Precept 7 Code

Ziyao Tian

October 25, 2018

Coding with matrices (Useful Reference for Problem 1)

First, we're going to code in some toy matrices we can use to practice matrix operations.

```
A <- matrix(c(1,2,3,4), nrow = 2, ncol = 2, byrow = TRUE)
B <- matrix(c(1,0,1,0), nrow = 2, ncol = 2, byrow = TRUE)
C <- matrix(c(6,5,4,3,2,1), nrow = 3, ncol = 2, byrow = TRUE)
```

Now we're going to practice doing matrix operations in `R`. Here are some of the functions and operations you'll need:

- Addition: `+`
- Subtraction: `-`
- Multiplication: `%*%`
- Inverse: `solve()`
- Transpose: `t()`
- Extract the diagonal of a matrix **A**: `diag(A)`
- Make a k by k identity matrix: `diag(k)`

Try filling in the code yourself!

```
## Add A and B together
A + B
```

```
##      [,1] [,2]
## [1,]    2    2
## [2,]    4    4
```

```
## A minus B
A - B
```

```
##      [,1] [,2]
## [1,]    0    2
## [2,]    2    4
```

```
## A times B
B %*% A
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    1    2
```

```
## B times A

## C times A

## A times C
A %*% t(C)
```

```
##      [,1] [,2] [,3]
## [1,]   16   10    4
## [2,]   38   24   10
```

```
## Inverse of A (A^(-1))
solve(A)
```

```
##      [,1] [,2]
## [1,] -2.0  1.0
## [2,]  1.5 -0.5
```

```
A %*% solve(A)
```

```
##      [,1]         [,2]
## [1,]    1 1.110223e-16
## [2,]    0 1.000000e+00
```

```
## C'
t(C)
```

```
##      [,1] [,2] [,3]
## [1,]    6    4    2
## [2,]    5    3    1
```

```
## A four by four identity matrix
diag(4)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
```

```
diag(diag(4))
```

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

```
## The diagonal of matrix C

##Verify that A times its inverse gives you the identity matrix
```

Multiple Regression: Interpretation & F-test (Useful Reference for Problem 2)

Interpretation

First, we'll load the data and run some unrestricted and restricted models.

```
## Let's load your favorite dataset!!!
billionaires <- read.csv("Billionaires.csv", header = TRUE)

bill <- billionaires %>%
  filter(year == 2014, !is.na(age), !is.na(networthusbillion)) %>%
  dplyr::select(year, name, rank, citizenship, networthusbillion,
         selfmade, typeofwealth, gender, age, foundingdate) %>%
  mutate(wealth = networthusbillion*1000000000,
         logwealth = log(wealth),
         woman = ifelse(gender == "female", 1, 0),
         inherit = ifelse(selfmade == "inherited", 1, 0))
```

```r
## We're going to run a model with lots of covariates we'll consider the "unrestricted model"
unrestrict <- lm(data = bill, logwealth ~ age + inherit + woman + woman:inherit)
summary(unrestrict)
```

```
##
## Call:
## lm(formula = logwealth ~ age + inherit + woman + woman:inherit,
##     data = bill)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3275 -0.5641 -0.1755  0.3551  3.4688
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   21.133839   0.095859 220.467  < 2e-16 ***
## age            0.007782   0.001481   5.255 1.68e-07 ***
## inherit        0.193152   0.048177   4.009 6.37e-05 ***
## woman         -0.145810   0.145342  -1.003    0.316
## inherit:woman  0.098744   0.164902   0.599    0.549
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7715 on 1585 degrees of freedom
## Multiple R-squared:  0.0321, Adjusted R-squared:  0.02966
## F-statistic: 13.14 on 4 and 1585 DF,  p-value: 1.559e-10
```

Think-pair-share: how do we interpret the interaction coefficient?

