



# STOR 320 Modeling V

Lecture 20

Yao Li

Department of Statistics and Operations Research

UNC Chapel Hill

# Introduction

- Now We Consider
  - Categorical Response Variables
  - Numerical/Categorical Explanatory Variables
- Focus is on Classification
- Read Chapter 4 in ISLR

# Introduction

- Basic Case: Binary Response
  - Variable Has Two Possible Outcomes
  - Typically, Yes or No Responses to a Question
  - Example
    - $Y$  = Who Will Win the 2020 Presidential Election?
    - $Y$  = Did You Pass Your STOR 320 Class?
    - $Y$  = What Factors Influence the Admission into Graduate School?

# Scenario

- Question: Are Students Who Get Good Grades Likely to be Admitted to Graduate School?
  - $Y$  = Would the Student be Admitted to a Graduate School?
  - $X$  = College GPA
- Why is Linear Regression Inappropriate?

$$P(\text{Admission}|X) = \beta_0 + \beta_1 X$$

# Problem Setting

- Bernoulli Random Variable

$$Y = \begin{cases} 1 & \text{if Yes} \\ 0 & \text{if No} \end{cases}$$
$$p = E(Y) = P(Y = 1)$$

- Sample  $n$  Students

$$Y' = \sum Y_i \sim \text{Binomial}(n, p)$$

$$\hat{p} = \frac{\sum y_i}{n}$$

Estimated Probability that a Student Would  
be Admitted to a Graduate School

- Analyze the Effect of  $X$  on  $p$ :  $p = E(Y|X) \neq \beta_0 + \beta_1 X$

# Logit Link

- Modeling the Mean

- Logit Link Function

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X$$



Odds of  
Admission

- Understanding Odds
  - Odds of Admission = 1
  - Odds of Admission < 1
  - Odds of Admission > 1

# Model Construction

- Solving for  $\frac{p}{1-p}$

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X$$

$$\frac{p}{1-p} = e^{\beta_0 + \beta_1 X} \longrightarrow$$

Odds of Admission Given  
the Student's GPA

- Solving for  $p$

$$p = e^{\beta_0 + \beta_1 X} - p e^{\beta_0 + \beta_1 X}$$

$$p(1 + e^{\beta_0 + \beta_1 X}) = e^{\beta_0 + \beta_1 X}$$

$$p = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}} \longrightarrow$$

Probability of Admission Given the  
Student's GPA

# Logistic Regression for Classification

- Recall:  $Y = \begin{cases} 1 & \text{if Yes} \\ 0 & \text{if No} \end{cases}$
- After Getting Data, We Estimate
  - $\hat{\beta}_0$
  - $\hat{\beta}_1$
  - $\hat{p} = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X}} \rightarrow$ 

Estimated Probability of Admission Given the Student's GPA
- Two Scenarios
  - $\hat{p} < 0.5 \Rightarrow \hat{Y} = 0$
  - $\hat{p} > 0.5 \Rightarrow \hat{Y} = 1$



# Evaluating the LR Model

- Two Methods
  - Leave Out Data Intentionally
  - Use Cross-Validation
- Positives and Negatives
  - True Positive = Predicted an Admission and the Student Got Admitted
  - False Positive = Predicted an Admission and the Student Didn't Get Admitted
  - False Negative = Predicted a Student Wouldn't be Admitted and They Did Get Admitted
  - True Negative = Predicted a Student Wouldn't be Admitted and They Didn't Get Admitted

# Confusion Matrix

- Confusion Matrix

	Predicted	
	Will be Admitted	Won't be Admitted
Actual Admission	$n_{11}$	$n_{12}$
Isn't Admitted	$n_{21}$	$n_{22}$

- Sensitivity:

$$n_{11}/(n_{11} + n_{12})$$

- Specificity:

$$n_{22}/(n_{21} + n_{22})$$

- False Positive Rate:

$$n_{21}/(n_{21} + n_{22})$$

- False Negative Rate:

$$n_{12}/(n_{11} + n_{12})$$

# Titanic: Data

- Titanic Survival Data `> library(titanic)`

- Response Variable

$$Y = \begin{cases} 1 & \text{if Survived} \\ 0 & \text{if Did Not Survive} \end{cases}$$

