- 1000
beta0 <- 1
beta1 <- 0.9
sigma.u <- 1
mu <- beta0/(1-beta1)
interval_collect=NULL
for (i in (1:numRep)) {
  iSeq <- (1:n)
  u <- rnorm(n, mean=0, sd=sigma.u)
  y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-
    beta1^2)
  up.bound <- mean(y)+1.96/sqrt(n)*sigma.u/(1-beta1)
  low.bound <- mean(y)-1.96/sqrt(n)*sigma.u/(1-beta1)
  interval_collect[i] <- (mu>low.bound)&(mu<up.bound)}
sum(interval_collect)/length(interval_collect) # 0.942, the result
     is around 0.95
```

C

I

Verifying via summation whether both sides of the equality are equal. Calculating each term, we notice that the sum of the terms (50.19136) is really close to the variance of s_y^2 (50.16381).

```
set.seed(4110)
m <- 5000000
n <- 2*m
iSeq <- (1:n)
beta0 <- 1
beta1 <- 0.9
sigma.u <- 1
u <- rnorm(n, mean=0, sd=sigma.u)
y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-beta1
    ^2)
```

```
term1 <- ((1+beta1^2)/(1-beta1^2)^2)*(1/(n-1))*sum(u^2)
term2 <- 0
for (i in (1:m)) {
  term2 = term2+(1/(n-1))*u[2*i]*u[(2*i-1)]
}
term2<-4*beta1/((1-beta1)^2)*term2
term3 <- 1/((1-beta1)^2)*n/(n-1)*mean(u)^2
var(y) #50.16381
(term1+term2-term3) #50.19136
# From the two values above we can see that they are not exactly
  the same, but pretty close.
```

II

Simulatting and comparing it with $\frac{1+\beta_1^2}{(1-\beta_1^2)^2}\sigma_\mu^2$ one can see that it is pretty close to s_y^2 50.1385 , although it is far from $\sigma_\mu^2/(1-\beta_1)^2$ which equals to 100.

```
set.seed(4110)
m <- 5000000
n <- 2*m
beta0 <- 1
beta1 <- 0.9
sigma.u <- 1
mu <- beta0/(1-beta1)
iSeq <- (1:n)
u <- rnorm(n, mean=0, sd=sigma.u)
y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-beta1
  ^2)
var(y) #50.16381
(1+beta1^2)/(1-beta1^2)^2*sigma.u^2 #50.1385
sigma.u^2/(1-beta1)^2 #100
# From the three values above we can see that var(y) is close to
  (1+beta1^2)/(1-beta1^2)^2*sigma.u^2 which is 50.1385 but not
  even close to sigma.u^2/(1-beta1)^2 which is 100.
```

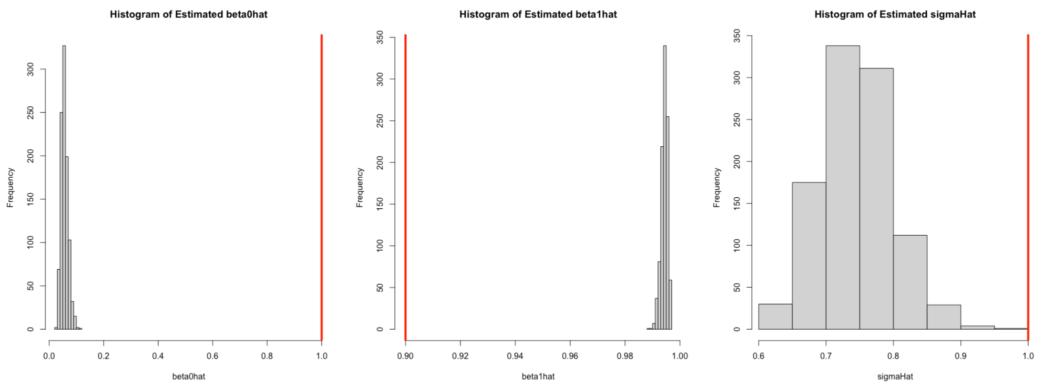
D

I

Looking at the Plot of β_0, β_1 and also the σ_μ estimates (Figure 11). The estimates of β_0 are pretty far from the actual value (represented by a vertical red line). The same applies to β_1 and σ_μ as well. Hence, it is indeed naive and far from reality estimation.

