Recitation 4: Nonlinear regressions and variable interactions

Matthew Alampay Davis

June 20, 2023

Regressions with interactions

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 few equivalent ways to do this. Starting with what I think is the most convenient:

Method (1)

The easiest is to put an asterisk '*' in between the two variables you want to interact:

```
int.model.1 <- lm_robust(ahe ~ female*bachelor, cps, se_type = 'HC1')
summary(int.model.1)
```

```
##
## Call:
## lm_robust(formula = ahe ~ female * bachelor, data = cps, se_type = "HC1")
##
## 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
```

Conveniently, this automatically adds the two interacting variables separately so no need to think about whether you've included all the 'main' effects

Method (2)

A second way is to use the colon ':' in between the two variables you want to interact. The difference here is doing so does not automatically include the main effects:

```
int.model.2a <- lm_robust(ahe ~ female:bachelor, cps, se_type = 'HC1')
summary(int.model.2a)
```

```
##
## Call:
## lm_robust(formula = ahe ~ female:bachelor, data = cps, se_type = "HC1")
##
## Standard error type:  HC1
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper   DF
## (Intercept)       20.618     0.1701 121.215 0.00e+00   20.285   20.951 7096
## female:bachelor    2.421     0.3134   7.725 1.27e-14    1.807    3.035 7096
##
## Multiple R-squared:  0.007592 ,  Adjusted R-squared:  0.007452
## F-statistic: 59.68 on 1 and 7096 DF,  p-value: 1.27e-14
```

We'll almost always want to include the main effects so a complete implementation has a longer formula:

```
int.model.2b <- lm_robust(ahe ~ female + bachelor + female:bachelor, cps, se_type = 'HC1')
summary(int.model.2b)
```

```
##
## Call:
## lm_robust(formula = ahe ~ female + bachelor + female:bachelor,
##     data = cps, se_type = "HC1")
##
## 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
```

Method (3)

Finally, we could also 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.new <- mutate(cps, female.bachelor = female*bachelor)
int.model.3 <- lm_robust(ahe ~ female + bachelor + female.bachelor, cps.new, se_type = 'HC1')
summary(int.model.3)
```

```
##
## Call:
## lm_robust(formula = ahe ~ female + bachelor + female.bachelor,
##     data = cps.new, se_type = "HC1")
##
## 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
```

A bit more circuitous but all three methods are equivalent: you can see the ones that include the main effects all produce the exact same estimates. The only difference is that in the third method, the name of the interactive term uses a period '.' instead of a colon ':'. I'll tend to favor method 1.

Here's how we'd conduct a linear Hypothesis test including female:bachelor":

```
linearHypothesis(int.model.1, c('bachelor = 0', 'female:bachelor = 0'), test = 'F')
```

```
## Linear hypothesis test
##
## Hypothesis:
## bachelor = 0
## female:bachelor = 0
##
## Model 1: restricted model
## Model 2: ahe ~ female * bachelor
##
##   Res.Df Df      F    Pr(>F)
## 1   7096
## 2   7094  2 733.72 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Some more data cleaning

Summarizing data 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 1 for an example that also demonstrates the efficiency and readability of using piping and the "group_by" function from tidyverse.

Plotting data 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 differently from male observations). ggplot2 lets us do this very efficiently as we'll see in Practice Question 1 part b-ii.

Practice Question 1: Stock-Watson Empirical Exercise E8.1

Preview the data:

```
head(lead)
```

```
## # A tibble: 6 x 15
##    year city      state   age hardness    ph infrate typhoid_rate np_tub_rate
##   <dbl> <chr>     <chr> <dbl>    <dbl> <dbl>   <dbl>        <dbl>       <dbl>
## 1  1900 Alameda   CA     29.0       97  7.60   0.110       0.0244     0.0305
## 2  1900 Albany    NY     30.3       43  7.30   0.299       0.0414     0.0138
## 3  1900 Allegheny PA     27.1      111  7.30   0.447       0.0940     0.0277
## 4  1900 Allentown PA     27.8      176  7.70   0.384       0.0282     0.00565
## 5  1900 Altoona   PA     27.0      111  7.30   0.468       0.0437     0.00771
## 6  1900 Amsterdam NY     28.6       43  7.30   0.306       0.0144     0.0191
## # i 6 more variables: mom_rate <dbl>, population <dbl>, precipitation <dbl>,
## #   temperature <dbl>, lead <dbl>, foreign_share <dbl>
```

We will be investigating the effects of early-20th century lead contamination on infant mortality.

