PS4 R Notes

Matthew Alampay Davis

October 24, 2021

Notice I've started my preamble chunk with an argument *include* = *F* (only visible if you're reading this as an .Rmd file; if you're reading the pdf, then you won't see a code chunk above). This just makes it so that when the pdf is knitted, the preamble doesn't show up. All these packages are still loaded, it just doesn't get printed. Obviously irrelevant to grading, but you might prefer your problem sets this way.

R commands

Let's use a test dataset of workers in 2015 using their average hourly earnings (ahe) as our outcome variable of interest.

```
cps <- read.dta13('CPS2015.dta')
head(cps)
```

```
##   year       ahe bachelor female age
## 1 2015 11.778846        0      0  26
## 2 2015  9.615385        0      1  33
## 3 2015 12.019231        0      0  31
## 4 2015 18.376068        0      0  32
## 5 2015 41.836735        0      0  28
## 6 2015 19.230770        0      1  31
```

```
# Creating new variables needed for the regressions
cps %<>% mutate(log.ahe = log(ahe),
                log.age = log(age),
                age2 = age^2)
```

Selecting elements from a vector by indexing

We can pick out the nth element of a vector or dataframe just by using square brackets:

```
cps$age2[1]
```

```
## [1] 676
```

gives us the squared age of the first observation. This index works because we are looking at a vector, which is one dimensional. In comparison a data.frame is two-dimensional:

```
head(cps)
```

```
##   year       ahe bachelor female age  log.ahe  log.age age2
## 1 2015 11.778846        0      0  26 2.466305 3.258097  676
## 2 2015  9.615385        0      1  33 2.263364 3.496508 1089
## 3 2015 12.019231        0      0  31 2.486508 3.433987  961
## 4 2015 18.376068        0      0  32 2.911049 3.465736 1024
## 5 2015 41.836735        0      0  28 3.733775 3.332205  784
## 6 2015 19.230770        0      1  31 2.956512 3.433987  961
```

```
dim(cps)
```

```
## [1] 7098    8
```

If we wanted to pick the element in the fifth row and sixth column, we'd just use [5,6] as so:

```
cps[5,6]
```

```
## [1] 3.733775
```

Just remember the first number is the row and the second is the column. We can also say

```
cps[5, 'age2']
```

```
## [1] 784
```

And if we wanted, for example, every column pertaining to the fifth observation, we would just leave the second argument blank to include all columns available:

```
cps[5,]
```

```
##   year      ahe bachelor female age  log.ahe  log.age age2
## 5 2015 41.83673        0      0  28 3.733775 3.332205  784
```

Selecting specific variables

We already know how to subset data by rows using the filter function:

```
test <- filter(cps, bachelor == 0)
table(test$bachelor)
```

```
## 
##    0 
## 3365
```

This gives the subset with no bachelor's degree.

Similarly, we can subset data by columns using the select function:

```
test <- select(cps, bachelor, female, age)
head(test)
```

```
##   bachelor female age
## 1        0      0  26
## 2        0      1  33
## 3        0      0  31
## 4        0      0  32
## 5        0      0  28
## 6        0      1  31
```

The first argument is the dataset we want to subset, then every argument after that separated by commas are the variables we want to keep.

Interaction terms

Regressions with interaction terms are a type of non-linear regression and they involve multiplying two covariates together:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 (X_{1i} \times X_{2i}) + u_i$$

Obviously, the last covariate is the interaction term. They are relevant when we think that X_1's effect on Y depends on the value of X_2. For example, in our data, we may think that having a college degree affects earnings and that being a woman affects earnings, but we may also suspect that the effect a college education has on earnings is different for men and women. Similarly, we may think the effect being a woman has on earnings is different for those with and without a college degree. If so, then we want to include an interaction term to capture this relationship.

There are a couple ways to do this. The first is that we define a new variable that is the product of the female and bachelor variables then include it as a regressor in our regression formula:

```
cps %<>% mutate(female.bachelor = female*bachelor)
int.model.1 <- lm_robust(ahe ~ female + bachelor + female.bachelor, cps, se_type = 'stata')
summary(int.model.1)
```

```
##
## Call:
## lm_robust(formula = ahe ~ female + bachelor + female.bachelor,
##     data = cps, se_type = "stata")
##
## Standard error type:  HC1
##
## Coefficients:
##                 Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper   DF
## (Intercept)       17.498     0.1915   91.37 0.000e+00   17.123  17.8739 7094
## female            -3.289     0.2820  -11.67 3.669e-31   -3.842  -2.7367 7094
## bachelor          10.557     0.3799   27.79 2.156e-161    9.812  11.3017 7094
## female.bachelor   -1.727     0.5064   -3.41 6.535e-04   -2.720  -0.7341 7094
##
## Multiple R-squared:  0.175 , Adjusted R-squared:  0.1746
## F-statistic:   531 on 3 and 7094 DF,  p-value: < 2.2e-16
```