```
numRep=1000
m <- 100
n <- 2*m
beta0 <- 1
```

```
beta1 <- 0.9
sigma.u <- 1
mu <- beta0/(1-beta1)
beta0hat_collect=NULL
beta1hat_collect=NULL
sd.u.hat_collect=NULL
for (i in (1:numRep)) {
  iSeq <- (1:n)
  u <- rnorm(n, mean=0, sd=sigma.u)
  y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-beta1^2)
  lmfit <- lm(y[iSeq] ~ y[iSeq-(-1)^iSeq])
  beta0hat <- lmfit$coefficients[1]
  beta1hat <- lmfit$coefficients[2]
  sd.u.hat <- sigma(lmfit)
  beta1hat_collect[i] <- beta1hat
  beta0hat_collect[i] <- beta0hat
  sd.u.hat_collect[i] <- sd.u.hat
}
hist(beta0hat_collect,xlim = c(min(beta0hat_collect),1),col="white",main = "Histogram of Estimated beta0hat",xlab = "beta0hat")
abline(v=beta0,col="red",lwd=4)
hist(beta1hat_collect,xlim = c(0.9,max(beta1hat_collect)),col="white",main = "Histogram of Estimated beta1hat",xlab = "beta1hat")
abline(v=0.9,col="red",lwd=4)
hist(sd.u.hat_collect,xlim = c(min(sd.u.hat_collect),1),col="white",main = "Histogram of Estimated sigmaHat",xlab = "sigmaHat")
abline(v=sigma.u,col="red",lwd=4)
```



(a) beta0hat (b) beta1hat (c) sigmaHat

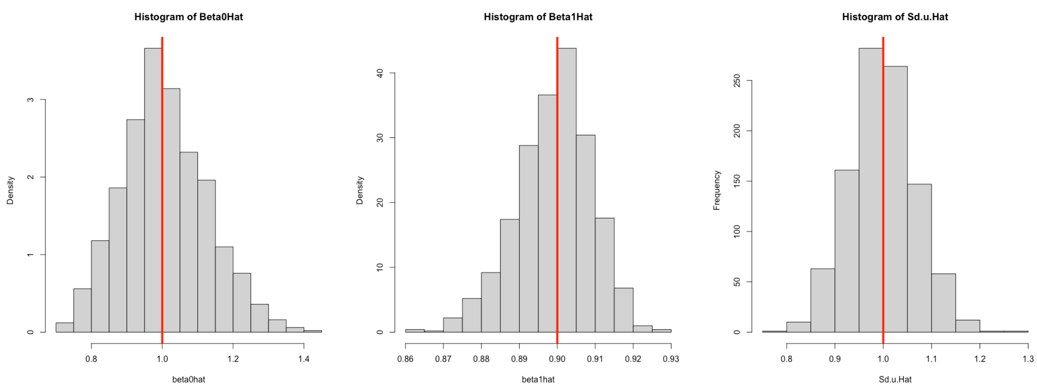
Figure 11

II

In this case, the histograms (Figure 12) confirm that estimates for β_0, β_1, and σ_μ are all approximately reasonable estimates. The red vertical line indicates the actual β_0,

β_1, σ_μ respectively. One can further confirm it by looking at the mean of estimates. All of them are really close to the actual values. Hence we can verify that they are much better estimates than what we have seen previously.

```r
numRep=1000
m <- 100
n <- 2*m
beta0 <- 1
beta1 <- 0.9
sigma.u <- 1
mu <- beta0/(1-beta1)
beta1hat_collect=NULL
sd.u.hat_collect=NULL
for (i in (1:numRep)) {
  iSeq <- (1:n)
  u <- rnorm(n, mean=0, sd=sigma.u)
  y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-
    beta1^2)
  ratio <- var(y)/var(y[2*(1:m)] - y[2*(1:m)-1])
  beta1hat <- (2*ratio-sqrt(4*ratio-1))/(2*ratio-1)
  beta0hat <- mean(y)*(1-beta1hat)
  #extract residuals:
  u.hat <- y[iSeq] - (beta0hat + beta1hat*y[iSeq-(-1)^iSeq])
  sd.u.hat <- sd(u.hat)
  beta1hat_collect[i] <- beta1hat
  sd.u.hat_collect[i] <- sd.u.hat
}
hist(beta1hat_collect,freq = F,main = "Histogram of Beta1Hat")
abline(v=beta1,col="red",lwd=4)
hist(sd.u.hat_collect,main = "Histogram of Sd.u.Hat",xlab = "Sd.u.
    Hat")
abline(v=sigma.u,col="red",lwd=4)
```



(a) beta0hat (b) beta1hat (c) sigmaHat

Figure 12

III

Simulating 10^4 times whether the upper limit of the confidence interval is larger than μ, at the same time, the lower limit is smaller than μ. In 95% of the cases, it is true.