Multiple regression meets matrix: manually coding the betas (Useful Reference for Problem 3)

```r
#X
bill_m <- bill %>%
  mutate(woman.inherit = woman * inherit) %>%
  select(age, inherit, woman, woman.inherit, logwealth)
head(bill_m)
```

```
##   age inherit woman woman.inherit logwealth
## 1  83       0     0             0  21.67878
## 2  54       0     0             0  20.90559
## 3  52       1     1             1  23.57397
## 4  77       1     0             0  21.75289
## 5  83       0     0             0  21.19327
## 6  71       1     0             0  21.97603
```

```r
X <- as.matrix(cbind(1, bill_m[, c("age", "inherit", "woman", "woman.inherit")]))
#y
y <- bill_m$logwealth
#betas: X'X^(-1)X'y
betas <- solve(t(X) %*% X) %*% t(X) %*% y
betas
```

```
##                       [,1]
```

```
## 1             21.133838561
## age            0.007782127
## inherit        0.193152150
## woman         -0.145809557
## woman.inherit  0.098744159
```

```
#compare with lm results
summary(unrestrict)$coefficients[,1]
```

```
##   (Intercept)           age       inherit         woman inherit:woman
##  21.133838561   0.007782127   0.193152150  -0.145809557   0.098744159
```

Comparing restricted and unrestricted models

```
## Now we're going to look at a couple of restricted models

## we can use the first one to test whether we need any covariates at all
## note that the intercept is just the mean!
restrict1 <-  lm(data = bill, logwealth ~ 1)

## we can use the second to test whether we need gender or the interaction term
restrict2 <- lm(data = bill, logwealth ~ age + inherit)
```

Let's start by comparing our results with one big stargazer table:

```
stargazer(restrict1, restrict2, unrestrict,
          title = "Comparing our linear models of billionaire wealth",
          star.cutoffs = c(0.05, 0.01, 0.001),
          header = FALSE,
          table.placement = "!h")
```

Now let's look visually at how well these models are doing at predicting our outcomes by looking at the actual vs. predicted outcomes for our data:

```
##Extracting the fitted values from the model and adding them to the dataframe
bill$fit_un <- unrestrict$fitted.values
bill$fit_r1 <- restrict1$fitted.values
bill$fit_r2 <- restrict2$fitted.values

##Plotting actual vs. predicted values for each model
un <- ggplot(data = bill) +
        geom_point(aes(x = logwealth, y = fit_un),
                   alpha = .2, color = "darkgreen") +
        ylim(min(bill$logwealth), max(bill$logwealth)) +
        xlab("Actual log wealth") +
        ylab("Predicted log wealth") +
        ggtitle("Unrestricted model") +
        theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
              text = element_text(family = "Helvetica"))

r1 <-   ggplot(data = bill) +
        geom_point(aes(x = logwealth, y = fit_r1),
                   alpha = .2, color = "darkgreen") +
        ylim(min(bill$logwealth), max(bill$logwealth)) +
        xlab("Actual log wealth") +
```

