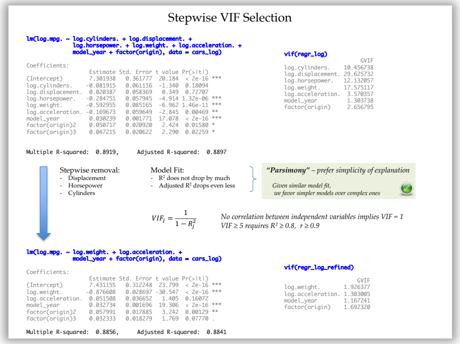
Business Analytics Using Computational Statistics

Week 11
Applied Regression

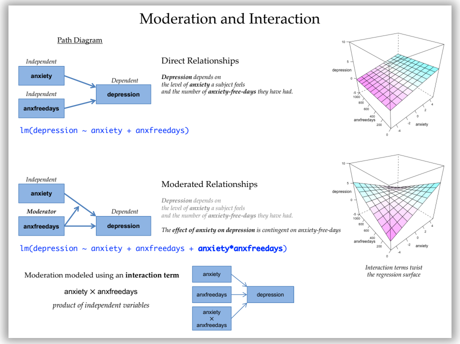
Week 12
Moderation and Mediation

Week 13
Composites and Components

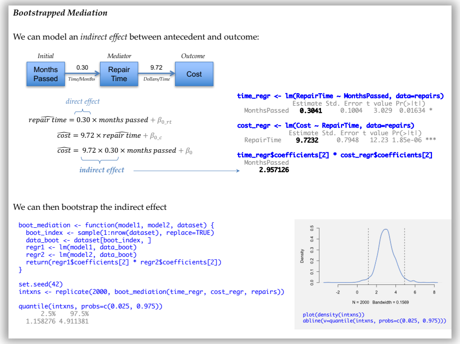
Linear Regression Revisited



Moderation



Mediation



Regression Models

***Explaining the relationship** of the*
***dependent variable**…*

"error"
captures all other explanations
we have missed

*… to **independent variables***

$$\boldsymbol{mpg} = b_0 + b_1\boldsymbol{cylinders} + b_2\boldsymbol{displacement} + b_3\boldsymbol{horsepower} + b_4\boldsymbol{weight} + b_5\boldsymbol{acceleration} + b_6\boldsymbol{model\_year} + b_7\boldsymbol{origin} + \boldsymbol{\varepsilon}$$

Regression coefficients capture
the degree and direction of relationship

🤔

Why did we pick these indpendent variables?

🤔

Are there other factors we might have missed?

Explanatory Models

Why did we pick these independent variables?

$mpg, cylinders, displacement, horsepower,$
$weight, acceleration, model\_year, origin$

Are there other factors we might have missed?

$technologies\ (e.g., fuelinjection)$
$aero\_dynamic, engine\_design, production\_quality$

Useful

Convenient & Available

Meaningful to Experts

Traditionally Understood / Used

Not Useful?

Hard to Measure

Hard to Understand / Interpret

Not Well Understood

🤔

What is the purpose of an ***explanatory "model"***

What is the Purpose of Explanatory Models?

Which model is correct?

https://goo.gl/LSNfSA

"All models are wrong but some are useful"

--George Box

***Explanatory models** are meant to be **interpretable and useful**, not "accurate"*

What makes a model explanatory?

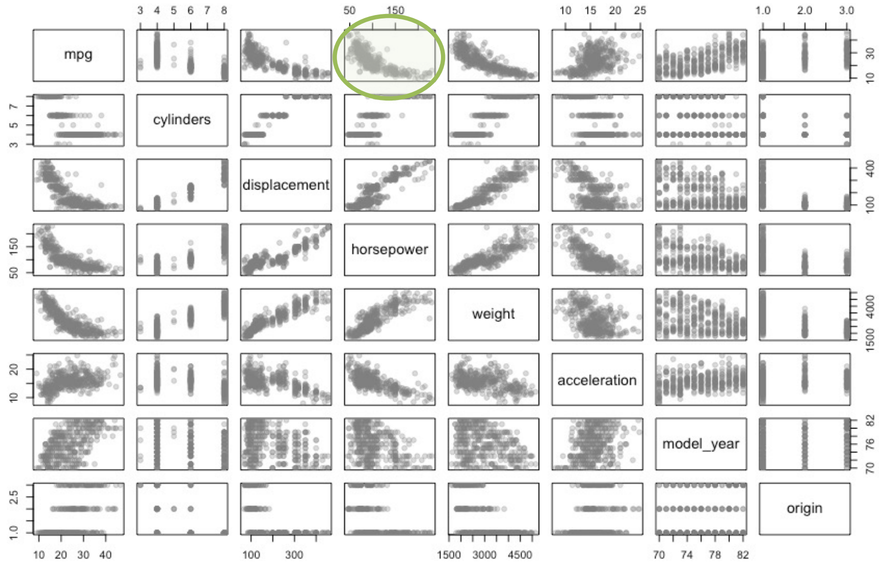
Box, G. E. (1976). Science and statistics.
Journal of the American Statistical Association, *71*(356), 791–799.

Revisiting Non-Linearity

Raw unit multiple regression

```
regr <- lm(mpg ~ cylinders + displacement + horsepower + weight +
                acceleration + model_year + factor(origin),
           data=cars)
```

```
(Intercept)      -1.795e+01  4.677e+00  -3.839 0.000145 ***
cylinders        -4.897e-01  3.212e-01  -1.524 0.128215
displacement      2.398e-02  7.653e-03   3.133 0.001863 **
horsepower       -1.818e-02  1.371e-02  -1.326 0.185488
weight           -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
acceleration      7.910e-02  9.822e-02   0.805 0.421101
model_year        7.770e-01  5.178e-02  15.005  < 2e-16 ***
factor(origin)2   2.630e+00  5.664e-01   4.643 4.72e-06 ***
factor(origin)3   2.853e+00  5.527e-01   5.162 3.93e-07 ***

Multiple R-squared:  0.8242, Adjusted R-squared:  0.8205
```



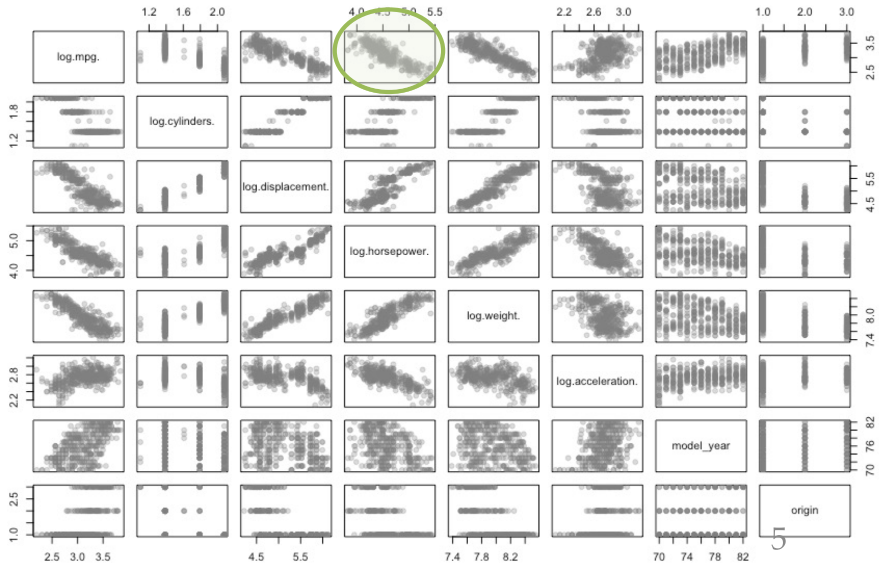
Different factors are now significant!
Notice that many relationships are now more linear

Log transformed multiple regression

```
cars_log <- with(cars, data.frame(log(mpg), log(cylinders),
                  log(displacement), log(horsepower), log(weight),
                  log(acceleration), model_year, origin))

regr_log <- lm(
  log.mpg. ~ log.cylinders. + log.displacement. +
             log.horsepower. + log.weight. + log.acceleration. +
             model_year + factor(origin), data=cars_log)
```

```
(Intercept)        7.301938   0.361777  20.184  < 2e-16 ***
log.cylinders.    -0.081915   0.061116  -1.340  0.18094
log.displacement.  0.020387   0.058369   0.349  0.72707
log.horsepower.   -0.284751   0.057945  -4.914 1.32e-06 ***
log.weight.       -0.592955   0.085165  -6.962 1.46e-11 ***
log.acceleration. -0.169673   0.059649  -2.845  0.00469 **
model_year         0.030239   0.001771  17.078  < 2e-16 ***
factor(origin)2    0.050717   0.020920   2.424  0.01580 *
factor(origin)3    0.047215   0.020622   2.290  0.02259 *

Multiple R-squared:  0.8919, Adjusted R-squared:  0.8897
```