- Explanatory Variables
  - Passenger Class
  - Sex
  - Age
  - Siblings/Spouses Aboard
  - Parents/Children Aboard
  - Passenger Fare
  - Port of Embarkation

# Titanic: Data

- Titanic Survival Data (Continued)
  - Selecting Variables of Interest

```
> TRAIN=titanic_train[,c(2,3,5,6,7,8,10,12)]
> TEST=titanic_test[,c(2,4,5,6,7,9,11)]
```

- Glimpse of Data

```
glimpse (TRAIN)
```

```
## Observations: 891
## Variables: 8
## $ Survived <int> 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1,...
## $ Pclass <int> 3, 1, 3, 1, 3, 3, 1, 3, 3
## $ Sex <chr> "male", "female", "female"
## $ Age <dbl> 22, 38, 26, 35, 35, NA, 5
## $ SibSp <int> 1, 1, 0, 1, 0, 0, 0, 3, 0
## $ Parch <int> 0, 0, 0, 0, 0, 0, 0, 1, 2
## $ Fare <dbl> 7.2500, 71.2833, 7.9250,
## $ Embarked <chr> "S", "C", "S", "S", "S",
glimpse(TEST)
## Observations: 418
## Variables: 7
## $ Pclass <int> 3, 3, 2, 3,
## $ Sex <chr> "male", "fe
```

```
glimpse (TEST)
```

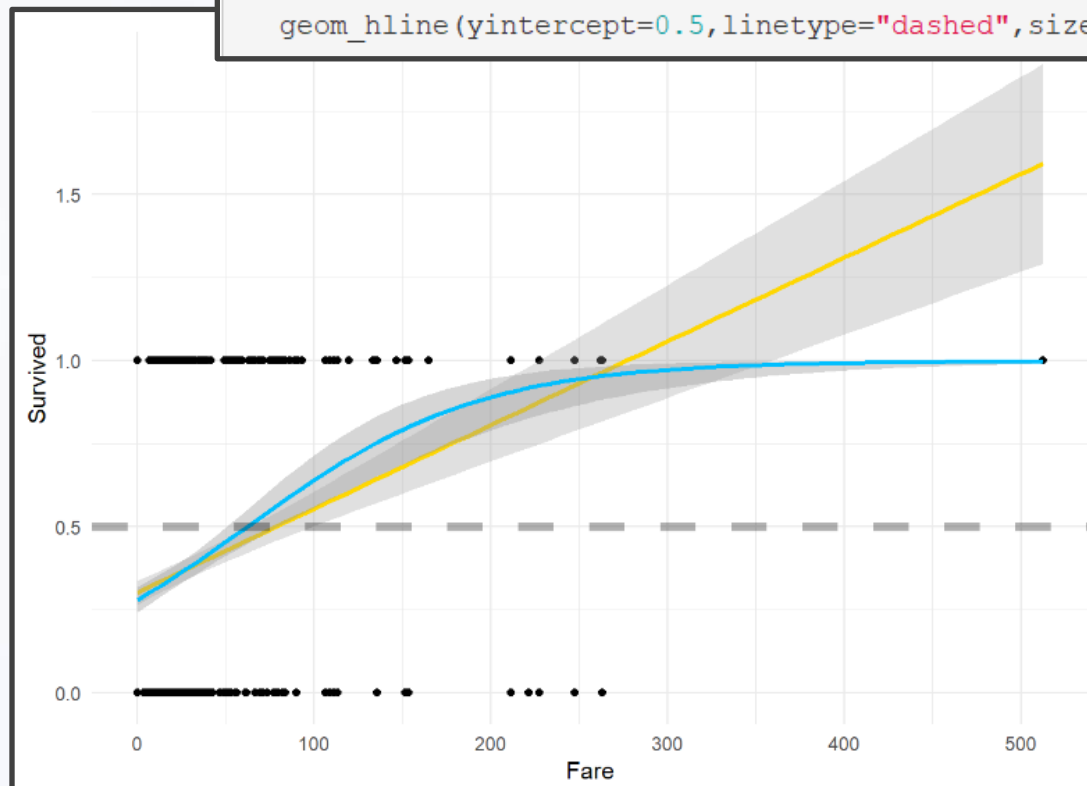
# Problem?

```
## Observations: 418
## Variables: 7
## $ Pclass    <int> 3, 3, 2, 3, 3, 3, 3, 2, 3, 3, 3, 1, 1, 2, 1, 2, 2, 3,...
## $ Sex       <chr> "male", "female", "male", "male", "female", "male", "...
## $ Age       <dbl> 34.5, 47.0, 62.0, 27.0, 22.0, 14.0, 30.0, 26.0, 18.0,...
## $ SibSp     <int> 0, 1, 0, 0, 1, 0, 0, 1, 0, 2, 0, 0, 1, 1, 1, 1, 0, 0,...
## $ Parch     <int> 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Fare      <dbl> 7.8292, 7.0000, 9.6875, 8.6625, 12.2875, 9.2250, 7.62...
## $ Embarked  <chr> "Q", "S", "Q", "S", "S", "S", "Q", "S", "C", "S", "S"...
```

