



# MULTIPLE REGRESSION WITH INTERACTION TERMS

---

Luis Castro-de-Araujo<sup>a</sup>

11/07/2022

Virginia Institute for Psychiatric and Behavioral Genetics

---

<sup>a</sup>Post-doc T32. [luis.araujo@vcuhealth.org](mailto:luis.araujo@vcuhealth.org)

Multiple regression recap

Interaction terms

Visualizing interactions

Marginal effects

## MULTIPLE REGRESSION RECAP

---



- Extension of the simple linear regression model to two or more independent variables

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon$$

- Cognitive performance = Baseline + Depression + Age + Sex + Error
- Partial Regression Coefficients: effect on the dependent variable when increasing the  $i^{\text{th}}$  independent variable by 1 unit, **holding all other predictors constant**



- Qualitative variables are easily incorporated in regression framework through ***dummy variables***
- Simple example: sex can be coded as 0/1
- What if my categorical variable contains three levels:

$$x_1 = \begin{cases} 0 & \text{if AA} \\ 1 & \text{if AG} \\ 2 & \text{if GG} \end{cases}$$



- Previous coding would result in **colinearity**
- Solution is to set up a series of dummy variable.
- for k levels you need k-1 dummy variables

	x1	x2
AA	1	0
AG	0	1
GG	0	0

$$x_1 = \begin{cases} 1 & \text{if AA} \\ 0 & \text{otherwise} \end{cases}$$

$$x_2 = \begin{cases} 1 & \text{if AG} \\ 0 & \text{otherwise} \end{cases}$$



**Validity** Does the data we're modeling matches the problem we're actually trying to solve?

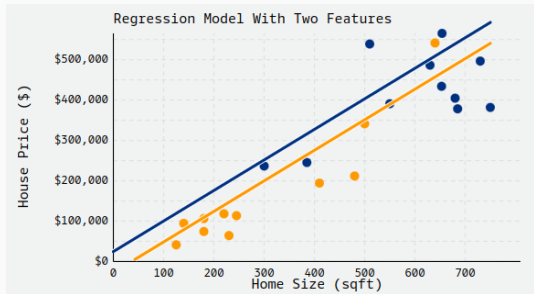
**Representativeness** Is the sample data used in the regression model representative of the population to which it will be applied?

**Linearity** The deterministic component of a regression model is a linear function of the separate predictors:  $y = B_0 + B_1x_1 + \dots + B_px_p$

**Independence of Errors** The errors from our model are independent.

**Homoscedasticity** The errors from our model have equal variance.

**Normality of Errors** The errors from our model are normally distributed.



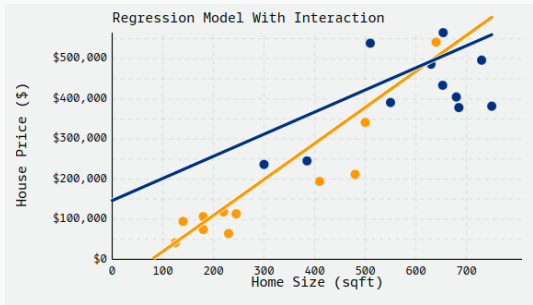
$$\text{houseprice} = -27154 + 757 * \text{sqft} + 51867 * \text{pool}$$

- In our example, we model home prices as a function of both the size of the house (sqft) and whether or not it has a pool

- intercept:  $-\$27,154$ , the predicted average housing price for houses with all  $x_i = 0$ . Or the cost of houses with no pools and a square-footage of zero.
- coefficient of pool:  $\$51,867$ , average expected price difference in houses of the same size (in sqft) if they do or do not have a pool. In other words, we expect, on average, houses of the same size to cost  $\$51,867$  more if they have a pool than if they do not.
- coefficient of sqft:  $\$757$ , average expected price difference in housing price for houses that have the same value of pool but differ in size by one square-foot.
- We assume the same slope for sqft. Hence, two lines. This isn't always a valid assumption to make.



