

## Example 2a: Predicting Binary Outcomes via STATA LOGIT, R GLM, and SAS GLIMMIX (complete syntax, data, and output available for STATA, R, and SAS electronically))

The (fake) data for this example demonstrating “logistic regression” (i.e., using a logit link function and Bernoulli conditional response distribution to predict a binary outcome) came from: <https://stats.idre.ucla.edu/sas/dae/ordinal-logistic-regression/>. In this example we will predict a student’s **binary decision** of how likely they are to apply to grad school (0=no, >0=pry) using undergraduate GPA (centered at 3.0), whether at least one of their parents has a graduate degree (0=no, 1=yes), and whether they attended a private university (0=no, 1=yes).

For the polychoric and polyserial correlations, I am using a user-created STATA command POLYCHORIC and POLYCOR in R. For the predictive models, I am using LOGIT in STATA (along with a user-created command FITSTAT) and GLM in R, which do not use denominator degrees of freedom. Consequently, single slopes will be tested using univariate Wald tests (i.e., the z-tests as given directly in the program output for Estimate/SE), and sets of slopes will be tested using likelihood ratio tests (the difference in  $-2LL$  between models, a more general approach that should work better theoretically, and that also gives more consistent results across packages than do multivariate Wald tests using  $F$  or  $\chi^2$ ). A version of the last model using a probit link function instead is available in the electronic materials. (Syntax and output for SAS GLIMMIX is available in the electronic materials, which I used because it has more helpful options even though these are not mixed-effects models.)

### STATA Syntax for Importing and Preparing Data for Analysis:

```
// Defining global variable for file location to be replaced in code below
// \\Client\ precedes path in Virtual Desktop outside H drive
global filesave "C:\Dropbox\24_PSQF6270\PSQF6270_Example2ab"

// Open Example 2 STATA dataset and clear away any existing data
use "$filesave\STATA_Example2.dta", clear // Has converted all variables to lower-case

// Create ID variable
gen PersonID = _n
// Rename ordinal outcome to distinguish as 3-category version
gen apply3=apply

// Create new binary outcome JUST to demonstrate logistic regression
gen apply2=. // New empty variable
replace apply2=0 if apply==0
replace apply2=1 if apply>0

// Rename and center predictors
gen parD=pared
gen gpa3=gpa-3

// Recode ref for public to create positive slope
gen priv=.
replace priv=0 if public==1
replace priv=1 if public==0

// Label variables
label variable apply3 "apply3: 0=Not, 1=Eh, 2=Very"
label variable apply2 "apply2: 0=No, 1=Pry"
label variable parD "parD: Parent Has Graduate Degree (0=N,1=Y)"
label variable priv "priv: Student Attends Private University (0=N,1=Y)"
label variable gpa3 "gpa3: Student GPA (0=3)"

// Filter to only cases complete on all variables to be used below
egen nmiss=rowmiss(apply3 parD priv gpa3)
drop if nmiss>0
```

## R Syntax for Importing and Preparing Data for Analysis (after loading packages *readxl*, *TeachingDemos*, *psych*, *polychor*, *multcomp*, *prediction*, and *DescTools* as shown online):

```
# Define variables for working directory and data name
filesave = "C:\\Dropbox\\24_PSQF6270\\PSQF6270_Example2ab/"
filename = "Excel_Example2_Data.xlsx"
setwd(dir=filesave)

# Import Example2a and Example2b Excel data
Example2 = read_excel(path=paste0(filesave,filename))
# Convert to data frame without labels to use for analysis
Example2 = as.data.frame(Example2)

# Create ID variable
Example2$PersonID <- seq.int(nrow(Example2))

# Rename ordinal outcome to distinguish as 3-category version
Example2$apply3=Example2$apply

# Create new binary outcome JUST to demonstrate logistic regression
Example2$apply2 = NA # New empty variable
Example2$apply2[which(Example2$apply==0)]=0
Example2$apply2[which(Example2$apply>0)]=1

# Rename and center predictors
Example2$parD=Example2$pared
Example2$gpa3=Example2$gpa-3

# Recode ref for public to create positive slope
Example2$priv = NA # New empty variable
Example2$priv[which(Example2$public==1)]=0
Example2$priv[which(Example2$public==0)]=1

# Filter to only cases complete on all variables to be used below
Example2 = Example2[complete.cases(Example2[ c("apply3","parD","priv","gpa3") ]),]

# Subset data to just needed columns for Example 2a for correlations
Example2a = Example2[c("apply2","parD","priv","gpa3")]
```

---

## STATA and R Syntax and Output for Descriptive Statistics:

```
display "STATA Descriptive Statistics"
summarize apply2 parD priv gpa3
```

Variable	Obs	Mean	Std. Dev.	Min	Max
apply2	400	.45	.4981168	0	1
parD	400	.1575	.3647277	0	1
priv	400	.8575	.35	0	1
gpa3	400	-.001075	.3979409	-1.1	1

