Roy Model

SAI OMKAR,KANDUKURI - PROBLEM SET 1

Monday, October 8 5:00 - 6:20 PM

Setup

The section loads libraries that we will need to use to run the code below.

```
# For this session we will load the tidyverse, a commonly used set of R libraries
# Find more information here: https://www.tidyverse.org/packages/
library('tidyverse')
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library('MASS')
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

Roy Model

In the context of lecture, the Roy Model was used to motivate the need for economic and statistical models to underly our analysis of observed phenomena in public policy. To (hopefully) illustrate the model as well as some of the power of computational tools, here is a brief simulation of the model. The expectation is NOT that you can put this together yourself right now.

Generate Random Data

The first step is to generate random data from two correlated distributions with the characteristics described in class, i.e. Economist earnings are normally distributed with mean = 60k and stdev = 10k, and Accountant earnings are normally distributed with mean = 65k and st dev = 5k. This generation will produce data that pairs draws from both distribution, i.e. we are observing for every theoretical individual in our sample both their potential earnings as an accountant and their potential earnings as an economist.

```
set.seed(10042018)



samples <- 1000000  # Sample size or size of population
cor <- 0.9 # correlation coefficient

# Generate correlated std normal random sample using the mvnorm function from MASS package
data <- mvrnorm(n=samples, mu=c(0, 0), Sigma=matrix(c(1, cor, cor, 1), nrow=2), empirical=FALSE)

# Convert to dataframe
df1 <- as_data_frame(data)
```

```
## Warning: `as_data_frame()` was deprecated in tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
```

```
## Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if `.name_repair` is
## Using compatibility `.name_repair`.
```

```
# Make the dataframe easier to use
df1 <- df1 %>% dplyr::rename(accnt = V1, econ = V2) # rename columns

# Change distributions from std normal to those specified in lecture:
# Accounting ~ N(65000, 5000)
# Economics ~ N(60000, 10000)
mu_econ <- 60000
sigma_econ <- 10000
mu_accnt <- 65000
sigma_accnt <- 5000

df1 <- df1 %>% mutate(
        accnt = accnt*sigma_accnt + mu_accnt, # update accounting variable
        econ = econ*sigma_econ + mu_econ # update econ variable
)
```

Sanity Checks

Let's run a few checks to make sure data looks reasonable. First we'll use the `head` and `tail` commands to get a look at the data. Then, we will make sure that the correlation between the accoutant and economist distributions is 0.86, the value we set when we generated the data, using the `cor` command. Finally, we will run the `summarise` command to see the mean, min, max, and quartiles of each of the distributions.

```r
# Look at the first and last 6 rows of the dataframe
head(df1)
```

```
## # A tibble: 6 x 2
##    accnt   econ
##    <dbl>  <dbl>
## 1 65037. 58736.
## 2 66280. 65719.
## 3 64605. 61839.
## 4 65908. 63792.
## 5 58921. 49206.
## 6 63169. 48734.
```

```r
tail(df1)
```

```
## # A tibble: 6 x 2
##    accnt   econ
##    <dbl>  <dbl>
## 1 58520. 43816.
## 2 68335. 64071.
## 3 62921. 54838.
## 4 61847. 51383.
## 5 70867. 69141.
## 6 68559. 68678.
```

```r
# Compare the correlation we set to the correlation we calculate
check <- round(cor(df1$accnt, df1$econ)) == round(cor)
print(ifelse(check, "The correlations are the same!", "Oops, the correlations are not the same."))
```

```
## [1] "The correlations are the same!"
```

```r
# Generate summary statistics
print("Economists Summary Stats")
```

```
## [1] "Economists Summary Stats"
```

```r
summary(df1$econ)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11271   53253   59997   59997   66726  106422
```

```r
print("Accountant Summary Stats")
```

```
## [1] "Accountant Summary Stats"
```

```r
summary(df1$accnt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   40451   61637   64999   65002   68365   90649
```