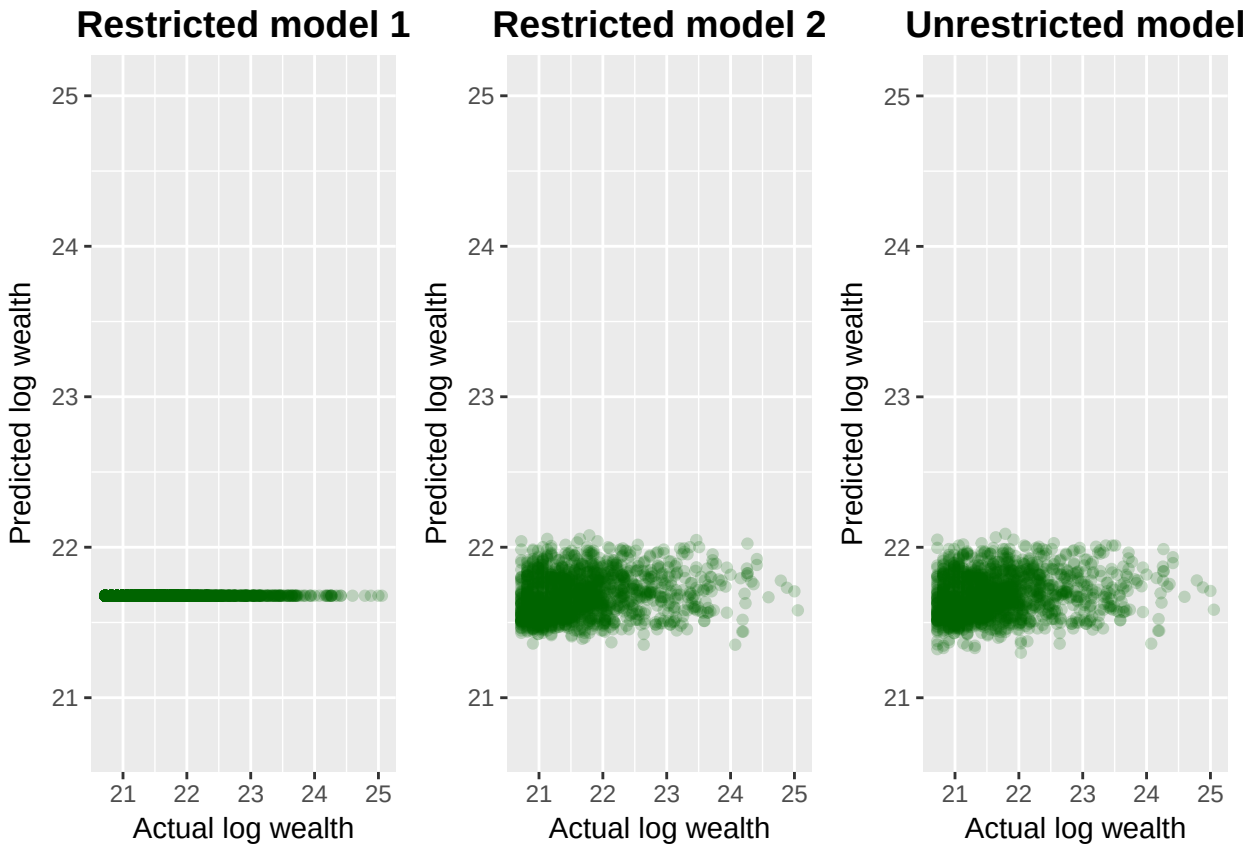
Table 1: Comparing our linear models of billionaire wealth

	Dependent variable:		
	logwealth		
	(1)	(2)	(3)
age		0.008***	0.008***
		(0.001)	(0.001)
inherit		0.183***	0.193***
		(0.042)	(0.048)
woman			−0.146
			(0.145)
inherit:woman			0.099
			(0.165)
Constant	21.678***	21.123***	21.134***
	(0.020)	(0.095)	(0.096)
Observations	1,590	1,590	1,590
R^2	0.000	0.031	0.032
Adjusted R^2	0.000	0.030	0.030
Residual Std. Error	0.783 (df = 1589)	0.771 (df = 1587)	0.772 (df = 1585)
F Statistic		25.610*** (df = 2; 1587)	13.142*** (df = 4; 1585)
Note:			*p<0.05; **p<0.01; ***p<0.001

```
        ylab("Predicted log wealth") +
        ggtitle("Restricted model 1") +
        theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
              text = element_text(family = "Helvetica"))

r2 <-   ggplot(data = bill) +
        geom_point(aes(x = logwealth, y = fit_r2),
                   alpha = .2, color = "darkgreen") +
        ylim(min(bill$logwealth), max(bill$logwealth)) +
        xlab("Actual log wealth") +
        ylab("Predicted log wealth") +
        ggtitle("Restricted model 2") +
        theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
              text = element_text(family = "Helvetica"))

grid.arrange(r1, r2, un, nrow = 1, ncol = 3)
```



Now we'll do a much more precise test- the F-test.

Recall that the **F statistic** can be calculated by the following procedure:

1. Fit the **Unrestricted Model (UR)** which *does not* impose H_0
2. Fit the **Restricted Model (R)** which *does* impose H_0
3. From the two results, compute the **F Statistic:**

$$F_0 = \frac{(SSR_r - SSR_{ur})/q}{SSR_{ur}/(n-k-1)}$$

where **SSR** = sum of squared residuals, **q** = number of restrictions, **k** = number of predictors in the unrestricted model, and **n** = # of observations.