Part a: Compute the average infant mortality rate (*Inf*) for cities with lead pipes and for cities with nonlead pipes. Is there a statistically significant difference in the averages?

Question's simple enough so just for kicks, here's two equivalent ways of answering the question:

Method 1: using the group_by and reframe functions from tidyverse:

```
# Method 1: in one command
lead %>% # The dataset lead
  group_by(lead) %>% # Here, lead refers to the variable lead contained in the dataset also named lead
  reframe(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: doing separate calculations for each category of lead:

```
# Lead pipes
filter(lead, lead == 1) %$%  # %$% pipe to select the variable infrate
  infrate %>%
  mean
```

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

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

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

```
# Non-lead pipes
filter(lead, lead == 0) %$%
  infrate %>%
  mean
```

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

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

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

Part b: The amount of lead leached from lead pipes depends on the chemistry of the water running through the pipes. The more acidic the water is (that is, the lower its *pH*), the more lead is leached. Run a regression of *Inf* on *Lead*, *pH*, and the interaction term $Lead \times pH$.

Running the regression:

```
lead.mod <- lm_robust(infrate ~ lead*ph, lead, se_type = 'HC1')
summary(lead.mod)
```

```
##
## Call:
## lm_robust(formula = infrate ~ lead * ph, data = lead, se_type = "HC1")
##
## 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
```

b-i) The regression includes four coefficients (the intercept and the three coefficients multiplying the regressors). Explain what each coefficient measures.

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 two cities, one with lead pipes ($lead = 1$) and one without lead pipes ($lead = 0$), but with the same pH, the difference in predicted infant mortality rate is

$$0.46180 - 0.05686 \times pH$$

Thus, the effect of lead contamination depends on the level of acidity that we're holding fixed.

The third and fourth coefficients measure the effect of pH on the infant mortality rate. Comparing two 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

$$-0.075 - 0.057 \times lead$$

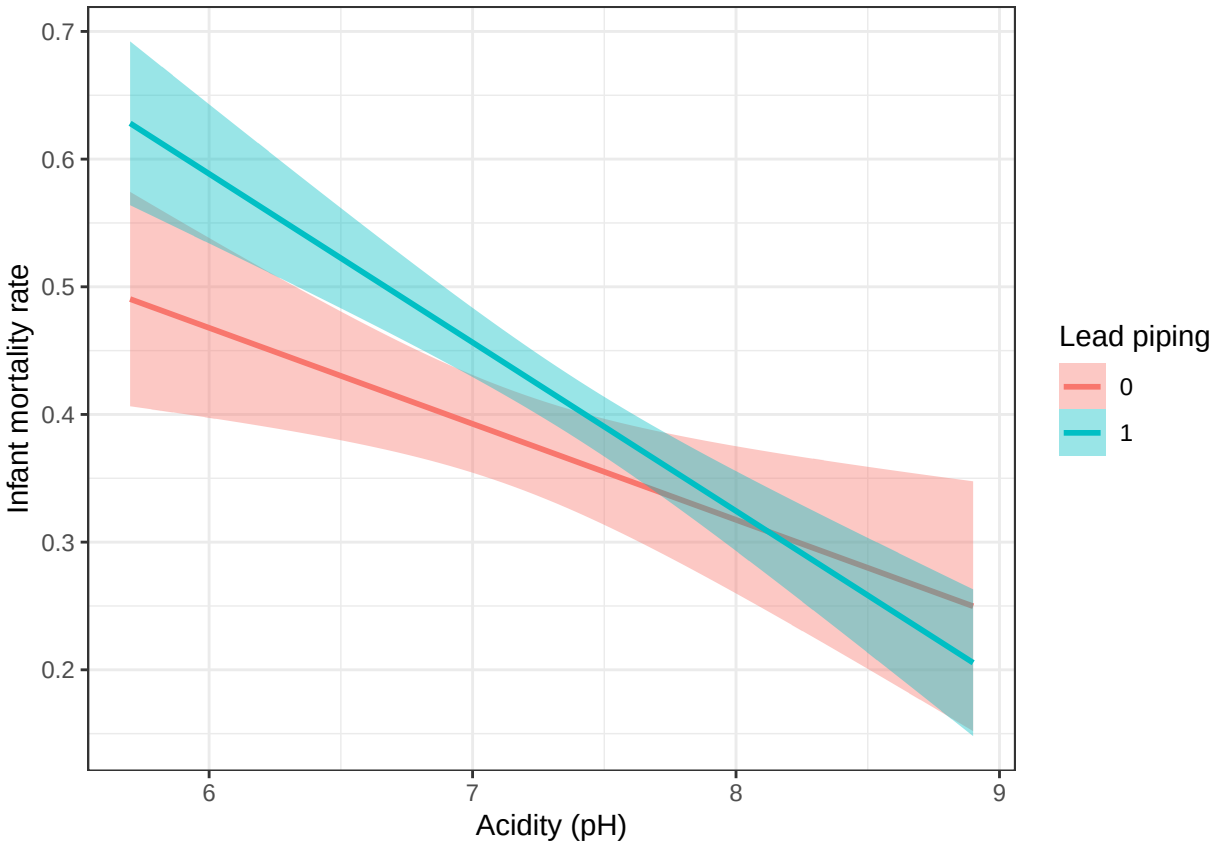
so the infant mortality increase associated with a one-unit increase in pH is -0.075 for cities without lead pipes and -0.075-0.057=-0.132 for cities with lead pipes.

b-ii) Plot the estimated regression function relating *Inf* to *pH* for *Lead* = 0 and for *Lead* = 1. Describe the differences in the regression functions, and relate these differences to the coefficients you discussed in b-i).

You can of course plot two separate graphs for the two cases. But here's an opportunity to try out plotting two lines on the same graph:

```
ggplot(lead, aes(x = ph, y = infrate, fill = factor(lead), color = factor(lead))) +
  theme_bw() +
  geom_smooth(method = 'lm') +
  # Give title to the axes and the legend
  labs(x = 'Acidity (pH)',
       y = 'Infant mortality rate',
       fill = 'Lead piping', color = 'Lead piping')
```

```
## `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 we set the *fill* and *color* colors to depend on this factor so that when we add any plot element, it will color the two groups differently. In addition, it means that if 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) since geom_smooth() also takes *fill* and *color* as arguments.

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

b-iii) Does *lead* have a statistically significant effect on *Infrate*?

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

```
## Linear hypothesis test
##
## Hypothesis:
## lead = 0
## lead:ph = 0
##
## Model 1: restricted model
## Model 2: infrate ~ 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 = 3.936, which has a p- value of 0.02, so the lead coefficients are jointly significant at the 5% but not the 1% significance level.

b-iv) Does the effect of lead on infant mortality depend on pH? Is this dependence statistically significant?

```
lead.mod$p.value
```

```
##  (Intercept)         lead           ph      lead:ph
## 6.865990e-09 2.745960e-02 4.369094e-04 4.447784e-02
```

```
coeftable(lead.mod) # from fixest package
```

```
##                Estimate Std. Error   t value     Pr(>|t|)    CI Lower
## (Intercept)  0.91890383 0.15049414  6.105911 6.865990e-09  0.62180053
## lead         0.46179846 0.20761355  2.224317 2.745960e-02  0.05193085
## ph          -0.07517915 0.02095325 -3.587948 4.369094e-04 -0.11654474
## lead:ph     -0.05686222 0.02808375 -2.024737 4.447784e-02 -0.11230474
##                 CI Upper  DF
## (Intercept)  1.216007135 168
## lead         0.871666061 168
## ph          -0.033813564 168
## lead:ph     -0.001419694 168
```

The interaction term has a t-statistic of t = -2.02, corresponding to a p-value of 0.448 so the coefficient is significant at the 5% but not the 1% significance level.

b-v) Average value of pH

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

What is the average value of pH in the sample?

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

At this pH level, what is the estimated effect of Lead on infant mortality? Method 1:

```
# Create two observations with the mean ph level but different values of lead:
mean.ph <- data.frame(lead = c(1,0),
                      ph = mean(lead$ph))
# Estimate their infant mortality
inf.meanph <- predict(lead.mod, newdata = mean.ph)
inf.meanph
```

```
##         1         2
## 0.4138063 0.3683914
```

```r
# Take the difference between these estimates
inf.meanph[1]-inf.meanph[2]
```

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

Careful with the sign here: the interpretation here is that at the mean pH level, an observation with lead (our first observation) is predicted to have a 0.0454 *higher* infant mortality than an observation without lead (our second observation)

Method 2:

```r
# Estimated mortality with lead
lead.1 <- data.frame(lead = 1, ph = mean(lead$ph))
# Estimated mortality with lead
lead.0 <- data.frame(lead = 0, ph = mean(lead$ph))
# Difference between estimates
predict(lead.mod, newdata = lead.1)-predict(lead.mod, newdata = lead.0)
```

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

What is the standard deviation of pH? The standard deviation of pH is

```r
ph.sd <- sd(lead$ph)
ph.sd
```

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

Suppose the pH level is one standard deviation lower than the average level of pH in the sample: What is the estimated effect of Lead on infant mortality? 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

```r
# Method 1
ph.1sd.lower <- data.frame(lead = c(1,0),
                           ph = mean(lead$ph)-sd(lead$ph))
preds.1sd.lower <- predict(lead.mod, newdata = ph.1sd.lower)
preds.1sd.lower[1]-preds.1sd.lower[2]
```

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

```r
# Method 2
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
```

```r
# Method 1
ph.1sd.higher <- data.frame(lead = c(1,0),
                            ph = mean(lead$ph)+sd(lead$ph))
preds.1sd.higher <- predict(lead.mod, newdata = ph.1sd.higher)
preds.1sd.higher[1]-preds.1sd.higher[2]
```

What if pH is one standard deviation higher than the average value?

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

```r
# Method 2
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
```

b-vi) Constructing a 95% confidence interval for the effect of lead on infant mortality when pH = 6.5

The given regression equation is

$$\text{Infrate} = \beta_0 + \beta_1 \text{lead} + \beta_2 \text{pH} + \beta_3 (\text{lead} \times \text{pH}) + u_i$$

```r
mod.lead <- lm_robust(infrate ~ lead*ph, lead, se_type = 'HC1')
summary(mod.lead)
```

Method 1: the margins package

```
##
## Call:
## lm_robust(formula = infrate ~ lead * ph, data = lead, se_type = "HC1")
##
## 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
```

The given coefficient on lead is the marginal effect of lead but when pH = 0. We want the marginal effect of lead when pH = 6.5:

```
marg.ph65 <- margins(mod.lead, at = list(ph = 6.5))
marg.ph65
```

```
## Average marginal effects at specified values

## lm_robust(formula = infrate ~ lead * ph, data = lead, se_type = "HC1")

##  at(ph)    lead      ph
##     6.5 0.09219 -0.1139
```

The marginal effect of lead is estimated to be 0.092. We can construct the corresponding confidence intervals using confint():

```
confint(marg.ph65, level = 0.95)
```

```
##            lower       upper
## lead  0.02778994  0.15659815
## ph   -0.14203619 -0.08568118
```

This gives a confidence interval of 0.028 to 0.157

Method 2: Transforming the regression Referring to method 2 of section 7.3 of Stock-Watson, we add and subtract $6.5\beta_3$lead to the regression:

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

Estimating this regression

```
lead %<>% mutate(lead, x3 = lead*ph-6.5*lead)
mod.lead2 <- lm_robust(infrate ~ lead + ph + x3, lead, se_type = 'HC1')
summary(mod.lead2)
```

```
##
## Call:
## lm_robust(formula = infrate ~ lead + ph + x3, data = lead, se_type = "HC1")
##
## 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         0.09219    0.03286   2.806 5.614e-03  0.02732  0.15707 168
## ph          -0.07518    0.02095  -3.588 4.369e-04 -0.11654 -0.03381 168
## x3          -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
```

Then we get exactly the same estimated effet, here presented as the coefficient on lead. The corresponding confidence interval is the one for the coefficient on lead: 0.027 to 0.157, essentially the same as through Method 1.

Part c: The analysis in (b) may suffer from omitted variable bias because it neglects factors that affect infant mortality and that might potentially be correlated with Lead and pH. Investigate this concern, using the other variables in the data set.

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

Practice Question 2: Stock-Watson Empirical Exercise E8.2

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.

Transform data:

```
head(cps)
```

```
## # A tibble: 6 x 5
##    year   ahe bachelor female   age
##   <dbl> <dbl>    <dbl>  <dbl> <dbl>
## 1  2015 11.8         0      0    26
## 2  2015  9.62        0      1    33
## 3  2015 12.0         0      0    31
## 4  2015 18.4         0      0    32
## 5  2015 41.8         0      0    28
## 6  2015 19.2         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:

```
# You can create five separate model objects as usual:
mod.a <- lm_robust(ahe ~ age + female + bachelor, cps, se_type = 'HC1')
mod.b <- lm_robust(log.ahe ~ age + female + bachelor, cps, se_type = 'HC1')
mod.c <- lm_robust(log.ahe ~ log.age + female + bachelor, cps, se_type = 'HC1')
mod.d <- lm_robust(log.ahe ~ age + age2 + female + bachelor, cps, se_type = 'HC1')
mod.i <- lm_robust(log.ahe ~ age + age2 + female*bachelor, cps, se_type = 'HC1')
# And then running five separate summary() commands

# Here's a convenient way to do this in one command and one object using the fixest package:
## Models a, b: different LHS, same RHS
mods.ab <- feols(c(ahe, log.ahe) ~ age + female + bachelor,
                 data = cps, se = 'HC1')
## Models c, d, i: same LHS, different RHS
mods.cdi <- feols(log.ahe ~ sw(log.age + female + bachelor,
                               age + age2 + female + bachelor,
```

```
                                age + age2 + female*bachelor,
                                female*bachelor + female*age + female*age2),
                  data = cps, se = 'HC1')
```

With the latter, we can call any of the models estimated using "mods$" and selecting the relevant model or call mods[[number]] where number is the index of the model in order of estimation.

It also becomes convenient for making compact tables:

```
etable(mods.ab, mods.cdi, markdown = T)
```

Dependent Variables:	ahe			log.ahe		
Model:	(1)	(2)	(3)	(4)	(5)	(6)
Variables						
Constant	2.045	2.027***	0.3233	0.4187	0.4119	0.2905
	(1.324)	(0.0600)	(0.1986)	(0.6696)	(0.6694)	(0.9145)
age	0.5313***	0.0242***		0.1341***	0.1348***	0.1392**
	(0.0446)	(0.0020)		(0.0456)	(0.0456)	(0.0622)
female	-4.144***	-0.1776***	-0.1775***	-0.1774***	-0.1903***	-0.0345
	(0.2624)	(0.0115)	(0.0115)	(0.0115)	(0.0161)	(1.336)
bachelor	9.846***	0.4615***	0.4615***	0.4616***	0.4521***	0.4514***
	(0.2613)	(0.0115)	(0.0115)	(0.0115)	(0.0155)	(0.0155)
log.age			0.7154***			
			(0.0586)			
age2				-0.0019**	-0.0019**	-0.0019*
				(0.0008)	(0.0008)	(0.0010)
female × bachelor					0.0235	0.0231
					(0.0229)	(0.0229)
female × age						-0.0013
						(0.0911)
female × age2						-0.0001
						(0.0015)
Fit statistics						
Observations	7,098	7,098	7,098	7,098	7,098	7,098
R^2	0.18964	0.20837	0.20863	0.20901	0.20913	0.20971
Adjusted R^2	0.18930	0.20804	0.20829	0.20857	0.20857	0.20893

Heteroskedasticity-robust standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Parts a-d:

All these subquestions also ask us to look at the effect of age increasing from 25 to 26 and from 33 to 34 for each of the different models so we also define those cases below. Since we will be interested in the age effect which does not interact with our control variables female and bachelor, we arbitrarily set sex to female and bachelor to 1.