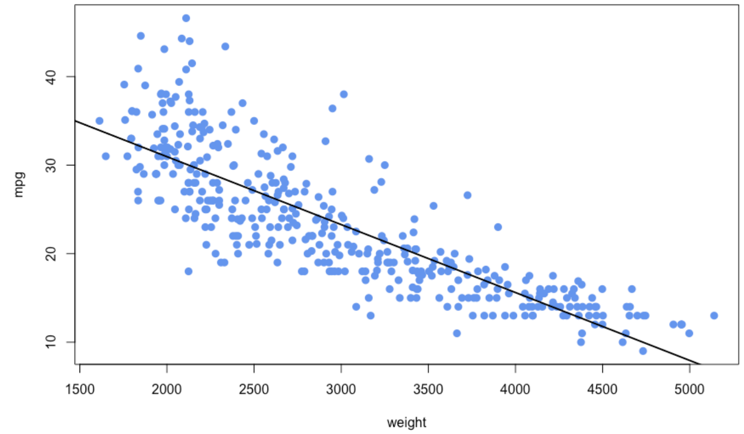
Variance Explained (R^2) has improved



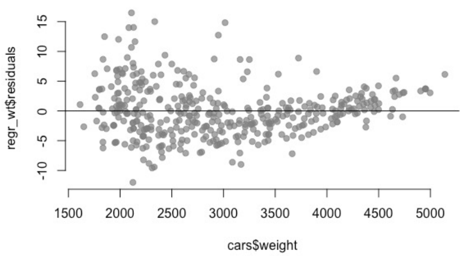
Regressing mpg over weight

```
regr_wt <- lm(mpg ~ weight, data=cars)
summary(regr_wt)

               Estimate Std. Error t value Pr(>|t|)
(Intercept) 46.3173644  0.7952452   58.24   <2e-16 ***
weight      -0.0076766  0.0002575  -29.81   <2e-16 ***

Multiple R-squared:  0.6918,     Adjusted R-squared:  0.691
```

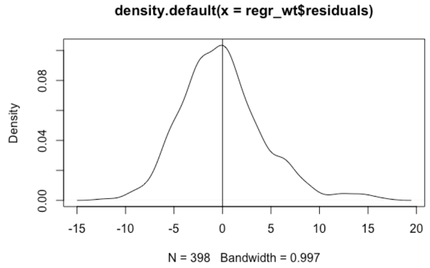
```
with(cars, plot(weight, mpg))
abline(regr_wt)
```




```
plot(cars$weight, regr_wt$residuals)
abline(h=mean(regr_wt$residuals))
```



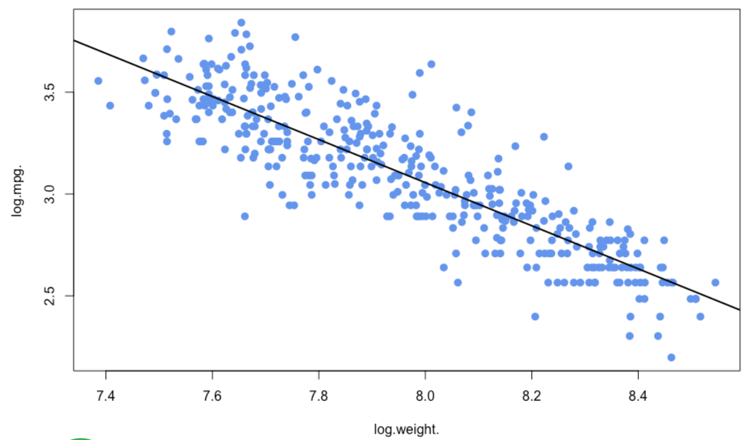

```
plot(density(regr_wt$residuals))
abline(v=mean(regr_wt$residuals))
```



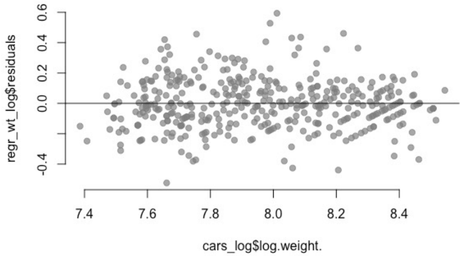
Regressing log(mpg) over log(weight)

```
regr_wt_log <- lm(log.mpg. ~ log.weight., data=cars_log)
summary(regr_wt_log)
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  11.5219     0.2349   49.06   <2e-16 ***
log.weight.  -1.0583     0.0295  -35.87   <2e-16 ***

Multiple R-squared:  0.7647,     Adjusted R-squared:  0.7641
```

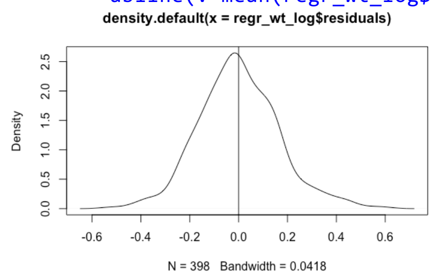
```
with(cars_log, plot(log.weight., log.mpg.))
abline(regr_wt_log)
```




```
plot(cars_log$log.weight., regr_wt_log$residuals)
abline(h=mean(regr_wt_log$residuals))
```




```
plot(density(regr_wt_log$residuals))
abline(v=mean(regr_wt_log$residuals))
```



Log transformation improved:

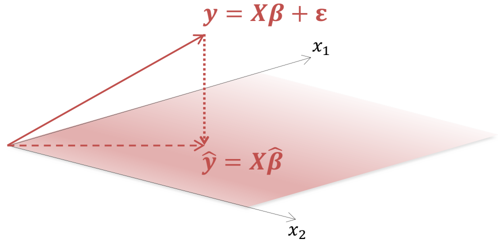
- *the distribution of residuals around 0*
- *the independence of IV and errors*
- *the variance explained (R^2) of model*

Deeper look at: Multicollinearity

Let's see how multicollinearity affects the linear algebra of regression:

$$\hat{y} = X\hat{\beta}; \qquad \hat{y} = Hy$$

$$\hat{y} = X\left(X^T X\right)^{-1} X^T y$$



$X =$

log.weight.	log.acceleration.
8.161660	2.484907
8.214194	2.442347
8.142063	2.397895
8.141190	2.484907
8.145840	2.351375
8.375860	2.302585

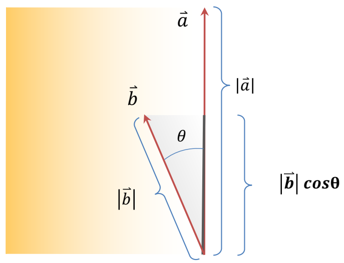
The "Dot Product":

$$\vec{a} \cdot \vec{b} = |\vec{a}||\vec{b}|cos\theta = \sum_i a_i b_i$$

*Scalar that tells us the **similarity of two vectors***

```
sum(t(log_weight) * log_acclxn)
log_weight %*% log_acclxn
```

8501.307 *cosine normalizes this to [-1,1]*



The "Gram Matrix": $X^T X$

***Matrix of dot-products** of all column vectors*

```
gram <- function(X) { t(X) %*% X }
```

	log.weight.	log.acceleration.
log.weight.	**24863.547**	**8501.307**
log.acceleration.	**8501.307**	**2928.969**

$\left(X^T X\right)^{-1}$

***Inverse** of Gram Matrix – **reverses** the transformation by Gram Matrix*

```
solve(gram(X))
```

	log.weight.	log.acceleration.
log.weight.	**0.005**	**-0.015**
log.acceleration.	**-0.015**	**0.045**

*We can only **solve for the inverse** if $X^T X$ is **non-singular**: $det\left(X^T X\right) \neq 0$*

```
det(gram(X1))
```

552322.2

Let's look at just the X variables log-weight and log-acceleration

```
X1 <- as.matrix(na.omit(cars_power)[,4:5])
gram(X1)
solve(gram(X1))

# Determinant
det(gram(X1)) # 552322.2
```