Let's extract and calculate all the relevant values first:

```
## Calculate the sum of squared residuals for each model
SSR_un <- sum(resid(unrestrict) ^ 2)
SSR_res1 <- sum(resid(restrict1) ^ 2)
SSR_res2 <- sum(resid(restrict2) ^ 2)

## Calculate n - k - 1 for unrestricted model
nk1 <- nrow(bill) - 4 - 1
## There is a slightly easier way to do that
nk1 == df.residual(unrestrict)
```

```
## [1] TRUE
```

```
## Find q for the restricted models
q1 <- 4
q2 <- 2
```

Now we'll use the formula to calculate our first F-Statistic. Note that we should get the same F-Statistic we see in our `lm()` output for the unrestricted model.

```
## F-test with first restricted model
f1 <- ((SSR_res1 - SSR_un) / q1) / (SSR_un / nk1)
f1
```

```
## [1] 13.14155
```

```
## we can use pf() to get the p-value for our f test
pf(f1, q1, nk1, lower.tail = FALSE)
```

```
## [1] 1.559003e-10
```

Now we can do the same for our second restricted model.

```
## F-test with first restricted model
f2 <- ((SSR_res2 - SSR_un) / q2) / (SSR_un / nk1)
f2
```

```
## [1] 0.6834552
```

```
## we can use pf() to get the p-value for our f test
pf(f2, q2, nk1, lower.tail = FALSE)
```

```
## [1] 0.5050183
```

We can also do this with the `anova()` function:

```
#recall that
#unrestrict <- lm(data = bill, logwealth ~ age + inherit + woman + woman:inherit)
#restrict1 <- lm(data = bill, logwealth ~ 1)
#restrict2 <- lm(data = bill, logwealth ~ age + inherit)

#F test 1
anova(restrict1, unrestrict)
```

```
## Analysis of Variance Table
##
## Model 1: logwealth ~ 1
## Model 2: logwealth ~ age + inherit + woman + woman:inherit
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   1589 974.75
## 2   1585 943.46  4     31.29 13.142 1.559e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#F test 2
anova(restrict2, unrestrict)
```

```
## Analysis of Variance Table
##
## Model 1: logwealth ~ age + inherit
## Model 2: logwealth ~ age + inherit + woman + woman:inherit
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1   1587 944.27
```

```
## 2   1585 943.46  2   0.81364 0.6835  0.505
```

Think-pair-share: - What is the null hypothesis H0 for F test 1 and 2? - how do we interpret the F test results?

Bootstrapping to get standard errors and confidence intervals (Useful Reference for Problem 4)

```
## Let's say we have a random sample of billionaires and we want to
## estimate the mean log wealth and the standard error of the mean
## Let's first implement this with a for loop
set.seed(12334)
# First create the random sample
n <- 1000
random_sample <- sample_n(bill, n)
mean(bill$logwealth)
```

```
## [1] 21.67788
```