The alternative is to just use a colon (":") in the formula without needing to define a new variable like so:

```
int.model.2 <- lm_robust(ahe ~ female + bachelor + female:bachelor, cps, se_type = 'stata')
summary(int.model.2)
```

```
##
## Call:
## lm_robust(formula = ahe ~ female + bachelor + female:bachelor,
##     data = cps, se_type = "stata")
##
## Standard error type:  HC1
##
## Coefficients:
##                 Estimate Std. Error t value   Pr(>|t|) CI Lower CI Upper   DF
## (Intercept)       17.498     0.1915   91.37  0.000e+00   17.123  17.8739 7094
## female            -3.289     0.2820  -11.67  3.669e-31   -3.842  -2.7367 7094
## bachelor          10.557     0.3799   27.79 2.156e-161    9.812  11.3017 7094
## female:bachelor   -1.727     0.5064   -3.41  6.535e-04   -2.720  -0.7341 7094
##
## Multiple R-squared:  0.175 , Adjusted R-squared:  0.1746
## F-statistic:   531 on 3 and 7094 DF,  p-value: < 2.2e-16
```

Writing dollar signs

If we want to write an amount of money outside of a chunk, we have to prefix the dollar sign with a backslash or else R will think you're trying to write an equation. So forty dollars is $40. A bit silly and trivial, but it comes up in this week's problem set.

Creating predictions from an lm or lm_robust model for certain values of the regressors

Occasionally, we'll get a question like "Based on the regression you just ran, what is the expected value of y for a hypothetical observation with these given values for x1, x2, and x3?" We've seen this in past problem sets, but it'll be convenient this week to use a new command.

As an example, refer to the interaction models we defined above where the y is hourly earnings and the covariates are female, bachelor, and their interaction.

Suppose we want to know the model's predicted value for a male college graduate. Then we could use the *predict* command and use the model as an input. First, let's create the corresponding observation:

```
male.college.graduate <- data.frame(female = 0,
                                    bachelor = 1)
```

We here defined a new observation in terms of the variables that went into defining the model. Now let's feed it into the predict command, whose first argument is the regression model and that takes in a new argument called "newdata" which we'll set to our new observation:

```
predict(int.model.2, newdata = male.college.graduate)
```

```
##        1
## 28.05536
```

This model predicts an average hourly earnings of $28.06 for a a person who is male and has a college degree.

Note that we had to use int.model.2 because int.model.1 also includes a third variable, the created interaction term. To use predict on that model, we would have to create the new observation accordingly:

```
male.college.graduate <- data.frame(female = 0,
                                    bachelor = 1,
                                    female.bachelor = 0)
predict(int.model.1, newdata = male.college.graduate)
```

```
##        1
## 28.05536
```

We can also create several observations to be predicted and the predict function will generate predictions for all of them:

```
new.data <- data.frame(female = c(0, 0, 1, 1),
                       bachelor = c(1, 0, 1, 0))
new.predict <- predict(int.model.2, newdata = new.data)
new.predict
```

```
##        1        2        3        4
## 28.05536 17.49846 23.03898 14.20896
```

These are predicted earnings for 1) a male college graduate, 2) a male without a college degree, 3) a female college graduate, and 4) a female without a college degree.

If we were asked how earnings are expected to change for a male individual if they were to get a college degree, we would subtract prediction 1 from prediction 2:

```
new.predict[1]-new.predict[2]
```

```
##       1
## 10.5569
```

And we'd say something like “Having a college degree increases expected hourly earnings by $10.5569.”

With this in mind, consider the following model, that adds the age regressor:

```
model.2 <- lm_robust(ahe ~ female + bachelor + female:bachelor + age, cps, se_type = 'stata')
summary(model.2)
```

```
##
## Call:
## lm_robust(formula = ahe ~ female + bachelor + female:bachelor +
##     age, data = cps, se_type = "stata")
##
## Standard error type:  HC1
##
## Coefficients:
##                 Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper   DF
## (Intercept)       1.8893    1.32158   1.430 1.529e-01  -0.7013   4.4800 7093
## female           -3.3073    0.28178 -11.737  1.606e-31  -3.8597  -2.7549 7093
## bachelor         10.4596    0.37584  27.830 7.301e-162   9.7229  11.1964 7093
## age               0.5269    0.04466  11.798  7.930e-32   0.4394   0.6145 7093
## female:bachelor  -1.5145    0.50379  -3.006  2.654e-03  -2.5021  -0.5270 7093
##
## Multiple R-squared:  0.1906 ,    Adjusted R-squared:  0.1901
## F-statistic: 423.3 on 4 and 7093 DF,  p-value: < 2.2e-16
```

Now suppose we are again asked how earnings are expected to change for a male individual if they were to get a college degree. To use the predict command, we would do the same thing but we'd also have to include a value for age. However, we aren't given a particular value of age to fix it to so what do we do?