```r
### Picking a Career

#Our assumption is that every person will choose to be an accountant or an economist based on what will
#```{r jobs}
# Assign job labels
df1 <- df1 %>% mutate(job = ifelse(econ > accnt, "econ", "accnt"))

# Take a look at the change to the dataframe using head
head(df1)
```

```
## # A tibble: 6 x 3
##    accnt   econ job
##    <dbl>  <dbl> <chr>
## 1 65037. 58736. accnt
## 2 66280. 65719. accnt
## 3 64605. 61839. accnt
## 4 65908. 63792. accnt
## 5 58921. 49206. accnt
## 6 63169. 48734. accnt
```

Results

```r
# Make results dataframe
results <- df1 %>%
            group_by(job) %>% # Group all of the rows with the same "job" together
            summarise('Economist Earnings' = mean(econ), 'Accountant Earnings' = mean(accnt),

            n=n()) %>% # Calculate means and counts for economicts and accountants
            mutate(job = c("Accountant", "Economist")) %>% # add labels
            t() # transpose

colnames(results) <- c('Accountant', 'Economist')




results <- as.data.frame(results) %>%
            slice(2:4) %>%
            mutate(
              x= c('Economist Earnings', 'Accountant Earnings', 'n')
            ) %>%
            dplyr::select(x, Accountant, Economist)

results
```

```
##                                        x Accountant Economist
## Economist Earnings   Economist Earnings   56756.85  73066.65
## Accountant Earnings Accountant Earnings   63822.98  69759.32
## n                                      n     801309    198691
```

Answer to Question1:

Answer for Question1 Two reasons: 1)Seed is different in R and Stata. Seed is a dynamic number that can be set which al lows us to replicate the observations generated b y the random number generator. It is helpful for simulations to replicate the result s which can aid simulation analysis.

2)However, even with same seed, different results are observed in R and Stata. This c an be because the Normal distribution generated i n R differs from what is generated in Stata. This shows that the underlying distribution generator algorithm functions different ly in R(mvrnorm) and Stata(drawnorm).

Question2

#Question2 #set.seed(02101870) #Question2

Monday, October 8 5:00 - 6:20 PM

Setup

The section loads libraries that we will need to use to run the code below.

```
# For this session we will load the tidyverse, a commonly used set of R libraries
# Find more information here: https://www.tidyverse.org/packages/
library('tidyverse')
library('MASS')
```

Roy Model

In the context of lecture, the Roy Model was used to motivate the need for economic and statistical models to underly our analysis of observed phenomena in public policy. To (hopefully) illustrate the model as well as some of the power of computational tools, here is a brief simulation of the model. The expectation is NOT that you can put this together yourself right now.

Generate Random Data

The first step is to generate random data from two correlated distributions with the characteristics described in class, i.e. Economist earnings are normally distributed with mean = 60k and stdev = 10k, and Accountant earnings are normally distributed with mean = 65k and st dev = 5k. This generation will produce data that pairs draws from both distribution, i.e. we are observing for every theoretical individual in our sample both their potential earnings as an accountant and their potential earnings as an economist.

```
set.seed(02101870)


samples <- 1000000  # Sample size or size of population
cor <- 0.9 # correlation coefficient

# Generate correlated std normal random sample using the mvnorm function from MASS package
data <- mvrnorm(n=samples, mu=c(0, 0), Sigma=matrix(c(1, cor, cor, 1), nrow=2), empirical=FALSE)

# Convert to dataframe
df1 <- as_data_frame(data)

# Make the dataframe easier to use
```

```r
df1 <- df1 %>% dplyr::rename(accnt = V1, econ = V2) # rename columns

# Change distributions from std normal to those specified in lecture:
# Accounting ~ N(65000, 5000)
# Economics ~ N(60000, 10000)
mu_econ <- 60000
sigma_econ <- 10000
mu_accnt <- 65000
sigma_accnt <- 5000

df1 <- df1 %>% mutate(
        accnt = accnt*sigma_accnt + mu_accnt, # update accounting variable
        econ = econ*sigma_econ + mu_econ # update econ variable
)
```

Sanity Checks

Let's run a few checks to make sure data looks reasonable. First we'll use the `head` and `tail` commands to get a look at the data. Then, we will make sure that the correlation between the accoutant and economist distributions is 0.86, the value we set when we generated the data, using the `cor` command. Finally, we will run the `summarise` command to see the mean, min, max, and quartiles of each of the distributions.