## BACK TO OUR HOUSING EXAMPLE, NOW WITH INTERACTIONS



- interaction term:  $-\$347$ , represents the difference in the slope for sqft, comparing houses that do and do not have pools. Visually, this represents the difference between the slopes of the two lines.
- intercept:  $-\$70,296$ , represents the predicted housing price for houses with no pools and a square-footage of zero.
- coefficient of pool:  $\$217,111$ , represents the average expected difference in houses of the same size (0 sqft) that differed in whether or not they had a pool. (It's not super useful since we don't have houses with 0 square-feet).
- coefficient of sqft:  $\$899$ , represents the average expected difference in housing price for houses that do not have a pool (pool=0) but differ in size by one square-foot.

$$\text{houseprice} = -70296 + 899 * \text{sqft} + 217111 * \text{pool} - 347 * (\text{sqft} : \text{pool})$$

- If we believe that the slope for sqft should differ between houses that do have pools and houses that do not, we can add an interaction term to our model, (sqft:pool).

## **INTERACTION TERMS**

---



- An interaction is a predictor that is some combination of the other predictors.



- Interactions are often the product of two or more predictors.
- Can be written as,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon$$



- Conditional effects: the effect of a predictor on the response, holding all other predictors constant.
- Marginal effects: the effect of a predictor on the response, averaged over all values of the other predictors.



- If the conditional effects of  $X_1$  on  $Y$  at different levels of  $X_2$  are all the same then there is no interaction.



Parameter Meaning		Where people (used to) go awry
$\beta_0$	Expected value of the DV when $X_1$ and $X_2 == 0$	People get this
$\beta_1$	Effect of $X_1$ when $X_2 == 0$	Not marginal effects!
$\beta_2$	Effect of $X_2$ when $X_1 == 0$	Not marginal effects!
$\beta_3$	The addition to the conditional effect when both $X_1$ and $X_2$ are 1	People just look at the significance of the interaction parameter and do not calculate the underlying marginal or conditional effects or standard errors

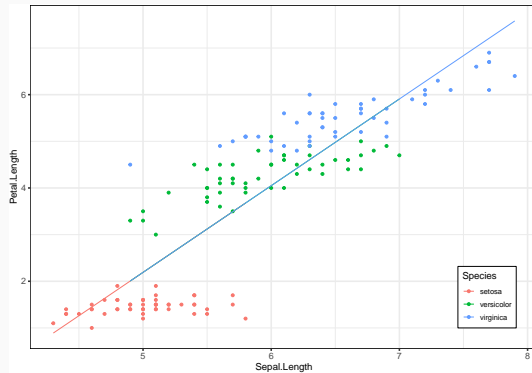


- A common mistake that people make when interpreting interaction models is using the wrong standard errors.
- The standard errors that are printed in every regression table are the positive square roots of the diagonal elements of the variance- covariance matrix of  $\beta$
- This does not matter anymore because of `margins()`





$$\widehat{petal.length}_i = \beta_0 + \beta_1 sepal.length_i$$





### Creating the dummy

$$\text{setosa}_i = \begin{cases} 1 & \text{if species of flower } i = \text{setosa}, \forall i \in [1, 150] \\ 0 & \text{otherwise} \end{cases}$$

$$\text{versicolor}_i = \begin{cases} 1 & \text{if species of flower } i = \text{versicolor}, \forall i \in [1, 150] \\ 0 & \text{otherwise} \end{cases}$$

### Our formula is then

$$\widehat{\text{petal.length}}_i = \beta_0 + \beta_1 \text{sepal.length}_i + \beta_2 \text{setosa}_i + \beta_3 \text{versicolor}_i$$



### If it is setosa

$$\begin{aligned}\widehat{petal.length}_i &= \beta_0 + \beta_1 sepal.length_i + \beta_2 setosa_i + \beta_3 versicolor_i \\ &= \beta_0 + \beta_1 sepal.length_i + \beta_2 1 + \beta_3 0 \\ &= \beta_0 + \beta_1 sepal.length_i + \beta_2 \\ &= (\beta_0 + \beta_2) + \beta_1 sepal.length_i\end{aligned}$$

