

DATA CONSIDERATIONS

Dr. Aric LaBarr

Institute for Advanced Analytics

RARE EVENT MODELING

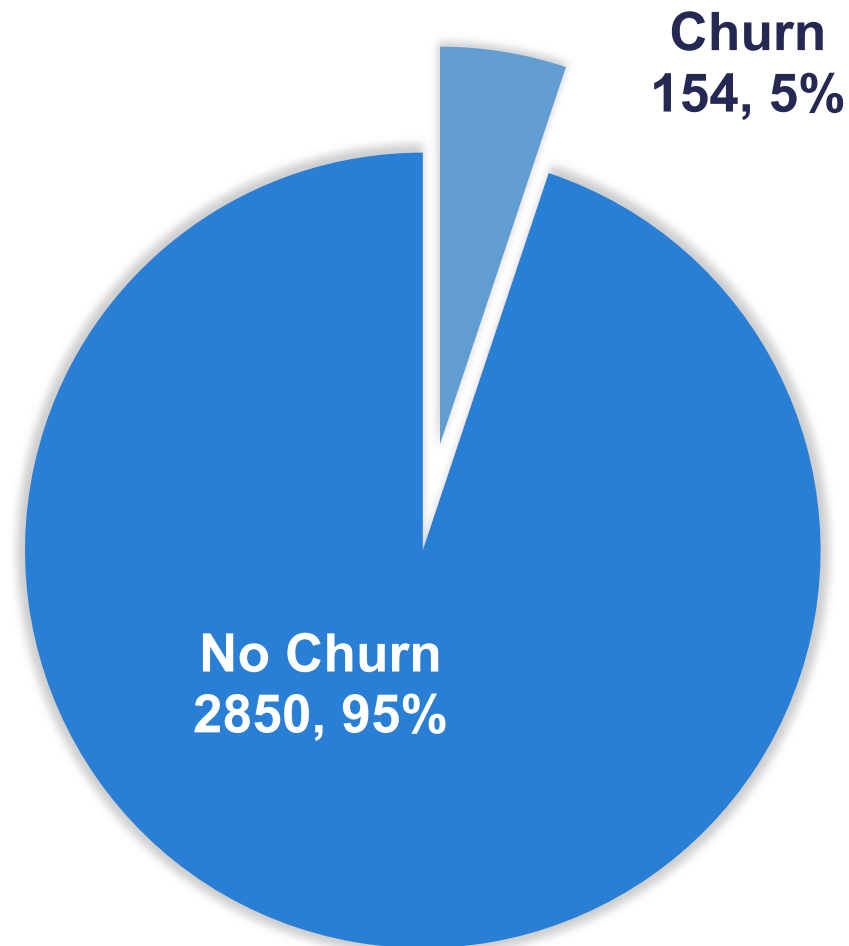
Rare Event Modeling

- 5% or smaller in a category can lead to classification problems.
- Common Situations:
 - Fraud
 - Default
 - Marketing Response
 - Weather Event