```
numRep=10000
m <- 100
beta0 = 1
n <- 2*m
beta1 = 0.9
sigma.u = 1
mu <- beta0/(1-beta1)
confinterval_collect=NULL
for (i in (1:numRep)) {
  iSeq <- (1:n)
  u <- rnorm(n, mean=0, sd=sigma.u)
  y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-
    beta1^2)
  ratio <- var(y)/var(y[2*(1:m)] - y[2*(1:m)-1])
  beta1hat <- (2*ratio-sqrt(4*ratio-1))/(2*ratio-1)
  beta0hat <- mean(y)*(1-beta1hat)
  #extract residuals:
  u.hat <- y[iSeq] - (beta0hat + beta1hat*y[iSeq-(-1)^iSeq])
  sd.u.hat <- sd(u.hat)
  up.bound <- mean(y)+1.96/sqrt(n)*sd.u.hat/(1-beta1hat)
  low.bound <- mean(y)-1.96/sqrt(n)*sd.u.hat/(1-beta1hat)
  confinterval_collect[i] <- (mu>low.bound)&(mu<up.bound)
}
sum(confinterval_collect)/length(confinterval_collect) #0.9438
    which is around 95%
```

Appendix

```
rm(list=ls())

# Assignment 1
## (A)
### (I)
data <- read.csv("ReplicationProcessedfinaldata04202018.csv")
miss <- data.frame(columns=colnames(data),missing.num=NA)
for (column in miss$columns){
  miss[miss$columns==column,2] <- sum(is.na(data[,column]))
}
rows.before <- nrow(data)
miss
### (II)
data <- data[-c(which((data$mate_age<12)|(is.na(data$mate_age))))
    ,]
rows.after <- nrow(data)
proportion.removed <- (rows.before-rows.after)/rows.before
proportion.removed #0.401764
### (III)
countries <- data.frame(country.name=unique(data$country),
    nCountries=NA)
for (name in countries$country.name) {
  countries[countries$country.name==name,2]=sum(data$country==name
    )
}
countries <- countries[-c(which(countries$nCountries<350)),]

## (B)
### (I)
### NB: sex = 0 are women, sex = 1 are men
with(data, plot(age, mate_age, cex = 1.2, pch = 21, bg=(sex+2),
    main="Male and Female"))
legend("topleft",c("Female","Male"),fill=c(2,3))
with(data[data$sex == 0,], lines(smooth.spline(age, mate_age), col
     = 2,lwd=5))
with(data[data$sex == 1,], lines(smooth.spline(age, mate_age), col
     = 3,lwd=5))

i <- 0
with(data[data$sex == i,], plot(age, mate_age, cex = 1.2, pch =
    21,bg=2,main="Female"))
with(data[data$sex == i,], lines(smooth.spline(age, mate_age), col
     = "lightgreen",lwd=5))
lmFemale <- lm(mate_age ~ age, data = data[data$sex== i,])
summary(lmFemale)
lmFemaleQuadratic <- lm(mate_age ~ age + I(age^2), data = data[
    data$sex == i,])
```

```
summary(lmFemaleQuadratic)
#quadratic term not significant.
#Let's consider the fit of the simple linear regression:
x <- seq(min(data[data$sex == i,]$age), max(data[data$sex == i,]$age), length.out=400)
betahat <- lmFemale$coefficients
points(x, betahat[1]+betahat[2]*x, lwd=5, lty=2, col="black",type="l")
abline(a=0,b=1,col="orange",lwd=5)
legend("topleft",c("Smooth Spline","Simple Linear","mate_age=age"),lty = c(1,2,1),col=c("lightgreen","black","orange"),lwd=c(3,3,3))
#perfect match

i <- 1
lmMale <- lm(mate_age ~ age, data = data[data$sex== i,])
summary(lmMale)
lmMaleQuadratic <- lm(mate_age ~ age + I(age^2), data = data[data$sex == i,])
summary(lmMaleQuadratic) #quadratic term seems quite significant.
with(data[data$sex == i,],cor(age,age^2)) # 0.9861473
# the correlation between the age and quadratic age is quite large, so we cannot trust the p-value easily
#Let's further try the cubic term
lmMaleCubic <- lm(mate_age ~ age + I(age^2) + I(age^3), data = data[data$sex == i,])
summary(lmMaleCubic)
#In this case the linear term is not significant anymore, to figure out whether we should use the quadratic term or the cubic term,
#we will visually assess which one is better
with(data[data$sex == i,], plot(age, mate_age, cex = 1.2, pch = 21,bg=3,main="Male"))
with(data[data$sex == i,], lines(smooth.spline(age, mate_age), col = "red",lwd=5))
# plot the quadratic line
x <- seq(min(data[data$sex == i,]$age), max(data[data$sex == i,]$age), length.out=400)
betahat <- lmMaleQuadratic$coefficients
points(x, betahat[1]+betahat[2]*x+betahat[3]*x^2, lwd=5, col="black",type="l")
x <- seq(min(data[data$sex == i,]$age), max(data[data$sex == i,]$age), length.out=400)
betahat <- lmMaleCubic$coefficients
points(x, betahat[1]+betahat[2]*x+betahat[3]*x^2+betahat[4]*x^3, lwd=5, col="skyblue",type="l")
abline(a=0,b=1,col="orange",lwd=5)
abline(lmMale,col="purple",lwd=5)
legend("topleft",c("Smooth Spline","Quadratic Linear","Cubic
```