### If it is versicolor

$$\begin{aligned}\widehat{petal.length}_i &= \beta_0 + \beta_1 sepal.length_i + \beta_2 setosa_i + \beta_3 versicolor_i \\ &= \beta_0 + \beta_1 sepal.length_i + \beta_2 0 + \beta_3 1 \\ &= \beta_0 + \beta_1 sepal.length_i + \beta_3 \\ &= (\beta_0 + \beta_3) + \beta_1 sepal.length_i\end{aligned}$$

### If it is virginica

$$\begin{aligned}\widehat{petal.length}_i &= \beta_0 + \beta_1 sepal.length_i + \beta_2 setosa_i + \beta_3 versicolor_i \\ &= \beta_0 + \beta_1 sepal.length_i + \beta_2 0 + \beta_3 0 \\ &= \beta_0 + \beta_1 sepal.length_i \\ &= \beta_0 + \beta_1 sepal.length_i\end{aligned}$$

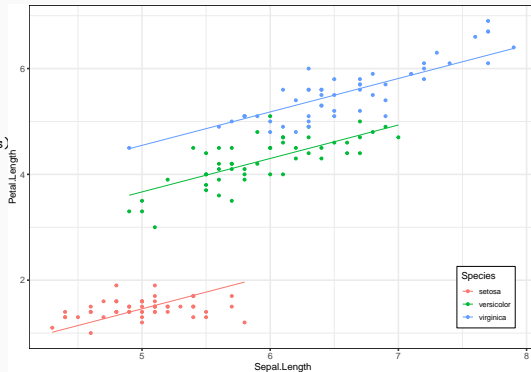
# SAME SLOPE, DIFFERENT INTERCEPTS



```
iris$pred <- predict(lm(Petal.Length ~ Sepal.Length+Species,  
                        data = iris))
```

*# plot in ggplot*

```
ggplot(iris, aes(x = Sepal.Length, y = Petal.Length, color = Species))  
  geom_point() +  
  geom_line(aes(Sepal.Length, pred )) +  
  theme_luis() +  
  theme( legend.position = c(0.9, 0.15))
```





$$\widehat{petal.length}_i = \beta_0 + \beta_1 sepal.length_i + \beta_2 setosa_i + \beta_3 versicolor_i \\ + \beta_4 sepal.length_i setosa_i + \beta_5 sepal.length_i versicolor_i$$

- this will result in three unique lines depending on the species of the flower.
- both the intercepts and the slopes will be allowed to be different.

### Does it make sense to retain the interaction?

```
inter <- lm(Petal.Length ~ Sepal.Length + Species + Sepal.Length:Species, data = iris)
```

# DOES IT MAKE SENSE TO RETAIN THE INTERACTION?



```
broom::tidy(inter) |> kable()
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.803	0.531	1.512	0.133
Sepal.Length	0.132	0.106	1.244	0.216
Speciesversicolor	-0.618	0.684	-0.904	0.368
Speciesvirginica	-0.193	0.658	-0.293	0.770
Sepal.Length:Speciesversicolor	0.555	0.128	4.330	0.000
Sepal.Length:Speciesvirginica	0.618	0.121	5.111	0.000

```
anova(nospecies, w_species, inter)
```

## Analysis of Variance Table

Model 1: Petal.Length ~ Sepal.Length

Model 2: Petal.Length ~ Sepal.Length + Species

Model 3: Petal.Length ~ Sepal.Length + Species + Sepal.Length:Species

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	148	111.5				
2	146	11.7	2	99.8	731.9	< 0.0000000000000002 ***
3	144	9.8	2	1.8	13.5	0.0000043 ***

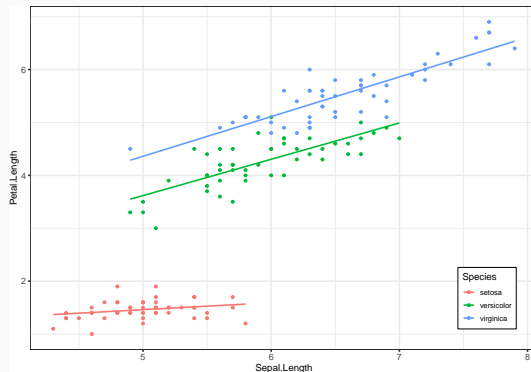
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# NOW WE CAN ADD AN INTERACTION



$$\widehat{petal.length}_i = \beta_0 + \beta_1 sepal.length_i + \beta_2 setosa_i + \beta_3 versicolor_i + \beta_4 sepal.length_i setosa_i + \beta_5 sepal.length_i versicolor_i$$

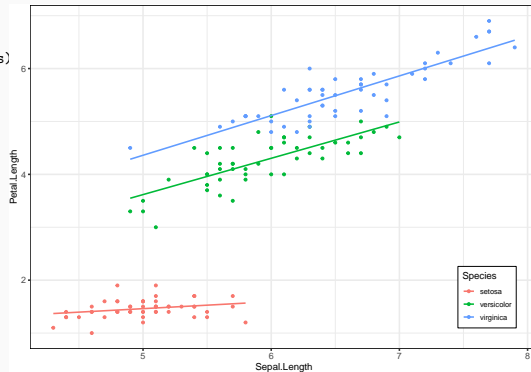
- this will result in three unique lines depending on the species of the flower.
- both the intercepts and the slopes will be allowed to be different.
- ggplot geom\_smooth does this by default if color is used



# NOW WE CAN ADD AN INTERACTION