Telecomm Churn Data Set

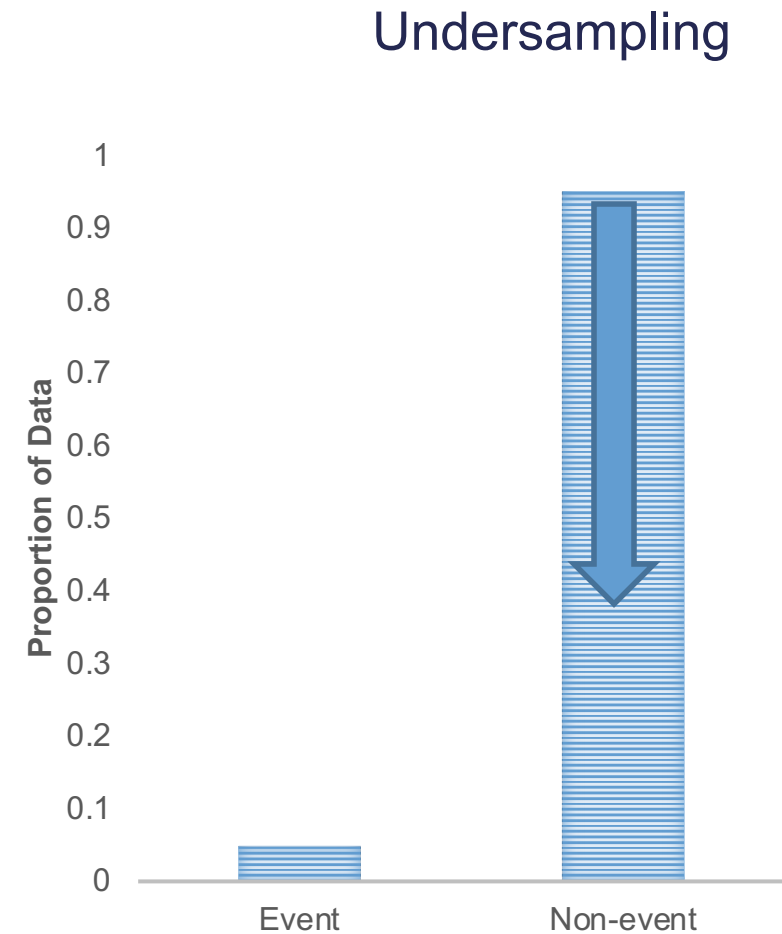
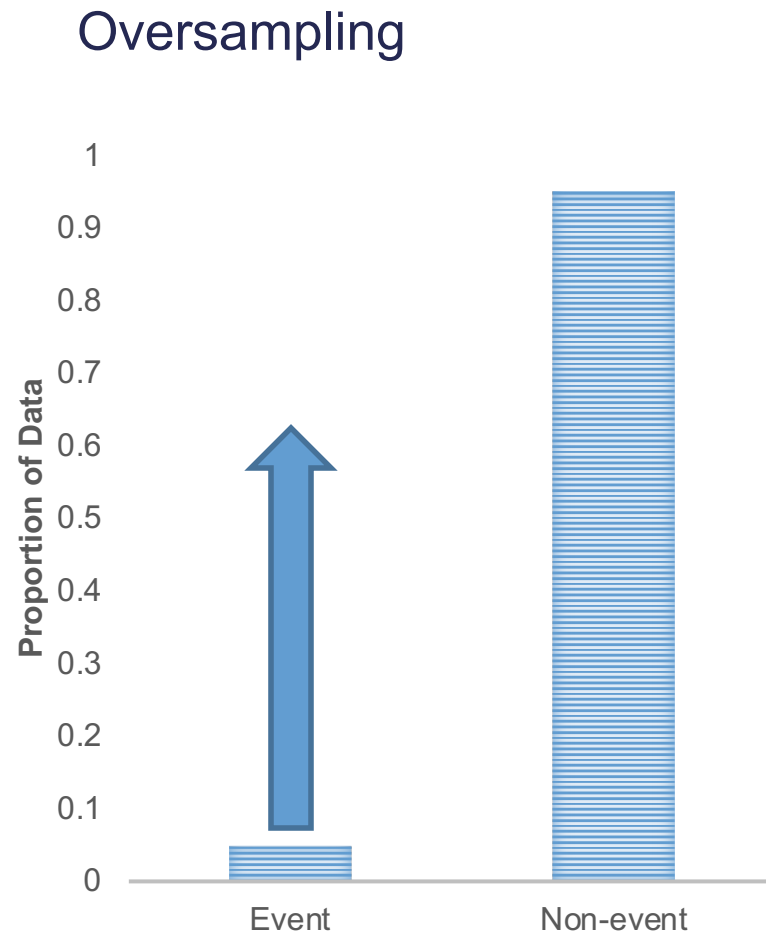
- Model the association between various factors and a customer churning (leaving the company)
- 3004 observations in the data set



Telecomm Churn Data Set

- Model the association between various factors and a customer churning (leaving the company)
- Predictors:
 - **account_length**: length of time with company
 - **international_plan**: yes, no
 - **voice_mail_plan**: yes, no
 - **customer_service_calls**: number of service calls
 - **total_day_minutes**: minutes used during daytime
 - **total_day_calls**: calls used during daytime
 - **total_day_charge**: cost of usage during daytime
 - Same as previous three for evening, night, international

Rare Event Sampling Correction



Rare Event Sampling Correction

Oversampling

- Duplicate current event cases in training set to balance better with non-event cases.
- Keep test set as original population proportion.

Undersampling

- Randomly sample current non-event cases to keep in the training set to balance with event cases.
- Keep test set as original population proportion.

Oversampling

```
set.seed(12345)
train_o <- churn %>%
  sample_frac(0.70)

train_o_T <- train_o %>%
  filter(churn == TRUE) %>%
  slice(rep(1:n(), each = 10))

train_o_F <- train_o %>%
  filter(churn == FALSE)

train_o <- rbind(train_o_F, train_o_T)

test_o <- churn[-train_o$id,]
```

```
table(train_o$churn)
```

FALSE	TRUE
1996	1070

```
table(test_o$churn)
```

FALSE	TRUE
854	47

Undersampling

```
set.seed(12345)
```

```
train_u <- churn %>%  
  group_by(churn) %>%  
  sample_n(104)
```

```
test_u <- churn[-train_u$id,]
```

```
table(train_u$churn)
```

FALSE	TRUE
104	104

```
table(test_u$churn)
```

FALSE	TRUE
2746	50

Telecomm Model

```
logit.model <- glm(churn ~ factor(international.plan) +  
                    factor(voice.mail.plan) +  
                    total.day.charge +  
                    customer.service.calls,  
                    data = train_u, family = binomial(link = "logit"))  
  
summary(logit.model)
```

Telecomm Model

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.19331	-0.74911	-0.01389	0.73301	2.51757

Coefficients:

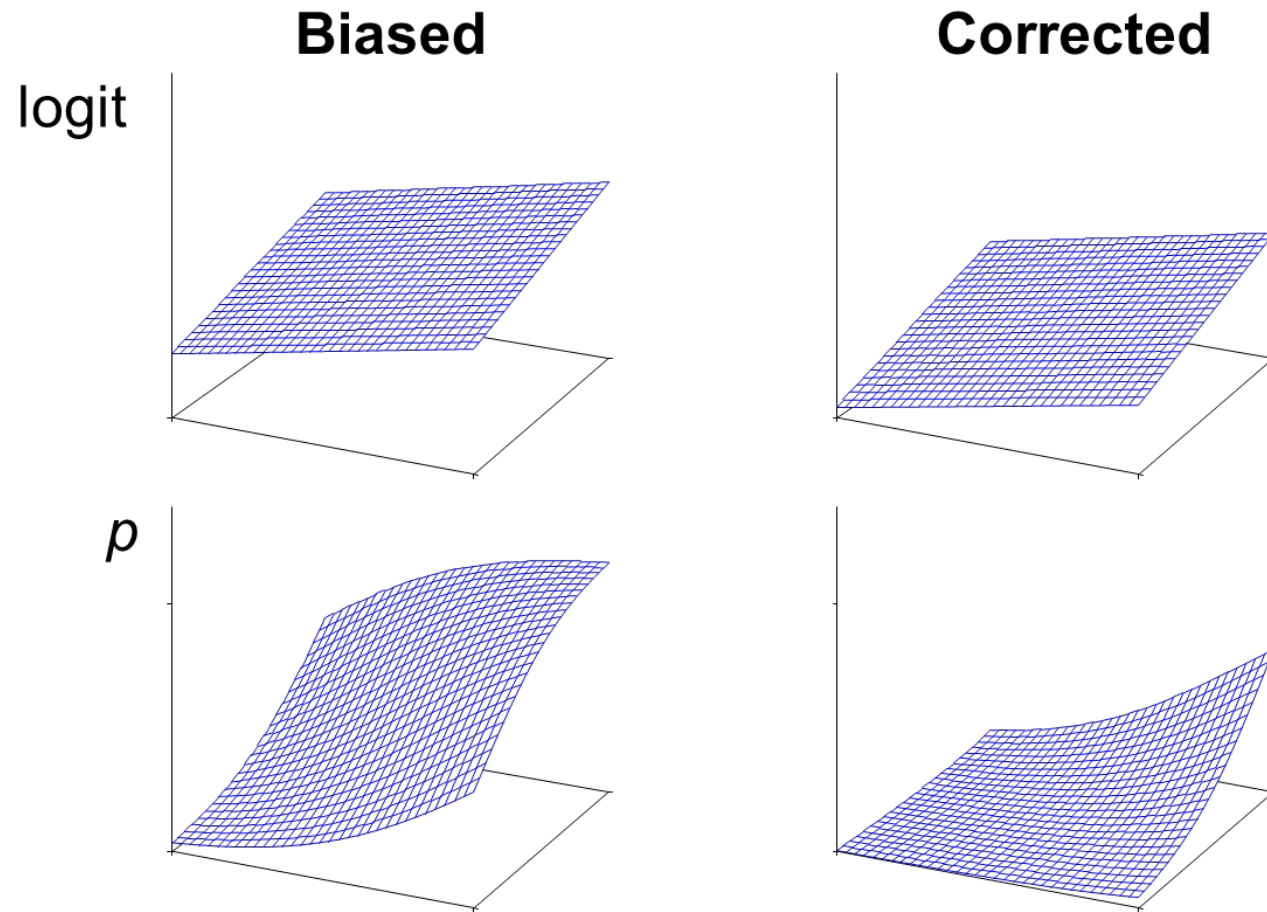
	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-5.81880	0.95939	-6.065	1.32e-09	***
factor(international.plan)yes	2.97995	0.57057	5.223	1.76e-07	***
factor(voice.mail.plan)yes	-0.85107	0.41372	-2.057	0.0397	*
total.day.charge	0.12898	0.02234	5.773	7.79e-09	***
customer.service.calls	0.78520	0.14947	5.253	1.50e-07	***

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 288.35 on 207 degrees of freedom
 Residual deviance: 195.24 on 203 degrees of freedom
 AIC: 205.24