```r
Linear","Simple Linear","mate_age=age"),col=c("red","black","skyblue","purple","orange"),lwd=c(3,3,3,3))
# Seems like the cubic estimation is not as good as the quadratic estimation, so let's go for the quadratic term

#Let's consider the fit of the quadratic linear regression:
with(data[data$sex == i,], plot(age, mate_age, cex = 1.2, pch = 21,bg=3))
with(data[data$sex == i,], lines(smooth.spline(age, mate_age), col = "red",lwd=5))
x <- seq(min(data[data$sex == i,]$age), max(data[data$sex == i,]$age), length.out=400)
betahat <- lmMaleQuadratic$coefficients
points(x, betahat[1]+betahat[2]*x+betahat[3]*x^2, lwd=3, lty=2, col="black",type="l")
abline(a=0,b=1,col="skyblue",lwd=5)
legend("topleft",c("Smooth Spline","Quadratic Estimate","mate_age=age"),lty = c(1,2,1),col=c("red","black","skyblue"),lwd=c(3,3,3))
# perfect match

# For the full data, this motivates the following model
lm.joint <- with(data,lm(mate_age ~ age + sex*age + sex*I(age^2)-I(age^2)))
summary(lm.joint)

### (II)
cuba.both <- data[data$country=="Cuba",]
cuba.male <- cuba.both[cuba.both$sex==1,]
cuba.female <- cuba.both[cuba.both$sex==0,]
lm.cuba <- with(cuba.both,lm(mate_age~age))
lm.cuba.male <- with(cuba.male,lm(mate_age~age))
lm.cuba.female <- with(cuba.female,lm(mate_age~age))
with(cuba.both,plot(age, mate_age, cex = 1.2, pch = 21,bg="orange",main="Cuba Male and Female"))
abline(lm.cuba,col="blue",lwd=5)
abline(a=0,b=1,col="red",lwd=5)
legend("topleft",c("Simple linear regression","mate_age=age"),lty=c(1,1),col=c("blue","red"),lwd=c(3,3))
with(cuba.male,plot(age, mate_age, cex = 1.2, pch = 21,bg="skyblue",main="Cuba Male"))
legend("topleft",c("Simple linear regression","mate_age=age"),lty=c(1,1),col=c("blue","red"),lwd=c(3,3))
abline(lm.cuba.male,col="blue",lwd=5)
abline(a=0,b=1,col="red",lwd=5)
with(cuba.female,plot(age, mate_age, cex = 1.2, pch = 21,bg="yellow",main="Cuba Female"))
abline(lm.cuba.female,col="blue",lwd=5)
abline(a=0,b=1,col="red",lwd=5)
```

```
legend("topleft",c("Simple linear regression","mate_age=age"),lty=c(1,1),col=c("blue","red"),lwd=c(3,3))

lm.cuba.null <- with(cuba.both, lm(mate_age ~ offset(1*age) - 1))
lm.cuba.null.male <- with(cuba.male, lm(mate_age ~ offset(1*age) - 1))
lm.cuba.null.female <- with(cuba.female, lm(mate_age ~ offset(1*age) - 1))

anova1 <- anova(lm.cuba,lm.cuba.null)
anova2 <- anova(lm.cuba.male,lm.cuba.null.male)
anova3 <- anova(lm.cuba.female,lm.cuba.null.female)
max(residuals(lm.cuba.null)- with(cuba.both, mate_age - age)) == 0