```r
# Look at the first and last 6 rows of the dataframe
head(df1)
```

```
## # A tibble: 6 x 2
##    accnt   econ
##    <dbl>  <dbl>
## 1 66606. 63314.
## 2 65084. 63758.
## 3 70090. 65585.
## 4 56853. 50869.
## 5 60755. 53398.
## 6 73071. 80320.
```

```r
tail(df1)
```

```
## # A tibble: 6 x 2
##    accnt   econ
##    <dbl>  <dbl>
## 1 74382. 75437.
## 2 59409. 55834.
## 3 67347. 56298.
## 4 58116. 40943.
## 5 64621. 65722.
## 6 59699. 56242.
```

```r
# Compare the correlation we set to the correlation we calculate
check <- round(cor(df1$accnt, df1$econ)) == round(cor)
print(ifelse(check, "The correlations are the same!", "Oops, the correlations are not the same."))
```

```
## [1] "The correlations are the same!"
```

```r
# Generate summary statistics
print("Economists Summary Stats")
```

```
## [1] "Economists Summary Stats"
```

```r
summary(df1$econ)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15650   53234   59972   59988   66729  107892
```

```r
print("Accountant Summary Stats")
```

```
## [1] "Accountant Summary Stats"
```

```r
summary(df1$accnt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   41577   61628   64995   64999   68367   89273
```

```r
### Picking a Career

#Our assumption is that every person will choose to be an accountant or an economist based on what will
#```{r jobs}
# Assign job labels
df1 <- df1 %>% mutate(job = ifelse(econ > accnt, "econ", "accnt"))

# Take a look at the change to the dataframe using head
head(df1)
```

```
## # A tibble: 6 x 3
##    accnt   econ job
##    <dbl>  <dbl> <chr>
## 1 66606. 63314. accnt
## 2 65084. 63758. accnt
## 3 70090. 65585. accnt
## 4 56853. 50869. accnt
## 5 60755. 53398. accnt
## 6 73071. 80320. econ
```

Results

```r
# Make results dataframe
results <- df1 %>%
            group_by(job) %>% # Group all of the rows with the same "job" together
            summarise('Economist Earnings' = mean(econ), 'Accountant Earnings' = mean(accnt),
            n=n()) %>% # Calculate means and counts for economicts and accountants
            mutate(job = c("Accountant", "Economist")) %>% # add labels
            t() # transpose
```

```
colnames(results) <- c('Accountant', 'Economist')




results <- as.data.frame(results) %>%
            slice(2:4) %>%
            mutate(
              x= c('Economist Earnings', 'Accountant Earnings', 'n')
            ) %>%
            dplyr::select(x, Accountant, Economist)

results
```

```
##                                       x Accountant Economist
## Economist Earnings   Economist Earnings   56745.38  73043.11
## Accountant Earnings Accountant Earnings   63819.94  69745.89
## n                                     n     801019    198981
```

Answer for Question2 Reset the seed to original seed value 195912191 1) Reason for not giving birthdate of self could be construed a data privacy issue or data security issue. Also it will produce similar results as fellow mates in the class because there is a greater probablity for repetition of birthdays when the sample size exceeds 23, class sample size is somewhere between 300-400.This is called birthday paradox.

2) Seed set is different in question 1 and question 2. As seed is changed, the distribution generated changes slightly causing the results to change slightly. However the mean and variance still remains same as they are fixed. Only the observations changes with change in seed.

Question 3

Monday, October 8 5:00 - 6:20 PM

Setup

The section loads libraries that we will need to use to run the code below.