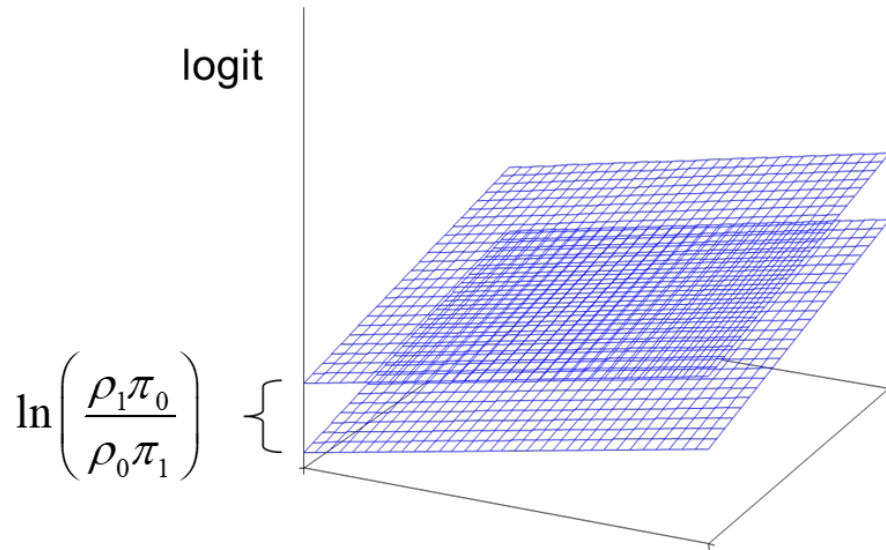
Effect of Oversampling



Adjustments to Oversampling

- When the sample proportion is out of line with the population proportion, adjustments need to be made to correct the bias.
- 2 Methods:
 1. Adjusting the intercept
 2. Weighting observations

Adjusting the Intercept



- Population proportion: π_1, π_0
- Sample proportion: ρ_1, ρ_0
- Unadjusted predictions: \hat{p}_i^*

- Need to correct for the bias created by oversampling.
- Adjustment is only applied to intercept.
- This adjusts the predicted values:

$$\hat{p}_i = \frac{\hat{p}_i^* \rho_0 \pi_1}{(1 - \hat{p}_i^*) \rho_1 \pi_0 + \hat{p}_i^* \rho_0 \pi_1}$$

Adjusting the Intercept

```
test_u_p_bias <- predict(logit.model, newdata = test_u, type = "response")
test_u_p <- (test_u_p_bias*(104/208)*(154/3004)) /
            ((1-test_u_p_bias)*(104/208)*(2850/3004) +
             test_u_p_bias*(104/208)*(154/3004))
```

```
test_u <- data.frame(test_u, 'Pred' = test_u_p)
```

```
head(test_u_p)
```

1	2	3	4	5	6
0.04788873	0.00516951	0.03230002	0.91516214	0.56312205	0.29766875

Weighting Observations

- Instead of adjusting the model after it is built, weighting observations adjusts while the model is being built.
- Uses **weighted MLE** instead – each observation has potentially different weight to the MLE calculation.
- Need to create a weight variable in the oversampled data set:

$$weight = \begin{cases} 1, & y = 1 \\ \rho_1 \pi_0 / \rho_0 \pi_1, & y = 0 \end{cases}$$


Bigger than 1!



Weighting Observations

- Instead of adjusting the model after it is built, weighting observations adjusts while the model is being built.
- Uses **weighted MLE** instead – each observation has potentially different weight to the MLE calculation.
- Need to create a weight variable in the oversampled data set:

$$weight = \begin{cases} 1, & y = 1 \\ \rho_1\pi_0/\rho_0\pi_1, & y = 0 \end{cases}$$



Need to overweight the 0's,
since their effect was reduced
in the sampling!

Weighted Observations

```
train_u$weight <- ifelse(train_u$churn == 'TRUE', 1, 18.49)

logit.model.w <- glm(churn ~ factor(international.plan) +
                     factor(voice.mail.plan) +
                     total.day.charge +
                     customer.service.calls,
                     data = train_u, family = binomial(link = "logit"),
                     weights = weight)

summary(logit.model.w)
```

Weighted Observations

Deviance Residuals:

Min	1Q	Median	3Q	Max
-4.5172	-0.7988	0.0717	1.8224	3.7129

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-9.76831	0.69648	-14.025	< 2e-16	***
factor(international.plan)yes	3.33560	0.32429	10.286	< 2e-16	***
factor(voice.mail.plan)yes	-1.07451	0.27107	-3.964	7.37e-05	***
total.day.charge	0.16320	0.01647	9.911	< 2e-16	***
customer.service.calls	0.72693	0.08810	8.251	< 2e-16	***

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 820.31 on 207 degrees of freedom
 Residual deviance: 585.33 on 203 degrees of freedom
 AIC: 590.92