### (III)
five_country <- data[(data$country=="Hungary") | (data$country=="Pakistan")|(data$country=="Poland")|(data$country=="Slovenia")|(data$country=="Turkey"),]
five_country$Hungary <- (five_country$country=="Hungary")*1
five_country$Pakistan <- (five_country$country=="Pakistan")*1
five_country$Poland <- (five_country$country=="Poland")*1
five_country$Slovenia <- (five_country$country=="Slovenia")*1
five_country$Turkey <- (five_country$country=="Turkey")*1
with(five_country, plot(age,mate_age, cex = 1, pch = 20, col=(1*Hungary+2*Pakistan+3*Poland+4*Slovenia+7*Turkey),main="Scatter Plot of 5 Countries"))
with(five_country[five_country$Hungary == 1,], lines(smooth.spline(age, mate_age), col = (1*Hungary+2*Pakistan+3*Poland+4*Slovenia+7*Turkey),lwd=20))
with(five_country[five_country$Pakistan == 1,], lines(smooth.spline(age, mate_age), col = (1*Hungary+2*Pakistan+3*Poland+4*Slovenia+7*Turkey),lwd=5))
with(five_country[five_country$Poland == 1,], lines(smooth.spline(age, mate_age), col = (1*Hungary+2*Pakistan+3*Poland+4*Slovenia+7*Turkey),lwd=2))
with(five_country[five_country$Slovenia == 1,], lines(smooth.spline(age, mate_age), col = (1*Hungary+2*Pakistan+3*Poland+4*Slovenia+7*Turkey),lwd=4))
with(five_country[five_country$Turkey == 1,], lines(smooth.spline(age, mate_age), col = (1*Hungary+2*Pakistan+3*Poland+4*Slovenia+7*Turkey),lwd=2))
legend("topleft",c("Hungary","Pakistan","Poland","Slovenia","Turkey"),fill=c(1,2,3,4,7))

# For Hungary, Pakistan and Slovenia together
threeCountry = five_country[(five_country$country!="Poland")&(five_country$country!="Turkey"),]
with(threeCountry, plot(age, mate_age, cex = 1.2, pch = 21, main="Three Countries"))
```

```
with(threeCountry, lines(smooth.spline(age, mate_age),col = (1*Hungary+2*Pakistan+4*Slovenia),lwd=5))
lmThree <- lm(mate_age ~ age, data = threeCountry)
summary(lmThree)
lmThreeQuadratic <- lm(mate_age ~ age + I(age^2), data = threeCountry)
summary(lmThreeQuadratic)
lmThreeCubic <- lm(mate_age ~ age + I(age^2) +I(age^3), data = threeCountry)
summary(lmThreeCubic)
with(threeCountry,cor(age,age^2))
with(threeCountry,cor(age,age^3))
with(threeCountry,cor(age^2,age^3))
betahat<-lmThree$coefficients
#quadratic and cubic term are significant.
#However, the correlations among these three terms are extremely high, so we should be carefully with the quadratic and cubic terms
#Let's plot the three linear lines to see which one fits the best.
x <- seq(min(threeCountry$age), max(threeCountry$age), length.out=400)
with(threeCountry,points(x, betahat[1]+betahat[2]*x, lwd=5, col="red",type="l"))
betahatQua <- lmThreeQuadratic$coefficients
with(threeCountry,points(x, betahatQua[1]+betahatQua[2]*x+betahatQua[3]*x^2, lwd=5, col="blue",type="l"))
betahatCub <- lmThreeCubic$coefficients
with(threeCountry,points(x, betahatCub[1]+betahatCub[2]*x+betahatCub[3]*x^2+betahatCub[4]*x^3, lwd=5, col="green",type="l"))
legend("topleft", legend=c("Smooth", "Simple Linear","Quadratic Linear","Cubic Linear"),lty = c(1,1,1,), col=c("black", "red","blue","green"), lwd=c(3,3,3,3))
# Seems like the simple linear fits the best, so let's go for it

# For Poland
with(five_country[five_country$Poland == 1,], plot(age, mate_age, cex = 1.2, pch = 21,bg=(1*Hungary+2*Pakistan+3*Poland+4*Slovenia+5*Turkey), main="Poland"))
with(five_country[five_country$Poland == 1,], lines(smooth.spline(age, mate_age), col = (1*Hungary+2*Pakistan+3*Poland+4*Slovenia+5*Turkey+1),lwd=5))
lmPoland <- lm(mate_age ~ age, data = five_country[five_country$Poland== 1,])
summary(lmPoland)
lmPolandQuadratic <- lm(mate_age ~ age + I(age^2), data = five_country[five_country$Poland == 1,])
summary(lmPolandQuadratic)
lmPolandCubic <- lm(mate_age ~ age + I(age^2) +I(age^3), data =
```