```
ggplot(iris, aes(x = Sepal.Length, y = Petal.Length, color = Species))  
  geom_point() +  
  geom_smooth(method = "lm", se = FALSE) +  
  theme_luis() +  
  theme( legend.position = c(0.9, 0.15))
```





## **VISUALIZING INTERACTIONS**

---



```
fit1 <- glm(qsec ~ wt*as.factor(cyl), data = mtcars)
broom::tidy(fit1) |> kable()
```

- Note: not significant, but we will return to this later  
summary(margins(fit1))

term	estimate	std.error	statistic	p.value
(Intercept)	14.83	1.500	9.884	0.000
wt	1.89	0.639	2.951	0.007
as.factor(cyl)6	-9.78	4.395	-2.226	0.035
as.factor(cyl)8	-1.44	2.273	-0.632	0.533
wt:as.factor(cyl)6	2.26	1.464	1.546	0.134
wt:as.factor(cyl)8	-1.04	0.764	-1.360	0.185



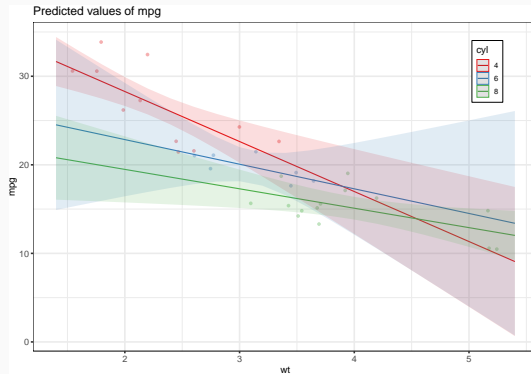
```
fit <- glm(mpg ~ wt*as.factor(cyl), data = mtcars)
broom::tidy(fit) |> kable()
```

term	estimate	std.error	statistic	p.value
(Intercept)	39.57	3.19	12.39	0.000
wt	-5.65	1.36	-4.15	0.000
as.factor(cyl)6	-11.16	9.36	-1.19	0.244
as.factor(cyl)8	-15.70	4.84	-3.25	0.003
wt:as.factor(cyl)6	2.87	3.12	0.92	0.366
wt:as.factor(cyl)8	3.46	1.63	2.12	0.043

# REGRESSION OF MPG ON WT\*CYL



```
pred <- ggpredict(fit, terms = c("wt", "cyl"))  
plot(pred, add.data = TRUE)+  
  theme_luis()+  
  theme( legend.position = c(0.1, 0.15))
```



# AN INTERACTION MAY NOT BE SIG ACROSS THE ENTIRE RANGE OF THE PREDICTOR



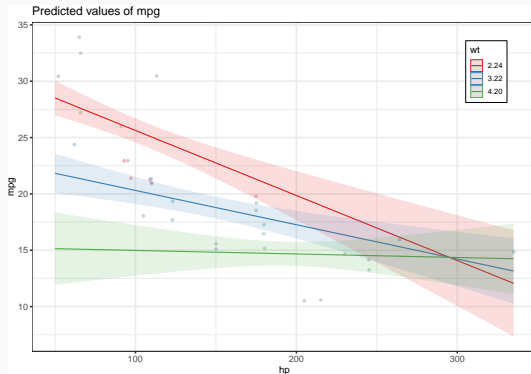
## Let's see mpg ~ hp + wt