So now we know that **45% of the sample has apply2=1, and 55% have apply2=0**. This information will come in handy in making sure we know which value our logistic regression models are predicting!

```
print("R Descriptive Statistics")
describe(x=Example2a)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
apply2	1	400	0.45	0.50	0.00	0.44	0.0	0.0	1	1.0	0.20	-1.96	0.02
parD	2	400	0.16	0.36	0.00	0.07	0.0	0.0	1	1.0	1.87	1.51	0.02
priv	3	400	0.86	0.35	1.00	0.95	0.0	0.0	1	1.0	-2.04	2.16	0.02
gpa3	4	400	0.00	0.40	-0.01	0.00	0.4	-1.1	1	2.1	-0.03	-0.43	0.02

Now let's see how related our variables are using different types of correlations. First, let's examine **Pearson correlations** assuming all variables are continuous (computed using just their means, variances, and covariances):

```
pwcorr apply2 parD priv gpa3, sig // STATA: Pearson correlations
```

	apply2	parD	priv	gpa3
apply2	1.0000			
parD	0.2021	1.0000		
priv	-0.0050	-0.0790	1.0000	
gpa3	0.1301	0.1856	-0.2275	1.0000

Remember that the maximum possible Pearson correlation will be smaller than  $\pm 1$  for any two binary variables whose means for the probability of  $y=1 \neq .50$ :

$$r_{x,y} = \sqrt{\frac{p_x(1-p_y)}{p_y(1-p_x)}}$$

```
cor(x=Example2a) # R Pearson correlations
```

	apply2	parD	priv	gpa3
apply2	1.0000000000	0.202099295	-0.0050314824	0.13014887
parD	0.2020992947	1.0000000000	-0.0789743993	0.18559072
priv	-0.0050314824	-0.078974399	1.0000000000	-0.22747377
gpa3	0.1301488722	0.185590719	-0.2274737694	1.00000000

Next, let's examine **polychoric** correlations (between ordinal variables with  $\leq 10$  categories) or **polyserial** correlations (between an ordinal variable and a continuous variable with  $> 10$  categories), computed here without  $p$ -values:

```
polychoric apply2 parD priv gpa3, pw // STATA: Polychoric or Polyserial (>10 options) correlations
```

	apply2	parD	priv	gpa3
apply2	1			
parD	.37557523	1		
priv	-.00977699	-.16969222	1	
gpa3	.16472965	.27952343	-.35043179	1

```
# Recognize categorical variables as factor variables
Example2a$apply2 = as.factor(Example2a$apply2)
Example2a$parD = as.factor(Example2a$parD)
Example2a$priv = as.factor(Example2a$priv)
print("hetcor determines correlation type based on variable type")
hetcor(data=Example2a, ML=TRUE, std.err=TRUE,
use="pairwise.complete.obs")
```

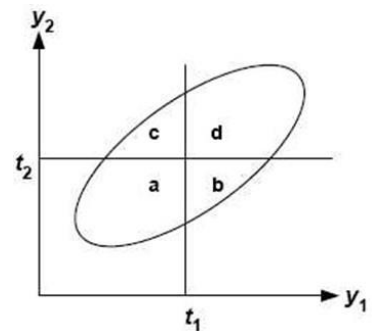
Correlations/Type of Correlation:

	apply2	parD	priv	gpa3
apply2	1	Polychoric	Polychoric	Polyserial
parD	0.37519	1	Polychoric	Polyserial
priv	-0.0098198	-0.16975	1	Polyserial
gpa3	0.16456	0.27905	-0.35099	1

Data	$y_2 = 0$	$y_2 = 1$
$y_1 = 0$	a	c
$y_1 = 1$	b	d

Most of the relations among variables are stronger when indexed by these correlations that use a bivariate normal distribution to describe what the correlation would be for their "underlying" unobserved continuous distributions:

- Tetrachoric = binary with binary (as a special case of "polychoric" here)
- Polychoric = ordinal with ordinal
- Biserial = binary with continuous (as a special case of "polyserial" here)
- Polyserial = ordinal with continuous



**STATA Syntax and Partial Output for an Empty Model—*which value is the model predicting?***

$$\text{Log} \left( \frac{\text{Apply2}_i=1}{\text{Apply2}_i=0} \right) = \text{Logit}(\text{Apply2}_i = 1) = \beta_0 \rightarrow \text{Probability}(\text{Apply2}_i = 1) = \frac{\exp(\beta_0)}{1+\exp(\beta_0)}$$

```
display "STATA Empty Model Predicting Binary Apply2"
logit apply2, nolog
```

```
Logistic regression      Number of obs      =      400
                        LR chi2(0)                    =      0.00 → LRT for MODEL
                        Prob > chi2                    =      .
Log likelihood = -275.25553 (so you need to *-2)  Pseudo R2          =      0.0000
```

apply2	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
_cons	-.2006707	.1005038	-2.00	0.046	-.3976545 - .0036869

→ Beta0 in logits!

```
display "-2LL= " e(11)*-2 // Print -2LL for model
```

-2LL= 550.51105

```
estat ic, n(400) // AIC and BIC using N=400
```

Akaike's information criterion and Bayesian information criterion

Model	Obs	ll(null)	ll(model)	df	AIC	BIC
.	400	-275.2555	-275.2555	1	552.5111	556.5025

Note: N=400 used in calculating BIC.