$X^T X$

	log.weight.	log.acceleration.
log.weight.	24863.547	8501.307
log.acceleration.	8501.307	2928.969

$(X^T X)^{-1}$

	log.weight.	log.acceleration.
log.weight.	0.005	-0.015
log.acceleration.	-0.015	0.045

Let's add a "highly collinear" variable to X

```
noise <- rnorm(nrow(X1), mean=0, sd=0.001)
X2 <- cbind(X1, collinear = X1[, 'log.weight.'] + noise)
gram(X2)
solve(gram(X2))

# Determinant
det(gram(X2)) # 220.5415
```

$X^T X$

	log.weight.	log.acceleration.	collinear
log.weight.	24863.547	8501.307	24863.288
log.acceleration.	8501.307	2928.969	8501.218
collinear	24863.288	8501.218	24863.029

$(X^T X)^{-1}$

	log.weight.	log.acceleration.	collinear
log.weight.	2504.398	-0.094	-2504.391
log.acceleration.	-0.094	0.045	0.079
collinear	-2504.391	0.079	2504.391

Determinant drops in size; inverse of Gram matrix changes dramatically

Let's add a "perfectly collinear" variable to X

```
X3 <- cbind(X1, collinear = X1[, 'log.weight.'])
gram(X3)
solve(gram(X3))

# Determinant
det(gram(X3)) # 0
```

$X^T X$

	log.weight.	log.acceleration.	collinear
log.weight.	24863.547	8501.307	24863.547
log.acceleration.	8501.307	2928.969	8501.307
collinear	24863.547	8501.307	24863.547

$(X^T X)^{-1}$

```
Error in solve.default(dot_products(X3)) :
  LAPACK routine dgesv: system is exactly singular
```

Determinant becomes zero; inverse of Gram matrix cannot be solved

NOTE: linear algebra routines are performed by your operating system, not R:
low-level packages *(e.g.,* ***BLAS****) do vector/matrix math;* ***high-level packages*** *(****LAPACK****) solve equations, etc.*

Multicollinearity: Stepwise VIF Selection

```
lm(log.mpg. ~ log.cylinders. + log.displacement. +
              log.horsepower. + log.weight. + log.acceleration. +
              model_year + factor(origin), data = cars_log)
```

```
Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        7.301938   0.361777  20.184  < 2e-16 ***
log.cylinders.    -0.081915   0.061116  -1.340  0.18094
log.displacement.  0.020387   0.058369   0.349  0.72707
log.horsepower.   -0.284751   0.057945  -4.914 1.32e-06 ***
log.weight.       -0.592955   0.085165  -6.962 1.46e-11 ***
log.acceleration. -0.169673   0.059649  -2.845  0.00469 **
model_year         0.030239   0.001771  17.078  < 2e-16 ***
factor(origin)2    0.050717   0.020920   2.424  0.01580 *
factor(origin)3    0.047215   0.020622   2.290  0.02259 *

Multiple R-squared:  0.8919, Adjusted R-squared:  0.8897
```

```
vif(regr_log)
                       GVIF
log.cylinders.    10.456738
log.displacement. 29.625732
log.horsepower.   12.132057
log.weight.       17.575117
log.acceleration.  3.570357
model_year         1.303738
factor(origin)     2.656795
```

First candidate for removal

$$VIF_j = \frac{1}{1 - R_j^2}$$

VIF = 1 implies no multicollinearity
VIF ≥ 5 requires $R^2 \geq 0.8$
VIF ≥ 10 requires $R^2 \geq 0.9$

Stepwise removal:

- *Displacement*
- *Horsepower*
- *Cylinders*

Model Fit: 0.8919 → 0.8856
- R^2 does not drop by much
- We prefer the simpler explanation

"Principle of Parsimony"
*Given similar model fit (explanatory power), we favor **simpler explanations** over complex ones*

```
lm(log.mpg. ~ log.weight. + log.acceleration. +
              model_year + factor(origin), data = cars_log)
```

```
Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        7.431155   0.312248  23.799  < 2e-16 ***
log.weight.       -0.876608   0.028697 -30.547  < 2e-16 ***
log.acceleration.  0.051508   0.036652   1.405  0.16072
model_year         0.032734   0.001696  19.306  < 2e-16 ***
factor(origin)2    0.057991   0.017885   3.242  0.00129 **
factor(origin)3    0.032333   0.018279   1.769  0.07770 .

Multiple R-squared:  0.8856, Adjusted R-squared:  0.8841
```

```
vif(regr_log_refined)
                      GVIF
log.weight.       1.926377
log.acceleration. 1.303005
model_year        1.167241
factor(origin)    1.692320
```

Is it reasonable to drop explanatory variables?
Is the regression model still useful?

Read more on VIF Metrics : https://stats.stackexchange.com/questions/70679/which-variance-inflation-factor-should-i-be-using-textgvif-or-textgvif

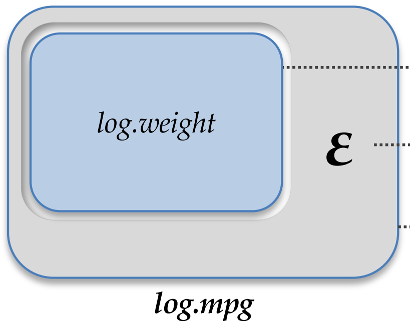
Residuals of Regression

$$\boldsymbol{log.mpg} = b_0 + b_1(\boldsymbol{log.weight}) + \boldsymbol{\varepsilon}$$

Residuals capture the portion of the dependent variable ***unexplained*** *by independent variables*

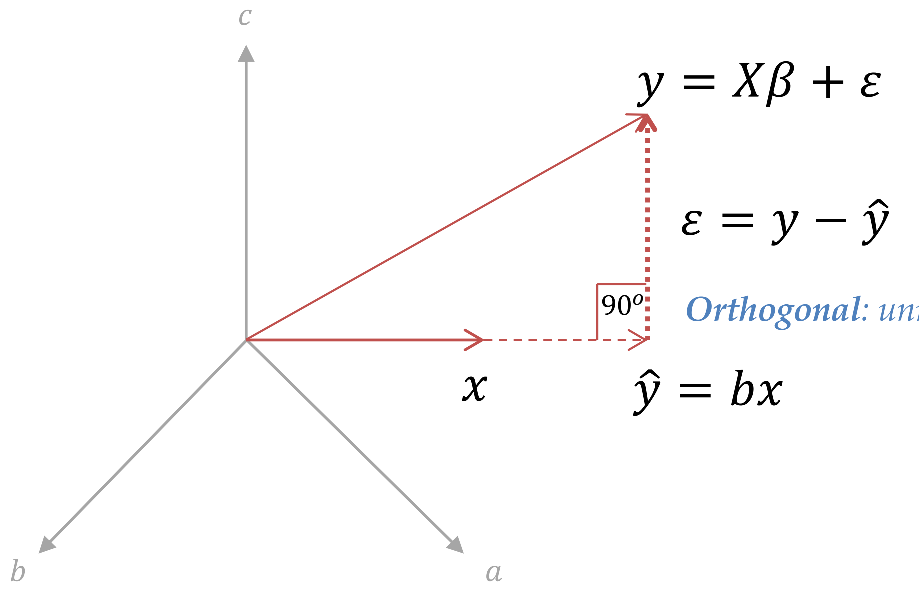
```
mpg_wt_residuals <-
  with(cars_log,
       data.frame(log.mpg., log.weight.,
                  residuals = regr_wt_log$residuals))

round(cor(mpg_wt_residuals), 2)
```

```
             log.mpg. log.weight. residuals
log.mpg.         1.00
log.weight.     -0.87        1.00
residuals        0.49        0.00      1.00
```



Residuals of regression are always ***orthogonal*** ***to independent*** *variables*

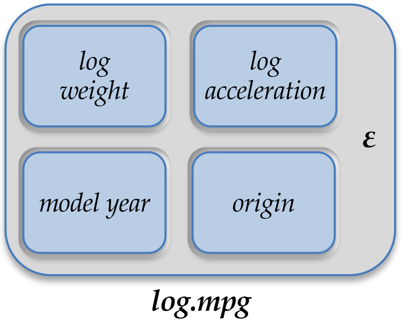
Residuals are still ***related to dependent*** *variables (they are a component of dependent variable)*