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

```r
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)

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

Effect of age increases from 25 to 26 and from 33 to 34 on expected earnings:

```r
ages.preds <- data.frame(ages,
                  pred.a = predict(mods.ab[[1]], newdata = ages),
                  pred.b = predict(mods.ab[[2]], newdata = ages),
                  pred.c = predict(mods.cdi[[1]], newdata = ages),
                  pred.d = predict(mods.cdi[[2]], newdata = ages),
                  pred.i = predict(mods.cdi[[3]], newdata = ages))

ages.preds
```

```
##   age age2  log.age female bachelor   pred.a   pred.b   pred.c   pred.d
## 1  25  625 3.218876      1        1 21.02880 2.916019 2.909950 2.893209
## 2  26  676 3.258097      1        1 21.56007 2.940210 2.938008 2.932450
## 3  33 1089 3.496508      1        1 25.27900 3.109548 3.108561 3.102957
## 4  34 1156 3.526361      1        1 25.81027 3.133739 3.129917 3.112433
##     pred.i
## 1 2.898165
## 2 2.937560
## 3 3.108545
## 4 3.118003
```

Then we can get their predictions for the differences by subtracting row 2 by row 1 and row 4 by row 3:

```r
# Age effect from age 25 to age 26
(ages.preds[2,]-ages.preds[1,])[,6:10]
```

```
##      pred.a     pred.b     pred.c    pred.d    pred.i
## 2 0.5312752 0.02419116 0.02805752 0.0392405 0.0393943
```

```r
# Age effect from age 33 to 34
(ages.preds[4,]-ages.preds[3,])[,6:10]
```

```
##      pred.a     pred.b     pred.c      pred.d      pred.i
## 4 0.5312752 0.02419116 0.02135606 0.009475901 0.009458534
```

Part e: Do you prefer the regression in (c) to the regression in (b)? Explain.

Part f: Do you prefer the regression in (d) to the regression in (b)? Explain.

Part g: Do you prefer the regression in (d) to the regression in (c)? Explain.

Displaying them again:

```
etable(mods.ab, mods.cdi, markdown = T)
```

Dependent Variables:	ahe			log.ahe		
Model:	(1)	(2)	(3)	(4)	(5)	(6)
Variables						
Constant	2.045	2.027***	0.3233	0.4187	0.4119	0.2905
	(1.324)	(0.0600)	(0.1986)	(0.6696)	(0.6694)	(0.9145)
age	0.5313***	0.0242***		0.1341***	0.1348***	0.1392**
	(0.0446)	(0.0020)		(0.0456)	(0.0456)	(0.0622)
female	-4.144***	-0.1776***	-0.1775***	-0.1774***	-0.1903***	-0.0345
	(0.2624)	(0.0115)	(0.0115)	(0.0115)	(0.0161)	(1.336)
bachelor	9.846***	0.4615***	0.4615***	0.4616***	0.4521***	0.4514***
	(0.2613)	(0.0115)	(0.0115)	(0.0115)	(0.0155)	(0.0155)
log.age			0.7154***			
			(0.0586)			
age2				-0.0019**	-0.0019**	-0.0019*
				(0.0008)	(0.0008)	(0.0010)
female × bachelor					0.0235	0.0231
					(0.0229)	(0.0229)
female × age						-0.0013
						(0.0911)
female × age2						-0.0001
						(0.0015)
Fit statistics						
Observations	7,098	7,098	7,098	7,098	7,098	7,098
R^2	0.18964	0.20837	0.20863	0.20901	0.20913	0.20971
Adjusted R^2	0.18930	0.20804	0.20829	0.20857	0.20857	0.20893

Heteroskedasticity-robust standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

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

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.

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

Part h: Plot the regression relation between *Age* and *ln(AHE)* from (b), (c), and (d) for males with a high school diploma. Describe the similarities and differences between the estimated regression functions. Would your answer change if you plotted the regression function for females with college degrees?

The regression functions are very similar, particularly for Age between 27 and 33 years. The quadratic regression shows somewhat more curvature than the log-log regression, but the difference is small. The regression functions for a female with a high school diploma will look just like these, but they will be shifted by the amount of the coefficient on the binary regressor Female. The regression functions for workers with a bachelor's degree will also look just like these, but they would be shifted by the amount of the coefficient on the binary variable Bachelor.

Part i: Run a regression of *ln(AHE)* on *Age*, *Age2*, *Female*, *Bachelor*, and the interaction term *Female * Bachelor.*