```
margins // Intercept in probability
```

Expression : Pr(apply2), predict()

$$\begin{aligned} &\text{Probability of } (\text{Apply2}_i = 1) \\ &= \frac{\exp(-0.2007)}{[1 + \exp(-0.2007)]} = 0.450 \end{aligned}$$

	Margin	Delta-method Std. Err.	z	P> z	[95% Conf. Interval]
_cons	.45	.0248747	18.09	0.000	.4012465 .4987535

→ Beta0 in probability!

**R Syntax and Partial Output for an Empty Model—*which value is the model predicting?***

$$\text{Log} \left( \frac{\text{Apply2}_i=1}{\text{Apply2}_i=0} \right) = \text{Logit}(\text{Apply2}_i = 1) = \beta_0 \rightarrow \text{Probability}(\text{Apply2}_i = 1) = \frac{\exp(\beta_0)}{1+\exp(\beta_0)}$$

```
print("R Empty Model Predicting Binary Apply2")
Model2Empty = glm(data=Example2, family=binomial(link="logit"), formula=apply2~1)
summary(Model2Empty);
```

```
Coefficients:
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.20067    0.10050 -1.9967  0.04586 → Beta0 in logits!
```

```
Null deviance: 550.511 on 399 degrees of freedom → Is always empty model -2LL
Residual deviance: 550.511 on 399 degrees of freedom → Is always current model -2LL
```

AIC: 552.511

```
BIC(Model2Empty) # Get BIC too
```

[1] 556.50252

```
print("Convert logits to probability to check interpretation")
Model2EmptyProb=1/(1+exp(-1*coefficients(Model2Empty))); Model2EmptyProb
```

0.45 → Beta0 in probability!

**STATA Syntax and Partial Output for a Main-Effects Model with 3 Predictors—to what extent do undergraduate GPA, parent education, and undergraduate school type uniquely predict the probability of applying to graduate school?**

$$\text{Logit}(\text{Apply2}_i = 1) = \beta_0 + \beta_1(\text{GPA}_i - 3) + \beta_2(\text{ParentGD}_i) + \beta_3(\text{Private}_i)$$

```
display "STATA Main-Effects Model Predicting Binary Apply2"
logit apply2 c.gpa3 c.parD c.priv, nolog
estimates store Main // Save for next LRT
```

This likelihood ratio test (LRT) is analogous to an F-test for the model R<sup>2</sup> in general linear models.

<b>Logistic regression</b>	Number of obs	=	400		
	LR chi2(3)	=	<b>20.59</b>	→	<b>LRT for MODEL</b>
	Prob > chi2	=	<b>0.0001</b>		
Log likelihood = <b>-264.9624 * -2 = -2LL</b>	Pseudo R2	=	0.0374	→	<b>McFadden's R2</b>

	apply2	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
	gpa3	.5482457	.2724341	2.01	0.044	.0142846 1.082207	<b>Beta1</b>
	parD	1.059612	.2973854	3.56	0.000	.4767471 1.642476	<b>Beta2</b>
	priv	.2005571	.3053354	0.66	0.511	-.3978894 .7990035	<b>Beta3</b>
	_cons	-.5387909	.287416	-1.87	0.061	-1.102116 .0245341	<b>Beta0</b>

**Interpret each fixed effect...**

**Intercept:**

**GPA3:**

**parentGD:**

**private:**

```
display "-2LL= " e(11)*-2 // Print -2LL for model
```

-2LL= 529.92481

```
estat ic, n(400) // AIC and BIC using N=400
```

Akaike's information criterion and Bayesian information criterion

Model	Obs	ll(null)	ll(model)	df	AIC	BIC
.	400	-275.2555	-264.9624	4	537.9248	553.8907

Note: N=400 used in calculating BIC.

```
fitstat // Additional R2 and fit stats (user-defined function)
```

Measures of Fit for logit of apply2

Log-Lik Intercept Only:	-275.256	Log-Lik Full Model:	-264.962	
D(396):	529.925	LR(3):	20.586	
		Prob > LR:	0.000	
<b>McFadden's R2:</b>	<b>0.037</b>	McFadden's Adj R2:	0.023	
Maximum Likelihood R2:	0.050	Cragg & Uhler's R2:	0.067	
McKelvey and Zavoina's R2:	0.063	Efron's R2:	0.051	
Variance of y*:	3.512	Variance of error:	3.290	→ pi <sup>2</sup> /3 because logit
Count R2:	0.608	Adj Count R2:	0.128	
AIC:	1.345	AIC*n:	537.925	
BIC:	-1842.695	BIC':	-2.612	

Look at how many flavors of Pseudo-R<sup>2</sup> there are! If you choose to use one, make sure to specify which one, how it's computed, and be prepared to defend why you chose that one (in case Reviewer 2 prefers a different one).

```
// For at, (from(by)to) for range of predictors
margins, at(c.gpa3=(-1(1)1) c.parD=0 c.priv=0) predict(xb) // Example yhat in logits
```

	Margin	Delta-method Std. Err.	z	P> z	[95% Conf. Interval]	
_at						
1	-1.087037	.4312041	-2.52	0.012	-1.932181	-.241892
2	-.5387909	.287416	-1.87	0.061	-1.102116	.0245341
3	.0094548	.3573788	0.03	0.979	-.6909948	.7099044

→ Diff = .5483  
→ Diff = .5483

The difference in the predicted logit outcome for each unit of GPA is  $\beta_1 = .548$ , which is a **linear** slope in predicting the **logit** outcome.