Orthogonal: unrelated or 90°

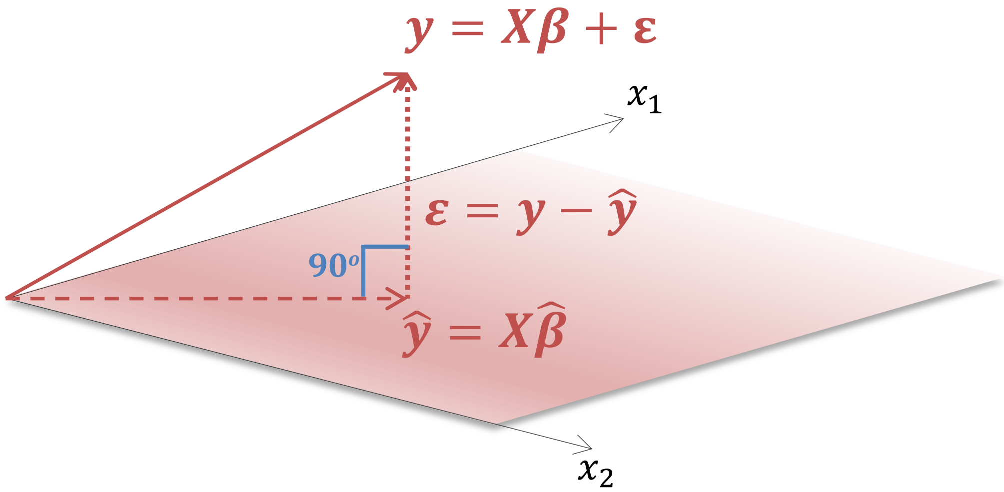
The same is true in a larger model (higher dimensions)

$$\boldsymbol{log.mpg} = b_0 + b_1(\boldsymbol{log.weight}) + b_2(\boldsymbol{log.acceleration})$$
$$+ b_3(\boldsymbol{model\_year}) + b_4(\boldsymbol{origin}) + \boldsymbol{\varepsilon}$$



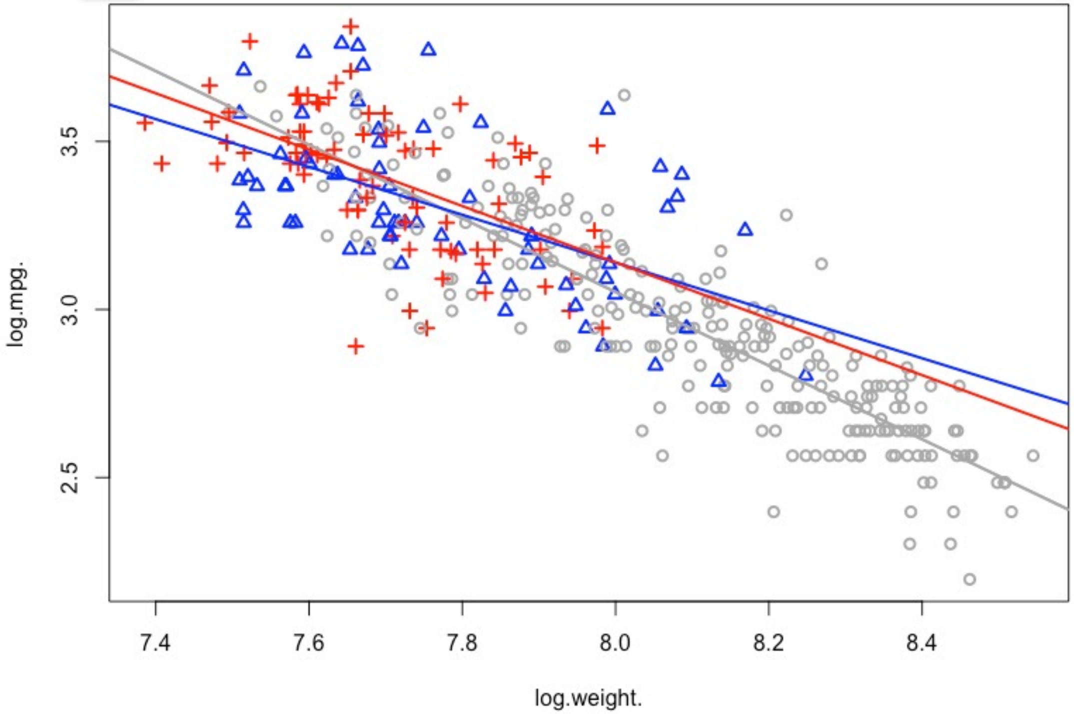

```
refined_data_residuals <-
  with(cars_log,
       data.frame(log.mpg., log.weight.,
                  log.acceleration., model_year, origin,
                  resdiuals = regr_log_refined$residuals))

round(cor(refined_data_residuals), 2)
```

```
                  log.mpg. log.weight. log.acceleration. model_year origin resdiuals
log.mpg.              1.00
log.weight.          -0.87        1.00
log.acceleration.     0.46       -0.43              1.00
model_year            0.58       -0.28              0.31       1.00
origin                0.56       -0.60              0.22       0.18   1.00
residuals             0.34        0.00              0.00       0.00   0.00      1.00
```



Contingency Perspective

*The effect of **weight** on **mpg** depends on **origin** of car*




```
# subset cars dataset by origin
cars_us <- subset(cars_log, origin==1)
cars_eu <- subset(cars_log, origin==2)
cars_jp <- subset(cars_log, origin==3)

# separate regressions of log.mpg ~ log.weight by origin
wt_regr_us <- lm(log.mpg. ~ log.weight., data=cars_us)
wt_regr_eu <- lm(log.mpg. ~ log.weight., data=cars_eu)
wt_regr_jp <- lm(log.mpg. ~ log.weight., data=cars_jp)

# plot points colorized by origin
origin_colors = c("darkgray", "blue", "red")
with(cars_log,
     plot(log.weight., log.mpg.,
          pch=origin, col=origin_colors[origin], lwd=2))

# plot separate regression lines colorized by origin
abline(wt_regr_us, col=origin_colors[1], lwd=2)
abline(wt_regr_eu, col=origin_colors[2], lwd=2)
abline(wt_regr_jp, col=origin_colors[3], lwd=2)
```

Is it normal for a relationship between two variables to depend on the level of a third variable?

Can regression model different levels of a relationship between two factors?

Contingency Perspective of Causal Science

Perspective that one set of rules does not universally apply

Outcome depends on situational factors

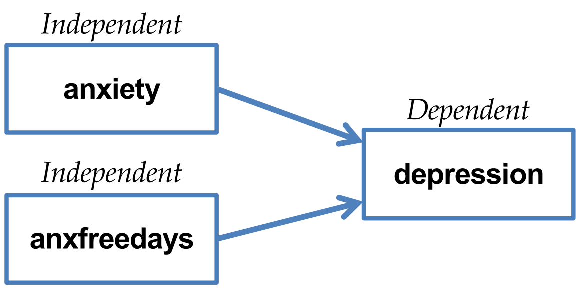
Path Diagrams and Causal Modeling

Direct Relationships: *Anxiety Example (fictional!)*

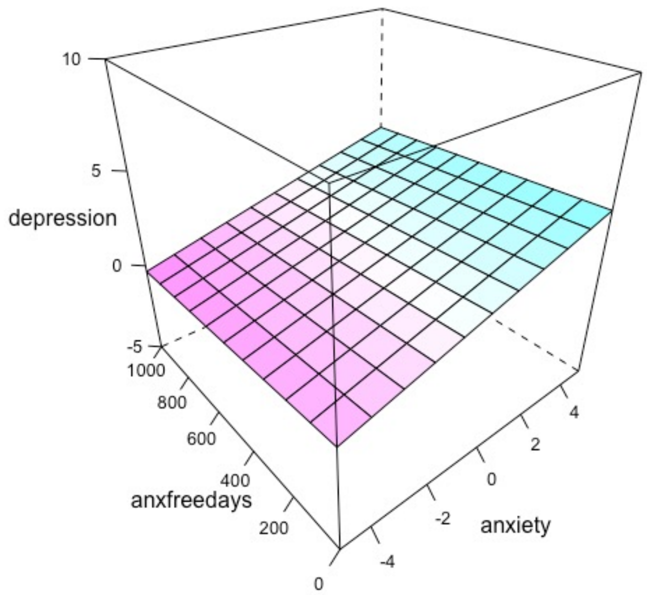
anxiety	*How anxious (worried, nervous) a person is feeling*	Concepts of interest
anxfreedays	*How many days since person last felt anxious*	
depression	*How depressed a person feels right now*	