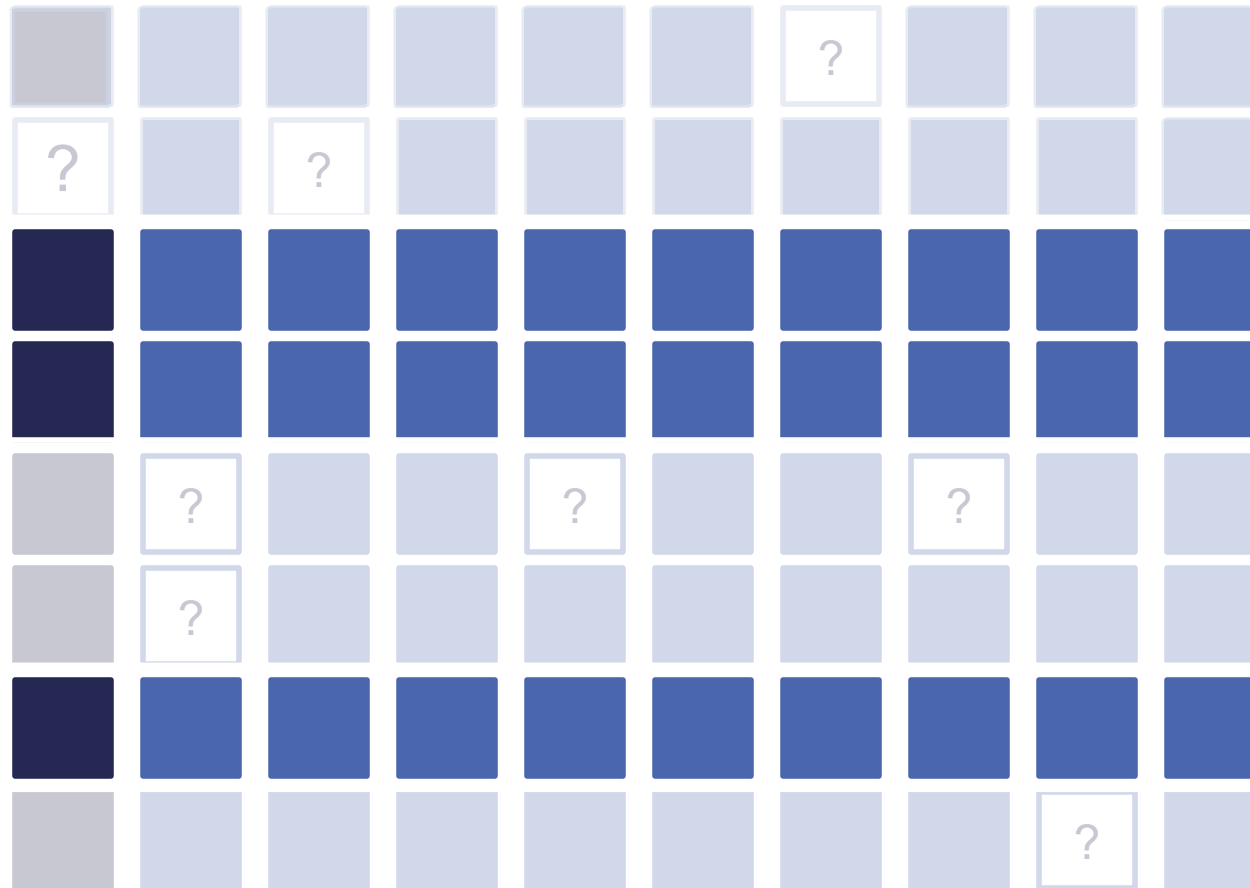
When to Use Which Technique?

	Model Correct	Model Misspecified
Small Sample ($n \leq 1000$)	Adjust Intercept	Weighted Observations
Large Sample ($n > 1000$)	Either	Weighted Observations



MISSING VALUES

Complete Case Analysis



Handling Missing Values

- Complete cases analysis isn't necessarily bad if you have enough observations.
- However, how to handle scoring new observations with missing values?
- Solutions to missing values:
 - Delete
 - Keep
 - Replace