# Visualization: Survival vs. Fare

- Visualizing the Data

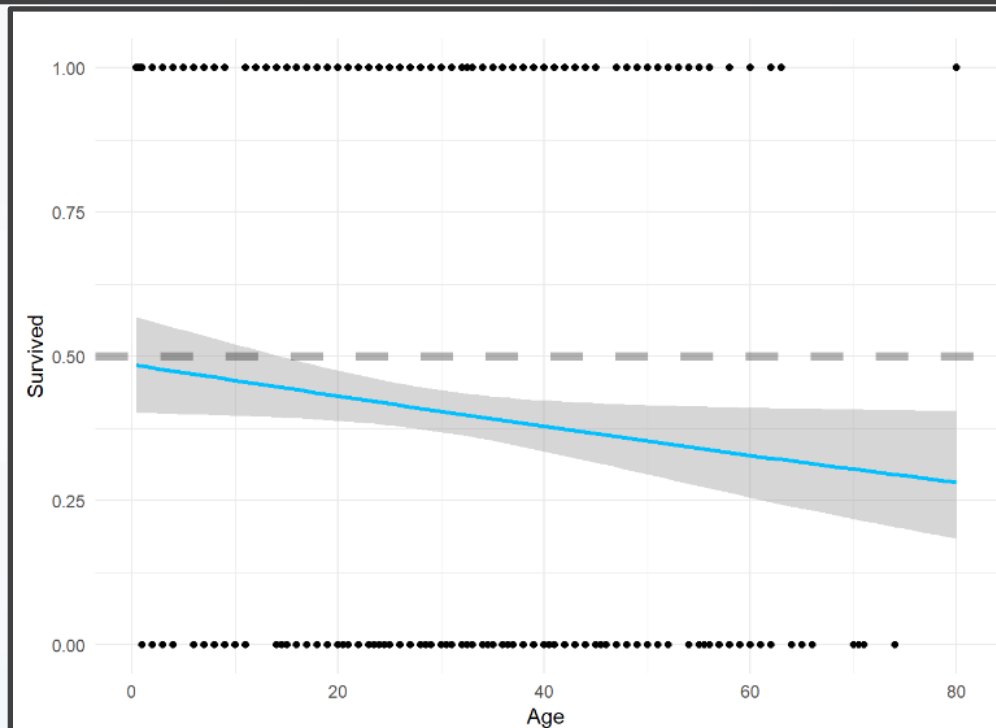
```
ggplot(TRAIN) + geom_point(aes(x=Fare,y=Survived)) + theme_minimal() +  
  geom_smooth(aes(x=Fare,y=Survived),method="lm",alpha=0.3,color="gold") +  
  geom_smooth(aes(x=Fare,y=Survived),method="glm",  
              method.args=list(family="binomial"),color="deepskyblue1") +  
  geom_hline(yintercept=0.5,linetype="dashed",size=2,alpha=0.3)
```