```
    five_country[five_country$Poland == 1,])
summary(lmPolandCubic)
#quadratic and cubic term are not significant.
#Let's consider the fit of the simple linear regression:
x <- seq(min(five_country[five_country$Poland == 1,]$age), max(
    five_country[five_country$Poland == 1,]$age), length.out=400)
betahat <- lmPoland$coefficients
with(five_country,points(x, betahat[1]+betahat[2]*x, lwd=5, lty=2,
     col="red",type="l"))
#perfect match
legend("topleft", legend=c("Smooth", "Simple Linear"),lty = c(1,2)
    , col=c(4, "red"), lwd=c(3,3))

# For Turkey
with(five_country[five_country$Turkey == 1,], plot(age, mate_age,
    cex = 1.2, pch = 21,bg=(1*Hungary+2*Pakistan+3*Poland+4*
    Slovenia+5*Turkey), main="Turkey"))
with(five_country[five_country$Turkey == 1,], lines(smooth.spline(
    age, mate_age), col = (1*Hungary+2*Pakistan+3*Poland+4*Slovenia
    +5*Turkey+1),lwd=5))
lmTurkey <- lm(mate_age ~ age, data = five_country[five_country$
    Turkey== 1,])
summary(lmTurkey)
lmTurkeyQuadratic <- lm(mate_age ~ age + I(age^2), data = five_
    country[five_country$Turkey == 1,])
summary(lmTurkeyQuadratic)
lmTurkeyCubic <- lm(mate_age ~age+ I(age^2) +I(age^3), data = five
    _country[five_country$Turkey == 1,])
summary(lmTurkeyCubic)
#Let's consider the fit of the cubic linear regression:
x <- seq(min(five_country[five_country$Turkey == 1,]$age), max(
    five_country[five_country$Turkey == 1,]$age), length.out=400)
betahat.quadratic <- lmTurkeyQuadratic$coefficients
with(five_country[five_country$Turkey == 1,],points(x, betahat.
    quadratic[1]+betahat.quadratic[2]*x+betahat.quadratic[3]*x^2,
    lwd=5, lty=1, col="blue",type="l"))
betahat.cubic <- lmTurkeyCubic$coefficients
with(five_country[five_country$Turkey == 1,],points(x, betahat.
    cubic[1]+betahat.cubic[2]*x+betahat.cubic[3]*x^2+betahat.cubic
    [4]*x^3, lwd=5, lty=1, col="green",type="l"))
legend("topleft", legend=c("Smooth", "Quadratic Linear","Cubic
    Linear"), col=c(6,"blue","green"), lwd=c(3,3,3))

# add the interaction terms
# the linear age term of Turkey Cubic regression is not
    significant, here we don't include the simple linear
    interaction term for Turkey (Turkey*age)
# since it is not clear whether we need the interaction term for
    Poland (Poland*age)
```

```r
# we first run the model without the interaction term
lm1 <- with(five_country,lm(mate_age~age+Poland+Turkey+Turkey*I(
    age^2)+Turkey*I(age^3)-I(age^3)-I(age^2)))
summary(lm1)
lm2 <- with(five_country,lm(mate_age~age+Poland+Poland*age+Turkey+
    Turkey*I(age^2)+Turkey*I(age^3)-I(age^3)-I(age^2)))
summary(lm2)
stargazer(lm2,lm1)
anova(lm2,lm1)
# seems like perfect


# Assignment 2
## (A) See the main body
## (B) See the main body
## (C) See the main body
## (D)
set.seed(4110)
beta0 <- 1
beta1 <- 0.9
sigma.u <- 1
m <- 1000
n <- 2*m
iSeq <- (1:n)
u <- rnorm(n, mean=0, sd=sigma.u)
y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-beta1
    ^2)
### (I)
### (II)
### (III)
check <- NULL
y.calculated <- NULL
for (i in (1:n)) {
  check[i] <- (beta0+beta1*y[i-(-1)^i]+u[i]-y[i])<1e-10
  y.calculated[i] <- beta0+beta1*y[i-(-1)^i]+u[i]
}
sum(check)==n
plot(y.calculated,y,cex=1.5,pch=21,main = "Simulated y and
    calculated y",xlab="Y Calculated",ylab="Y Simulated")
abline(a=0,b=1,col="red",lwd=4)
# which verifies that every y satisfies the equation