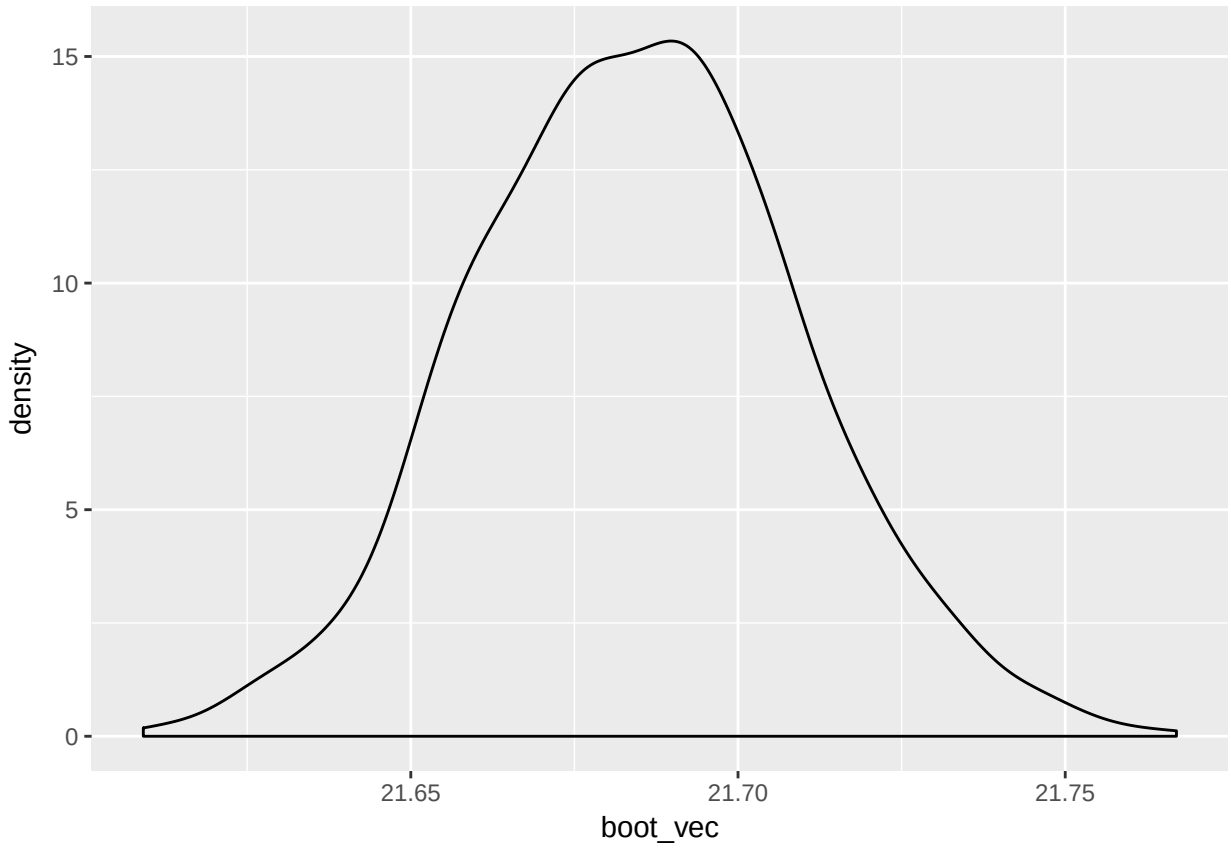
```
mean(random_sample$logwealth)
```

```
## [1] 21.68432
```

```
# We can do it by for loop
reps <- 1000
boot_vec <- rep(0, reps)
for (i in 1:reps) {
  boot_samp <- sample_n(random_sample, nrow(random_sample), replace = T)
  boot_vec[i] <- mean(boot_samp$logwealth)
}

# Or replicate
calc_boot <- function() {
  boot_samp <- sample_n(random_sample, nrow(random_sample), replace = T)
  boot_samp_mean <- mean(boot_samp$logwealth)
  return(boot_samp_mean)
}
boot_vec <- replicate(reps, calc_boot())

# the distribution of our results
ggplot() + geom_density(aes(x = boot_vec))
```




```
# their mean and standard deviation
mean(boot_vec) # about the same as the true mean in our sample
```

```
## [1] 21.68457
```

```
sd(boot_vec)
```

```
## [1] 0.0249139
```

```
# and the 95% confidence interval
quantile(boot_vec, probs = c(0.005, 0.995))
```

```
##     0.5%    99.5%
## 21.62231 21.74828
```

```
# How does our bootstrap SE compare to ...
sd(boot_vec)
```

```
## [1] 0.0249139
```

```
##to (1) the estimated SE from one sample
sd(boot_samp$logwealth)/sqrt(nrow(boot_samp))
```

```
## [1] 0.02450389
```

```
##to (2) the true SE, which we can calculate from the population dist. SE(X_bar)^2 = true variance / n
true.var <- sum((bill$logwealth - mean(bill$logwealth))^2)/nrow(bill)
sqrt(true.var/(nrow(bill)))
```

```
## [1] 0.01963581
```

```
## We can use bootstrapping to estimate standard errors for lots of things
## Let's go back to our unrestricted linear model
## And try to retrieve the standard errors of our coefficients
## we're also going to add some tests of inputs and outputs
reps <- 1000
set.seed(12334)
boot_coefs <- function(data) {
  if (!is.data.frame(data)) stop("Data is not a data frame")
  boot_samp <- sample_n(data, nrow(data), replace = T)
  model <-  lm(data = boot_samp, logwealth ~ age + inherit + woman + woman:inherit)
  coefs <- coef(model)
  if (anyNA(coefs)) stop("Missing values in coefficients")
  return(coefs)
}
boot_coefs_out <- replicate(reps, boot_coefs(random_sample))

# Get mean and variance
apply(boot_coefs_out, MARGIN = 1, FUN = mean) #1 indicates applying the FUN=mean by ROW
```

```
##   (Intercept)           age       inherit         woman inherit:woman
##  21.153575173   0.007354533   0.255268351  -0.219598563   0.096036656
```

```
apply(boot_coefs_out, MARGIN = 1, FUN = sd)
```

```
##   (Intercept)           age       inherit         woman inherit:woman
##   0.125110982   0.001975574   0.062776227   0.147445913   0.182380861
```

```
