Logistic Regression Extensions

Prof Wells

STA 295: Stat Learning

April 4th, 2024

Outline

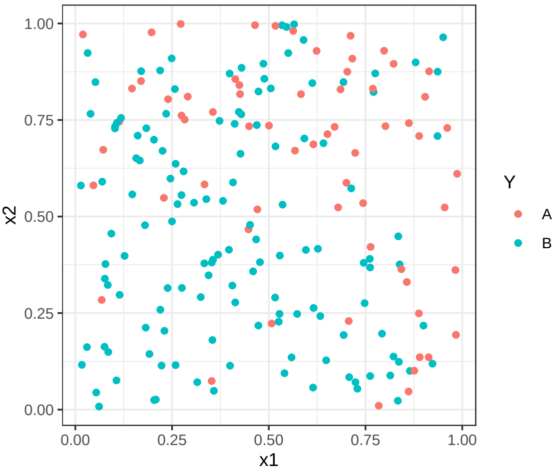
- Implement logistic regression in R
- Discuss extensions of logistic regression:
  - Transformations
  - Multinomial logistic regression
  - Penalized logistic regression

Section 1

Logistic Regression

Logistic Regression in R

Recall the simulation of 200 points from the model $p = \frac{x_1^2 + x_2^2}{2}$:



Logistic Regression in R

Before we fit the model, we need to pay attention to the response variable:

```
str(sim_data$Y)
```

```
##  Factor w/ 2 levels "A","B": 1 2 2 2 2 2 1 1 2 2 ...
```

Logistic Regression in R

Before we fit the model, we need to pay attention to the response variable:

```
str(sim_data$Y)
```

```
##  Factor w/ 2 levels "A","B": 1 2 2 2 2 2 1 1 2 2 ...
```

- Logistic regression requires the response to either be binary numeric (0 or 1) or a binary **factor**

Logistic Regression in R

Before we fit the model, we need to pay attention to the response variable:

```
str(sim_data$Y)
```

```
##  Factor w/ 2 levels "A","B": 1 2 2 2 2 2 1 1 2 2 ...
```

- Logistic regression requires the response to either be binary numeric (0 or 1) or a binary **factor**
    - The model will estimate the probability of the second level (i.e. $P(Y = B)$)

Logistic Regression in R

Before we fit the model, we need to pay attention to the response variable:

```
str(sim_data$Y)
```

```
##  Factor w/ 2 levels "A","B": 1 2 2 2 2 2 1 1 2 2 ...
```

- Logistic regression requires the response to either be binary numeric (0 or 1) or a binary **factor**
    - The model will estimate the probability of the second level (i.e. $P(Y = B)$)
- To change this, we can either recode the response as numeric:

```
sim_data$Y <- ifelse(sim_data$Y == "A", 1, 0)
head(sim_data$Y)
```

```
## [1] 1 0 0 0 0 0
```

Logistic Regression in R

Before we fit the model, we need to pay attention to the response variable:

```
str(sim_data$Y)
```

```
##  Factor w/ 2 levels "A","B": 1 2 2 2 2 2 1 1 2 2 ...
```

- Logistic regression requires the response to either be binary numeric (0 or 1) or a binary **factor**
  - The model will estimate the probability of the second level (i.e. $P(Y = B)$)
- To change this, we can either recode the response as numeric:

```
sim_data$Y <- ifelse(sim_data$Y == "A", 1, 0)
head(sim_data$Y)
```

```
## [1] 1 0 0 0 0 0
```

- Or we can relevel the factor:

```
sim_data$Y <- factor(sim_data$Y, levels = c("B", "A"))
head(sim_data$Y)
```

```
## [1] A B B B B B
## Levels: B A
```

Logistic Regression in R

We fit a logistic regression model using the `glm` function.

Logistic Regression in R

We fit a logistic regression model using the `glm` function.

```
sim_logistic <- glm(Y ~ x1 + x2, data = sim_data, family = "binomial")
```

Logistic Regression in R

We fit a logistic regression model using the glm function.

```
sim_logistic <- glm(Y ~ x1 + x2, data = sim_data, family = "binomial")
```

- We need to include `family = "binomial"` to tell R we want **logistic** regression

Logistic Regression in R

We fit a logistic regression model using the `glm` function.

```
sim_logistic <- glm(Y ~ x1 + x2, data = sim_data, family = "binomial")
```

- We need to include `family = "binomial"` to tell R we want **logistic** regression
- We can view the fitted model using `summary`, or just the coefficient estimates using `$coefficients`

Logistic Regression in R

We fit a logistic regression model using the glm function.

```
sim_logistic <- glm(Y ~ x1 + x2, data = sim_data, family = "binomial")
```

- We need to include `family = "binomial"` to tell R we want **logistic** regression
- We can view the fitted model using `summary`, or just the coefficient estimates using `$coefficients`

```
summary(sim_logistic)$coefficients
```

```
##               Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -3.472875  0.5685977 -6.107789 1.010206e-09
## x1           2.746111  0.6570948  4.179170 2.925746e-05
## x2           2.448198  0.5996131  4.082962 4.446520e-05
```

Logistic Regression in R

We fit a logistic regression model using the `glm` function.

```
sim_logistic <- glm(Y ~ x1 + x2, data = sim_data, family = "binomial")
```

- We need to include `family = "binomial"` to tell R we want **logistic** regression
- We can view the fitted model using `summary`, or just the coefficient estimates using `$coefficients`

```
summary(sim_logistic)$coefficients
```

```
##              Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -3.472875  0.5685977 -6.107789 1.010206e-09
## x1           2.746111  0.6570948  4.179170 2.925746e-05
## x2           2.448198  0.5996131  4.082962 4.446520e-05
```

- From the table, our logistic regression model is

$$\log \frac{p(X_1, X_2)}{1 + p(X_1, X_2)} = -3.47 + 2.75 \cdot X_1 + 2.45 \cdot X_2$$

Classification

To classify using logistic regression, we set a classification threshhold (usually 0.5) and predict $Y = 1$ if $P(x) > 0.5$.