Notice that age enters independently here: it does not interact with any other covariates. So no matter what age the observation is, as long as we keep it fixed, the effect of age will disappear once we difference the predictions since both predictions will have +0.5269 times age in it. In other words, we can set it to any number as long as it's the same. Let's say 20 years old:

```
new.data <- data.frame(female = c(0, 0, 1, 1),
                       bachelor = c(1, 0, 1, 0),
                       age = c(20, 20, 20, 20))
new.predict <- predict(model.2, newdata = new.data)
new.predict
```

```
##         1         2         3         4
## 22.887703 12.428066 18.065876  9.120782
```

Then again, we'd subtract 2 from 1 to get the effect on average hourly earnings of attaining a college degree, now controlling for age effects.

```
new.predict[1]-new.predict[2]
```

```
##        1
## 10.45964
```

Practice Problems

There was some confusion about the practice problem solutions not matching the questions in Stock and Watson so these questions are not the ones listed on the problem set but do solve the questions they are named after in the 2019 edition of Stock and Watson. They correspond to the practice questions listed in the problem set from last spring, which you can find in my Recitation 5 folder along with the solutions to the non-empirical questions from last year.

Some topics these questions cover:

Multicollinearity: regressions dropping the intercept and regressions dropping a covariate

We've talked about dealing with multicollinearity. We've also demonstrated that the two ways of dealing with collinearity are exactly equivalent (see Points of Emphasis from Recitation 4). Still, we may be interested in using a specific one of the two. Practice Question 1c shows us how to do this.

Summarizing by group

We know how to take means and standard deviations of variables. But suppose we want to calculate separate means and standard deviations for different subsets. See Practice Question 2 for an example that also demonstrates the efficiency and readability of using piping and the "group_by" function from dplyr.

Plotting by group

Suppose we want to draw a plot with separate lines of best fit for different subsets of the data so we can compare their slopes. Or suppose we want to draw a scatter plot and color points according to different values they take on (for example, color female observations different from male observations). ggplot lets us do this very efficiently See Practice Question 2bii.

Question 1: see the old problem set questions

Load the data

```
tanf <- read.dta13('tanf2.dta')
head(tanf)
```

```
##   obs      black blue mdinc midwest northeast      obsid south tanfreal west
## 1   1 0.25237498    0 20800       0         0 -1.0000000     1 186.4334    0
## 2   2 0.04035139    0 33000       0         0 -1.0000000     0 499.9504    1
## 3   3 0.03002869    0 25100       0         0 -0.9999999     0 279.0307    1
## 4   4 0.15886758    0 19500       0         0 -0.9999999     1 231.9922    0
## 5   5 0.07388321    1 28000       0         0 -0.9999999     0 394.9782    1
## 6   6 0.03983221    0 29000       0         0 -0.9999999     0 267.6726    1
```

Part a

We want to test whether there is a difference between the welfare programs of Midwest states and all other states

$$H_0 : \beta_{\text{midwest}} = \beta_{\text{black}\times\text{midwest}} = \beta_{\text{blue}\times\text{midwest}} = 0$$

Under the null hyptohesis, the expected level of benefits does not depend on whether the state is in the Midwest or not and so the coefficient on midwest will not be significantly different from zero and all the coefficients on all the interactions with midwest will be captured by their main effects.

Part b

The regression we run is the following:

```
q1.mod1 <- lm_robust(tanfreal ~ black + blue + midwest + black:midwest + blue:midwest, tanf, se_type =
summary(q1.mod1)
```

```
##
## Call:
## lm_robust(formula = tanfreal ~ black + blue + midwest + black:midwest +
##     blue:midwest, data = tanf, se_type = "stata")
```

```
##
## Standard error type:  HC1
##
## Coefficients:
##                Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper DF
## (Intercept)      347.53      25.84  13.450 5.253e-17   295.42   399.64 43
## black           -522.03     112.36  -4.646 3.196e-05  -748.62  -295.44 43
## blue              31.76      27.47   1.156 2.539e-01   -23.63    87.16 43
## midwest          141.42      34.07   4.150 1.539e-04    72.70   210.14 43
## black:midwest -1420.53     350.34  -4.055 2.070e-04 -2127.05  -714.01 43
## blue:midwest     204.14      42.86   4.763 2.194e-05   117.70   290.59 43
##
## Multiple R-squared:  0.7066 ,    Adjusted R-squared:  0.6725
## F-statistic: 45.59 on 5 and 43 DF,  p-value: 4.148e-16
```

Writing the model as a regression equation with standard errors in parentheses underneath each coefficient:

I really don't like this question asking for parentheses right beneath it because it's hard to implement on statistical software. This might be better done by hand or Microsoft Word, but here's my attempt in R:

$$\hat{\text{tanfreal}} = 347.53 - 522.03 \times \text{black} + 31.76 \times \text{blue} + 141.42 \times \text{midwest} - 1420.53 \times \text{black} \times \text{midwest} + 204.14 \times \text{blue} \times \text{midwest}$$

$$(25.84)(112.36)(27.47)(34.07)(350.34)(42.86)$$

Performing the test for the null hypothesis in a:

```
linearHypothesis(q1.mod1, c('midwest = 0', 'black:midwest = 0', 'blue:midwest = 0'), test = 'F')
```

```
## Linear hypothesis test
##
## Hypothesis:
## midwest = 0
## black:midwest = 0
## blue:midwest = 0
##
## Model 1: restricted model
## Model 2: tanfreal ~ black + blue + midwest + black:midwest + blue:midwest
##
##   Res.Df Df      F    Pr(>F)
## 1     46
## 2     43  3 26.684 6.615e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We reject the null hypothesis and conclude that Midwestern states have significantly different welfare programs compared with non-Midwest ones.

Part c

Note for this question, we want the intercept omitted

```
tanf %<>% mutate(nonmidwest = 1-midwest)
q1.mod2 <- lm_robust(tanfreal ~ 0 + nonmidwest + black:nonmidwest + blue:nonmidwest + midwest + black:mi
summary(q1.mod2)
```

```
##
## Call:
## lm_robust(formula = tanfreal ~ 0 + nonmidwest + black:nonmidwest +
##     blue:nonmidwest + midwest + black:midwest + blue:midwest,
##     data = tanf)
##
## Standard error type:  HC2
##
## Coefficients:
##                   Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper DF
## nonmidwest          347.53      25.17  13.808 2.090e-17   296.77    398.3 43
## midwest             488.96      24.23  20.181 1.489e-23   440.10    537.8 43
## nonmidwest:black  -522.03     110.30  -4.733 2.415e-05  -744.47   -299.6 43
## nonmidwest:blue      31.76      26.84   1.183 2.432e-01   -22.37     85.9 43
## black:midwest    -1942.56     376.70  -5.157 6.049e-06 -2702.25  -1182.9 43
## blue:midwest        235.91      37.88   6.228 1.705e-07   159.51    312.3 43
##
## Multiple R-squared:  0.9692 ,    Adjusted R-squared:  0.9649
## F-statistic: 274.6 on 6 and 43 DF,  p-value: < 2.2e-16
```

Adding the "0 +" term to the formula ensures there is no intercept term. Alternatively, we could have also added a "-1" to the formula and accomplished the same thing.

The hypothesis of no differences in welfare programs would equate main effects and interaction effects

$$H_0 : \gamma_{nonmidwest} = \gamma_{midwest}, \gamma_{nonmidwest \times black} = \gamma_{midwest \times black}, \gamma_{nonmidwest \times blue} = \gamma_{midwest \times blue}$$

The relationship between the parameters of this model and those of the original are such that

$$\begin{aligned}
\beta_0 &= \gamma_{nonmidwest} \\
\beta_{black} &= \gamma_{black \times nonmidwest} \\
\beta_{blue} &= \gamma_{blue \times nonmidwest} \\
\beta_{midwest} &= \gamma_{midwest} - \gamma_{nonmidwest} \\
\beta_{black \times midwest} &= \gamma_{black \times midwest} - \gamma_{black \times nonmidwest} \\
\beta_{blue \times midwest} &= \gamma_{blue \times midwest} - \gamma_{blue \times nonmidwest}
\end{aligned}$$

So the two models are just different parameterizations of the same equations. The coefficients and their numbers are different insofar as they measure effects relative to different baselines. In the model with an intercept but with an omitted category, the effects are relative to the omitted category. In the model without an intercept but not omitting a category, the effects are relative to average.

I'm not gonna bother writing the regression equation here since you get the point.

Part d

By including an intercept term in the model, the model will suffer from multicollinearity. Thus, we cannot estimate the model by OLS since the model is overparameterized.

Question 2: Stock-Watson Empirical Exercise E8.1

Loading the data:

```
lead <- read.dta13('Lead_Mortality.dta')
head(lead)
```

```
##   year      city state      age hardness  ph   infrate typhoid_rate np_tub_rate
## 1 1900   Alameda    CA 28.95484       97 7.6 0.1097561   0.02439024 0.030487806
## 2 1900    Albany    NY 30.34768       43 7.3 0.2986185   0.04144527 0.013815090
## 3 1900 Allegheny    PA 27.08730      111 7.3 0.4468413   0.09399076 0.027734976
## 4 1900 Allentown    PA 27.76405      176 7.7 0.3841808   0.02824859 0.005649718
## 5 1900   Altoona    PA 27.03753      111 7.3 0.4678663   0.04370180 0.007712082
## 6 1900 Amsterdam    NY 28.60989       43 7.3 0.3062201   0.01435407 0.019138755
##    mom_rate population precipitation temperature lead foreign_share
## 1 0.1951219        164      1.850307    59.02617    0    0.19354838
## 2 0.1795962        941      3.278735    45.26791    1    0.18488371
## 3 0.1926040       1298      3.350943    48.97149    1    0.23730159
## 4 0.1977401        354      3.350943    48.97149    1    0.07865169
## 5 0.1825193        389      3.350943    48.97149    0    0.11260054
## 6 0.1626794        209      3.278735    45.26791    1    0.25274727
```

Part a

