Week 9 Lab

Interaction and Non-Linear Relations

Micaela Wood
03/03/2022

Today

- For loops and functions
- Non linear relationships
- Interaction Relationships

- Loops and Functions can be difficult to learn without concrete examples so today we will incorporate them into our code for non-linear and interaction models.

Non-Linear Relations

Sometimes relations appear non-linearly

Taking logs of the equations often help us fix this

This changes how we interpret our results

Non-Linear Relations

Back on homework set 3 you were asked to log transform your data.

Let's revisit this assignment

```
library(tidyverse, huxtable, broom, 'haven')

nlsy ← read.csv('nlsy79.csv')
```

Non-Linear Relations

This time we are going to run four different regressions and interpret the results.

1. Linear
2. Log - Linear
3. Linear-Log
4. Log-Log

We can do this step by step

We could also make a function to run all of the models and interpret them for us

We will do both!

Linear Model

For the linear model will we need to create any new variables?

No, but for consistency I will go ahead and get rid of zero and missing earnings

```
nlsy ← nlsy %>% filter(earn2009>0 & is.na(earn2009)==0)%>%
  filter(hgc>0 & is.na(hgc)==0)
```

Linear Model

```
lin = lm(data = nlsy, earn2009 ~ hgc)%>%tidy()
huxreg(lin)
```

	(1)
(Intercept)	-78514.290 ***
	(5592.149)
hgc	10327.696 ***
	(407.425)
N	0

*** p < 0.001; ** p < 0.01; * p < 0.05.

Linear Model

How can we interpret $\hat{\beta}$?

	(1)
(Intercept)	-78514.290 ***
	(5592.149)
hgc	10327.696 ***
	(407.425)
N	0

*** p < 0.001; ** p < 0.01; * p < 0.05.

One more **year** of schooling leads to an increase in annual earnings of **$10,261**

Log-Linear

For a log-linear model will we need to transform any variables?

Yes! We will want to take the log of earnings

```
nlsy ← nlsy %>% mutate(
  log_earn = log(earn2009)
)
```

Log-Linear

Now we can run our regression

```
log_lin = lm(data = nlsy, log_earn ~ hgc)%>%tidy()
huxreg(log_lin)
```

	(1)
(Intercept)	8.649 ***
	(0.095)
hgc	0.143 ***
	(0.007)
N	0

*** p < 0.001; ** p < 0.01; * p < 0.05.

Log-Linear

How should we interpret these results?

	(1)
(Intercept)	8.649 ***
	(0.095)
hgc	0.143 ***
	(0.007)
N	0

*** p < 0.001; ** p < 0.01; * p < 0.05.

One extra **year** of school increases earnings by...0.142(100%) = **14.2%**

Linear-Log

Now we will create a logged version of years of schooling and run a new regression

```
nlsy ← nlsy %>% mutate(
  log_hgc = log(hgc)
)

lin_log = lm(data = nlsy, earn2009 ~ log_hgc)%>%tidy()
```

Linear-Log

How can we interpret this result?

	(1)
(Intercept)	-279153.650 ***
	(14314.009)
log_hgc	131477.586 ***
	(5519.220)
N	0

*** p < 0.001; ** p < 0.01; * p < 0.05.

A **1%** increase in schooling leads to an increase in earnings by...131477/100= **$1,314.77**

Log-Log

Do we need any new variables for this regression?

No because we have already created both logged variables

```
log_log = lm(data = nlsy, log_earn ~ log_hgc)%>%tidy()
```

Log-Log

How can we interpret these results?

	(1)
(Intercept)	5.742 ***
	(0.243)
log_hgc	1.873 ***
	(0.094)
N	0

*** p < 0.001; ** p < 0.01; * p < 0.05.

A **1%** increase in earnings increases earnings by...**1.87%**

Function

Now we can try to make a function that will run all four scenarios for us.

Let's try to make a function that takes in the two variables and gives a table of results and interpretations

Function

Remember a function takes the form

```
function(inputs){
expressions
return(output)
}
```

Function

Step 1: Create the data frame

```
df = tibble(y,x)

#get rid of na and 0
df = df %>% filter(y>0 & is.na(y)==0)%>% filter(x>0 & is.na(x)==0)

#create logs
df = df %>%mutate(
  log_y = log(y),
  log_x = log(x)
)
```

Function

Step 2: Run Regressions

```
lin = lm(data = df, y ~ x)%>%tidy()
log_lin = lm(data = df, log_y ~ x)%>%tidy()
lin_log = lm(data = df, y ~ log_x)%>%tidy()
log_log = lm(data = df, log_y ~ log_x)%>%tidy()
```