```
margins, at(c.gpa3=(-1(1)1) c.parD=0 c.priv=0) // Example yhat in probability
```

	Margin	Delta-method Std. Err.	z	P> z	[95% Conf. Interval]	
_at						
1	.2521767	.081318	3.10	0.002	.0927963	.4115571
2	.3684689	.0668816	5.51	0.000	.2373834	.4995544
3	.5023637	.0893427	5.62	0.000	.3272552	.6774722

→ Diff = .1163  
→ Diff = .1339

If I convert the logit slope = .548 into a probability, I get .389, but that is NOT the expected change in probability per unit GPA!

The difference in the predicted probability outcome for each unit of GPA is NOT constant. This is why you cannot **“unlogit” a slope to compute a slope for change in probability**—it doesn’t make sense! The effect of a predictor on probability depends on depends where you are on the probability scale (biggest impact is near probability = .50).

```
// Must re-estimate with 'or' added to first line to get odds ratios
display "STATA Main-Effects Model Predicting Binary Apply2"
display "Get Odds Ratios Instead of Logit Fixed Effects"
logit apply2 c.gpa3 c.parD c.priv, or nolog
```

apply2	Odds Ratio	Std. Err.	z	P> z	[95% Conf. Interval]	
gpa3	1.730215	.4713696	2.01	0.044	1.014387	2.951185
parD	2.885251	.8580314	3.56	0.000	1.610826	5.167952
priv	1.222083	.3731454	0.66	0.511	.6717363	2.223324
_cons	.5834533	.1676938	-1.87	0.061	.3321675	1.024838

exp(Beta1)  
exp(Beta2)  
exp(Beta3)  
exp(Beta0)

**R Syntax and Partial Output for a Main-Effects Model with 3 Predictors—to what extent do undergraduate GPA, parent education, and undergraduate school type uniquely predict the probability of applying to graduate school?**

$$\text{Logit}(\text{Apply2}_i = 1) = \beta_0 + \beta_1(\text{GPA}_i - 3) + \beta_2(\text{ParentGD}_i) + \beta_3(\text{Private}_i)$$

```
print("R Main-Effects Model Predicting Binary Apply2")
Model2Main = glm(data=Example2, family=binomial(link="logit"),
  formula=apply2~1+gpa3+parD+priv); summary(Model2Main);
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )	
(Intercept)	-0.53879	0.28742	-1.8746	0.0608473	Beta0
gpa3	0.54825	0.27243	2.0124	0.0441780	Beta1
parD	1.05961	0.29739	3.5631	0.0003665	Beta2
priv	0.20056	0.30534	0.6568	0.5112826	Beta3

Null deviance: 550.511 on 399 degrees of freedom → Is always empty model -2LL  
Residual deviance: 529.925 on 396 degrees of freedom → Is always current model -2LL  
AIC: 537.925

```
BIC(Model2Main) # Get BIC too
[1] 553.89066
```

```
confint.default(Model2Main) # Get 95% CIs for logit parameters
                2.5 %      97.5 %
(Intercept) -1.102115790 0.024534011
gpa3         0.014284691 1.082206667
parD        0.476747275 1.642476234
priv        -0.397889291 0.799003465
```

```
print("Likelihood Ratio Test of Predictors (analogous to F-test for model R2 in general LM)")
anova(Model2Empty, Model2Main, test="LRT") # Nested "fewer" model goes first
```

```
Analysis of Deviance Table
Model 1: apply2 ~ 1
Model 2: apply2 ~ 1 + gpa3 + parD + priv

  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         399     550.511          20.5862 0.0001283
2         396     529.925          3
```

“Deviance” is the difference in  $-2LL$  between nested models, which is treated as a  $\chi^2$  test statistic with degrees of freedom equal to the number of new parameters (3 fixed slopes here).

It’s written like this:  $-2\Delta LL(3) = 20.59, p < .001$   
 or like this:  $\chi^2(3) = 20.59, p < .001$

```
print("Pseudo-R2 values"); PseudoR2(x=Model2Main, which="all")
```

	McFadden	McFaddenAdj	CoxSnell	Nagelkerke	AldrichNelson
	<b>0.037394791</b>	0.022862839	0.050163690	0.067110121	0.048946550
VeallZimmermann		Efron	McKelveyZavoina	Tjur	AIC
	0.084510995	0.050966133	0.063047115	0.050877331	537.924805234
	BIC	logLik	logLik0	G2	
	553.890663422	-264.962402617	-275.255525485	20.586245737	

Look at how many flavors of Pseudo-R<sup>2</sup> there are! If you choose to use one, make sure to specify which one, how it’s computed, and be prepared to defend why you chose that one (in case Reviewer 2 prefers a different one).

```
print("Odds ratios with 95% CIs using standard errors")
OddsRatio(x=Model2Main, conf.level=.95, digits=5, use.profile=FALSE)
```

```
Odds Ratios:
              or  or.lci  or.uci  Pr(>|z|)
(Intercept) 0.58345 0.33217 1.02484 0.0608 exp(Beta0)
gpa3         1.73022 1.01439 2.95118 0.0442 exp(Beta1)
parD         2.88525 1.61083 5.16795 3.67e-04 exp(Beta2)
priv         1.22208 0.67174 2.22332 0.5113 exp(Beta3)
```