## (E)
### (I)
numRep=1000
m <- 100
n <- 2*m
beta0 = 1
beta1 = 0.9
```

```r
sigma.u = 1
mu = beta0/(1-beta1)
mean.y.collect <- NULL
for (i in (1:numRep)) {
  iSeq <- (1:n)
  u <- rnorm(n, mean=0, sd=sigma.u)
  y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-
    beta1^2)
  mean.y.collect[i] <- mean(y)
}
hist(mean.y.collect,main = "Histogram of y means",col = "white",
    xlab = "mean of y")
abline(v=mu,col="red",lwd=4)
### (II)
numRep=1000
m <- 100
n <- 2*m
beta0 = 1
beta1 = 0.9
sigma.u = 1
counter <- NULL
for (i in (1:numRep)) {
  iSeq <- (1:n)
  u <- rnorm(n, mean=0, sd=sigma.u)
  y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-
    beta1^2)
  lo.lim <- mean(y) - 1.96*sqrt(var(y))/sqrt(n)
  up.lim <- mean(y) + 1.96*sqrt(var(y))/sqrt(n)
  counter[i] <- (mu>lo.lim) & (mu<up.lim)
}
sum(counter)/length(counter) #0.837

# Assignment 3
## (A)
### (I) See the main body
### (II)
m <- 1000
n <- 2*m
numRep <- 1000
beta0 <- 1
beta1 <- 0.9
sigma.u <- 1
mu <- beta0/(1-beta1)
xy.collect <- NULL
for (i in (1:numRep)) {
  iSeq <- (1:n)
  u <- rnorm(n, mean=0, sd=sigma.u)
  y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-
    beta1^2)
```

```
  x <- mu + 1/(1-beta1)*u
  xy.collect[i] <- (mean(x)-mean(y))<1e-10
}
sum(xy.collect) #10000 which equals to the length of xy.collect, so for all the 1000 simulations, mean(x)=mean(y)
## (B)
### (I)
m <- 1000
n <- 2*m
numRep <- 5000
beta0 <- 1
beta1 <- 0.9
sigma.u <- 1
mu <- beta0/(1-beta1)
mean_collect=NULL
for (i in (1:numRep)) {
  iSeq <- (1:n)
  u <- rnorm(n, mean=0, sd=sigma.u)
  y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-beta1^2)
  mean_collect[i] <- mean(y)
}

hist(mean_collect,freq = F,main = "Histogram of the means of y", col = "white",xlab = "mean of y")
x.plot<-seq(mu -3*sqrt((sigma.u/(1-beta1))^2/n),mu + 3*sqrt((sigma.u/(1-beta1))^2/n),length.out = 400)
y.plot<-(1/(sqrt(2*pi)*(sqrt((sigma.u/(1-beta1))^2/n))))*exp(-0.5*(x_plot - mu)^2/((sigma.u/(1-beta1))^2/n))
points(x.plot,y.plot, type="l", col="red",lwd=4)

### (II) See the main body
### (III)
m <- 1000
n <- 2*m
numRep <- 1000
beta0 <- 1
beta1 <- 0.9
sigma.u <- 1
mu <- beta0/(1-beta1)
interval_collect=NULL
for (i in (1:numRep)) {
  iSeq <- (1:n)
  u <- rnorm(n, mean=0, sd=sigma.u)
  y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-beta1^2)
  up.bound <- mean(y)+1.96/sqrt(n)*sigma.u/(1-beta1)
  low.bound <- mean(y)-1.96/sqrt(n)*sigma.u/(1-beta1)
  interval_collect[i] <- (mu>low.bound)&(mu<up.bound)
```

```
}
sum(interval_collect)/length(interval_collect) # 0.942, the result is around 0.95
## (C)
### (I)
set.seed(4110)
m <- 5000000
n <- 2*m
iSeq <- (1:n)
beta0 <- 1
beta1 <- 0.9
sigma.u <- 1
u <- rnorm(n, mean=0, sd=sigma.u)
y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-beta1^2)
term1 <- (1+beta1^2)/(1-beta1^2)^2*(1/(n-1))*sum(u)^2
term2 <- 0
for (i in (1:m)) {
  term2 = term2+u[2*i]*u[(2*i-1)]
}
term2<-4*(1/(n-1))*beta1/((1-beta1)^2)*term2
term3 <- 1/((1-beta1)^2)*n/(n-1)*(mean(u))^2
var(y) #50.16381
(term1+term2-term3) #50.19136

# From the two values above we can see that they are not exactly the same, but pretty close.(mean(u)^2)
### (II)
set.seed(4110)
m <- 10000000
n <- 2*m
beta0 <- 1
beta1 <- 0.9
sigma.u <- 1
mu <- beta0/(1-beta1)
iSeq <- (1:n)
u <- rnorm(n, mean=0, sd=sigma.u)
y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-beta1^2)
var(y) #50.14622
(1+beta1^2)/(1-beta1^2)^2*sigma.u^2 #50.1385
sigma.u^2/(1-beta1)^2 #100
# From the three values above we can see that var(y) is close to (1+beta1^2)/(1-beta1^2)^2*sigma.u^2 which is 50.1385 but not even close to
# sigma.u^2/(1-beta1)^2 which is 100.