# Visualization: Survival vs. Age

- Visualizing the Data (Continued)

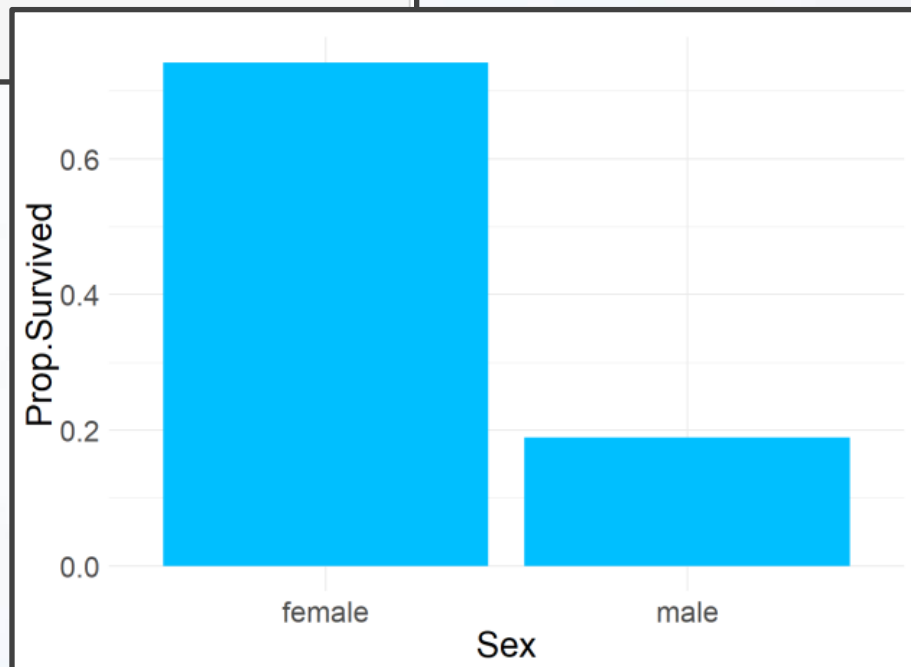
```
ggplot(TRAIN) + geom_point(aes(x=Age,y=Survived)) + theme_minimal() +  
  geom_smooth(aes(x=Age,y=Survived),method="glm",  
              method.args=list(family="binomial"),color="deepskyblue1") +  
  geom_hline(yintercept=0.5,linetype="dashed",size=2,alpha=0.3)
```



# Visualization: Survival vs. Sex

- Visualizing the Data (Continued)

```
TRAIN %>%  
  mutate(Sex=factor(Sex)) %>%  
  group_by(Sex) %>%  
  summarize(Prop.Survived=mean(Survived)) %>%  
  ggplot() +  
  geom_bar(aes(x=Sex,y=Prop.Survived),  
           stat="Identity",fill="deepskyblue1") +  
  theme_minimal() +  
  theme(text=element_text(size=20))
```



# Data Splitting

- Logistic Regression Models
  - Split Training Set Up

```
> set.seed(216)
> sample.in=sample(1:dim(TRAIN)[1],
                  size=floor(0.8*dim(TRAIN)[1]))
> TRAIN.IN=TRAIN[sample.in,
                  c("Survived","Fare","Sex","Age")]
> TRAIN.OUT=TRAIN[-sample.in,
                  c("Survived","Fare","Sex","Age")]
```

- Modeling the Probability of Survival Given the Ticket Fare, the Sex of the Passenger, and the Age of the Passenger



# Model 1

- Logistic Regression Models (Cont.)
  - Including 3-Way Interaction

```
logmod1=glm(Survived~.^3,family="binomial",data=TRAIN.IN)  
tidy(logmod1)[,c("term","estimate","p.value")]
```

```
## # A tibble: 8 x 3  
##   term                estimate p.value  
##   <chr>              <dbl>    <dbl>  
## 1 (Intercept)        0.959     0.0719  
## 2 Fare              -0.0132    0.357  
## 3 Sexmale           -1.54      0.0182  
## 4 Age               -0.0362    0.0745  
## 5 Fare:Sexmale       0.0180    0.255  
## 6 Fare:Age           0.00177   0.00684  
## 7 Sexmale:Age       -0.000359 0.988  
## 8 Fare:Sexmale:Age -0.00168   0.0140
```