# If the OddsRatio function doesn't work, use this instead  
`exp(cbind(OR=coef(Model2Main), confint(Model2Main)))`

```
print("Example yhat in logits for specific values of predictors")
Main2Logits = prediction(model=Model2Main, type="link",
                        at=list(gpa3=-1:1, parD=0, priv=0));
summary(Main2Logits)
```

at(gpa3)	at(parD)	at(priv)	Prediction	SE	z	p	lower	upper
-1	0	0	-1.087037	0.4312	-2.52093	0.0117044	-1.9322	-0.24189
0	0	0	-0.538791	0.2874	-1.87460	0.0608473	-1.1021	0.02453 → Diff = .5483
1	0	0	0.009455	0.3574	0.02646	0.9788937	-0.6910	0.70990 → Diff = .5483

The difference in the predicted logit outcome for each unit of GPA is  $\beta_1 = .548$ , which is a linear slope in predicting the logit outcome.

```
print("Example yhat in probability for specific values of predictors")
Main2Probs = prediction(model=Model2Main, type="response",
                       at=list(gpa3=-1:1, parD=0, priv=0));
summary(Main2Probs)
```

at(gpa3)	at(parD)	at(priv)	Prediction	SE	z	p	lower	upper
-1	0	0	0.2522	0.08132	3.101	1.928e-03	0.0928	0.4116
0	0	0	0.3685	0.06688	5.509	3.603e-08	0.2374	0.4996 → Diff = .1163
1	0	0	0.5024	0.08934	5.623	1.878e-08	0.3273	0.6775 → Diff = .1339

If I convert the logit slope = .548 into a probability, I get .389, but that is NOT the expected change in probability per unit GPA!

The difference in the predicted probability outcome for each unit of GPA is NOT constant. This is why you cannot “unlogit” a slope to compute a slope for change in probability—it doesn’t make sense! The effect of a predictor on probability depends on where you are on the probability scale (bigger impact near probability = .50).

**STATA Syntax and Partial Output adding 2 New Interactions—to what extent does the effect of undergraduate GPA differ by parent education and undergraduate school type?**

$$\text{Logit}(\text{Apply2}_i = 1) = \beta_0 + \beta_1(\text{GPA}_i - 3) + \beta_2(\text{ParentGD}_i) + \beta_3(\text{Private}_i) + \beta_4(\text{GPA}_i - 3)(\text{ParentGD}_i) + \beta_5(\text{GPA}_i - 3)(\text{Private}_i)$$

Model-implied GPA Slope:  $\beta_1 + \beta_4(\text{ParentGD}_i) + \beta_5(\text{Private}_i)$

This likelihood ratio test (LRT) is analogous to an F-test for the model  $R^2$  in general linear models. To assess the contribution of new predictors (analogous to an F-test for the change in  $R^2$ ) we must do our own LRT (see below).

```
display "STATA Interaction Model Predicting Binary Apply2"
logit apply2 c.gpa3 c.parD c.priv c.gpa3#c.parD c.gpa3#c.priv, nolog
estimates store Interact // Save for this LRT
display "-2LL= " e(11)*-2 // Print -2LL for model
estat ic, n(400) // AIC and BIC using N=400
```

Logistic regression	Number of obs	=	
	LR chi2(5)	=	<b>22.34</b> → Is LRT for MODEL
	Prob > chi2	=	<b>0.0005</b>
Log likelihood = -264.08684 * -2 = -2LL	Pseudo R2	=	0.0406

apply2	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
gpa3	1.25645	.7736689	1.62	0.104	-.2599127 2.772814	<b>Beta1</b>
parD	1.162334	.3196688	3.64	0.000	.5357943 1.788873	<b>Beta2</b>
priv	.3197742	.351815	0.91	0.363	-.3697705 1.009319	<b>Beta3</b>
c.gpa3#c.parD	-.8358888	.7695577	-1.09	0.277	-2.344194 .6724166	<b>Beta4</b>
c.gpa3#c.priv	-.6821162	.8077466	-0.84	0.398	-2.265271 .9010381	<b>Beta5</b>
_cons	-.6594466	.3373972	-1.95	0.051	-1.320733 .0018397	<b>Beta0</b>

```
lrtest Interact Main // LRT for two new interactions ("fewer" model goes LAST)
```

Likelihood-ratio test	LR chi2(2)	=	<b>1.75</b>
(Assumption: Main nested in Interact)	Prob > chi2	=	<b>0.4166</b>

The two new interaction slopes do not significantly improve the model prediction,  $-2\Delta LL(2) = 1.75, p = .417$ .

```
// For at, (from(by)to) for range of predictors
margins, at(c.gpa3=(-1(1)1) c.parD=(0(1)1) c.priv=(0(1)1)) predict(xb) // Yhat in logits
margins, at(c.gpa3=(-1(1)1) c.parD=(0(1)1) c.priv=(0(1)1)) // Yhat in probability
lincom c.gpa3*1 + c.gpa3#c.parD*0 + c.gpa3#c.priv*0 // GPA slope: Ndeg Pub = B1
lincom c.gpa3*1 + c.gpa3#c.parD*0 + c.gpa3#c.priv*1, or // GPA slope: Ndeg Pri = B1+B5
lincom c.gpa3*1 + c.gpa3#c.parD*1 + c.gpa3#c.priv*0 // GPA slope: Ydeg Pub = B1+B4
lincom c.gpa3*1 + c.gpa3#c.parD*1 + c.gpa3#c.priv*1 // GPA slope: Ydeg Pri = B1+B4+B5
```