```
etable(mods.cdi[[3]], markdown = T)
```

Dependent Variable:	log.ahe
Model:	(1)
Variables	
Constant	0.4119
	(0.6694)
age	0.1348***
	(0.0456)
age2	-0.0019**
	(0.0008)
female	-0.1903***
	(0.0161)
bachelor	0.4521***
	(0.0155)
female × bachelor	0.0235
	(0.0229)
Fit statistics	
Observations	7,098
R^2	0.20913
Adjusted R^2	0.20857

Heteroskedasticity-robust standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

What does the coefficient on the interaction term measure?

The coefficient on the interaction term $Female \cdot Bachelor$ shows the "extra effect" of *Bachelor* on *ln(AHE)* for women relative to that for men.

Alexis is a 30-year-old female with a bachelor's degree. What does the regression predict for her value of *ln(AHE)*?

Jane is a 30-year-old female with a high school diploma. What does the regression predict for her value of *ln(AHE)*?

Bob is a 30-year-old male with a bachelor's degree. What does the regression predict for his value of ln(AHE)?

Jim is a 30-year-old male with a high school diploma. What does the regression predict for his value of ln(AHE)?

```
people <- data.frame(name = c('Alexis', 'Jane', 'Bob', 'Jim'),
                     age = c(30, 30, 30, 30),
                     female = c(1, 1, 0, 0),
                     bachelor = c(1, 0, 1, 0)) %>%
  mutate(age2 = age^2)

preds <- data.frame(people,
                    predictions = predict(mods.cdi[[3]], newdata = people))
preds
```

```
##     name age female bachelor age2 predictions
## 1 Alexis  30      1        1  900    3.057717
## 2   Jane  30      1        0  900    2.582129
## 3    Bob  30      0        1  900    3.224567
## 4    Jim  30      0        0  900    2.772454
```

What is the predicted difference between Alexis's and Jane's earnings?

```
preds$prediction[1]-preds$prediction[2]
```

```
## [1] 0.4755878
```

Alexis' predicted earnings are 0.476 higher than Jane's

What is the predicted difference between Bob's and Jim's earnings?

```
preds$prediction[3]-preds$prediction[4]
```

```
## [1] 0.4521137
```

Bob's predicted earnings are 0.452 higher than Jim's

Part j: Is the effect of Age on earnings different for men than for women? Specify and estimate a regression that you can use to answer this question.

Parts j, k, and l all ask for additional regressions so combining them into one command:

```
mods.jkl <- feols(log.ahe ~ sw(female*bachelor + female*age + female*age2,
                               female*bachelor + bachelor*age + bachelor*age2,
                               female*bachelor + female*age + female*age2 + bachelor*age + bachelor*age2
                  cps, se = 'HC1')
```

For model j, we include two additional regressors: the interactions of female and the two age variables, age and age2.

```
etable(mods.jkl[[1]], markdown = T)
```

Dependent Variable:	log.ahe
Model:	(1)
Variables	
Constant	0.2905
	(0.9145)
female	-0.0345
	(1.336)
bachelor	0.4514***
	(0.0155)
age	0.1392**
	(0.0622)
age2	-0.0019*
	(0.0010)
female × bachelor	0.0231
	(0.0229)
female × age	-0.0013
	(0.0911)
female × age2	-0.0001
	(0.0015)
Fit statistics	
Observations	7,098
R^2	0.20971
Adjusted R^2	0.20893

Heteroskedasticity-robust standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

```
# Re-estimating model j asn lm_robust object so we can use linearHypothesis
mod.j <- lm_robust(log.ahe ~ female*bachelor + female*age + female*age2, cps, se_type = 'HC1')
# Testing joint significance of the age-female interactions
linearHypothesis(mod.j, c('female:age = 0', 'female:age2 = 0'), test = 'F')
```

```
## Linear hypothesis test
##
## Hypothesis:
## female:age = 0
```

```
## female:age2 = 0
##
## Model 1: restricted model
## Model 2: log.ahe ~ female * bachelor + female * age + female * age2
##
##   Res.Df Df      F  Pr(>F)
## 1   7092
## 2   7090  2 2.6372 0.07163 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F-statistic testing the null hypothesis that the coefficients on these interaction terms are both equal to zero is F = 2.64 with a p-value of 0.07. This implies that there isn't sufficient evidence to reject the null hypothesis that there is a different effect of *Age* on *ln(AHE)* for men compared to women at the 1% confidence level.