Classification

To classify using logistic regression, we set a classification threshhold (usually 0.5) and predict $Y = 1$ if $P(x) > 0.5$.

- If $P(x) = 0.5$, then odds are $\frac{P(x)}{1-P(x)} = \frac{0.5}{0.5} = 1$, and log odds are $\log(1) = 0$.

Classification

To classify using logistic regression, we set a classification threshhold (usually 0.5) and predict $Y = 1$ if $P(x) > 0.5$.

- If $P(x) = 0.5$, then odds are $\frac{P(x)}{1-P(x)} = \frac{0.5}{0.5} = 1$, and log odds are $\log(1) = 0$.
  - Thus, we classify $Y = 1$ if $\log \mathrm{odds} > 0$.

Classification

To classify using logistic regression, we set a classification threshhold (usually 0.5) and predict $Y = 1$ if $P(x) > 0.5$.

- If $P(x) = 0.5$, then odds are $\frac{P(x)}{1-P(x)} = \frac{0.5}{0.5} = 1$, and log odds are $\log(1) = 0$.
  - Thus, we classify $Y = 1$ if $\log \mathrm{odds} > 0$.
- Our fitted model predicting whether $Y = A$ was

$$\log \frac{p(X_1, X_2)}{1 + p(X_1, X_2)} = -3.47 + 2.75 \cdot X_1 + 2.45 \cdot X_2$$

and so we classify $Y = A$ if

$$0 < -3.47 + 2.75 \cdot X_1 + 2.45 \cdot X_2$$

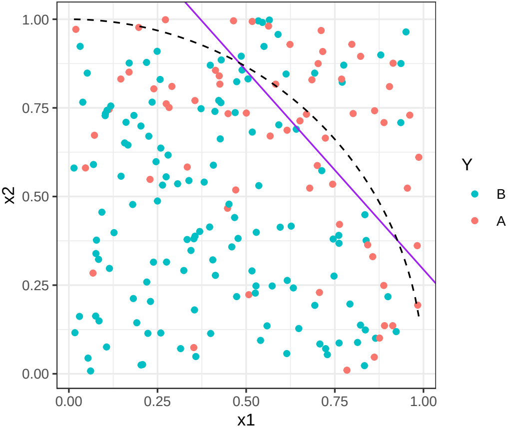
or equivalently, if

$$X_2 > (3.47 - 2.75 \cdot X_1)/2.45$$

Decision Boundary

The logistic decision boundary is $X_2 = (3.47 - 2.75 \cdot X_1)/2.45$ (purple)

- We classify as A all points above this line, and classify as B all points below this line.
- The Bayes Classifier decision boundary shown in black



Prediction

We can use the `predict` function to classify points using logistic regression.

Prediction

We can use the `predict` function to classify points using logistic regression.

```
my_preds <- predict(sim_logistic, newdata = test_data)
head(my_preds)
```

```
##          1           2           3           4           5           6
##  0.77874924 -0.03902659 -0.43933156 -0.53148993 -0.03576242 -1.62153528
```

- By default, `predict` will output the estimated log-odds for a point

Prediction

We can use the `predict` function to classify points using logistic regression.

```
my_preds <- predict(sim_logistic, newdata = test_data)
head(my_preds)
```

```
##           1           2           3           4           5           6
##  0.77874924 -0.03902659 -0.43933156 -0.53148993 -0.03576242 -1.62153528
```

- By default, `predict` will output the estimated log-odds for a point
- To instead output estimated probabilities, include `type = "response"`

Prediction

We can use the predict function to classify points using logistic regression.

```
my_preds <- predict(sim_logistic, newdata = test_data)
head(my_preds)
```

```
##          1           2           3           4           5           6
##  0.77874924 -0.03902659 -0.43933156 -0.53148993 -0.03576242 -1.62153528
```

- By default, predict will output the estimated log-odds for a point
- To instead output estimated probabilities, include type = "response"

```
my_preds_prob <- predict(sim_logistic, newdata = test_data, type = "response")
head(my_preds_prob)
```

```
##         1         2         3         4         5         6
## 0.6854105 0.4902446 0.3919003 0.3701695 0.4910603 0.1649932
```

Prediction

We can use the predict function to classify points using logistic regression.

```
my_preds <- predict(sim_logistic, newdata = test_data)
head(my_preds)
```

```
##          1           2           3           4           5           6
##  0.77874924 -0.03902659 -0.43933156 -0.53148993 -0.03576242 -1.62153528
```

- By default, predict will output the estimated log-odds for a point
- To instead output estimated probabilities, include type = "response"

```
my_preds_prob <- predict(sim_logistic, newdata = test_data, type = "response")
head(my_preds_prob)
```

```
##         1         2         3         4         5         6
## 0.6854105 0.4902446 0.3919003 0.3701695 0.4910603 0.1649932
```