The rest of the (very long) STATA output for this model is available electronically...

```
// Must re-estimate with 'or' added to first line to get odds ratios
display "STATA Interaction Model Predicting Binary Apply2"
display "Get Odds Ratios Instead of Logit Fixed Effects"
logit apply2 c.gpa3 c.parD c.priv c.gpa3#c.parD c.gpa3#c.priv, or
lincom c.gpa3*1 + c.gpa3#c.parD*0 + c.gpa3#c.priv*0, or // GPA slope: Ndeg Pub = B1
lincom c.gpa3*1 + c.gpa3#c.parD*0 + c.gpa3#c.priv*1, or // GPA slope: Ndeg Pri = B1+B5
lincom c.gpa3*1 + c.gpa3#c.parD*1 + c.gpa3#c.priv*0, or // GPA slope: Ydeg Pub = B1+B4
lincom c.gpa3*1 + c.gpa3#c.parD*1 + c.gpa3#c.priv*1, or // GPA slope: Ydeg Pri = B1+B4+B5
```

apply2	Odds Ratio	Std. Err.	z	P> z	[95% Conf. Interval]	
gpa3	3.51293	2.717845	1.62	0.104	.7711189 16.0036	<b>exp(Beta1)</b>
parD	3.197386	1.022105	3.64	0.000	1.708805 5.982706	<b>exp(Beta2)</b>
priv	1.376817	.4843848	0.91	0.363	.6908928 2.743732	<b>exp(Beta3)</b>
c.gpa3#c.parD	.433489	.3335948	-1.09	0.277	.0959245 1.958966	<b>exp(Beta4)</b>
c.gpa3#c.priv	.505546	.4083531	-0.84	0.398	.1038019 2.462158	<b>exp(Beta5)</b>
_cons	.5171374	.1744807	-1.95	0.051	.2669396 1.001841	<b>exp(Beta0)</b>



**R Syntax and Partial Output adding 2 New Interactions—to what extent does the effect of undergraduate GPA differ by parent education and undergraduate school type?**

$$\text{Logit}(\text{Apply2}_i = 1) = \beta_0 + \beta_1(\text{GPA}_i - 3) + \beta_2(\text{ParentGD}_i) + \beta_3(\text{Private}_i) \\ + \beta_4(\text{GPA}_i - 3)(\text{ParentGD}_i) + \beta_5(\text{GPA}_i - 3)(\text{Private}_i)$$

```
print("R Interaction Model Predicting Binary Apply2")
Model2Int = glm(data=Example2, family=binomial(link="logit"),
               formula=apply2~1+gpa3+parD+priv+gpa3:parD+gpa3:priv)
summary(Model2Int); BIC(Model2Int) # Get BIC too
confint.default(Model2Int) # Get 95% CIs for logit parameters (not shown here)
```

```
Coefficients:
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.65945    0.33740 -1.9545 0.0506406 Beta0
gpa3         1.25645    0.77367  1.6240 0.1043722 Beta1
parD        1.16233    0.31967  3.6361 0.0002768 Beta2
priv        0.31977    0.35181  0.9089 0.3633882 Beta3
gpa3:parD   -0.83589    0.76956 -1.0862 0.2773930 Beta4
gpa3:priv   -0.68212    0.80775 -0.8445 0.3984075 Beta5

Null deviance: 550.511 on 399 degrees of freedom → Is empty model -2LL
Residual deviance: 528.174 on 394 degrees of freedom → Is current model -2LL

AIC: 540.174
BIC: 564.12247
```

**Interpret each simple effect and interaction...**

**GPA3:**

**parentGD:**

**private:**

**GPA3\*parentGD:**

**GPA3\*private:**

Model-implied GPA Slope:  $\beta_1 + \beta_4(\text{ParentGD}_i) + \beta_5(\text{Private}_i)$

```
print("Simple slopes for GPA by moderators")
Int2Slopes = (summary(glht(model=Model2Int, linfct=rbind(
  "GPA Slope: Ndeg Pub = B1      " = c(0,1,0,0,0,0), # in order of fixed effects
  "GPA Slope: Ndeg Pri = B1+B5   " = c(0,1,0,0,0,1),
  "GPA Slope: Ydeg Pub = B1+B4   " = c(0,1,0,0,1,0),
  "GPA Slope: Ydeg Pri = B1+B4+B5" = c(0,1,0,0,1,1))), test=adjusted("none"))
Int2Slopes
```

```
Linear Hypotheses:
              Estimate Std. Error z value Pr(>|z|)
GPA Slope: Ndeg Pub = B1      == 0  1.25645    0.77367  1.6240 0.10437
GPA Slope: Ndeg Pri = B1+B5   == 0  0.57433    0.30900  1.8587 0.06307
GPA Slope: Ydeg Pub = B1+B4   == 0  0.42056    0.94762  0.4438 0.65718
GPA Slope: Ydeg Pri = B1+B4+B5 == 0 -0.26155    0.73545 -0.3556 0.72211
(Adjusted p values reported -- none method)
```

```
print("Likelihood Ratio Test of New Interactions (analogous to F-test for R2 change in general LM")
anova(Model2Main, Model2Int, test="LRT") # Nested "fewer" model goes first
```

```
Analysis of Deviance Table
Model 1: apply2 ~ 1 + gpa3 + parD + priv
Model 2: apply2 ~ 1 + gpa3 + parD + priv + gpa3:parD + gpa3:priv
```