Function

Step 3: Store output

```
bind_rows(lin, log_lin, lin_log, log_log)%>%
    filter(term == 'x' | term == 'log_x')%>%
    mutate(
      interpret = c(estimate[1], estimate[2]*100, estimate[3]/100, estimate[4]),
      type = c('lin', 'log_lin', 'lin_log', 'log_log'),
      x_change = c("unit","unit",'percent','percent'),
      y_change = c('unit', 'percent', 'unit', 'percent')
    )
```

Now let's put it all together

Function

```
log_lin_fun = function(y, x){
  df = tibble(y,x)

  df = df %>% filter(y>0 & is.na(y)==0)%>% filter(x>0 & is.na(x)==0)

    df = df %>%mutate(
    log_y = log(y),
    log_x = log(x)
  )

  lin = lm(data = df, y ~ x)%>%broom::tidy()
  log_lin = lm(data = df, log_y ~ x)%>%broom::tidy()
  lin_log = lm(data = df, y ~ log_x)%>%broom::tidy()
  log_log = lm(data = df, log_y ~ log_x)%>%broom::tidy()

  bind_rows(lin, log_lin, lin_log, log_log)%>%
    filter(term == 'x' | term == 'log_x')%>%
    mutate(
      interpret = c(estimate[1], estimate[2]*100, estimate[3]/100, estimate[4]),
      type = c('lin', 'log_lin', 'lin_log', 'log_log'),
      x_change = c("unit","unit",'percent','percent'),
      y_change = c('unit', 'percent', 'unit', 'percent')
    )
```

Function

Now we can test it with the nlsy data to see if we get the same results

```
results = log_lin_fun(nlsy$earn2009, nlsy$hgc)
results[,6:9]
```

interpret	type	x_change	y_change
1.03e+04	lin	unit	unit
14.3	log_lin	unit	percent
1.31e+03	lin_log	percent	unit
1.87	log_log	percent	percent

Function

Let's load gapminder data and see if we can use the function on that to match the results from the slides in class

```
library(gapminder)
head(gapminder)
```

country	continent	year	lifeExp	pop	gdpPercap
Afghanistan	Asia	1952	28.8	8425333	779
Afghanistan	Asia	1957	30.3	9240934	821
Afghanistan	Asia	1962	32	10267083	853
Afghanistan	Asia	1967	34	11537966	836
Afghanistan	Asia	1972	36.1	13079460	740
Afghanistan	Asia	1977	38.4	14880372	786

Function

Let's make life expectancy the y variable and GDP per capita the x variable

```
results = log_lin_fun(gapminder$lifeExp, gapminder$gdpPercap)
results[,c(2,6:9)]
```

estimate	interpret	type	x_change	y_change
0.000765	0.000765	lin	unit	unit
1.29e-05	0.00129	log_lin	unit	percent
8.41	0.0841	lin_log	percent	unit
0.147	0.147	log_log	percent	percent

Quadratic Equations

In homework 3 you were also asked to run this model:

$$earning = \beta_0 + \beta_1 hgc + \beta_2 hgc^2$$

As a bonus you could solve for this increase in earning for different levels of education.

We can use a for loop to solve for all levels of education and then plot the results

Quadratic Equations

First we need to recreate the squared variable and run the regression

```
nlsy ← nlsy %>% mutate(
  hgc_sq = hgc^2
)

quad = lm(data = nlsy, earn2009 ~ hgc + hgc_sq)%>%tidy()
```

Quadratic Equations

How can we interpret these results?

	(1)
(Intercept)	92811.579 ***
	(23429.759)
hgc	-14822.121 ***
	(3365.913)
hgc_sq	893.966 ***
	(118.778)
N	0