```
fit <- glm(mpg ~ hp*wt, data = mtcars)
pred <- ggpredict(fit, terms = c("hp", "wt"))
```

```
# plot(pred, add.data = TRUE) +
#   theme_luis() +
#   theme(legend.position = c(0.9, 0.85))
```

```
broom::tidy(fit) |> kable()
```

term	estimate	std.error	statistic	p.value
(Intercept)	49.808	3.605	13.82	0.000
hp	-0.120	0.025	-4.86	0.000
wt	-8.217	1.270	-6.47	0.000
hp:wt	0.028	0.007	3.75	0.001

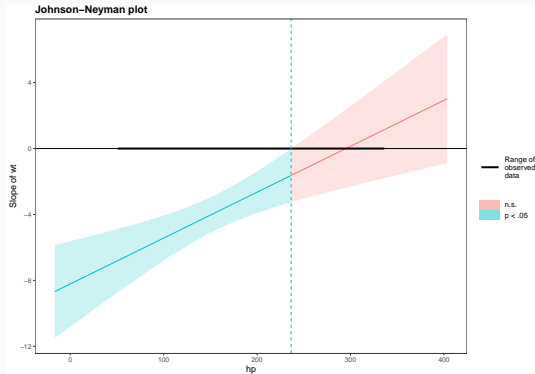


# NOW, FROM WHAT POINT THE SLOPE BECOMES NON SIGNIFICANT?



## JOHNSON-NEYMAN INTERVAL

```
jn <- johnson_neyman(fit, wt, hp, plot = TRUE)
jn
```



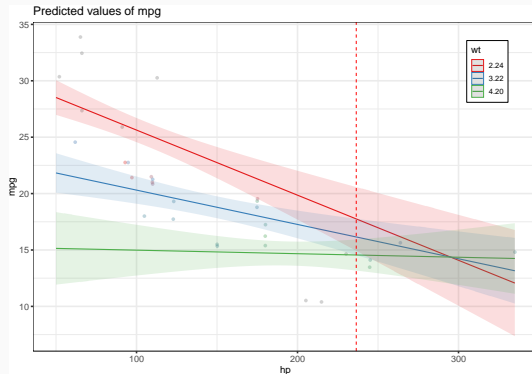
# NOW, FROM WHAT POINT THE SLOPE BECOMES NON SIGNIFICANT?



## JOHNSON-NEYMAN INTERVAL - Overlaid on data

```
fit <- glm(mpg ~ hp*wt, data = mtcars)
pred <- ggpredict(fit, terms = c("hp", "wt"))
jn <- johnson_neyman(fit, wt, hp, plot = TRUE)
jn_bound <- as.numeric(jn$bounds[1])

plot(pred, add.data=T) +
  geom_vline(xintercept = jn_bound, linetype = "dashed",
    color = "red") +
  theme_luis() +
  theme(legend.position = c(0.9, 0.85))
```

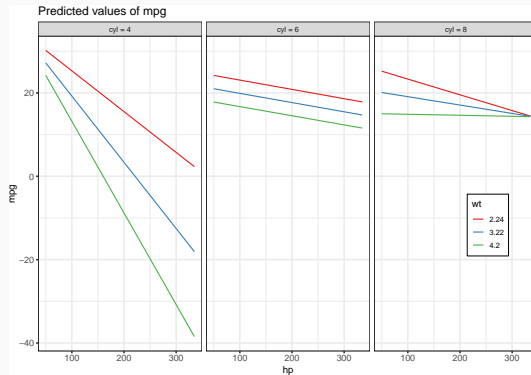


# THREE-WAY INTERACTIONS



```
fit <- glm(mpg ~ hp*wt*as.factor(cyl), data = mtcars)
```

```
dat <- ggpredict(fit, terms = c("hp", "wt", "cyl"))  
plot(dat, ci = FALSE)+  
  theme( legend.position = c(0.9, 0.85))
```