	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
1	396	529.925			
2	394	528.174	2	1.75113	0.41663

The 2 new interaction slopes do not significantly improve the model prediction,  $-2\Delta LL(2) = 1.75, p = .417$ .

```
print("Odds ratios with 95% CIs using standard errors")
OddsRatio(x=Model2Int, conf.level=.95, digits=5, use.profile=FALSE)
```

```
Odds Ratios:
```

	or	or.lci	or.uci	Pr(> z )	
(Intercept)	0.51714	0.26694	1.00184	0.0506	exp(Beta0)
gpa3	3.51293	0.77112	16.00359	0.1044	exp(Beta1)
parD	3.19739	1.70881	5.98270	2.77e-04	exp(Beta2)
priv	1.37682	0.69089	2.74373	0.3634	exp(Beta3)
gpa3:parD	0.43349	0.09592	1.95896	0.2774	exp(Beta4)
gpa3:priv	0.50555	0.10380	2.46216	0.3984	exp(Beta5)

# If the OddsRatio function doesn't work, use this instead  
`exp(cbind(OR=coef(Model2Int), confint(Model2Int)))`

```
print("Odds ratios for simple slopes using GLHT output")
data.frame(OR=exp(Int2Slopes$test$coefficients))
```

	OR
GPA Slope: Ndeg Pub = B1	3.51293093
GPA Slope: Ndeg Pri = B1+B5	1.77594769
GPA Slope: Ydeg Pub = B1+B4	1.52281717
GPA Slope: Ydeg Pri = B1+B4+B5	0.76985392

```
print("Yhat in logits for specific values of predictors")
Int2Logits = prediction(model=Model2Int, type="link",
                        at=list(gpa3=-1:1, parD=0:1, priv=0:1))
summary(Int2Logits)
```

at (gpa3)	at (parD)	at (priv)	Prediction	SE	z	p	lower	upper
-1	0	0	-1.91590	0.9908	-1.93362	0.053160	-3.8579	0.026104
0	0	0	-0.65945	0.3374	-1.95451	0.050641	-1.3207	0.001839
1	0	0	0.59700	0.6656	0.89692	0.369762	-0.7076	1.901588
<hr/>								
-1	1	0	0.08233	1.2300	0.06693	0.946638	-2.3285	2.493135
0	1	0	0.50289	0.4289	1.17249	0.241000	-0.3378	1.343526
1	1	0	0.92345	0.8068	1.14459	0.252377	-0.6578	2.504730
<hr/>								
-1	0	1	-0.91401	0.3194	-2.86135	0.004218	-1.5401	-0.287932
0	0	1	-0.33967	0.1187	-2.86067	0.004227	-0.5724	-0.106949
1	0	1	0.23466	0.3422	0.68567	0.492921	-0.4361	0.905434
<hr/>								
-1	1	1	1.08422	0.8967	1.20917	0.226598	-0.6732	2.841641
0	1	1	0.82266	0.3034	2.71185	0.006691	0.2281	1.417232
1	1	1	0.56111	0.6796	0.82566	0.408995	-0.7709	1.893064

```
print("Yhat in probability for specific values of predictors")
Int2Probs = prediction(model=Model2Int, type="response",
                       at=list(gpa3=-1:1, parD=0:1, priv=0:1))
summary(Int2Probs)
```

For comparison, the last column shows the predicted probabilities using a probit link instead of logit:

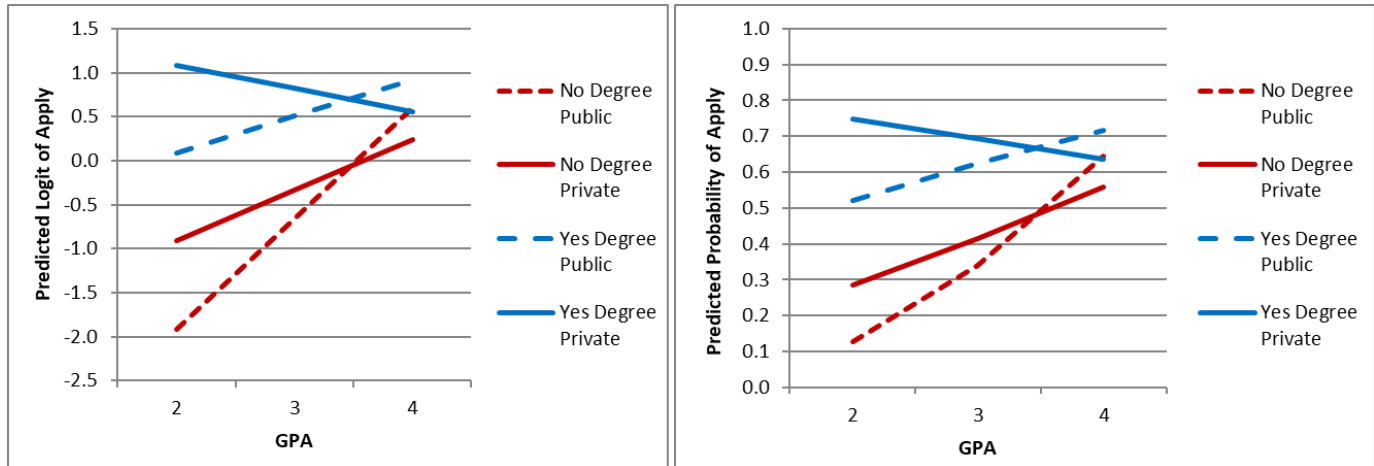
at (gpa3)	at (parD)	at (priv)	Prediction	SE	z	p	lower	upper	Prediction
-1	0	0	0.1283	0.11083	1.158	2.469e-01	-0.08890	0.3455	0.1250
0	0	0	0.3409	0.07580	4.497	6.905e-06	0.19229	0.4894	0.3446
1	0	0	0.6450	0.15242	4.232	2.320e-05	0.34624	0.9437	0.6370
<hr/>									
-1	1	0	0.5206	0.30699	1.696	8.993e-02	-0.08111	1.1223	0.5338
0	1	0	0.6231	0.10072	6.187	6.145e-10	0.42572	0.8206	0.6256
1	1	0	0.7157	0.16415	4.360	1.298e-05	0.39403	1.0375	0.7108
<hr/>									
-1	0	1	0.2862	0.06525	4.386	1.156e-05	0.15829	0.4141	0.2832
0	0	1	0.4159	0.02884	14.418	3.972e-47	0.35935	0.4724	0.4162
1	0	1	0.5584	0.08439	6.617	3.673e-11	0.39299	0.7238	0.5598
<hr/>									
-1	1	1	0.7473	0.16933	4.413	1.019e-05	0.41541	1.0792	0.7459
0	1	1	0.6948	0.06433	10.801	3.407e-27	0.56872	0.8209	0.6945
1	1	1	0.6367	0.15719	4.050	5.112e-05	0.32861	0.9448	0.6390

## Illustrating these simple (conditional) slopes for GPA at each combination of degree and school type:

Linear Hypotheses:

	Estimate	Std. Error	z value	Pr(> z )
GPA Slope: Ndeg Pub = B1	== 0 1.25645	0.77367	1.6240	0.10437
GPA Slope: Ndeg Pri = B1+B5	== 0 0.57433	0.30900	1.8587	0.06307
GPA Slope: Ydeg Pub = B1+B4	== 0 0.42056	0.94762	0.4438	0.65718
GPA Slope: Ydeg Pri = B1+B4+B5	== 0 -0.26155	0.73545	-0.3556	0.72211

(Adjusted p values reported -- none method)



The **model provides direct tests of the differences in logits** amongst the degree and school conditions, as well as for the simple slopes of GPA for each degree and school type. Model-predicted logit outcomes can then be converted through an inverse link (the “un-logit” back-transformation) into predicted probabilities for ease of interpretation, but the **slopes or mean differences themselves cannot be converted in differences in probabilities, only odds ratios.**

### Sample results section (should also report what software and version you used):

We examined the extent to which a binary decision to apply to graduate school (55% 0=No, 45% 1=Pry) could be predicted by a student’s undergraduate GPA ( $M = 3.00$ ,  $SD = 0.40$ , range = 1.90 to 4.00), whether at least one of their parents has a graduate degree (15.75% 0=No, 84.25% 1=Yes), and whether they attended a private university (14.25% 0=No, 85.75% 1=Yes). Specifically, we estimated generalized linear models using maximum likelihood, in which the conditional probability of applying to graduate school was predicted using a logit link function and a conditional Bernoulli distribution (i.e., logistic regressions). The GPA predictor was centered such that 0 indicated a GPA = 3. Effect sizes are provided using odds ratios (OR), in which OR values between 0 and 1 indicate negative effects, 1 indicates no effect, and values above 1 indicate positive effects. Nested model comparisons were conducted using likelihood ratio tests (i.e., the difference in  $-2LL$  between nested models with degrees of freedom equal to the number of new parameters).

The first model examined only the main effects of the three predictors, which together resulted in a significant prediction of the logit of the probability of applying to graduate school,  $-2\Delta LL(3) = 20.59$ ,  $p < .001$ . GPA had a significantly positive effect, such that for every unit higher GPA, the logit of applying to graduate school was greater by 0.548 ( $SE = 0.272$ ;  $OR = 1.730$ ). Likewise, the logit of applying to graduate school was significantly greater for students for whom at least one parent had a graduate degree by 1.060 ( $SE = 0.297$ ,  $OR = 2.882$ ). However, the logit of applying to graduate school was nonsignificantly greater for students who attended a private university by 0.201 ( $SE = 0.305$ ,  $OR = 1.222$ ).

The second model then included two-way interactions of GPA with parent graduate degree and GPA with university type. This augmented model was not a significant improvement over the main effects model,  $-2\Delta LL(2) = 1.75$ ,  $p = .417$ . Neither individual interaction term was significant, nor was the simple slope of GPA significant in any of the four subgroups (i.e., formed by parent graduate degree by university type).