Path Diagram: A *graphical* representation of researcher's *causal* model

- **Measured concepts** are shown as boxes
- **Causal relationships** are shown as arrows between boxes



Depression *depends on the level of* ***anxiety*** *a subject feels and the number of* ***anxiety-free-days****.*

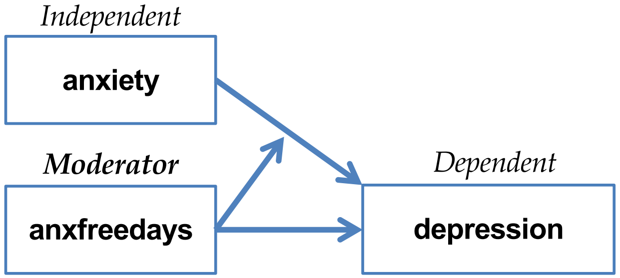


$$\boldsymbol{depression} = b_0 + b_1(\boldsymbol{anxiety}) + b_2(\boldsymbol{anxfreedays}) + \varepsilon$$

```
lm(depression ~ anxiety + anxfreedays)
```

Moderation and Interaction

Moderated Relationships



*The effect of an **independent** variable on a **dependent** variable is contingent on a **moderator**.*

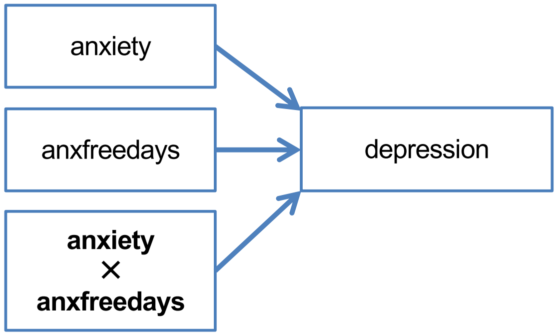
*The effect of **anxiety** on **depression** is contingent on **anxiety-free-days**.*

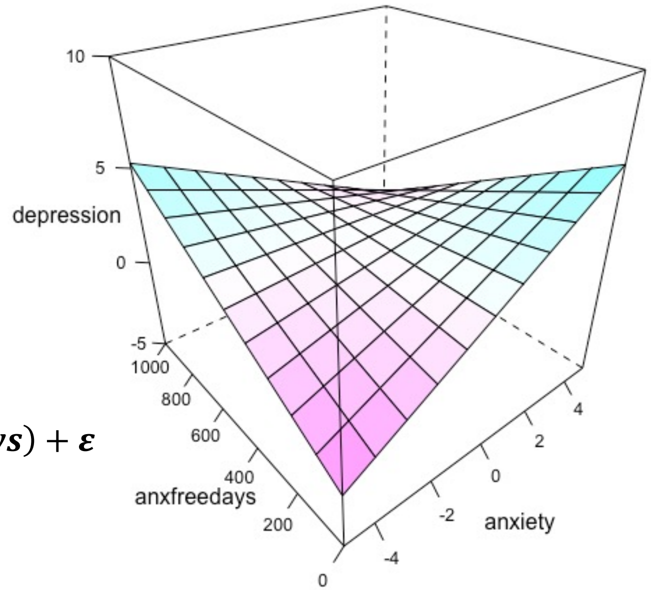
Interaction terms twist the regression surface

Moderation modeled in regression using an *interaction term*

anxiety × anxfreedays

***product** of independent variables*





$$\boldsymbol{depression} = b_0 + b_1(\boldsymbol{anxiety}) + b_2(\boldsymbol{anxfreedays}) + b_3(\boldsymbol{anxiety} \cdot \boldsymbol{anxfreedays}) + \boldsymbol{\varepsilon}$$

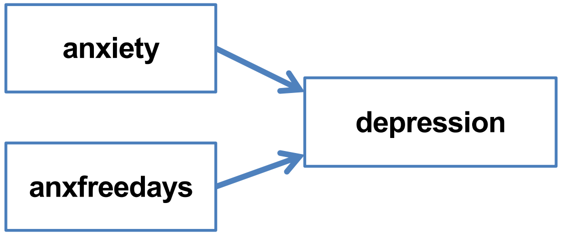
```
lm(depression ~ anxiety + anxfreedays + anxiety*anxfreedays)
```


Why interactions are products: https://stats.stackexchange.com/questions/393579/in-multiple-regression-why-are-interactions-modelled-as-products-and-not-somet

Multicollinearity in Interactions

```r
dep <- read.table("depression_intxn.txt", header=TRUE)
```

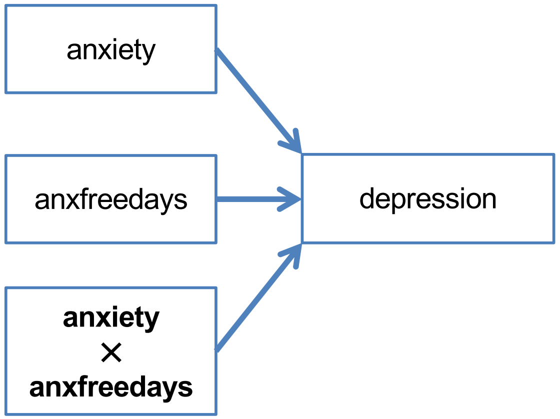
Modeling Direct Effects




```r
lm(depression ~ anxiety + anxfreedays, data=dep)
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.1871157  0.2759032   7.927 1.40e-11 ***
anxiety      0.3651384  0.0807465   4.522 2.19e-05 ***
anxfreedays -0.0006942  0.0008024  -0.865     0.39

Multiple R-squared:  0.2126, Adjusted R-squared:  0.1922
```

Adding the interaction




```r
lm(depression ~ anxiety + anxfreedays + anxiety*anxfreedays, data=dep)
```

```
Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          1.9141605  0.2978311   6.427 1.03e-08 ***
anxiety              0.8231915  0.2266133   3.633 0.000507 ***
anxfreedays          0.0001731  0.0008812   0.196 0.844791
anxiety:anxfreedays -0.0014553  0.0006749  -2.156 0.034231 *

Multiple R-squared:  0.258,  Adjusted R-squared:  0.2287
```

Symptoms of Multicollinearity:
- Change in size of estimates
- Inflation in standard error (variance inflation)
- Change in significance of estimates

Correlations between interaction and independent variables

```
cor(dep$anxiety, dep$anxiety * dep$anxfreedays)
```

```
[1] 0.9246831
```

```
cor(cbind(dep, intxn = dep$anxiety * dep$anxfreedays))
```

```
             depression     anxiety anxfreedays     intxn
depression   1.00000000 0.45272115 -0.05920345 0.3312457
anxiety      0.45272115 1.00000000  0.06210723 0.9246831
anxfreedays -0.05920345 0.06210723  1.00000000 0.2308910
intxn        0.33124565 0.92468309  0.23089096 1.0000000
```

⚠ *Even weakly correlated independent variables can produce highly correlated interactions*

Variance inflation with interactions

```
vif(dep_regr_intxn)
```

```
 anxiety  anxfreedays  anxiety:anxfreedays
8.281544     1.268106             8.714156
```

🤔 ***Dropping the interaction term is not an option***
We need way to improve the interpretability of coefficients

Mean-centered Interactions

Mean Centering

$$X_{mc} = X - \bar{X}$$

```
anxiety_mc <- scale(dep$anxiety, center=TRUE, scale=FALSE)
anxfreedays_mc <- scale(dep$anxfreedays, center=TRUE, scale=FALSE)
```

Mean-Centered Correlation

```
cor(anxiety_mc, anxiety_mc*anxfreedays_mc)
[1,] -0.1050647
```

Mean-centered Regression with Interaction

```
summary(lm(dep$depression ~ anxiety_mc + anxfreedays_mc + anxiety_mc*anxfreedays_mc))
```

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	2.0431974	0.0771099	26.497	< 2e-16	***
anxiety_mc	**0.3427213**	**0.0795806**	**4.307**	**4.89e-05**	*******
anxfreedays_mc	-0.0001321	0.0008262	-0.160	0.8734	
anxiety_mc:anxfreedays_mc	**-0.0014553**	**0.0006749**	**-2.156**	**0.0342**	*****

```
Multiple R-squared:  0.258,    Adjusted R-squared:  0.2287
```