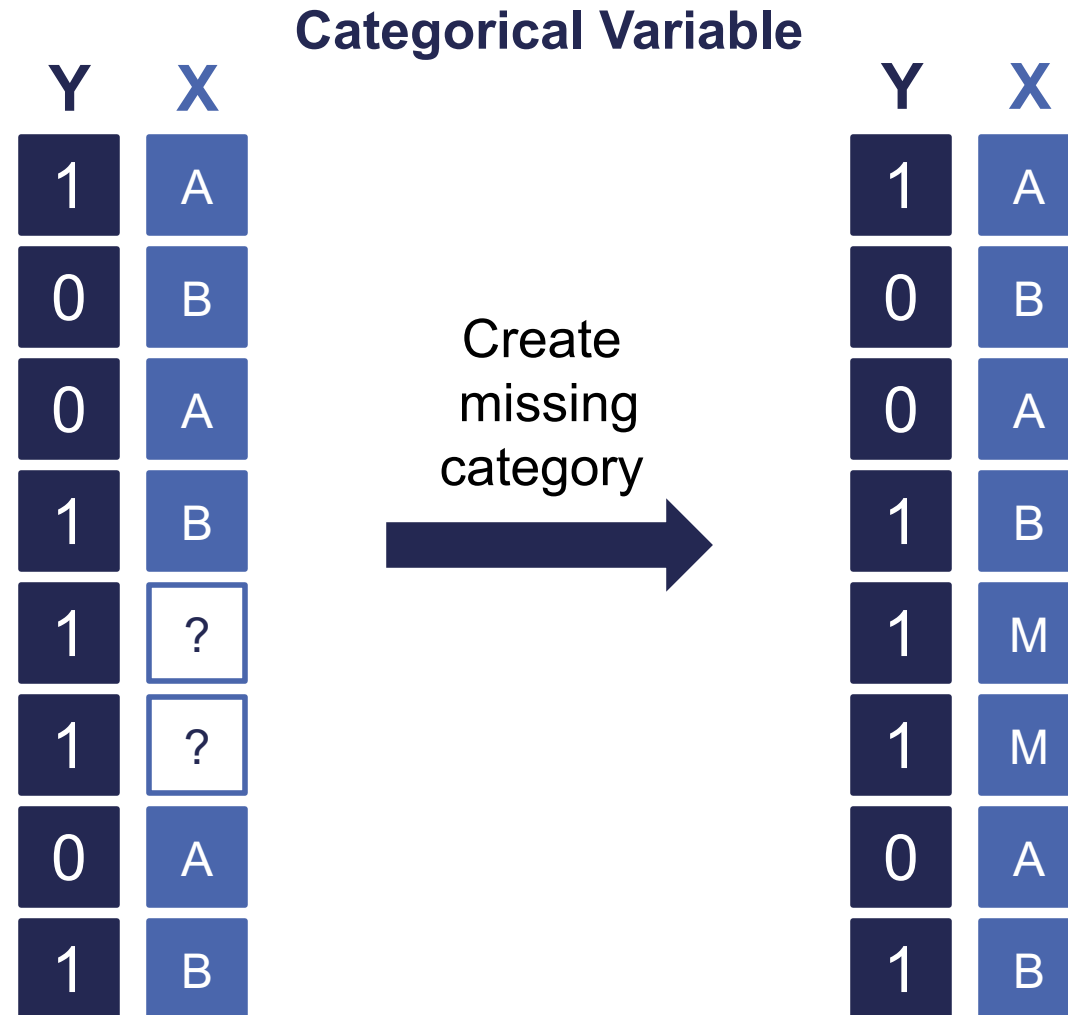
Delete

- If a majority of your data is missing, then consider deleting the variable all together.
- More than 50% missing → Remove

Y	X
1	14
0	?
0	?
1	18
1	?
1	?
0	31
1	?

Keep

- Missing values in predictor variables are not necessarily bad.
- In fact, they might even be predictive.
- Easy to handle with categorical variables!
- Add a missing category.




Replace

- Could estimate a missing value with **imputation**.
- Not best to do with categorical variables as you can just add missing category.
- Approaches:
 1. Simple mean/median replacement
 2. Predictive model using other variables (**not empirically shown to add value**)

Continuous Variable

Y	X
1	14
0	67
0	33
1	18
1	?
1	?
0	31
1	51

Impute
with
median



ALWAYS
add and
keep missing
flag variable

Y	X	X _M
1	14	0
0	67	0
0	33	0
1	18	0
1	32	1
1	32	1
0	31	0
1	51	0

Summary of General (not Strict) Imputation Rules

- If variable has more than 50% missing, consider deleting from analysis.
- **Categorical:**
 - Create missing value category for categorical variables.
- **Continuous:**
 - Impute missing values for continuous variables (median is a popular choice)
 - Create a missing value binary variable for each of the continuous variables you impute.



CONVERGENCE PROBLEMS

Linear Separation

- **Complete linear separation** occurs when some combination of the predictors perfectly predict **every** outcome:

	Yes	No
Group A	100	0
Group B	0	50

- **Quasi-complete separation** occurs when the outcome can be perfectly predicted for only a subset of the data:

	Yes	No
Group A	77	23
Group B	0	50

Linear Separation

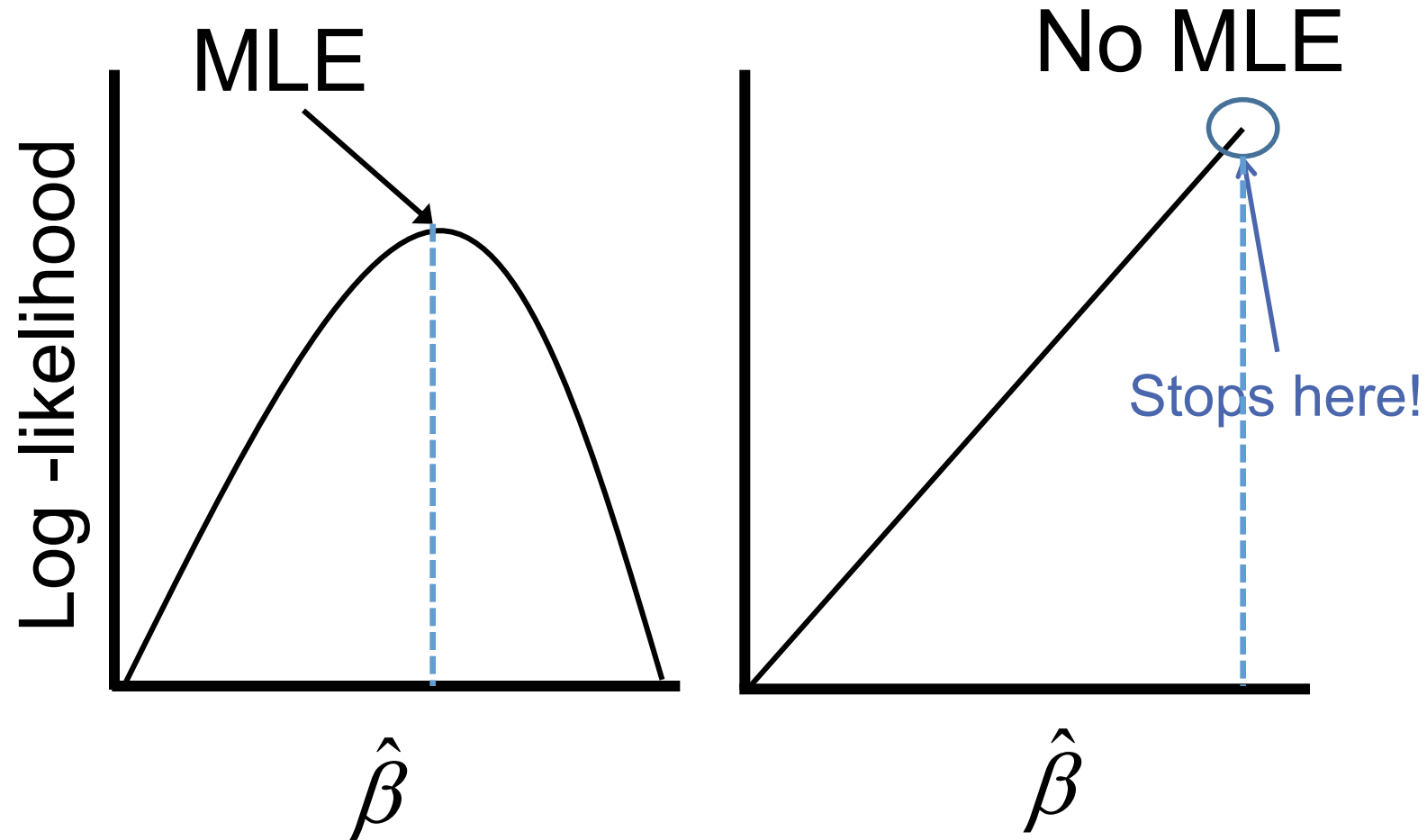
- **Complete linear separation** occurs when some combination of the predictors perfectly predict **every** outcome:

	Yes	No	Logit
Group A	100	0	∞
Group B	0	50	$-\infty$

- **Quasi-complete separation** occurs when the outcome can be perfectly predicted for only a subset of the data:

	Yes	No	Logit
Group A	77	23	1.39
Group B	0	50	$-\infty$

Problems with Convergence



Linear Separation – SAS

- SAS Warning Message:
 - **WARNING:** There is a complete separation of data points. The maximum likelihood estimate does not exist.
 - **WARNING:** The LOGISTIC procedure continues in spite of the above warning. Results shown are based on the last maximum likelihood iteration. Validity of the model fit is questionable.

Linear Separation – R

- Typical R Warning Message:

Linear Separation – R

- Sometimes R warning message deals with letting you know that you have predictions of exactly 0 or 1.
- However, it is not reliable to trust R to give a warning message.
- **Always explore data ahead of time.**
- Logistic regression output might also gives signs of a problem with parameter estimates.

Convergence Problems

```
table(train_u$customer.service.calls, train_u$churn)
```

	FALSE	TRUE
0	29	25
1	34	25
2	23	20
3	15	12
4	2	13
5	1	4
6	0	3
7	0	2

← Problems with quasi-complete separation

Convergence Problems

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-10.14712	0.81612	-12.433	< 2e-16	***
factor(international.plan)yes	3.29304	0.34767	9.472	< 2e-16	***
factor(voice.mail.plan)yes	-0.99870	0.30331	-3.293	0.000993	***
total.day.charge	0.17914	0.01788	10.019	< 2e-16	***
factor(customer.service.calls)1	0.49904	0.36185	1.379	0.167847	
factor(customer.service.calls)2	1.44529	0.40013	3.612	0.000304	***
factor(customer.service.calls)3	1.22882	0.44653	2.752	0.005924	**
factor(customer.service.calls)4	3.61499	0.52008	6.951	3.63e-12	***
factor(customer.service.calls)5	2.38233	0.69880	3.409	0.000652	***
factor(customer.service.calls)6	22.86097	799.56689	0.029	0.977190	
factor(customer.service.calls)7	21.52660	1028.81105	0.021	0.983306	

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

Solutions

- Possible Solutions:
 - Collapse the categories of the predictor variable to eliminate the 0 cell count.
 - Penalized maximum likelihood.
 - Eliminate the category altogether – probably not reasonable since the category seems important!
 - Add a very small constant to the cell counts.

Solutions

- Possible Solutions:
 - Collapse the categories of the predictor variable to eliminate the 0 cell count.
 - Penalized maximum likelihood.
 - Eliminate the category altogether – probably not reasonable since the category seems important!
 - Add a very small constant to the cell counts.

