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.

\[
A \leftarrow \text{matrix}(c(1,2,3,4), \text{nrow} = 2, \text{ncol} = 2, \text{byrow} = \text{TRUE})
\]

\[
B \leftarrow \text{matrix}(c(1,0,1,0), \text{nrow} = 2, \text{ncol} = 2, \text{byrow} = \text{TRUE})
\]

\[
C \leftarrow \text{matrix}(c(6,5,4,3,2,1), \text{nrow} = 3, \text{ncol} = 2, \text{byrow} = \text{TRUE})
\]

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: \( \times \)
- Inverse: \text{solve}()
- Transpose: \text{t}()
- Extract the diagonal of a matrix \( A \): \text{diag}(A)
- Make a \( k \) by \( k \) identity matrix: \text{diag}(k)

Try filling in the code yourself!

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

```r
## [,1] [,2]
## [1,]  2  2
## [2,]  4  4
## A minus B
A - B
```

```r
## [,1] [,2]
## [1,]  0  2
## [2,]  2  4
## A times B
B %*% A
```

```r
## [,1] [,2]
## [1,]  1  2
## [2,]  1  2
## B times A
```

```r
## C times A
```

```r
## A times C
A %*% t(C)
```
## Inverse of A (A^(-1))

\[ \text{solve}(A) \]

\[
\begin{bmatrix}
1 & -2.0 \\
3 & 1.5
\end{bmatrix}
\]

A \%\% solve(A)

\[
\begin{bmatrix}
1 & 1.110223e-16 \\
0 & 1.000000e+00
\end{bmatrix}
\]

t(C)

\[
\begin{bmatrix}
6 & 4 & 2 \\
5 & 3 & 1
\end{bmatrix}
\]

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

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

diag(diag(4))

\[
\begin{bmatrix}
1 & 1 & 1 & 1
\end{bmatrix}
\]

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.

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

filter(year == 2014, !is.na(age), !is.na(networthusd)) %>%
dplyr::select(year, name, rank, citizenship, networthusd, selfmade, typeofwealth, gender, age, foundingdate) %>%
mutate(wealth = networthusd*1000000000, 
logwealth = log(wealth),
woman = ifelse(gender == "female", 1, 0),
inherit = ifelse(selfmade == "inherited", 1, 0))
```
## We're going to run a model with lots of covariates we'll consider the "unrestricted model"

```r
unrestrict <- lm(data = bill, logwealth ~ age + inherit + woman + woman:inherit)
summary(unrestrict)
```

```r
##
## Call:
## lm(formula = logwealth ~ age + inherit + woman + woman:inherit, 
##     data = bill)
##
## Residuals:
##     Min       1Q   Median       3Q      Max
## -1.32750 -0.56413 -0.17549  0.35509  3.46877
##
## Coefficients:  Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)   21.13384   0.09586   220.5  < 2e-16 ***
## age           0.00778   0.00148     5.3  1.68e-07 ***
## inherit       0.19315   0.04818     4.0  6.37e-05 ***
## woman        -0.14581   0.14534    -1.0  0.316
## inherit:woman  0.09874   0.16490     0.6  0.549
##
## 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

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]
```

```r
##
##                Estimate 
## (Intercept) 21.133848037
## age           0.007781897
## inherit       0.193149681
## woman        -0.145810121
## inherit:woman  0.098744147
```
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:

```r
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:

```r
##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") +
  ylab("Predicted log wealth") +
  ggtitle("Restricted model")
```

```r
```
Table 1: Comparing our linear models of billionaire wealth

<table>
<thead>
<tr>
<th></th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dependent variable:</strong></td>
<td>logwealth</td>
<td>logwealth</td>
<td>logwealth</td>
</tr>
<tr>
<td>age</td>
<td>0.008***</td>
<td>0.008***</td>
<td>0.008***</td>
</tr>
<tr>
<td></td>
<td>(0.001)</td>
<td>(0.001)</td>
<td>(0.001)</td>
</tr>
<tr>
<td>inherit</td>
<td>0.183***</td>
<td>0.193***</td>
<td>0.193***</td>
</tr>
<tr>
<td></td>
<td>(0.042)</td>
<td>(0.048)</td>
<td>(0.048)</td>
</tr>
<tr>
<td>woman</td>
<td>−0.146</td>
<td>−0.146</td>
<td>−0.146</td>
</tr>
<tr>
<td></td>
<td>(0.145)</td>
<td>(0.145)</td>
<td>(0.145)</td>
</tr>
<tr>
<td>inherit:woman</td>
<td>0.099</td>
<td>0.099</td>
<td>0.099</td>
</tr>
<tr>
<td></td>
<td>(0.165)</td>
<td>(0.165)</td>
<td>(0.165)</td>
</tr>
<tr>
<td></td>
<td>(0.020)</td>
<td>(0.095)</td>
<td>(0.096)</td>
</tr>
<tr>
<td>Observations</td>
<td>1,590</td>
<td>1,590</td>
<td>1,590</td>
</tr>
<tr>
<td>R²</td>
<td>0.000</td>
<td>0.031</td>
<td>0.032</td>
</tr>
<tr>
<td>Adjusted R²</td>
<td>0.000</td>
<td>0.030</td>
<td>0.030</td>
</tr>
<tr>
<td>Residual Std. Error</td>
<td>0.783 (df = 1589)</td>
<td>0.771 (df = 1587)</td>
<td>0.772 (df = 1585)</td>
</tr>
<tr>
<td>F Statistic</td>
<td>25.610*** (df = 2; 1587)</td>
<td>13.142*** (df = 4; 1585)</td>
<td>13.142*** (df = 4; 1585)</td>
</tr>
</tbody>
</table>

*Note:* p<0.05; **p<0.01; ***p<0.001

```r
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:

```r
## 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

\[
q_1 = 4 \\
q_2 = 2
\]

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.

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

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

```r
# 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.

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

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

```r
# 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:

```r
# 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
```
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)

```r
## 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

```r
mean(boot_vec)  # about the same as the true mean in our sample
```


```r
sd(boot_vec)
```

## [1] 0.0249139

# and the 95% confidence interval

```r
quantile(boot_vec, probs = c(0.005, 0.995))
```

## 0.5% 99.5%
## 21.62231 21.74828

# How does our bootstrap SE compare to ...

```r
sd(boot_vec)
```

## [1] 0.0249139

##to (1) the estimated SE from one sample

```r
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

```r
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

```r
tests <- 1000
set.seed(1234)

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(tests, boot_coefs(random_sample))

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

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

apply(boot_coefs_out, 1, quantile, probs = c(0.025, 0.975))

full_model <- lm(data = random_sample, logwealth ~ age + inherit + woman + woman:inherit)

summary(full_model)$coefficients

sqrt(diag(vcov(full_model)))

```

# How does this compare to the estimates we get from the original random sample?

```r
full_model <- lm(data = random_sample, logwealth ~ age + inherit + woman + woman:inherit)

summary(full_model)$coefficients

sqrt(diag(vcov(full_model)))

```

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

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

compare.SEhat
```
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** 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

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.