```
# Method 1: in one command
lead %>%
  group_by(lead) %>%
  summarize(mean = mean(infrate),
            sd = sd(infrate))
```

```
## # A tibble: 2 x 3
##    lead  mean    sd
##   <dbl> <dbl> <dbl>
## 1     0 0.381 0.148
## 2     1 0.403 0.153
```

```
# Method 2: creating two dataframes
filter(lead, lead == 1) %$%  # Pipe to select the variable infrate
  infrate %>% # Pipe to perform a function on this variable
  mean
```

```
## [1] 0.4032576
```

```
filter(lead, lead == 1) %$%
  infrate %>%
  sd
```

```
## [1] 0.1530873
```

```
filter(lead, lead == 0) %$%
  infrate %>%
  mean
```

```
## [1] 0.3811679
```

```
filter(lead, lead == 0) %$%
  infrate %>%
  sd
```

```
## [1] 0.1477588
```

Part b

Running the regression:

```
## Method 1
lead %<>% mutate(lead.ph = lead*ph)
lead.mod <- lm_robust(infrate ~ lead + ph + lead.ph, lead, se_type = 'stata')
## Method 2
lead.mod <- lm_robust(infrate ~ lead + ph + lead:ph, lead, se_type = 'stata')
summary(lead.mod)
```

```
##
## Call:
## lm_robust(formula = infrate ~ lead + ph + lead:ph, data = lead,
##     se_type = "stata")
##
## Standard error type:  HC1
##
## Coefficients:
##              Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper  DF
## (Intercept)  0.91890    0.15049   6.106 6.866e-09  0.62180  1.21601 168
## lead         0.46180    0.20761   2.224 2.746e-02  0.05193  0.87167 168
## ph          -0.07518    0.02095  -3.588 4.369e-04 -0.11654 -0.03381 168
## lead:ph     -0.05686    0.02808  -2.025 4.448e-02 -0.11230 -0.00142 168
##
## Multiple R-squared:  0.2719 ,    Adjusted R-squared:  0.2589
## F-statistic: 20.97 on 3 and 168 DF,  p-value: 1.366e-11
```

i) Explaining the coefficients The first coefficient is the intercept, which shows the level of Infrate when lead = 0 and pH = 0. It dictates the level of the regression line. The second coefficient and fourth coefficients measure the effect of lead on the infant mortality rate. Comparing 2 cities, one with lead pipes (lead = 1) and one without lead pipes (lead = 0), but the same of pH, the difference in predicted infant mortality rate is

$$0.462 - 0.057 \times pH$$

The third and fourth coefficients measure the effect of pH on the infant mortality rate. Comparing 2 cities, one with a pH of 6 and the other with a pH of 5, but the same leadedness, the difference in predicted infant mortality rate is

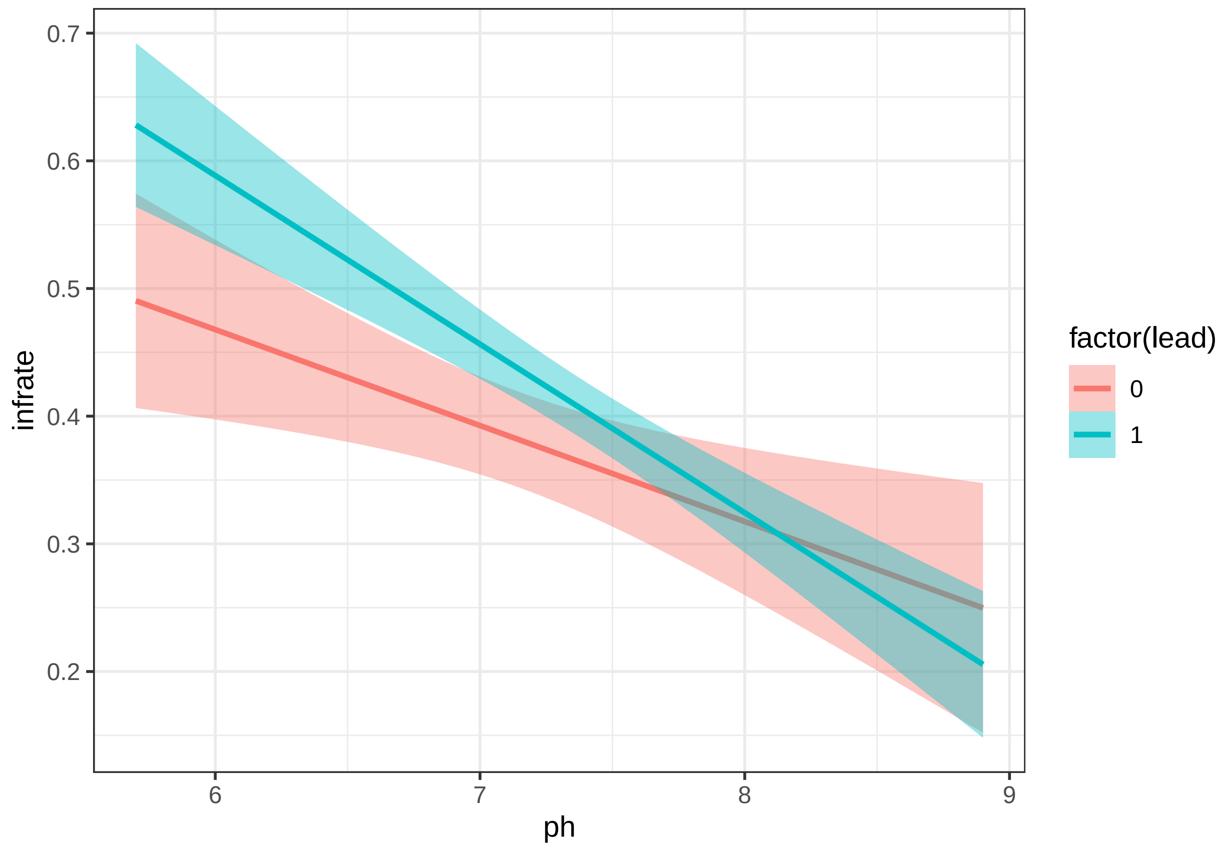
$$-0.075 - 0.057 \times lead$$

so the difference is -0.075 for cities without lead pipes and -0.132 for cities with lead pipes.