# Model 2

- Logistic Regression Models (Cont.)
  - Only 2-Way Interactions

```
logmod2=glm(Survived~.*.,family="binomial",data=TRAIN.IN)  
tidy(logmod2)[,c("term","estimate","p.value")]
```

```
## # A tibble: 7 x 3  
##   term          estimate p.value  
##   <chr>          <dbl>   <dbl>  
## 1 (Intercept)    0.0835    0.846  
## 2 Fare           0.0202    0.0459  
## 3 Sexmale       -0.472     0.355  
## 4 Age            0.00244   0.858  
## 5 Fare:Sexmale  -0.0204    0.0225  
## 6 Fare:Age       0.000255   0.188  
## 7 Sexmale:Age   -0.0456    0.00482
```

# Model 3

- Logistic Regression Models (Cont.)
  - No Way Interactions

```
logmod3=glm(Survived~., family="binomial", data=TRAIN.IN)  
tidy(logmod3)[,c("term", "estimate", "p.value")]
```

```
## # A tibble: 4 x 3  
##   term      estimate p.value  
##   <chr>      <dbl>    <dbl>  
## 1 (Intercept)  1.03    1.42e- 4  
## 2 Fare        0.0117  2.23e- 5  
## 3 Sexmale     -2.32    6.58e-28  
## 4 Age        -0.0157  2.87e- 2
```

# Predictions

- Getting Predictions

```
TRAIN.OUT2 = TRAIN.OUT %>%  
  mutate(p1=predict(logmod1,newdata=TRAIN.OUT,type="response"),  
         p2=predict(logmod2,newdata=TRAIN.OUT,type="response"),  
         p3=predict(logmod3,newdata=TRAIN.OUT,type="response")) %>%  
  select(Survived,p1,p2,p3) %>%  
  mutate(S1=ifelse(p1<0.5,0,1),  
         S2=ifelse(p2<0.5,0,1),  
         S3=ifelse(p3<0.5,0,1))  
head(TRAIN.OUT2,15)
```

##	Survived	p1	p2	p3	S1	S2	S3
## 1	1	0.9690919	0.9092749	0.7802745	1	1	1
## 2	1	0.7754082	0.7600334	0.6058744	1	1	1
## 3	1	0.2080353	0.2054202	0.2124202	0	0	0
## 4	0	0.6660041	0.6390900	0.7598035	1	1	1
## 5	0	NA	NA	NA	NA	NA	NA
## 6	1	NA	NA	NA	NA	NA	NA
## 7	0	0.5144529	0.6150895	0.6255526	1	1	1
## 8	0	NA	NA	NA	NA	NA	NA
## 9	0	0.3504463	0.3477779	0.2826244	0	0	0
## 10	0	0.2084528	0.2141609	0.1755685	0	0	0
## 11	0	0.3588175	0.3684181	0.2646063	0	0	0
## 12	0	0.2278485	0.2365545	0.1841222	0	0	0
## 13	0	0.1588185	0.1560858	0.1590190	0	0	0
## 14	1	0.2135621	0.2103355	0.2445736	0	0	0
## 15	1	NA	NA	NA	NA	NA	NA


Why?

# Predictions

- Getting Predictions

```
TRAIN.OUT3=na.omit(TRAIN.OUT2)
head(TRAIN.OUT3,20)
```

##	Survived		p1	p2	p3	s1	s2	s3
## 1	1	0.9690919	0.9092749	0.7802745	1	1	1	
## 2	1	0.7754082	0.7600334	0.6058744	1	1	1	
## 3	1	0.2080353	0.2054202	0.2124202	0	0	0	
## 4	0	0.6660041	0.6390900	0.7598035	1	1	1	
## 7	0	0.5144529	0.6150895	0.6255526	1	1	1	
## 9	0	0.3504463	0.3477779	0.2826244	0	0	0	
## 10	0	0.2084528	0.2141609	0.1755685	0	0	0	



```
mean(TRAIN.OUT3$s1==TRAIN.OUT3$s2)
```

```
## [1] 0.993007
```

```
mean(TRAIN.OUT3$s2==TRAIN.OUT3$s3)
```

```
## [1] 1
```

What Do You Notice About the Predictions?