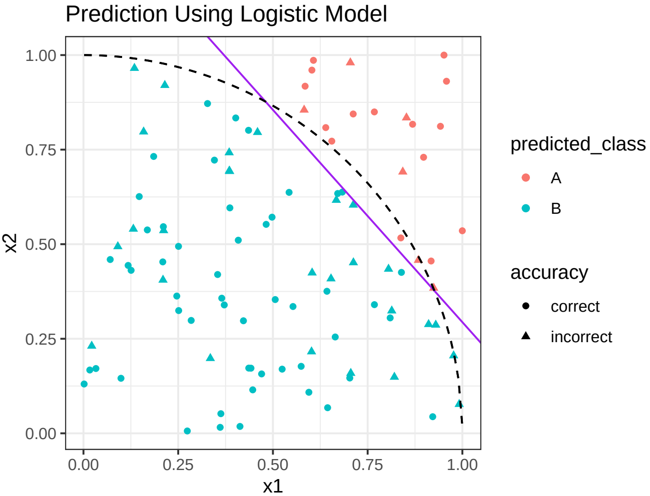
- To predict classes, apply the ifelse function to the probability vector

```
my_preds_class <- ifelse(my_preds_prob > 0.5, "A", "B")
head(my_preds_class)
```

```
##   1   2   3   4   5   6
## "A" "B" "B" "B" "B" "B"
```

Visualization

The following graph shows predicted classes for the test set, along with logistic classification boundary (purple) and theoretical Bayes classifier boundary (black)



Transformations

The decision boundary for every logistic regression model will always be linear.

- The rule: classify as 1 if $P(Y=1|X) > 0.5$" is equivalent to the rule: classify as 1 if

$$0 > \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

Transformations

The decision boundary for every logistic regression model will always be linear.

- The rule: classify as 1 if $P(Y = 1|X) > 0.5$" is equivalent to the rule: classify as 1 if

$$0 > \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

  - And this is a linear equation in the x's

Transformations

The decision boundary for every logistic regression model will always be linear.

- The rule: classify as 1 if $P(Y = 1|X) > 0.5$" is equivalent to the rule: classify as 1 if
$$0 > \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$
  - And this is a linear equation in the x's
- To create non-linear decision boundaries, we can instead write log-odds as a **non-linear** function of the predictors

Transformations

The decision boundary for every logistic regression model will always be linear.

- The rule: classify as 1 if $P(Y=1|X) > 0.5$" is equivalent to the rule: classify as 1 if
$$0 > \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$
  - And this is a linear equation in the x's
- To create non-linear decision boundaries, we can instead write log-odds as a **non-linear** function of the predictors
  - For example, we could use polynomial logistic regression:
$$\log \text{odds} = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_p x^p$$

Transformations

The decision boundary for every logistic regression model will always be linear.

- The rule: classify as 1 if $P(Y = 1|X) > 0.5$" is equivalent to the rule: classify as 1 if

$$0 > \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

  - And this is a linear equation in the x's
- To create non-linear decision boundaries, we can instead write log-odds as a **non-linear** function of the predictors
  - For example, we could use polynomial logistic regression:

$$\log \mathrm{odds} = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_p x^p$$

  - Or other non-linear transformations:

$$\log \mathrm{odds} = \beta_0 + \beta_1 e^{x_1} + \beta_2 \sqrt{x_2}$$

Circular Decision Boundaries

The Bayes Classifier decision boundary is an arc of a circle. Is there a way to use transformations to achieve this with logistic regression?

Circular Decision Boundaries

The Bayes Classifier decision boundary is an arc of a circle. Is there a way to use transformations to achieve this with logistic regression?

- Note that the equation of a circle is $r^2 = x_1^2 + x_2^2$, so we want our log-odds formula to involve sums of squares of predictors.

Circular Decision Boundaries

The Bayes Classifier decision boundary is an arc of a circle. Is there a way to use transformations to achieve this with logistic regression?

- Note that the equation of a circle is $r^2 = x_1^2 + x_2^2$, so we want our log-odds formula to involve sums of squares of predictors.

```
sim_mod_circ <- glm(Y ~ I(x1^2) + I(x2^2), data = sim_data, family = "binomial")
summary(sim_mod_circ)$coefficients
```

```
##              Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -2.505853  0.3842811 -6.520884 6.989438e-11
## I(x1^2)      2.725086  0.6206006  4.391046 1.128068e-05
## I(x2^2)      2.279677  0.5513573  4.134664 3.554747e-05
```

Circular Decision Boundaries

The Bayes Classifier decision boundary is an arc of a circle. Is there a way to use transformations to achieve this with logistic regression?

- Note that the equation of a circle is $r^2 = x_1^2 + x_2^2$, so we want our log-odds formula to involve sums of squares of predictors.

```
sim_mod_circ <- glm(Y ~ I(x1^2) + I(x2^2), data = sim_data, family = "binomial")
summary(sim_mod_circ)$coefficients
```

```
##                Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -2.505853  0.3842811 -6.520884 6.989438e-11
## I(x1^2)      2.725086  0.6206006  4.391046 1.128068e-05
## I(x2^2)      2.279677  0.5513573  4.134664 3.554747e-05
```

- Our model equation is $\log \text{odds} = -2.5 + 2.7x_1^2 + 2.3 \cdot x_2^2$

Circular Decision Boundaries

The Bayes Classifier decision boundary is an arc of a circle. Is there a way to use transformations to achieve this with logistic regression?