# And the 95% confidence interval
apply(boot_coefs_out, 1, quantile, probs = c(0.025, 0.975))
```

```
##       (Intercept)         age   inherit       woman inherit:woman
## 2.5%     20.91168 0.003500956 0.1287190 -0.50989817    -0.2718850
## 97.5%    21.38982 0.011235660 0.3814313  0.08067635     0.4343014
```

```
# How does this compare to the estimates we get from the original random sample?
full_model <- lm(data = random_sample, logwealth ~ age + inherit + woman + woman:inherit)

summary(full_model)$coefficients
```

```
##                   Estimate  Std. Error     t value     Pr(>|t|)
## (Intercept)   21.155185688 0.121408167 174.2484568 0.000000e+00
## age            0.007340779 0.001877477   3.9099168 9.858637e-05
## inherit        0.255466242 0.060378987   4.2310455 2.540839e-05
## woman         -0.221837183 0.200955609  -1.1039114 2.698985e-01
## inherit:woman  0.096781004 0.221953537   0.4360417 6.629010e-01
```

```
sqrt(diag(vcov(full_model)))
```

```
##   (Intercept)           age       inherit         woman inherit:woman
##   0.121408167   0.001877477   0.060378987   0.200955609   0.221953537
```

```
compare.SEhat <-
rbind(sqrt(diag(vcov(full_model))),
      apply(boot_coefs_out, MARGIN = 1, FUN = sd))

rownames(compare.SEhat) <- c("lm", "bootstrap")
compare.SEhat
```

```
##           (Intercept)         age    inherit     woman inherit:woman
## lm          0.1214082 0.001877477 0.06037899 0.2009556     0.2219535
## bootstrap   0.1251110 0.001975574 0.06277623 0.1474459     0.1823809
```

Answer to think-pair-share 1:

-for male billionaires of same age, we observe on average the logwealth of those who inherited their wealth to be ??? higher than those who don't inherited wealth

-for **female** billionairs of same age, we observe on average the logwealth of those who inherited their wealth to be ??? higher than those who don't inherited wealth

Answer to think-pair-share 2:

```
#unrestrict <- lm(data = bill, logwealth ~ age + inherit + woman + woman:inherit)
#restrict1 <- lm(data = bill, logwealth ~ 1)
#restrict2 <- lm(data = bill, logwealth ~ age + inherit)
```

What is the null hypothesis H_0 for F-test 1 and 2?

- H_0 for Ftest1: $\beta_{age} = \beta_{inherit} = \beta_{woman} = \beta_{woman:inherit} = 0$
- H_0 for Ftest1: $\beta_{woman} = \beta_{woman:inherit} = 0$

How do we interpret the F test results?

- F-test1: we reject the null hypothesis that none of the predictors will significantly improve model fit.
- F-test2: we fail to reject the null hypothesis that including women and the interaction between woman and inherit will not significantly improve our model fit.