```
# For this session we will load the tidyverse, a commonly used set of R libraries
# Find more information here: https://www.tidyverse.org/packages/
library('tidyverse')
library('MASS')
```

Roy Model

In the context of lecture, the Roy Model was used to motivate the need for economic and statistical models to underly our analysis of observed phenomena in public policy. To (hopefully) illustrate the model as well as some of the power of computational tools, here is a brief simulation of the model. The expectation is NOT that you can put this together yourself right now.

Generate Random Data

The first step is to generate random data from two correlated distributions with the characteristics described in class, i.e. Economist earnings are normally distributed with mean = 60k and stdev = 10k, and Accountant earnings are normally distributed with mean = 65k and st dev = 5k. This generation will produce data that pairs draws from both distribution, i.e. we are observing for every theoretical individual in our sample both their potential earnings as an accountant and their potential earnings as an economist.

```
set.seed(10042018)



samples <- 1000000  # Sample size or size of population
cor <- 0.9 # correlation coefficient

# Generate correlated std normal random sample using the mvnorm function from MASS package
data <- mvrnorm(n=samples, mu=c(0, 0), Sigma=matrix(c(1, cor, cor, 1), nrow=2), empirical=FALSE)

# Convert to dataframe
df1 <- as_data_frame(data)

# Make the dataframe easier to use
df1 <- df1 %>% dplyr::rename(accnt = V1, econ = V2) # rename columns

# Change distributions from std normal to those specified in lecture:
# Accounting ~ N(65000, 5000)
# Economics ~ N(60000, 10000)
mu_econ <- 60000
sigma_econ <- 10000

mu_accnt <- 65000
sigma_accnt <- 5000

df1 <- df1 %>% mutate(
        accnt = accnt*sigma_accnt + mu_accnt, # update accounting variable
        econ = econ*sigma_econ + mu_econ # update econ variable
)
```

Sanity Checks

Let's run a few checks to make sure data looks reasonable. First we'll use the `head` and `tail` commands to get a look at the data. Then, we will make sure that the correlation between the accoutant and economist distributions is 0.86, the value we set when we generated the data, using the `cor` command. Finally, we will run the `summarise` command to see the mean, min, max, and quartiles of each of the distributions.

```
# Look at the first and last 6 rows of the dataframe
head(df1)
```

```
## # A tibble: 6 x 2
##    accnt   econ
##    <dbl>  <dbl>
## 1 65037. 58736.
## 2 66280. 65719.
```

```
## 3 64605. 61839.
## 4 65908. 63792.
## 5 58921. 49206.
## 6 63169. 48734.
```

```r
tail(df1)
```

```
## # A tibble: 6 x 2
##    accnt   econ
##    <dbl>  <dbl>
## 1 58520. 43816.
## 2 68335. 64071.
## 3 62921. 54838.
## 4 61847. 51383.
## 5 70867. 69141.
## 6 68559. 68678.
```

```r
# Compare the correlation we set to the correlation we calculate
check <- round(cor(df1$accnt, df1$econ)) == round(cor)
print(ifelse(check, "The correlations are the same!", "Oops, the correlations are not the same."))
```

```
## [1] "The correlations are the same!"
```

```r
# Generate summary statistics
print("Economists Summary Stats")
```

```
## [1] "Economists Summary Stats"
```

```r
summary(df1$econ)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11271   53253   59997   59997   66726  106422
```

```r
print("Accountant Summary Stats")
```

```
## [1] "Accountant Summary Stats"
```

```r
summary(df1$accnt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   40451   61637   64999   65002   68365   90649
```

```r
### Picking a Career

#Our assumption is that every person will choose to be an accountant or an economist based on what will
#```{r jobs}
# Assign job labels
df1 <- df1 %>% mutate(job = ifelse(econ > accnt, "econ", "accnt"))