- Note that the equation of a circle is $r^2 = x_1^2 + x_2^2$, so we want our log-odds formula to involve sums of squares of predictors.

```
sim_mod_circ <- glm(Y ~ I(x1^2) + I(x2^2), data = sim_data, family = "binomial")
summary(sim_mod_circ)$coefficients
```

```
##              Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -2.505853  0.3842811 -6.520884 6.989438e-11
## I(x1^2)      2.725086  0.6206006  4.391046 1.128068e-05
## I(x2^2)      2.279677  0.5513573  4.134664 3.554747e-05
```

- Our model equation is $\log \mathrm{odds} = -2.5 + 2.7x_1^2 + 2.3 \cdot x_2^2$
- Setting log-odds equal to 0 actually gives the equation of an *ellipse*.

Circular Decision Boundaries

- If we insist on having circular decision boundaries, we could instead use

```
sim_mod_circ2 <- glm(Y ~ I(x1^2 + x2^2), data = sim_data, family = "binomial")
summary(sim_mod_circ2)$coefficients
```

```
##                   Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)    -2.497249  0.3829010 -6.521918 6.941413e-11
## I(x1^2 + x2^2)  2.469743  0.4578514  5.394201 6.882910e-08
```

Circular Decision Boundaries

- If we insist on having circular decision boundaries, we could instead use

```
sim_mod_circ2 <- glm(Y ~ I(x1^2 + x2^2), data = sim_data, family = "binomial")
summary(sim_mod_circ2)$coefficients
```

```
##                  Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)    -2.497249  0.3829010 -6.521918 6.941413e-11
## I(x1^2 + x2^2)  2.469743  0.4578514  5.394201 6.882910e-08
```

- Our model equation is $\log \mathrm{odds} = -2.5 + 2.5(x_1^2 + x_2^2)$

Circular Decision Boundaries

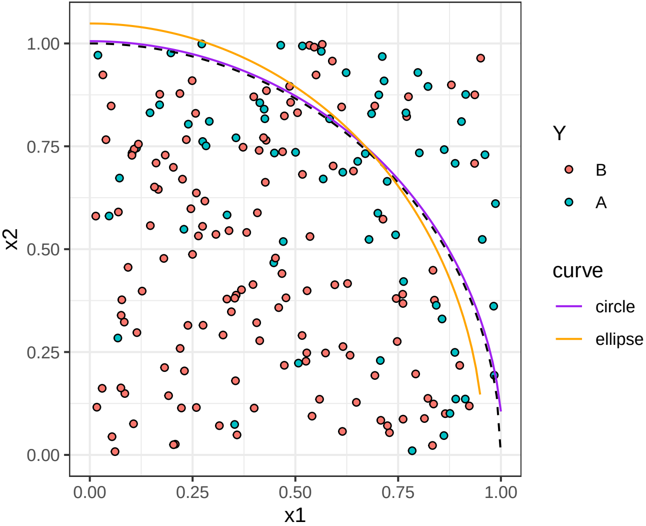
- If we insist on having circular decision boundaries, we could instead use

```
sim_mod_circ2 <- glm(Y ~ I(x1^2 + x2^2), data = sim_data, family = "binomial")
summary(sim_mod_circ2)$coefficients
```

```
##                  Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)     -2.497249  0.3829010 -6.521918 6.941413e-11
## I(x1^2 + x2^2)   2.469743  0.4578514  5.394201 6.882910e-08
```

- Our model equation is $\log \mathrm{odds} = -2.5 + 2.5(x_1^2 + x_2^2)$
- Setting log-odds equal to 0 actually indeed gives the equation of a *circle*.

Visualization



Section 2

Practice with Logistic Regression

The Unsinkable Example

The `Titanic` data set contains information on passengers of the *Titanic*

```
## Rows: 1,313
## Columns: 11
## $ row.names <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1~
## $ pclass    <chr> "1st", "1st", "1st", "1st", "1st", "1st", "1st", "1st", "1st~
## $ survived  <dbl> 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, ~
## $ name      <chr> "Allen, Miss Elisabeth Walton", "Allison, Miss Helen Loraine~
## $ age       <dbl> 29.0000, 2.0000, 30.0000, 25.0000, 0.9167, 47.0000, 63.0000,~
## $ embarked  <chr> "Southampton", "Southampton", "Southampton", "Southampton", ~
## $ home.dest <chr> "St Louis, MO", "Montreal, PQ / Chesterville, ON", "Montreal~
## $ room      <chr> "B-5", "C26", "C26", "C26", "C22", "E-12", "D-7", "A-36", "C~
## $ ticket    <chr> "24160 L221", NA, NA, NA, NA, NA, "13502 L77", NA, NA, NA, "~
## $ boat      <chr> "2", NA, "(135)", NA, "11", "3", "10", NA, "2", "(22)", "(12~
## $ sex       <chr> "female", "female", "male", "female", "male", "male", "femal~
```

- Goal: Build model for `survival` based on available predictors.