***Recall** our original correlation and direct effects*

```
cor(dep$anxiety, dep$anxiety * dep$anxfreedays)
[1] 0.9246831
lm(depression ~ anxiety + anxfreedays, data=dep)
```

	Estimate	Std. Error	t-value
anxiety	***0.3651384***	***0.0807465***	***4.522***
anxfreedays	*-0.0006942*	*0.0008024*	*-0.865*

*Mean-centering makes it **easier to interpret coefficients** despite multicollinearity*

Fully standardized Interactions

Fully Standardized Correlation

$$X_{std} = \frac{X - \bar{X}}{s_x}$$

```
with(as.data.frame(scale(dep)),
     cor(anxiety, anxiety*anxfreedays))
[1,] -0.1050647
```

Fully Standardized Regression with Interaction

```
summary(lm(depression ~ anxiety + anxfreedays + anxiety*anxfreedays),
           data=as.data.frame(scale(dep)))
```

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	0.01109	0.09832	0.113	0.9105	
anxiety_s	**0.43004**	**0.09986**	**4.307**	**4.89e-05**	*******
anxfreedays_s	-0.01669	0.10433	-0.160	0.8734	
anxiety_s:anxfreedays_s	**-0.18082**	**0.08386**	**-2.156**	**0.0342**	*****

```
Multiple R-squared:  0.258,  Adjusted R-squared:  0.2287
```

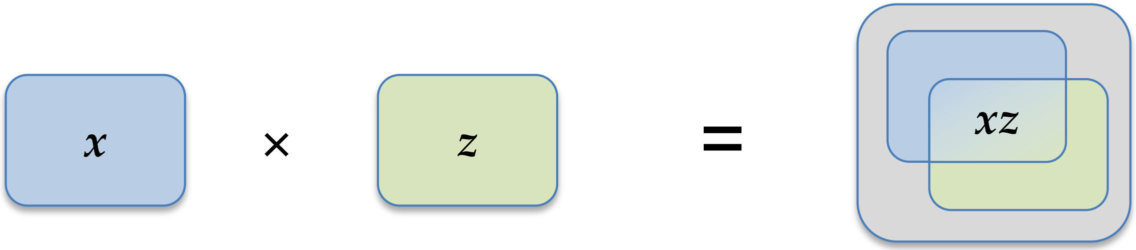
NOTE: We cannot statistically remove multicollinearity
We can only improve interpretability of coefficients
There is no change in significances or R^2

*Standardizing data requires mean-centering, so it **has the same effect on multicollinearity** as mean-centering*

More at: Echambadi, R., and Hess, J. D. 2007. "Mean-Centering Does Not Alleviate Collinearity Problems in Moderated Multiple Regression Models," Marketing Science (26:3), pp. 438–445.

Orthogonalized Interaction Terms

Consider an interaction between weakly correlated variables:

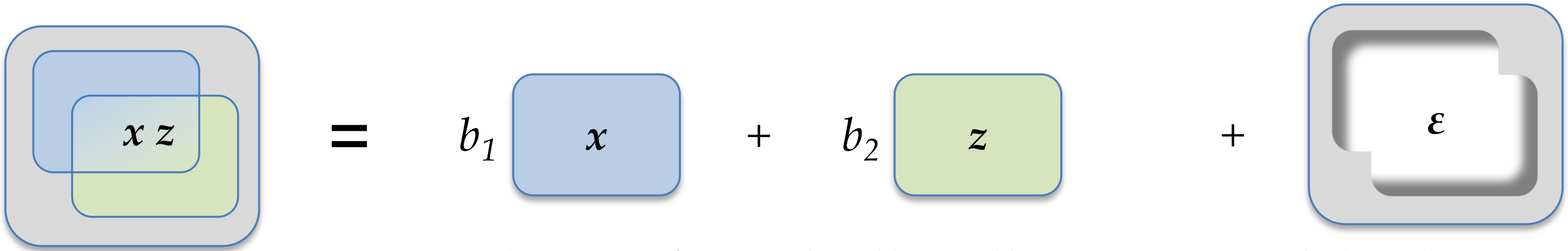


Even weakly correlated variables can produce highly correlated interaction terms

```
set.seed(53)
x = round(rnorm(100, mean=30, sd=5))
z = round(rnorm(100, mean=55, sd=7))

cor(x, z)
  [1] 0.08

xz = x*z

cor(cbind(x, z, xz))
```

	x	z	xz
x	1.00	0.08	0.82
z	0.08	1.00	0.62
xz	**0.82**	**0.62**	1.00

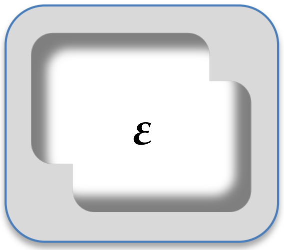
*Use **regression** to remove collinearity between interaction and its original variables:* $\boldsymbol{xz} = \beta_1 x + \beta_2 z + \boldsymbol{\varepsilon}$



The portions of $\boldsymbol{x \times z}$ *explained by* $\boldsymbol{x}$ *and by* $\boldsymbol{z}$ *are heavily correlated (variabilities overlap!)*

Residuals are the portion of $\boldsymbol{x \times z}$ *that is not explained by* $\boldsymbol{x}$ *or by* $\boldsymbol{z}$

orthogonalized interaction



a pure interaction term uncorrelated with its main variables!

```
oregr <- lm(xz ~ x + z)

xz_o <- oregr$residuals

cor(cbind(x, z, xz_o))
```

	x	z	xz_o
x	1.00	0.08	0.00
z	0.08	1.00	0.00
xz_o	0.00	0.00	1.00

Orthogonalized Moderation

We use the residual of the interaction as the interaction of the dependent

$$\boldsymbol{anxiety} \cdot \boldsymbol{anxfreedays} = b_0 + b_1(anxiety) + b_2(anxfreedays) + \boldsymbol{\varepsilon_{intxn}}$$

$$\boldsymbol{depression} = b_0 + b_1(anxiety) + b_2(anxfreedays) + b_3\boldsymbol{\varepsilon_{intxn}} + \varepsilon$$

Residuals of interaction's regression

```
anx_x_anxfree     <- dep$anxiety * dep$anxfreedays
interaction_regr  <- lm(anx_x_anxfree ~ dep$anxiety + dep$anxfreedays)
interaction_ortho <- interaction_regr$residuals
```

Correlation of residual

```
round(cor(cbind(dep, interaction_ortho)), 2)
                  depression anxiety anxfreedays interaction_ortho
depression              1.00    0.45       -0.06             -0.21
anxiety                 0.45    1.00        0.06              0.00
anxfreedays            -0.06    0.06        1.00              0.00
interaction_ortho      -0.21    0.00        0.00              1.00
```

Regression Model with Residual

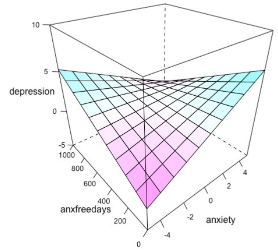
```
summary(lm(depression ~ anxiety + anxfreedays + interaction_ortho,
           data=dep))

  Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
  (Intercept)        2.1871157  0.2695889   8.113 6.70e-12 ***
  anxiety            0.3651384  0.0788986   4.628 1.49e-05 ***
  anxfreedays       -0.0006942  0.0007840  -0.885   0.3787
  interaction_ortho -0.0014553  0.0006749  -2.156   0.0342 *

  Multiple R-squared:  0.258,    Adjusted R-squared:  0.2287
```

Recall our original direct effects

```
lm(depression ~ anxiety + anxfreedays, data=dep)

 Coefficients:
               Estimate Std. Error
  (Intercept)  2.1871157  0.2759032
  anxiety      0.3651384  0.0807465
  anxfreedays -0.0006942  0.0008024

  Multiple R-squared:  0.2126
```