# Take a look at the change to the dataframe using head
head(df1)
```

```
## # A tibble: 6 x 3
##    accnt   econ job
##    <dbl>  <dbl> <chr>
## 1 65037. 58736. accnt
## 2 66280. 65719. accnt
## 3 64605. 61839. accnt
## 4 65908. 63792. accnt
## 5 58921. 49206. accnt
## 6 63169. 48734. accnt
```

Results

```r
# Make results dataframe
results <- df1 %>%
            group_by(job) %>% # Group all of the rows with the same "job" together
            #summarise('Economist Earnings' = mean(econ), 'Accountant Earnings' = mean(accnt),


              #Changing the code to calculate and display standard deviation
              summarise('Economist Earnings' = sd(econ), 'Accountant Earnings' = sd(accnt),

                      n=n()) %>% # Calculate sd and counts for economicts and accountants
            mutate(job = c("Accountant", "Economist")) %>% # add labels
            t() # transpose

colnames(results) <- c('Accountant', 'Economist')




results <- as.data.frame(results) %>%
            slice(2:4) %>%
            mutate(
              x= c('Economist Earnings', 'Accountant Earnings', 'n')
            ) %>%
            dplyr::select(x, Accountant, Economist)

results
```

```
##                                          x Accountant Economist
## Economist Earnings   Economist Earnings   7997.876  5709.471
## Accountant Earnings Accountant Earnings    4496.64   4020.58
## n                                        n     801309    198691
```

Answer for Question 3 The standard normal sample size is 1000000 It is observed that out of the sample size for economists only 198691 have become economists and hence the standard deviation we observed (5709.471) is significantly less compared to the conditional standard deviation of 100000 The same goes for accountants, out of the sample size for accountants its observed that 801309 have become accountants, hence the standard deviation we observed (4496.64) is less compared to the conditional standard deviation set of 5000

Question4

Monday, October 8 5:00 - 6:20 PM

Setup

The section loads libraries that we will need to use to run the code below.

```
# For this session we will load the tidyverse, a commonly used set of R libraries
# Find more information here: https://www.tidyverse.org/packages/
library('tidyverse')
library('MASS')
```

Roy Model

In the context of lecture, the Roy Model was used to motivate the need for economic and statistical models to underly our analysis of observed phenomena in public policy. To (hopefully) illustrate the model as well as some of the power of computational tools, here is a brief simulation of the model. The expectation is NOT that you can put this together yourself right now.

Generate Random Data

The first step is to generate random data from two correlated distributions with the characteristics described in class, i.e. Economist earnings are normally distributed with mean = 60k and stdev = 10k, and Accountant earnings are normally distributed with mean = 65k and st dev = 5k. This generation will produce data that pairs draws from both distribution, i.e. we are observing for every theoretical individual in our sample both their potential earnings as an accountant and their potential earnings as an economist.

```
set.seed(10042018)


samples <- 1000000  # Sample size or size of population
cor <- 0.9 # correlation coefficient

# Generate correlated std normal random sample using the mvnorm function from MASS package
data <- mvrnorm(n=samples, mu=c(0, 0), Sigma=matrix(c(1, cor, cor, 1), nrow=2), empirical=FALSE)

# Convert to dataframe
df1 <- as_data_frame(data)

# Make the dataframe easier to use
df1 <- df1 %>% dplyr::rename(accnt = V1, econ = V2) # rename columns

# Change distributions from std normal to those specified in lecture:
# Accounting ~ N(65000, 5000)
# Economics ~ N(60000, 10000)
mu_econ <- 60000

  sigma_econ <- 12000

mu_accnt <- 65000
sigma_accnt <- 5000
```

```r
df1 <- df1 %>% mutate(
        accnt = accnt*sigma_accnt + mu_accnt, # update accounting variable
        econ = econ*sigma_econ + mu_econ # update econ variable
)
```

Sanity Checks

Let's run a few checks to make sure data looks reasonable. First we'll use the `head` and `tail` commands to get a look at the data. Then, we will make sure that the correlation between the accoutant and economist distributions is 0.86, the value we set when we generated the data, using the `cor` command. Finally, we will run the `summarise` command to see the mean, min, max, and quartiles of each of the distributions.