The Unsinkable Example

The `Titanic` data set contains information on passengers of the *Titanic*

```
## Rows: 1,313
## Columns: 11
## $ row.names <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1~
## $ pclass    <chr> "1st", "1st", "1st", "1st", "1st", "1st", "1st", "1st", "1st~
## $ survived  <dbl> 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, ~
## $ name      <chr> "Allen, Miss Elisabeth Walton", "Allison, Miss Helen Loraine~
## $ age       <dbl> 29.0000, 2.0000, 30.0000, 25.0000, 0.9167, 47.0000, 63.0000,~
## $ embarked  <chr> "Southampton", "Southampton", "Southampton", "Southampton", ~
## $ home.dest <chr> "St Louis, MO", "Montreal, PQ / Chesterville, ON", "Montreal~
## $ room      <chr> "B-5", "C26", "C26", "C26", "C22", "E-12", "D-7", "A-36", "C~
## $ ticket    <chr> "24160 L221", NA, NA, NA, NA, NA, "13502 L77", NA, NA, NA, "~
## $ boat      <chr> "2", NA, "(135)", NA, "11", "3", "10", NA, "2", "(22)", "(12~
## $ sex       <chr> "female", "female", "male", "female", "male", "male", "femal~
```

- Goal: Build model for `survival` based on available predictors.
- Is this primarily an inference or prediction task?

The Unsinkable Example

The `Titanic` data set contains information on passengers of the *Titanic*

```
## Rows: 1,313
## Columns: 11
## $ row.names <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1~
## $ pclass    <chr> "1st", "1st", "1st", "1st", "1st", "1st", "1st", "1st", "1st~
## $ survived  <dbl> 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, ~
## $ name      <chr> "Allen, Miss Elisabeth Walton", "Allison, Miss Helen Loraine~
## $ age       <dbl> 29.0000, 2.0000, 30.0000, 25.0000, 0.9167, 47.0000, 63.0000,~
## $ embarked  <chr> "Southampton", "Southampton", "Southampton", "Southampton", ~
## $ home.dest <chr> "St Louis, MO", "Montreal, PQ / Chesterville, ON", "Montreal~
## $ room      <chr> "B-5", "C26", "C26", "C26", "C22", "E-12", "D-7", "A-36", "C~
## $ ticket    <chr> "24160 L221", NA, NA, NA, NA, NA, "13502 L77", NA, NA, NA, "~
## $ boat      <chr> "2", NA, "(135)", NA, "11", "3", "10", NA, "2", "(22)", "(12~
## $ sex       <chr> "female", "female", "male", "female", "male", "male", "femal~
```

- Goal: Build model for `survival` based on available predictors.
- Is this primarily an inference or prediction task?
  - Can it be neither?

Data Analysis

```
library(skimr)
Titanic %>% select(age, sex, survived) %>% summary()
```

```
##       age              sex               survived    
##  Min.   : 0.1667   Length:1313        Min.   :0.000  
##  1st Qu.:21.0000   Class :character   1st Qu.:0.000  
##  Median :30.0000   Mode  :character   Median :0.000  
##  Mean   :31.1942                      Mean   :0.342  
##  3rd Qu.:41.0000                      3rd Qu.:1.000  
##  Max.   :71.0000                      Max.   :1.000  
##  NA's   :680
```

```
Titanic %>% count(sex)
```

```
## # A tibble: 2 x 2
##   sex        n
##   <chr>  <int>
## 1 female   463
## 2 male     850
```

```
Titanic %>% count(survived)
```

```
## # A tibble: 2 x 2
##   survived     n
##      <dbl> <int>
## 1        0   864
## 2        1   449
```

- What are some concerns we may have about variables `sex`, age and `survival`?

Data Analysis

```
library(skimr)
Titanic %>% select(age, sex, survived) %>% summary()
```

```
##       age               sex               survived    
##  Min.   : 0.1667   Length:1313        Min.   :0.000  
##  1st Qu.:21.0000   Class :character   1st Qu.:0.000  
##  Median :30.0000   Mode  :character   Median :0.000  
##  Mean   :31.1942                      Mean   :0.342  
##  3rd Qu.:41.0000                      3rd Qu.:1.000  
##  Max.   :71.0000                      Max.   :1.000  
##  NA's   :680
```

```
Titanic %>% count(sex)
```

```
## # A tibble: 2 x 2
##   sex        n
##   <chr>  <int>
## 1 female   463
## 2 male     850
```

```
Titanic %>% count(survived)
```

```
## # A tibble: 2 x 2
##   survived     n
##      <dbl> <int>
## 1        0   864
## 2        1   449
```

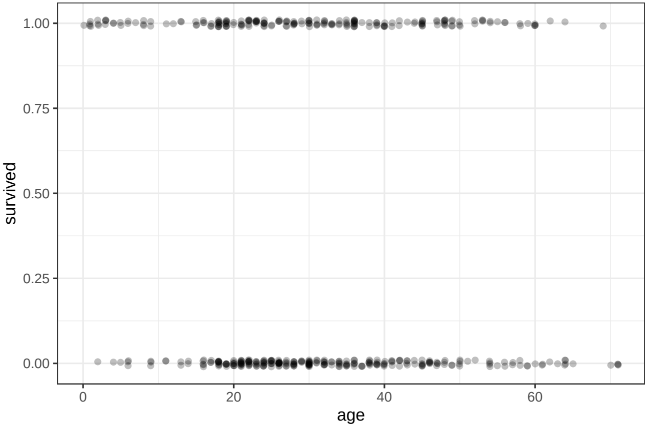
- What are some concerns we may have about variables `sex`, age and `survival`?

```
library(tidyr)
Titanic1<-Titanic %>% drop_na(age)

library(rsample)
set.seed(10)
Titanic1_split <- initial_split(Titanic1)
Titanic1_train <- training(Titanic1_split)
```