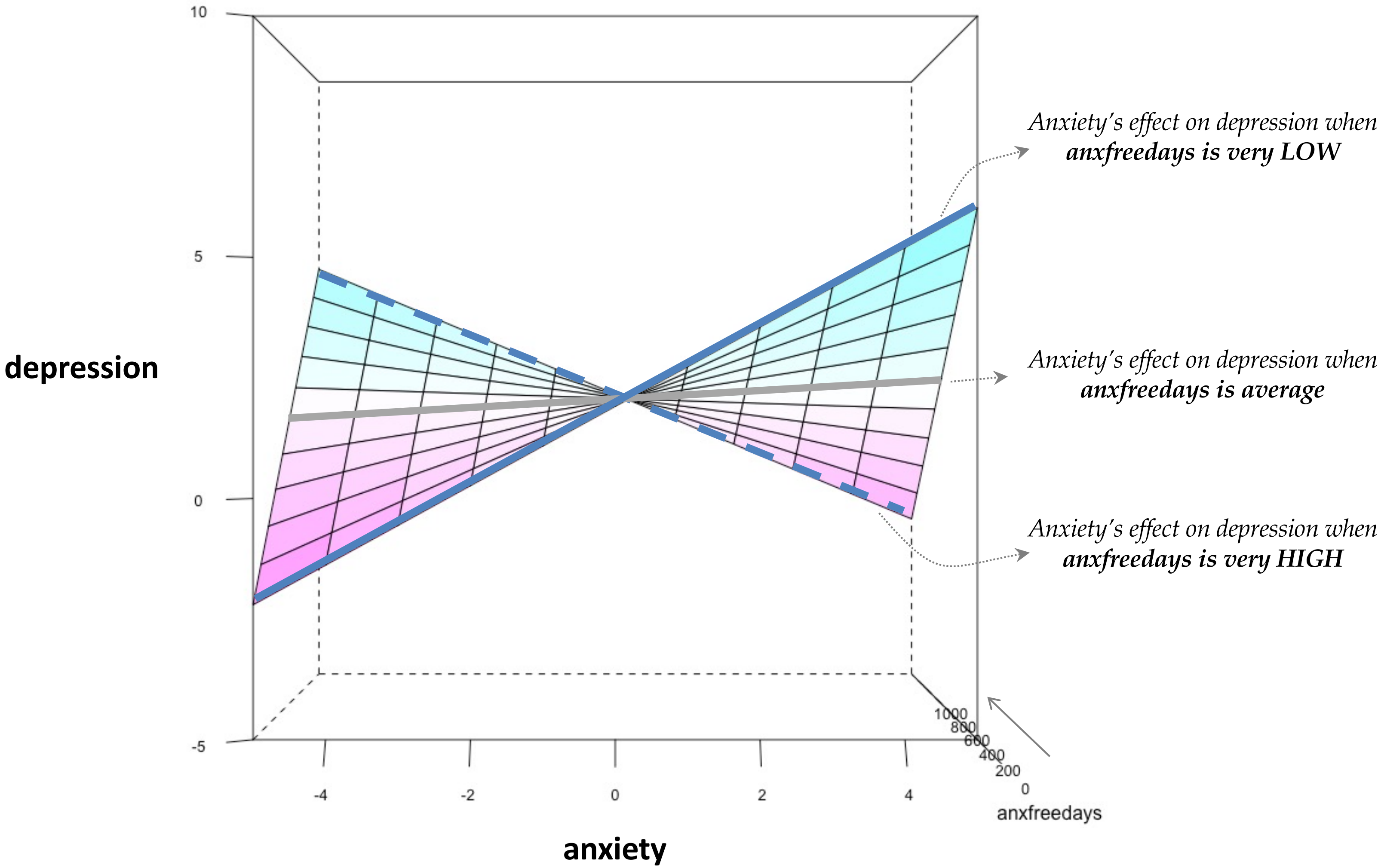
Orthogonalization gives us the most interpretable coefficients

NOTE: orthogonalization still does not statistically remove multicollinearity



Interpreting Interactions

```
Estimate Std. Error t value Pr(>|t|)
 (Intercept)                  2.1871157 ***
 anxiety                      0.3651384 ***
 anxfreedays                 -0.0006942
 anx_free_regr$residuals     -0.0014553 *
```



Levels of a moderator

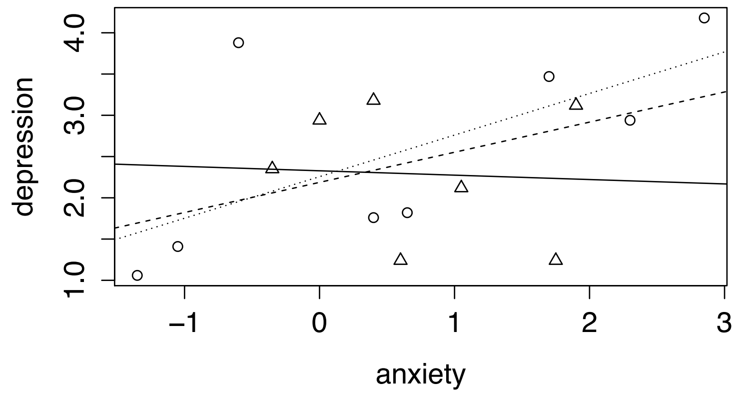
```
# Find subsets of data at different levels of the moderator
dep_low_anxfree <- subset(dep, scale(anxfreedays) < -1)
dep_high_anxfree <- subset(dep, scale(anxfreedays) > 1)
```

```
subset(dep, scale(anxfreedays) < -1)
```

*Get subset of data that where **anxfreedays** is 1 standard deviation or more below the mean*

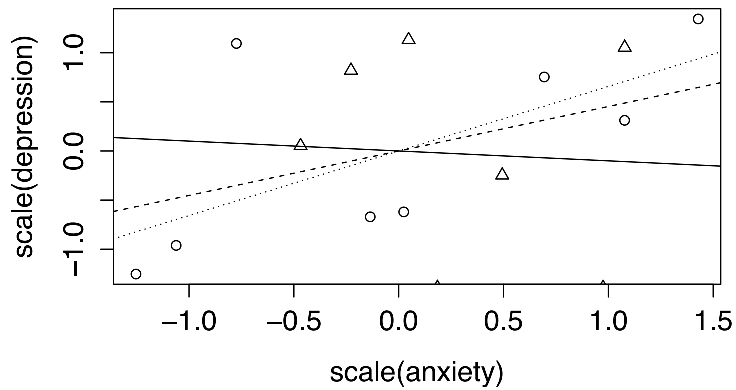
Unscaled Interaction plot

```
# Conduct unscaled regressions at different levels of anxfreedays
dep_regr <- with(dep, lm(depression ~ anxiety + anxfreedays))
dep_low_regr <- with(dep_low_anxfree, lm(depression ~ anxiety))
dep_high_regr <- with(dep_high_anxfree, lm(depression ~ anxiety))

# Plot unscaled low and high points based on anxfreedays
with(dep_low_anxfree, plot(anxiety, depression))
with(dep_high_anxfree, points(anxiety, depression, pch=19))

# Draw unscaled low-med-high regression lines based on anxfreedays
abline(dep_low_regr, lty="dotted")
abline(dep_regr, lty="dashed")
abline(dep_high_regr, lty="solid")
```



Fully Scaled Interaction plot

```
# Plot low and high points based on anxfreedays
with(dep_low_anxfree, plot(scale(anxiety), scale(depression)))
with(dep_high_anxfree, points(scale(anxiety), scale(depression), pch=2))

# Draw low-med-high regression lines based on anxfreedays
with(dep_low_anxfree, abline(lm(scale(depression)~scale(anxiety)), lty="dotted"))
with(dep, abline(lm(scale(depression)~scale(anxiety)), lty="dashed"))
with(dep_high_anxfree, abline(lm(scale(depression)~scale(anxiety)), lty="solid"))
```



How do these plots compare to the 3D visualization of moderation we saw?