```
ggplot(lead, aes(x = ph, y = infrate, fill = factor(lead), color = factor(lead))) +
  theme_bw() +
  geom_smooth(method = 'lm')
```

ii) Plotting estimated regression functions for different subgroups

```
## `geom_smooth()` using formula 'y ~ x'
```



Notice we have turned the variable *lead* into a factor. The factor function converts a numerical variable into a categorical variable so that all values that it takes on are distinct groups. In ggplot, this means that when we set the fill and color colors to depend on this factor so that when we use geom_smooth to plot lines of best fit, it'll create separate lines for the set of observations with lead == 1 and for lead == 0 (the only two values lead takes in this data).

The infant mortality rate is higher for cities with lead pipes, but the difference declines as the pH level increases.

```r
linearHypothesis(lead.mod, c('lead = 0', 'lead:ph = 0'), test = 'F')
```

iii) Does lead have a statistically significant effect on infant mortality?

```
## Linear hypothesis test
##
## Hypothesis:
## lead = 0
## lead:ph = 0
##
## Model 1: restricted model
## Model 2: infrate ~ lead + ph + lead:ph
##
##   Res.Df Df     F  Pr(>F)
## 1    170
## 2    168  2 3.936 0.02135 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F-statistic for the coefficient on lead and the interaction term is F = 3.94, which has a p- value of 0.02, so the coefficients are jointly statistically significantly different from zero at the 5% but not the 1% significance level.

iv) Does the effect of lead on infant mortality depend on pH? Is this dependence statistically significant? The interaction term has a t-statistic of t = -2.02, so the coefficient is significant at the 5% but not the 1% significance level.

```r
mean(lead$ph)
```

v) Average value of pH

```
## [1] 7.322674
```

At this level, the estimated effect of lead on infant mortality is

```r
lead.1 <- data.frame(lead = 1, ph = mean(lead$ph))
lead.0 <- data.frame(lead = 0, ph = mean(lead$ph))
predict(lead.mod, newdata = lead.1)-predict(lead.mod, newdata = lead.0)
```

```
##          1
## 0.04541495
```

The standard deviation of pH is

```
sd(lead$ph)
```

```
## [1] 0.6917288
```

The estimated effect of lead on infant mortality when the pH is one standard deviation lower than average level of pH in the sample is given by

```
lead.sd1 <- data.frame(lead = 1, ph = mean(lead$ph)-sd(lead$ph))
lead.sd0 <- data.frame(lead = 0, ph = mean(lead$ph)-sd(lead$ph))
predict(lead.mod, newdata = lead.sd1)-predict(lead.mod, newdata = lead.sd0)
```

```
##          1
## 0.08474818
```

If the pH were one standard deviation larger than average:

```
lead.sd1 <- data.frame(lead = 1, ph = mean(lead$ph)+sd(lead$ph))
lead.sd0 <- data.frame(lead = 0, ph = mean(lead$ph)+sd(lead$ph))
predict(lead.mod, newdata = lead.sd1)-predict(lead.mod, newdata = lead.sd0)
```

```
##           1
## 0.006081724
```

vi) Constructing a 95% confidence interval for the effect of lead on infant mortality when pH = 6.5 Referring to method 2 of section 7.3. We add and subtract $6.5\beta_3$ to the regression:

$$\text{Infrate} = \beta_0 + (\beta_1 + 6.5\beta_3)\text{lead} + \beta_2\text{pH} + \beta_3[\text{lead} \times \text{pH} - 6.5 \times \text{lead}]$$