## (D)
### (I)
```

```
numRep=1000
m <- 100
n <- 2*m
beta0 <- 1
beta1 <- 0.9
sigma.u <- 1
mu <- beta0/(1-beta1)
beta0hat_collect=NULL
beta1hat_collect=NULL
sd.u.hat_collect=NULL
for (i in (1:numRep)) {
  iSeq <- (1:n)
  u <- rnorm(n, mean=0, sd=sigma.u)
  y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-
    beta1^2)
  lmfit <- lm(y[iSeq] ~ y[iSeq-(-1)^iSeq])
  beta0hat <- lmfit$coefficients[1]
  beta1hat <- lmfit$coefficients[2]
  sd.u.hat <- sigma(lmfit)
  beta1hat_collect[i] <- beta1hat
  beta0hat_collect[i] <- beta0hat
  sd.u.hat_collect[i] <- sd.u.hat
}
hist(beta0hat_collect,xlim = c(min(beta0hat_collect),1),main = "
    Histogram of Estimated beta0hat",xlab = "beta0hat")
abline(v=beta0,col="red",lwd=4)
hist(beta1hat_collect,xlim = c(0.9,max(beta1hat_collect)),main = "
    Histogram of Estimated beta1hat",xlab = "beta1hat")
abline(v=0.9,col="red",lwd=4)
hist(sd.u.hat_collect,xlim = c(min(sd.u.hat_collect),1),main = "
    Histogram of Estimated sigmaHat",xlab = "sigmaHat")
abline(v=sigma.u,col="red",lwd=4)

### (II)
numRep=1000
m <- 100
n <- 2*m
beta0 <- 1
beta1 <- 0.9
sigma.u <- 1
mu <- beta0/(1-beta1)
beta1hat_collect=NULL
beta0hat_collect=NULL
sd.u.hat_collect=NULL
for (i in (1:numRep)) {
  iSeq <- (1:n)
  u <- rnorm(n, mean=0, sd=sigma.u)
  y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-
    beta1^2)
```

```
  ratio <- var(y)/var(y[2*(1:m)] - y[2*(1:m)-1])
  beta1hat <- (2*ratio-sqrt(4*ratio-1))/(2*ratio-1)
  beta0hat <- mean(y)*(1-beta1hat)
  #extract residuals:
  u.hat <- y[iSeq] - (beta0hat + beta1hat*y[iSeq-(-1)^iSeq])
  sd.u.hat <- sd(u.hat)
  beta0hat_collect[i] <- beta0hat
  beta1hat_collect[i] <- beta1hat
  sd.u.hat_collect[i] <- sd.u.hat
}
hist(beta0hat_collect,freq = F,main = "Histogram of Beta0Hat",xlab
    = "beta0hat")
abline(v=beta0,col="red",lwd=4)
hist(beta1hat_collect,freq = F,main = "Histogram of Beta1Hat",xlab
    = "beta1hat")
abline(v=beta1,col="red",lwd=4)
hist(sd.u.hat_collect,main = "Histogram of Sd.u.Hat",xlab = "Sd.u.
   Hat")
abline(v=sigma.u,col="red",lwd=4)

### (III)
numRep=10000
m <- 100
beta0 = 1
n <- 2*m
beta1 = 0.9
sigma.u = 1
mu <- beta0/(1-beta1)
confinterval_collect=NULL
for (i in (1:numRep)) {
  iSeq <- (1:n)
  u <- rnorm(n, mean=0, sd=sigma.u)
  y <- beta0/(1-beta1) + (u[iSeq]+beta1*u[iSeq-(-1)^iSeq])/(1-
    beta1^2)
  ratio <- var(y)/var(y[2*(1:m)] - y[2*(1:m)-1])
  beta1hat <- (2*ratio-sqrt(4*ratio-1))/(2*ratio-1)
  beta0hat <- mean(y)*(1-beta1hat)
  #extract residuals:
  u.hat <- y[iSeq] - (beta0hat + beta1hat*y[iSeq-(-1)^iSeq])
  sd.u.hat <- sd(u.hat)
  up.bound <- mean(y)+1.96/sqrt(n)*sd.u.hat/(1-beta1hat)
  low.bound <- mean(y)-1.96/sqrt(n)*sd.u.hat/(1-beta1hat)
  confinterval_collect[i] <- (mu>low.bound)&(mu<up.bound)
}
sum(confinterval_collect)/length(confinterval_collect) #0.9438
   which is around 95%
```