Part k: Is the effect of Age on earnings different for high school graduates than for college graduates? Specify and estimate a regression that you can use to answer this question.

Same as above but age interactions on bachelor instead:

```
etable(mods.jkl[[2]], markdown = T)
```

Dependent Variable:	log.ahe
Model:	(1)
Variables	
Constant	0.0783
	(0.9216)
female	-0.1903***
	(0.0161)
bachelor	1.093
	(1.336)
age	0.1604**
	(0.0627)
age2	-0.0024**
	(0.0011)
female × bachelor	0.0242
	(0.0229)
bachelor × age	-0.0492
	(0.0910)
bachelor × age2	0.0009
	(0.0015)
Fit statistics	
Observations	7,098
R^2	0.20936
Adjusted R^2	0.20858

Heteroskedasticity-robust standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

```r
# Re-estimating model k asn lm_robust object so we can use linearHypothesis
mod.k <- lm_robust(log.ahe ~ female*bachelor + bachelor*age + bachelor*age2, cps, se_type = 'HC1')
```

Testing the null hypothesis of no difference in the age association with log earnings between college-educated and non-college-educated people:

```r
linearHypothesis(mod.k, c('bachelor:age = 0', 'bachelor:age2 = 0'), test = 'F')
```

```
## Linear hypothesis test
##
## Hypothesis:
## bachelor:age = 0
## bachelor:age2 = 0
##
## Model 1: restricted model
## Model 2: log.ahe ~ female * bachelor + bachelor * age + bachelor * age2
##
##   Res.Df Df      F Pr(>F)
## 1   7092
## 2   7090  2 1.0333 0.3559
```

The associated p-value is 0.3559 so we cannot reject the null hypothesis at any reasonable confidence level.

Part l: After running all these regressions (and any others that you want to run), summarize the effect of age on earnings for young workers.

We'll run an additional regression with both sets of age interaction terms:

```r
etable(mods.jkl[[3]], markdown = T)
```

Dependent Variable:	log.ahe
Model:	(1)
Variables	
Constant	0.0302
	(1.051)
female	-0.0400
	(1.363)
bachelor	0.9409
	(1.363)
age	0.1602**
	(0.0715)
age2	-0.0023*
	(0.0012)
female × bachelor	0.0240
	(0.0229)
female × age	0.0002
	(0.0928)
female × age2	-0.0002
	(0.0016)
bachelor × age	-0.0405
	(0.0927)
bachelor × age2	0.0008
	(0.0016)
Fit statistics	
Observations	7,098
R^2	0.21007
Adjusted R^2	0.20907

Heteroskedasticity-robust standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

```
# mods.l <- lm_robust(log.ahe ~ female*bachelor + female*age + female*age2 + bachelor*age + bachelor*ag
# summary(mods.l)
```

This is a weirdly demanding question so no need to know the following commands. Including here just for completion.

Let's create a table of predicted values using this model for earnings from ages 25-35 for all possible combinations of female and bachelor

```
# Create a grid of combinations for the binary variables
combos <- expand.grid(female = c(0, 1), bachelor = c(0, 1))

# Create a data frame to store the predictions of the model for each combination from all ages from 25-
preds <- expand.grid(age = 20:50,
                     female = c(0,1),
                     bachelor = c(0,1)) %>%
  mutate(age2 = age^2)

# Generate predictions
preds$ahe.hat <- predict(mods.jkl[[3]], newdata = preds)
```

```r
# Plot these for different subgroups
ggplot(preds, aes(x = age, y = ahe.hat, color = female==1, lty = bachelor==1)) +
  theme_bw() +
  geom_smooth(method = 'lm', formula = y ~ poly(x, 2, raw = TRUE), se = FALSE) +
  labs(color = 'Female', lty = 'College-educated')
```