Estimating this regression

```
lead2 <- mutate(lead, x1 = lead, x2 = ph, x3 = lead*ph-6.5*lead)
lm_robust(infrate ~ x1 + x2 + x3, lead2)
```

```
##                Estimate Std. Error   t value     Pr(>|t|)    CI Lower
## (Intercept)  0.91890383 0.15207665  6.042373 9.498187e-09  0.61867637
## x1           0.09219405 0.03300108  2.793668 5.816998e-03  0.02704381
## x2          -0.07517915 0.02118934 -3.547970 5.034252e-04 -0.11701084
## x3          -0.05686222 0.02833735 -2.006617 4.639440e-02 -0.11280539
##                  CI Upper  DF
## (Intercept)  1.2191312933 168
## x1           0.1573442832 168
## x2          -0.0333474628 168
## x3          -0.0009190399 168
```

Then the relevant confidence interval is the one on x1: 0.027 to 0.157

Part c

There are several demographic variables in the dataset. You should add these and see if the conclusions from (b) change in an important way.

(Sorry for unsatisfying answer; these were the solutions provided)

Question 3: Stock-Watson Empirical Exercise 8.2

One thing to note here is that this data comes from 2015 whereas the solutions seem to use 2012 data so the estimates are slightly different. When I'm comparing models below, I"m copying and pasting the official answers provided so they may actually be incompatible with the results being displayed. I only include them because they show you how you'd want to answer the question, not because the results are necessarily correct.

Load data:

```
cps <- read.dta13('CPS2015.dta')
head(cps)
```

```
##   year       ahe bachelor female age
## 1 2015 11.778846        0      0  26
## 2 2015  9.615385        0      1  33
## 3 2015 12.019231        0      0  31
## 4 2015 18.376068        0      0  32
## 5 2015 41.836735        0      0  28
## 6 2015 19.230770        0      1  31
```

```
# Creating new variables needed for the regressions
cps %<>% mutate(log.ahe = log(ahe),
                log.age = log(age),
                age2 = age^2)
```

This question asks us to run several regressions so I think it's convenient to just run them all at the beginning then refer to them as needed:

```
q3.mod.a <- lm_robust(ahe ~ age + female + bachelor, cps, se_type = 'stata')
q3.mod.b <- lm_robust(log.ahe ~ age + female + bachelor, cps, se_type = 'stata')
q3.mod.c <- lm_robust(log.ahe ~ log.age + female + bachelor, cps, se_type = 'stata')
q3.mod.d <- lm_robust(log.ahe ~ age + age2 + female + bachelor, cps, se_type = 'stata')
q3.mod.i <- lm_robust(log.ahe ~ age + age2 + female + bachelor + female:bachelor, cps, se_type = 'stata
```

All the subquestions also ask us to look at the effect of age increasing from 25 to 26 and from 33 to 34 so we also define those scenarios here. Since we will be interested in changes, we set sex to female and bachelor to 1 arbitrarily (these will wash out when we take differences anyway).

```
age.25 <- data.frame(age = 25) %>%
  mutate(age2 = age^2, log.age = log(age),
         female = 1, bachelor = 1)
age.26 <- data.frame(age = 26) %>%
  mutate(age2 = age^2, log.age = log(age),
         female = 1, bachelor = 1)
age.33 <- data.frame(age = 33) %>%
  mutate(age2 = age^2, log.age = log(age),
         female = 1, bachelor = 1)
age.34 <- data.frame(age = 34) %>%
  mutate(age2 = age^2, log.age = log(age),
         female = 1, bachelor = 1)
```

Part a

```
summary(q3.mod.a)
```

```
##
## Call:
## lm_robust(formula = ahe ~ age + female + bachelor, data = cps,
##     se_type = "stata")
##
## Standard error type:  HC1
##
## Coefficients:
##             Estimate Std. Error t value   Pr(>|t|) CI Lower CI Upper   DF
## (Intercept)   2.0448    1.32418   1.544  1.226e-01  -0.5510   4.6406 7094
## age           0.5313    0.04456  11.924  1.816e-32   0.4439   0.6186 7094
## female       -4.1435    0.26235 -15.794  2.983e-55  -4.6578  -3.6292 7094
## bachelor      9.8456    0.26130  37.679 2.197e-283   9.3334  10.3579 7094
##
## Multiple R-squared:  0.1896 ,	Adjusted R-squared:  0.1893
## F-statistic: 519.1 on 3 and 7094 DF,  p-value: < 2.2e-16
```

```
predict(q3.mod.a, newdata = age.26)-predict(q3.mod.a, newdata = age.25)
```

```
##         1
## 0.5312752
```

```
predict(q3.mod.a, newdata = age.34)-predict(q3.mod.a, newdata = age.33)
```

```
##         1
## 0.5312752
```