```r
# Look at the first and last 6 rows of the dataframe
head(df1)
```

```
## # A tibble: 6 x 2
##    accnt   econ
##    <dbl>  <dbl>
## 1 65037. 58484.
## 2 66280. 66863.
## 3 64605. 62207.
## 4 65908. 64550.
## 5 58921. 47047.
## 6 63169. 46481.
```

```r
tail(df1)
```

```
## # A tibble: 6 x 2
##    accnt   econ
##    <dbl>  <dbl>
## 1 58520. 40580.
## 2 68335. 64885.
## 3 62921. 53806.
## 4 61847. 49660.
## 5 70867. 70969.
## 6 68559. 70414.
```

```r
# Compare the correlation we set to the correlation we calculate
check <- round(cor(df1$accnt, df1$econ)) == round(cor)
print(ifelse(check, "The correlations are the same!", "Oops, the correlations are not the same."))
```

```
## [1] "The correlations are the same!"
```

```r
# Generate summary statistics
print("Economists Summary Stats")
```

```
## [1] "Economists Summary Stats"
```

```r
summary(df1$econ)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1525   51903   59996   59997   68071  115707
```

```r
print("Accountant Summary Stats")
```

```
## [1] "Accountant Summary Stats"
```

```r
summary(df1$accnt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   40451   61637   64999   65002   68365   90649
```

```r
### Picking a Career

#Our assumption is that every person will choose to be an accountant or an economist based on what will
#```{r jobs}
# Assign job labels
df1 <- df1 %>% mutate(job = ifelse(econ > accnt, "econ", "accnt"))

# Take a look at the change to the dataframe using head
head(df1)
```

```
## # A tibble: 6 x 3
##    accnt   econ job
##    <dbl>  <dbl> <chr>
## 1 65037. 58484. accnt
## 2 66280. 66863. econ
## 3 64605. 62207. accnt
## 4 65908. 64550. accnt
## 5 58921. 47047. accnt
## 6 63169. 46481. accnt
```

Results

```r
# Make results dataframe
results <- df1 %>%
            group_by(job) %>% # Group all of the rows with the same "job" together
            summarise('Economist Earnings' = mean(econ), 'Accountant Earnings' = mean(accnt),


                      n=n()) %>% # Calculate means and counts for economicts and accountants
            mutate(job = c("Accountant", "Economist")) %>% # add labels
            t() # transpose

colnames(results) <- c('Accountant', 'Economist')
```

```r
results <- as.data.frame(results) %>%
            slice(2:4) %>%
            mutate(
              x= c('Economist Earnings', 'Accountant Earnings', 'n')
            ) %>%
            dplyr::select(x, Accountant, Economist)

results
```

```
##                                       x Accountant Economist
## Economist Earnings   Economist Earnings   54934.66  74370.56
## Accountant Earnings Accountant Earnings   63371.95  69632.12
## n                                     n     739539    260461
```

Answer to question 4 The data shows an increase in total economists and decrease in accountants. As the standard deviation of economist earnings is increased from $10000 to $12000, the economist earnings of economist have increased and accountants with economist earnings have reduced. This is due to the flattening of the curve that happens because of the increase in standard deviation causing the occurrences of high economist earnings has resulted accountants becoming economists, which means increase in total economists and decrease in total accountants. Due to the same, the mean of economists with economist earnings increases (Mean changed from 73066.65 to 74370.56) whereas the mean of accountants with accountant earn ings has reduced(Mean changed from 63822.98 to 63371.95). One another observation is the increase in people with economists earning s (198340 to 260672) and reduction in number of people with accountants earnings (801660 to 739328)