```
broom::tidy(fit) |> kable()
```

term	estimate	std.error	statistic	p.value
(Intercept)	34.972	10.883	3.213	0.004
hp	0.042	0.129	0.324	0.750
wt	0.054	5.010	0.011	0.991
as.factor(cyl)6	-2.340	77.278	-0.030	0.976
as.factor(cyl)8	5.829	42.608	0.137	0.893
hp:wt	-0.062	0.058	-1.068	0.298
hp:as.factor(cyl)6	-0.064	0.660	-0.098	0.923
hp:as.factor(cyl)8	-0.120	0.227	-0.527	0.604
wt:as.factor(cyl)6	-3.321	27.372	-0.121	0.905
wt:as.factor(cyl)8	-6.168	12.492	-0.494	0.627
hp:wt:as.factor(cyl)6	0.062	0.236	0.264	0.794
hp:wt:as.factor(cyl)8	0.080	0.078	1.026	0.317

## MARGINAL EFFECTS

---



- Marginal effects: the effect of a predictor on the response, averaged over all values of the other predictors.
- also, the instantaneous effect of “x” on “y”. Calculated with a derivative of x in respect to y, expressed mathematically as  $dy/dx$ .<sup>1</sup>
  - the “instantaneous rate of change” of y with respect to x.

---

<sup>1</sup>???



```
fit_m <- margins(fit2)
summary(fit_m)
```

factor	AME	SE	z	p	lower	upper
hp	-0.0305	0.0075	-4.0661	0.0000	-0.0452	-0.0158
wt	-4.1316	0.5296	-7.8021	0.0000	-5.1696	-3.0937

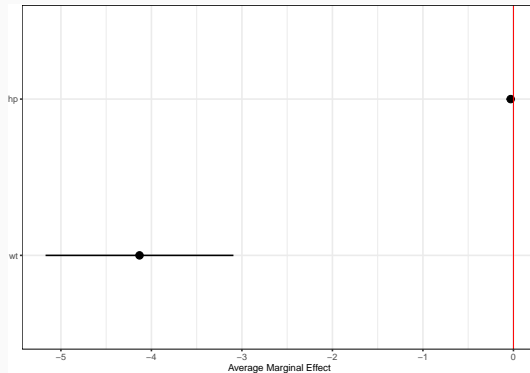
- Interpretation: for each unit increase of mpg an average reduction of hp 3% is expected.

# PLOTTING THE MARGINAL EFFECTS



```
fit_mo <- as_tibble(summary(fit_m))
p <- ggplot(data = fit_mo, aes(x = reorder(factor, AME),
                               y = AME, ymin = lower, ymax = upper))

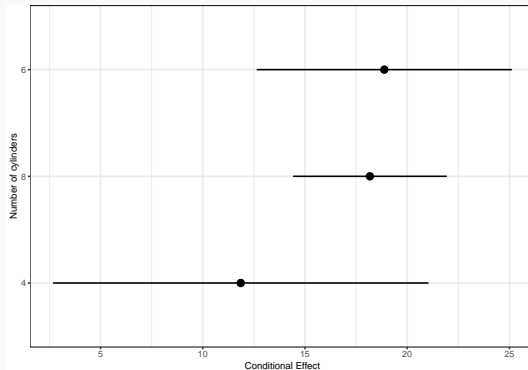
p + geom_hline(yintercept = 0, color = "gray80") +
  geom_pointrange(size = 1) + coord_flip() +
  labs(x = NULL, y = "Average Marginal Effect") +
  geom_hline(yintercept = 0, color = "red") +
  theme_luis()
```



# CAN WE PLOT THE CONDITIONAL EFFECTS TOO?