Part b

```
summary(q3.mod.b)
```

```
##
## Call:
## lm_robust(formula = log.ahe ~ age + female + bachelor, data = cps,
##     se_type = "stata")
##
## Standard error type:  HC1
##
## Coefficients:
##             Estimate Std. Error t value   Pr(>|t|) CI Lower CI Upper   DF
## (Intercept)  2.02736   0.060012   33.78 4.072e-232  1.90972  2.14500 7094
## age          0.02419   0.001997   12.12  1.848e-33  0.02028  0.02811 7094
## female      -0.17762   0.011504  -15.44  6.371e-53 -0.20017 -0.15507 7094
## bachelor     0.46150   0.011461   40.27 1.855e-319  0.43903  0.48397 7094
##
## Multiple R-squared:  0.2084 ,	Adjusted R-squared:  0.208
## F-statistic: 621.5 on 3 and 7094 DF,  p-value: < 2.2e-16
```

```
predict(q3.mod.b, newdata = age.26)-predict(q3.mod.a, newdata = age.25)
```

```
##         1
## -18.08859
```

```
predict(q3.mod.b, newdata = age.34)-predict(q3.mod.a, newdata = age.33)
```

```
##         1
## -22.14526
```

Part c

```
summary(q3.mod.c)
```

```
##
## Call:
## lm_robust(formula = log.ahe ~ log.age + female + bachelor, data = cps,
##     se_type = "stata")
##
## Standard error type:  HC1
##
## Coefficients:
##             Estimate Std. Error t value   Pr(>|t|) CI Lower CI Upper   DF
## (Intercept)   0.3233    0.19860   1.628  1.036e-01 -0.06605   0.7126 7094
## log.age       0.7154    0.05857  12.214  5.724e-34  0.60056   0.8302 7094
## female       -0.1775    0.01150 -15.435  6.894e-53 -0.20008  -0.1550 7094
## bachelor      0.4615    0.01146  40.276 1.329e-319  0.43906   0.4840 7094
##
## Multiple R-squared:  0.2086 ,    Adjusted R-squared:  0.2083
## F-statistic: 622.4 on 3 and 7094 DF,  p-value: < 2.2e-16
```

```
predict(q3.mod.c, newdata = age.26)-predict(q3.mod.a, newdata = age.25)
```

```
##         1
## -18.09079
```

```
predict(q3.mod.c, newdata = age.34)-predict(q3.mod.a, newdata = age.33)
```

```
##         1
## -22.14908
```

Part d

```
summary(q3.mod.d)
```

```
##
## Call:
## lm_robust(formula = log.ahe ~ age + age2 + female + bachelor,
##     data = cps, se_type = "stata")
##
## Standard error type:  HC1
##
## Coefficients:
##             Estimate Std. Error  t value   Pr(>|t|)  CI Lower   CI Upper   DF
## (Intercept)  0.41874  0.6695754   0.6254  5.317e-01 -0.893823  1.7313126 7093
## age          0.13412  0.0456102   2.9405  3.288e-03  0.044706  0.2235248 7093
## age2        -0.00186  0.0007713  -2.4118  1.590e-02 -0.003372 -0.0003483 7093
## female      -0.17736  0.0114993 -15.4240  8.079e-53 -0.199906 -0.1548224 7093
## bachelor     0.46163  0.0114557  40.2969 6.705e-320  0.439173  0.4840859 7093
##
## Multiple R-squared:  0.209 , Adjusted R-squared:  0.2086
## F-statistic: 467.9 on 4 and 7093 DF,  p-value: < 2.2e-16
```

```
predict(q3.mod.d, newdata = age.26)-predict(q3.mod.a, newdata = age.25)
```

```
##         1
## -18.09635
```

```
predict(q3.mod.d, newdata = age.34)-predict(q3.mod.a, newdata = age.33)
```

```
##         1
## -22.16657
```

Part e

The regressions differ in their choice of one of the regressors. They can be compared on the basis of the R2 . The regression in (3) has a (marginally) higher R2 , so it is preferred.

Part f

The regression in (4) adds the variable Age2 to regression (2). The coefficient on Age2 is not statistically significant (t = -1.72) and the estimated coefficient is very close to zero. This suggests that (2) is preferred to (4), the regressions are so similar that either may be used.

Part g

The regressions differ in their choice of the regressors (ln(Age) in (3) and Age and Age2 in 2 (4)). They can be compared on the basis of the R . The regression in (4) has a (marginally) 2 higher R , so it is preferred.

Parts h-l skipped in the solutions so skipped here as well

Question 4: Stock-Watson Empirical Exercise E9.1

This question uses the same data. This question doesn't require much programming so please just refer to the posted solutions in my recitation folder.