# Predictions

- Getting Predictions

```
TRAIN.OUT4=TRAIN.OUT3 %>% select(-p2,-S2)  
head(TRAIN.OUT4,8)
```

##	Survived		p1	p3	S1	S3
## 1	1	0.9690919	0.7802745	1	1	
## 2	1	0.7754082	0.6058744	1	1	
## 3	1	0.2080353	0.2124202	0	0	
## 4	0	0.6660041	0.7598035	1	1	
## 7	0	0.5144529	0.6255526	1	1	
## 9	0	0.3504463	0.2826244	0	0	
## 10	0	0.2084528	0.1755685	0	0	
## 11	0	0.3588175	0.2646063	0	0	



Where Do You See Error?

# Evaluation

- Evaluating Results
- Helpful Modifications

```
TRAIN.OUT5 = TRAIN.OUT4 %>%  
  select(-p1,-p3) %>%  
  mutate(Survived=factor(Survived),S1=factor(S1),S3=factor(S3)) %>%  
  mutate(Survived=fct_recode(Survived,"Survived"="1","Died"="0"),  
         S1=fct_recode(S1,"Will Survive"="1","Will Die"="0"),  
         S3=fct_recode(S3,"Will Survive"="1","Will Die"="0")) %>%  
  mutate(Survived=factor(Survived,levels=c("Survived","Died")),  
         S1=factor(S1,levels=c("Will Survive","Will Die")),  
         S3=factor(S3,levels=c("Will Survive","Will Die"))  
  
head(TRAIN.OUT5)
```

```
##   Survived      S1      S3  
## 1 Survived Will Survive Will Survive  
## 2 Survived Will Survive Will Survive  
## 3 Survived   Will Die   Will Die  
## 4    Died Will Survive Will Survive  
## 5    Died Will Survive Will Survive  
## 6    Died   Will Die   Will Die
```

# Evaluation: Confusion Matrix

- Evaluating Results (Continued)
  - Confusion Matrix
    - Including 3-Way Interactions

```
RESULTS1=table(TRAIN.OUT5$Survived,TRAIN.OUT5$S1) %>%  
  prop.table()  
print(RERESULTS1)
```

```
##  
##           Will Survive   Will Die  
##   Survived    0.32867133 0.13986014  
##   Died        0.07692308 0.45454545
```

- No Way Interactions

```
RESULTS3=table(TRAIN.OUT5$Survived,TRAIN.OUT5$S3) %>%  
  prop.table()  
print(RERESULTS3)
```

```
##  
##           Will Survive   Will Die  
##   Survived    0.33566434 0.13286713  
##   Died        0.07692308 0.45454545
```



# Evaluation: Rates

- Evaluating Results (Continued)

- Error Statistics

- Code

```
ERROR.RESULTS = tibble(  
  Model=c("3 Way", "No Way"),  
  Sensitivity=c(RESULTS1[1,1]/sum(RESULTS1[1,]), RESULTS3[1,1]/sum(RESULTS3[1,])),  
  Specificity=c(RESULTS1[2,2]/sum(RESULTS1[2,]), RESULTS3[2,2]/sum(RESULTS3[2,])),  
  FPR=c(RESULTS1[2,1]/sum(RESULTS1[2,]), RESULTS3[2,1]/sum(RESULTS3[2,])),  
  FNR=c(RESULTS1[1,2]/sum(RESULTS1[1,]), RESULTS3[1,2]/sum(RESULTS3[1,]))  
)  
print(ERROR.RESULTS)
```

- Results

Model	Sensitivity	Specificity	FPR	FNR
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
3 Way	0.701	0.855	0.145	0.299
No Way	0.716	0.855	0.145	0.284