*** p < 0.001; ** p < 0.01; * p < 0.05.

Quadratic Equations

We will need to take a **derivative** of our original equation

$$earning = \beta_0 + \beta_1 hgc + \beta_2 hgc^2$$

$$\frac{\partial earnings}{\partial hgc} = \beta_1 + 2\beta_2 hgc$$

Now we can use a for loop to get the derivative for different hgc

Quadratic Equations

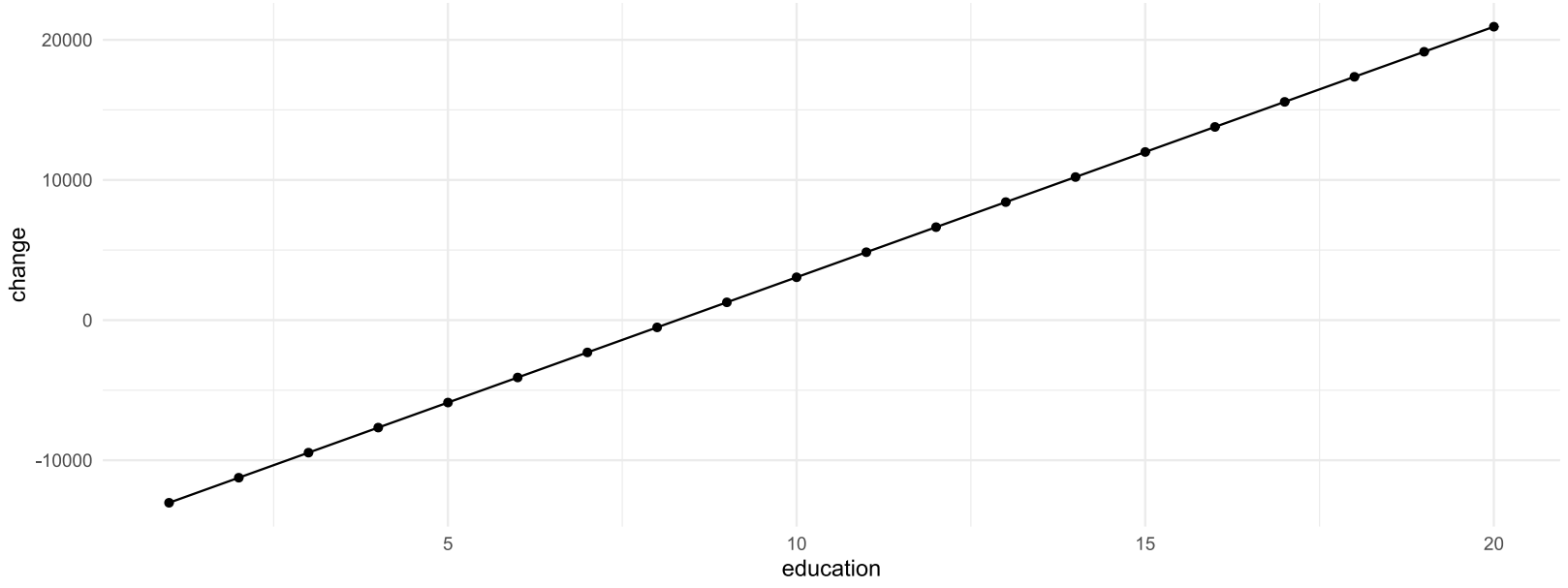
```
education = seq(1, 20, 1)
change = c()

for(i in education){
  change[i] = quad$estimate[2] + 2*quad$estimate[3]*education[i]
}

df = tibble(education, change)
head(df)
```

education	change
1	-1.3e+04
2	-1.12e+04
3	-9.46e+03
4	-7.67e+03

Quadratic Equations

We can plot these results in a graph to better visualize what is going on

```
ggplot(data = df, aes(x = education, y = change))+
  geom_point()+
  geom_line()+
  theme_minimal()
```



Interaction Variables

- Sometimes we may want to see how an event effects groups of people differently
- Some examples are:
  - Gender
  - Race
  - Age
  - Income Level
  - And many more

To see these differences we can use interaction variables

Interaction Variables

For this part let's use the nsw data from week 7

```
library(haven)
nsw ← read_dta("nsw.dta")
```

Let's look at the effects of the job training program for hispanic versus whites

Interaction Variables

The model we should use is:

$$re78 = \beta_0 + \beta_1 treat + \beta_2 hispanic + \beta_3 treat * hispanic$$

```
int = lm(data = nsw, re78 ~ treat + hispanic + treat:hispanic)
```

Interaction Variables

	(1)
(Intercept)	4896.377 ***
	(321.201)
treat	1058.593 *
	(497.757)
hispanic	1714.802
	(955.765)
treat:hispanic	-1488.000
	(1564.351)
N	0

Interaction Variables

What is the effect of the job training program if you are white?

β_1 = **1059**

What is the effect of the job training program if you are hispanic?

$\beta_1 + \beta_3$ = **-429**

Interaction Variables

Now we should find the standard errors at the margins

```
#install.packages(margins)
library(margins)

margins(int, at = list(hispanic = 0:1))%>%
  summary() %>% filter(factor == 'treat')
```

factor	hispanic	AME	SE	z	p	lower	upper
treat	0	1.06e+03	501	2.11	0.0346	76.5	2.04e+03
treat	1	-429	1.46e+03	-0.294	0.769	-3.29e+03	2.43e+03

Interaction Variables

Now we can make a plot of the marginal effects

factor	hispanic	AME	SE	z	p	lower	upper
treat	0	1.06e+03	501	2.11	0.0346	76.5	2.04e+03
treat	1	-429	1.46e+03	-0.294	0.769	-3.29e+03	2.43e+03

```
ggplot(data = plot_data, aes(x = AME, y = hispanic))+
  geom_pointrange(aes(ymin = lower, ymax = upper))+
  theme_minimal()+
  coord_flip()+
  geom_hline(yintercept = 0, linetype = 'dashed')
```

Interaction Variables