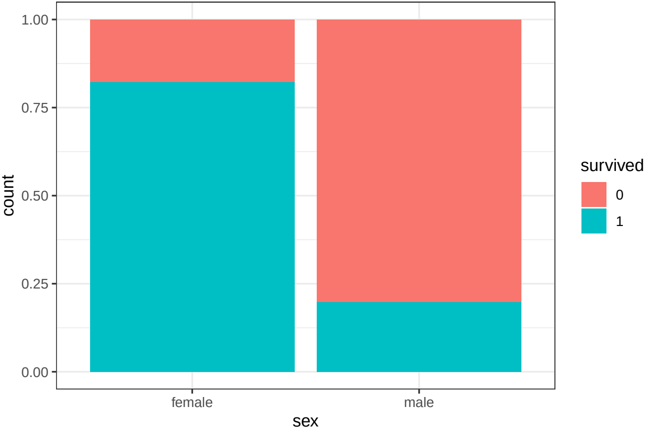
Children first?

- Who survived the Titanic?



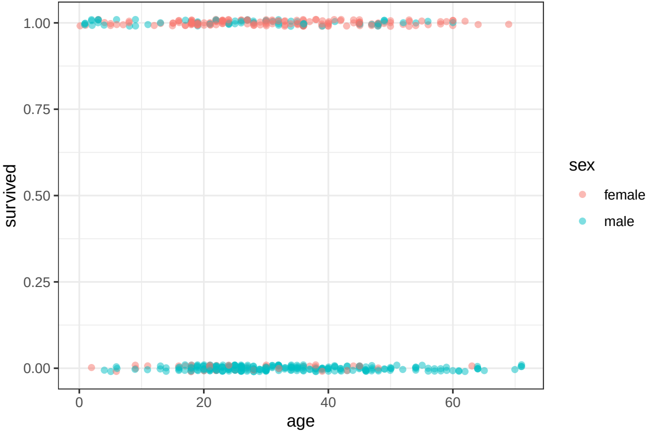
Women First?

- Who survived the Titanic?



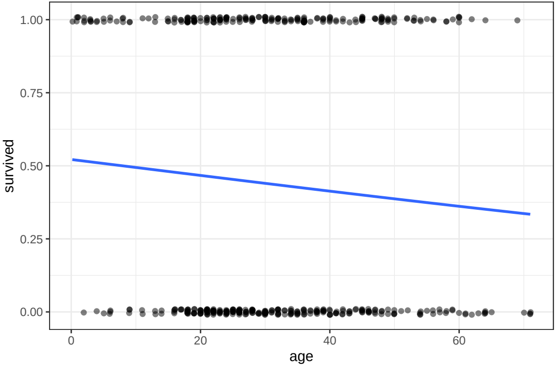
Women and Children First?

```
Titanic1_train %>% ggplot( aes( x = age, y = survived, color = sex))+
  geom_jitter(height = .01, alpha = .5)+theme_bw()
```



Logistic Model 1

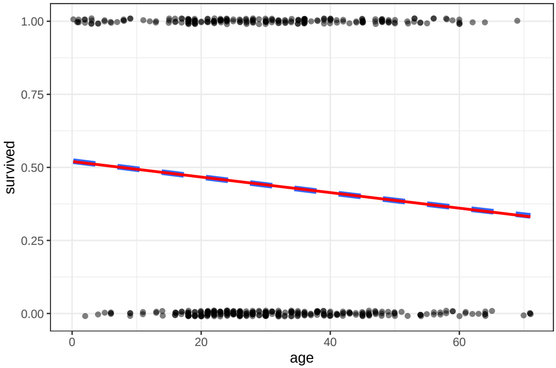
```
Titanic1_train %>% ggplot( aes( x = age, y = survived ))+
  geom_jitter(height = .01, alpha = .5)+theme_bw()+
  geom_smooth(method = "glm", method.args = list(family = "binomial"), se = F)
```



$$p(X) = \frac{e^{0.087-0.01X}}{1+e^{0.087-0.01X}}$$

VS Linear Model

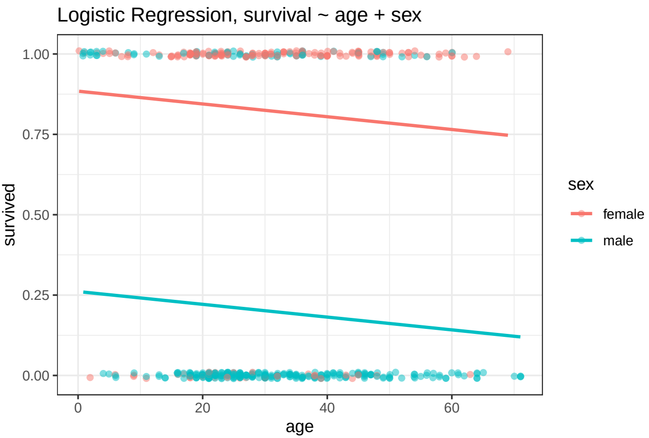
```
Titanic1_train %>% ggplot( aes( x = age, y = survived ))+
  geom_jitter(height = .01, alpha = .5)+theme_bw()+
  geom_smooth(method = "glm", method.args = list(family = "binomial"), se = F,size = 2,linetype
  geom_smooth(method = "lm", se = F, color  = "red")
```