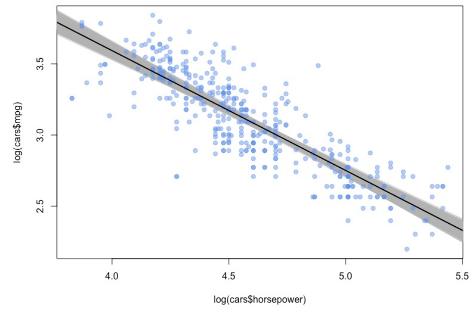
Uses of Regression in Modeling

Explanatory Modeling: *explaining relationship between dependent and independent variables*

*Regress **dependent variable** over **independent variables***

$$Y = \beta_0 + \beta_1 X_2 + \cdots \beta_k X_k + \varepsilon$$

Focus on coefficients and compute R^2



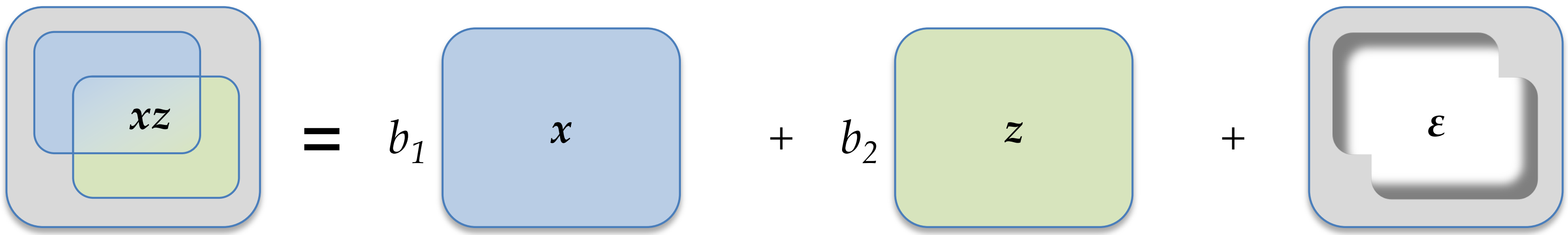
Variance Explained: *measuring variance shared by set of independent variables*

*Regress **independent variable** over **other independent variables** to **compute VIF***

$$X_j = \beta_0 + \beta_2 X_2 + \cdots \beta_k X_k + \varepsilon \qquad VIF_j = \frac{1}{1 - R_j^2}$$

Only compute R^2

Orthogonalization: *removing collinearity in data*



*Regress **interaction term** over its **main variables***

$$X_i X_j = \beta_0 + \beta_1 X_i + \beta_2 X_j + \underbrace{\varepsilon}$$

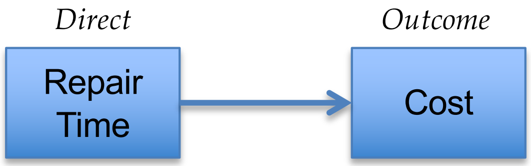
*Only retain **error term!***

Chain of Effects

Fictional case: An analyst at a car repair shop that does *annual inspections & repairs* wishes to explain how much the ***months passed since the last repair*** is related to the ***repair costs*** being high.

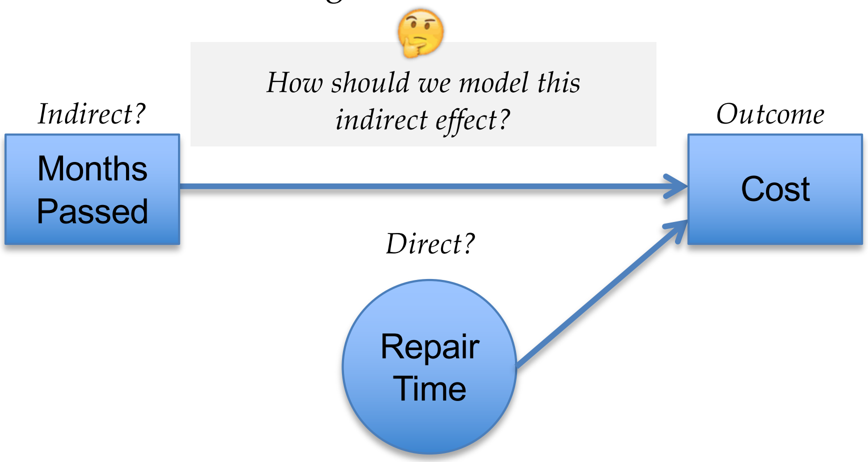

```
months_cost_regr <- lm(Cost ~ MonthsPassed, data=repairs)
summary(months_cost_regr)

             Estimate Std. Error t value Pr(>|t|)
(Intercept)    81.662      6.267  13.030 1.14e-06 ***
MonthsPassed    2.931      1.040   2.817   0.0226 *
```

A mechanic points out that the *direct reason* for **repairs costs** to be high is the total ***repair time*** it takes for mechanics to fix any issues.




```
time_cost_regr <- lm(Cost ~ RepairTime, data=repairs)
summary(time_cost_regr)

            Estimate Std. Error t value Pr(>|t|)
(Intercept)  60.6385     3.1432   19.29 5.41e-08 ***
RepairTime    9.7232     0.7948   12.23 1.85e-06 ***
```

However, another mechanic quickly points out that the **months passed** is an *indirect* explanation, because more things break over the months and so the **repair time** goes up, which *directly* affects **repair cost**.

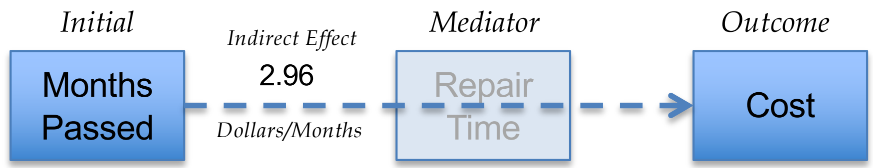



```
total_regr <- lm(Cost ~ MonthsPassed + RepairTime, data=repairs)
summary(total_regr)

             Estimate Std. Error t value Pr(>|t|)
(Intercept)  60.56983    3.41518  17.735 4.47e-07 ***
MonthsPassed -0.05681    0.51759  -0.110    0.916
RepairTime    9.82294    1.24384   7.897 9.90e-05 ***

cor(repairs$MonthsPassed, repairs$RepairTime)
[1] 0.7308738
```

Adding a correlated mediating explanation can cause multicollinearity problems

Calculating Indirect Effects

We can model an ***indirect effect*** between antecedent and outcome:



$$\widehat{repair\ time} = \overbrace{0.30}^{\textit{direct effect}} \times months\ passed + \beta_{0\_rt}$$

$$\widehat{cost} = 9.72 \times \widehat{repair\ time} + \beta_{0\_c}$$

$$\widehat{cost} = \underbrace{9.72 \times 0.30}_{\textit{indirect effect}} \times months\ passed + \beta_0$$

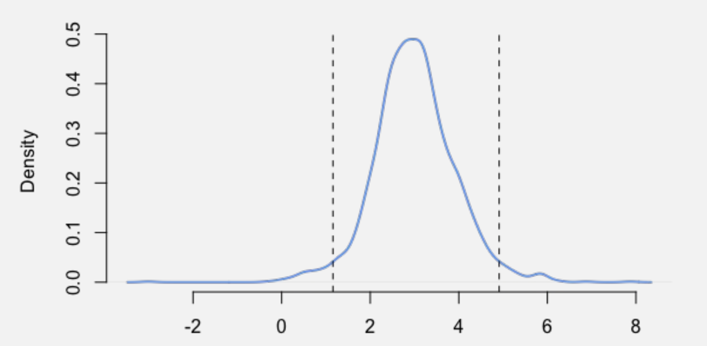
```
months_repair_regr<- lm(RepairTime ~ MonthsPassed, data=repairs)
             Estimate Std. Error t value Pr(>|t|)
MonthsPassed   0.3041     0.1004   3.029  0.01634 *

time_cost_regr <- lm(Cost ~ RepairTime, data=repairs)
             Estimate Std. Error t value Pr(>|t|)
RepairTime     9.7232     0.7948   12.23 1.85e-06 ***

months_repair_regr$coefficients[2] * time_cost_regr$coefficients[2]

MonthsPassed
    2.957126
```

We can compute the indirect effect
But we cannot always easily derive its significance

Bootstrapped Confidence Interval of Indirect Effects

We can ***bootstrap the significance*** of the indirect effect:

```
boot_mediation <- function(model1, model2, dataset) {
  boot_index <- sample(1:nrow(dataset), replace=TRUE)
  data_boot <- dataset[boot_index, ]
  regr1 <- lm(model1, data_boot)
  regr2 <- lm(model2, data_boot)
  return(regr1$coefficients[2] * regr2$coefficients[2])
}

set.seed(42)
indirect <- replicate(2000,
              boot_mediation(months_repair_regr, time_cost_regr, repairs))

quantile(indirect, probs=c(0.025, 0.975))
     2.5%    97.5%
  1.158276 4.911381
```




```
plot(density(indirect))
abline(v=quantile(indirect, probs=c(0.025, 0.975)))
```