Thresholding – Ordinal Option

Customer Service Calls	Sample Size	0	1
0	54	29	25
1	59	34	25
2	43	23	20
3	27	15	12
4	15	2	13
5	5	1	4
6	3	0	3
7	2	0	2

Thresholding – Ordinal Option

Customer Service Calls	Sample Size	0	1
0	54	29	25
1	59	34	25
2	43	23	20
3	27	15	12
4	15	2	13
5	5	1	4
6	3	0	3
7	2	0	2

Collapse the cells

Thresholding – Ordinal Option

Customer Service Calls	Sample Size	0	1
0	54	29	25
1	59	34	25
2	43	23	20
3	27	15	12
4+	25	3	22

Clustering Levels – Nominal Option

	0	1
A	28	7
B	16	0
C	94	11
D	23	21

Clustering Levels – Nominal Option

	0	1
A	28	7
B	16	0
C	94	11
D	23	21

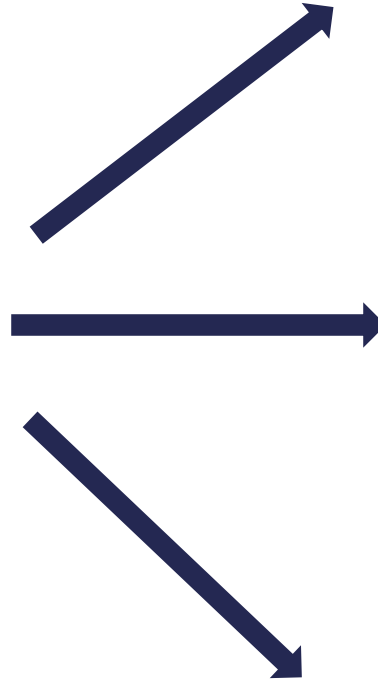
Most common categories

	0	1
A	28	7
B/C	110	11
D	23	21

Clustering Levels – Greenacre Method

	0	1
A	28	7
B	16	0
C	94	11
D	23	21

$$\chi^2 = 31.7$$



	0	1
A	28	7
B/C	110	11
D	23	21

$$\chi^2 = 30.7$$

	0	1
A/B	44	7
C	94	11
D	23	21

$$\chi^2 = 28.9$$

	0	1
A	28	7
C	110	11
B/D	39	21

$$\chi^2 = 18.3$$

Clustering Levels – Greenacre Method

	0	1
A	28	7
B	16	0
C	94	11
D	23	21

$$\chi^2 = 31.7$$

Least amount
information lost



	0	1
A	28	7
B/C	110	11
D	23	21

$$\chi^2 = 30.7$$

Combining Categories

```
train_u$customer.service.calls.c <- as.character(train_u$customer.service.calls)
train_u$customer.service.calls.c[which(train_u$customer.service.calls > 3)] <- "4+"

table(train_u$customer.service.calls.c, train_u$churn)
```

	FALSE	TRUE
0	29	25
1	34	25
2	23	20
3	15	12
4+	3	22

Combining Categories

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-9.17881	0.73429	-12.500	< 2e-16	***
factor(international.plan)yes	3.08119	0.33258	9.265	< 2e-16	***
factor(voice.mail.plan)yes	-1.14283	0.27822	-4.108	4.00e-05	***
total.day.charge	0.15725	0.01647	9.550	< 2e-16	***
factor(customer.service.calls.c)1	0.44968	0.35420	1.270	0.20424	
factor(customer.service.calls.c)2	1.25310	0.38494	3.255	0.00113	**
factor(customer.service.calls.c)3	1.03735	0.43343	2.393	0.01670	*
factor(customer.service.calls.c)4+	3.45969	0.42516	8.137	4.04e-16	***

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

Solutions

- Possible Solutions:
 - Collapse the categories of the predictor variable to eliminate the 0 cell count.
 - Penalized maximum likelihood.
 - Use the `brglm()` function in R in place of `glm()` function.
 - **Not covered here.**
 - Eliminate the category altogether – probably not reasonable since the category seems important!
 - Add a very small constant to the cell counts.

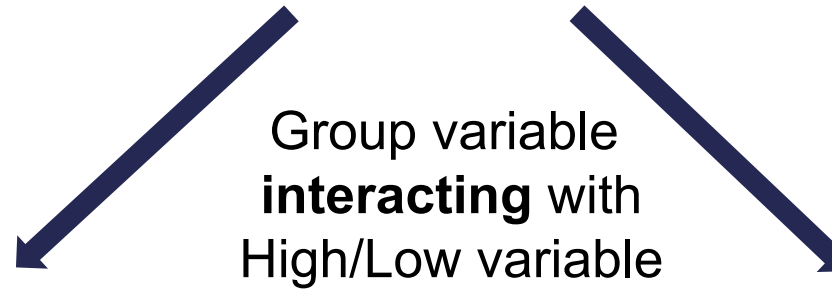
Watch out for Interactions!

	Yes	No
Group A	77	23
Group B	16	50

Group variable
seems good

Watch out for Interactions!

	Yes	No
Group A	77	23
Group B	16	50



	Yes	No
High	43	11
Low	0	41

	Yes	No
High	34	12
Low	16	9