$$p(X) = 0.520 - 0.003X$$

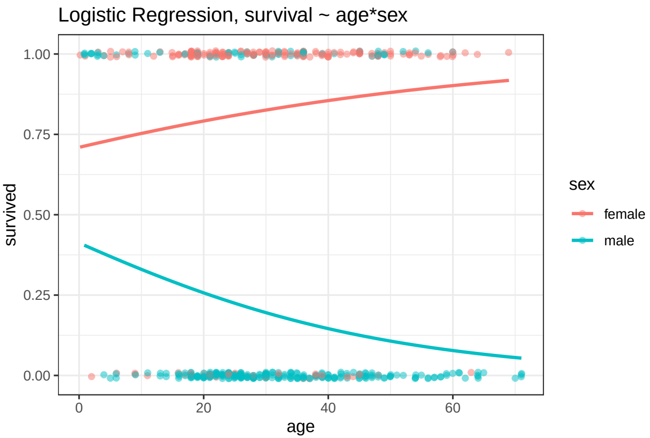
Logistic Model 2:

```
library(moderndive)
Titanic1_train %>% ggplot( aes( x = age, y = survived, color = sex ))+
  geom_jitter(height = .01, alpha = .5)+theme_bw()+
  geom_parallel_slopes(method = "glm", method.args = list(family = "binomial"), se = F)+
  labs(title = "Logistic Regression, survival ~ age + sex")
```



Logistic Model 3:

```
library(moderndive)
Titanic1_train %>% ggplot( aes( x = age, y = survived, color = sex ))+
  geom_jitter(height = .01, alpha = .5)+theme_bw()+
  geom_smooth(method = "glm", method.args = list(family = "binomial"), se = F)+
  labs(title = "Logistic Regression, survival ~ age*sex")
```



R code for Logistic Models

```
simple_logreg <- glm(survived ~ age, data = Titanic1_train, family = "binomial")
summary(simple_logreg)
```

```
##
## Call:
## glm(formula = survived ~ age, family = "binomial", data = Titanic1_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2049  -1.0857  -0.9893   1.2625   1.4708
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.086336   0.219226   0.394   0.6937
## age         -0.010926   0.006399  -1.707   0.0877 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 649.49  on 473  degrees of freedom
## Residual deviance: 646.54  on 472  degrees of freedom
## AIC: 650.54
##
## Number of Fisher Scoring iterations: 4
```

R code for Logistic Models

```
simple_logreg <- glm(survived ~ age, data = Titanic1_train, family = "binomial")
summary(simple_logreg)
```

```
##
## Call:
## glm(formula = survived ~ age, family = "binomial", data = Titanic1_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2049  -1.0857  -0.9893   1.2625   1.4708
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.086336   0.219226   0.394   0.6937
## age         -0.010926   0.006399  -1.707   0.0877 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 649.49  on 473  degrees of freedom
## Residual deviance: 646.54  on 472  degrees of freedom
## AIC: 650.54
##
## Number of Fisher Scoring iterations: 4
```

The logistic model is

$$\ln \frac{p(\mathrm{Age})}{1 - p(\mathrm{Age})} = 0.09 - 0.01 \cdot \mathrm{Age}$$

R code for Logistic Models

```
simple_logreg <- glm(survived ~ age, data = Titanic1_train, family = "binomial")
summary(simple_logreg)
```

```
##
## Call:
## glm(formula = survived ~ age, family = "binomial", data = Titanic1_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2049  -1.0857  -0.9893   1.2625   1.4708
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.086336   0.219226   0.394   0.6937
## age         -0.010926   0.006399  -1.707   0.0877 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 649.49  on 473  degrees of freedom
## Residual deviance: 646.54  on 472  degrees of freedom
## AIC: 650.54
##
## Number of Fisher Scoring iterations: 4
```

- The logistic model is

$$\ln \frac{p(\mathrm{Age})}{1 - p(\mathrm{Age})} = 0.09 - 0.01 \cdot \mathrm{Age}$$

- Since

$$e^{-0.011} = 0.989 = 1 - 0.011$$

increasing age by 1 year decreases survival odds by 1.1% **of the current odds**

R code for Logistic Models

```
simple_logreg <- glm(survived ~ age, data = Titanic1_train, family = "binomial")
summary(simple_logreg)
```

- Where is RSE? R^2? F-stat?

```
##
## Call:
## glm(formula = survived ~ age, family = "binomial", data = Titanic1_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2049  -1.0857  -0.9893   1.2625   1.4708
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.086336   0.219226   0.394   0.6937
## age         -0.010926   0.006399  -1.707   0.0877 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 649.49  on 473  degrees of freedom
## Residual deviance: 646.54  on 472  degrees of freedom
## AIC: 650.54
##
## Number of Fisher Scoring iterations: 4
```

R code for Logistic Models

```
simple_logreg <- glm(survived ~ age, data = Titanic1_train, family = "binomial")
summary(simple_logreg)
```

```
##
## Call:
## glm(formula = survived ~ age, family = "binomial", data = Titanic1_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2049  -1.0857  -0.9893   1.2625   1.4708
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.086336   0.219226   0.394   0.6937
## age         -0.010926   0.006399  -1.707   0.0877 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 649.49  on 473  degrees of freedom
## Residual deviance: 646.54  on 472  degrees of freedom
## AIC: 650.54
##
## Number of Fisher Scoring iterations: 4
```