```
glm(mpg ~ hp*wt*as.factor(cyl), data = mtcars) %>%  
  cplot(x = "cyl", draw = F) %>%  
  ggplot( aes(x = reorder(xvals, yvals),  
              y = yvals, ymin = lower, ymax = upper)) +  
    geom_pointrange(size = 1) + coord_flip() +  
    labs(x = "Number of cylinders", y = "Conditional Effect") +  
    theme_luis()
```





- At the beginning I said, keep the interaction that is significant, but<sup>2</sup>
  - In the transformation from log-odds to probabilities (AME) the interaction may not be significant anymore [vanhoveInteractionsLogisticRegression2019], or worse
  - The interaction may be significant in the AME, but not in conditional effects

---

<sup>2</sup>Bruin, "Deciphering Interactions in Logistic Regression," *Introduction to SAS. UCLA: Statistical Consulting Group.*; Vanhove, 2019, "Interactions in Logistic Regression Models," (2019).



## speed on wt\*cyl

```
fit1 <- glm(qsec ~ wt*as.factor(cyl), data = mtcars)
broom::tidy(fit1) |> kable()
```

term	estimate	std.error	statistic	p.value
(Intercept)	14.83	1.500	9.884	0.000
wt	1.89	0.639	2.951	0.007
as.factor(cyl)6	-9.78	4.395	-2.226	0.035
as.factor(cyl)8	-1.44	2.273	-0.632	0.533
wt:as.factor(cyl)6	2.26	1.464	1.546	0.134
wt:as.factor(cyl)8	-1.04	0.764	-1.360	0.185

```
summary(margins(fit1)) |> kable()
```

factor	AME	SE	z	p	lower	upper
cyl6	-2.50	0.825	-3.03	0.002	-4.12	-0.884
cyl8	-4.78	0.823	-5.81	0.000	-6.39	-3.169
wt	1.93	0.406	4.74	0.000	1.13	2.721





```
library(umx)
```

```
model <- "
  mpg ~ hp_on_mpg*hp + wt_on_mpg*wt
  moderation := hp_on_mpg*wt_on_mpg
"
```

```
m1 <- umxRAM(model, data = mtcars)
m1 <- umxC1(m1)
umxSummary(m1, std = T)
```

**Table 10:** Parameter loadings for model 'm1'

	name	Std.Estimate	Std.SE	CI	type
5	hp_with_wt	0.66	0.1	0.66 [0.46, 0.85]	Manifest Cov
1	hp_on_mpg	-0.36	0.1	-0.36 [-0.56, -0.17]	Manifest path
2	wt_on_mpg	-0.63	0.09	-0.63 [-0.81, -0.45]	Manifest path
7	one_to_mpg	6.28	0.7	6.28 [4.91, 7.64]	Mean
8	one_to_hp	2.17	0.32	2.17 [1.54, 2.81]	Mean
9	one_to_wt	3.34	0.45	3.34 [2.45, 4.23]	Mean
3	mpg_with_mpg	0.17	0.06	0.17 [0.06, 0.28]	Residual
4	hp_with_hp	1.00	0	1 [1, 1]	Residual
6	wt_with_wt	1.00	0	1 [1, 1]	Residual

Model Fit: Chi2(0) = 0, p = 1.000; CFI = 1; TLI = 1; RMSEA = 0 Algebra'moderation' = 0.123CI95[0.074, 0.173]. p-value < 0.001



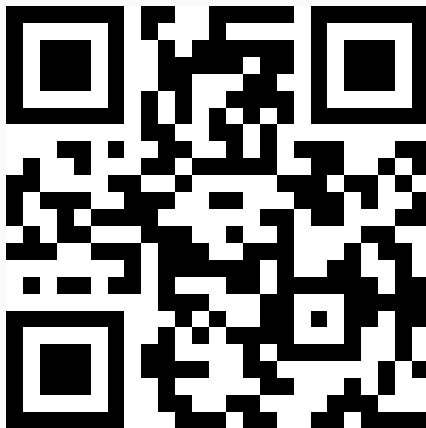
- We started reviewing multiple regression
- Then discussed the syntax and interpretation of parameters when an interaction term is included
- Finally, we discussed how to extract the marginal effects of the interaction term
- Luckily the package `margins()` makes this simple.



## Team

- Charles Gardner (2015)
  - Brad Verhulst (2013)
  - Joshua Pritkin.
  - Rob Kirkpatrick.
- 
- Michael C Neale.
  - NIH grant no R01 DA049867 and 5T32MH-020030

## Contact



- **THANK YOU**