- Where is RSE? R^2? F-stat?
- Logistic regression is from the family of *generalized linear models*
  - GLMs use *deviance* as metric of model fit.
  - Null deviance measures how well the null model (only intercept) predicts the data
  - Residual deviance measures how well the fitted model predicts the data

R code for Logistic Models

```
simple_logreg <- glm(survived ~ age, data = Titanic1_train, family = "binomial")
summary(simple_logreg)
```

```
##
## Call:
## glm(formula = survived ~ age, family = "binomial", data = Titanic1_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2049  -1.0857  -0.9893   1.2625   1.4708
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.086336   0.219226   0.394   0.6937
## age         -0.010926   0.006399  -1.707   0.0877 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 649.49  on 473  degrees of freedom
## Residual deviance: 646.54  on 472  degrees of freedom
## AIC: 650.54
##
## Number of Fisher Scoring iterations: 4
```

- Where is RSE? R^2? F-stat?
- Logistic regression is from the family of *generalized linear models*
  - GLMs use *deviance* as metric of model fit.
  - Null deviance measures how well the null model (only intercept) predicts the data
  - Residual deviance measures how well the fitted model predicts the data
- Fisher Scoring Iterations indicates the number of loops of numeric optimization algorithm

R code for Multiple Logistic Models

- Suppose we fit a logistic model for `survived ~ age + sex`:

```
logreg <- glm(survived ~ age + sex, data = Titanic1_train, family = "binomial")
summary(logreg)
```

```
##
## Call:
## glm(formula = survived ~ age + sex, family = "binomial", data = Titanic1_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0311  -0.6835  -0.5928   0.6363   1.9680
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.953077   0.329108   5.934 2.95e-09 ***
## age         -0.013107   0.008136  -1.611    0.107
## sexmale     -2.947348   0.245357 -12.012  < 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: 649.49  on 473  degrees of freedom
## Residual deviance: 457.81  on 471  degrees of freedom
## AIC: 463.81
##
## Number of Fisher Scoring iterations: 4
```

R code for Multiple Logistic Models

- Suppose we fit a logistic model for `survived ~ age + sex`:

```
logreg <- glm(survived ~ age + sex, data = Titanic1_train, family = "binomial")
summary(logreg)
```

```
##
## Call:
## glm(formula = survived ~ age + sex, family = "binomial", data = Titanic1_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0311  -0.6835  -0.5928   0.6363   1.9680
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.953077   0.329108   5.934 2.95e-09 ***
## age         -0.013107   0.008136  -1.611    0.107
## sexmale     -2.947348   0.245357 -12.012  < 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: 649.49  on 473  degrees of freedom
## Residual deviance: 457.81  on 471  degrees of freedom
## AIC: 463.81
##
## Number of Fisher Scoring iterations: 4
```

- What is the formula for the logistic model?

R code for Multiple Logistic Models

- Suppose we fit a logistic model for `survived ~ age + sex`:

```
logreg <- glm(survived ~ age + sex, data = Titanic1_train, family = "binomial")
summary(logreg)
```

```
##
## Call:
## glm(formula = survived ~ age + sex, family = "binomial", data = Titanic1_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0311  -0.6835  -0.5928   0.6363   1.9680
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.953077   0.329108   5.934 2.95e-09 ***
## age         -0.013107   0.008136  -1.611    0.107
## sexmale     -2.947348   0.245357 -12.012  < 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: 649.49  on 473  degrees of freedom
## Residual deviance: 457.81  on 471  degrees of freedom
## AIC: 463.81
##
## Number of Fisher Scoring iterations: 4
```

- What is the formula for the logistic model?
- What is the survival odds for a male child of age 5? A female child of age 5?

R code for Multiple Logistic Models

- Suppose we fit a logistic model for `survived ~ age + sex`:

```
logreg <- glm(survived ~ age + sex, data = Titanic1_train, family = "binomial")
summary(logreg)
```

```
##
## Call:
## glm(formula = survived ~ age + sex, family = "binomial", data = Titanic1_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0311  -0.6835  -0.5928   0.6363   1.9680
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.953077   0.329108   5.934 2.95e-09 ***
## age         -0.013107   0.008136  -1.611    0.107
## sexmale     -2.947348   0.245357 -12.012  < 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: 649.49  on 473  degrees of freedom
## Residual deviance: 457.81  on 471  degrees of freedom
## AIC: 463.81
##
## Number of Fisher Scoring iterations: 4
```

- What is the formula for the logistic model?
- What is the survival odds for a male child of age 5? A female child of age 5?
- What effect does being male have on survival odds?

R code for Multiple Logistic Models

- Suppose we fit a logistic model for `survived ~ age * sex`:

```r
logreg2 <- glm(survived ~ age * sex, data = Titanic1_train, family = "binomial")
summary(logreg2)
```

```
##
## Call:
## glm(formula = survived ~ age * sex, family = "binomial", data = Titanic1_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1814  -0.7023  -0.4754   0.6428   2.2616
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.89363    0.43623   2.048   0.0405 *
## age          0.02204    0.01402   1.572   0.1159
## sexmale     -1.24793    0.55518  -2.248   0.0246 *
## age:sexmale -0.05741    0.01797  -3.195   0.0014 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 649.49  on 473  degrees of freedom
## Residual deviance: 446.95  on 470  degrees of freedom
## AIC: 454.95
##
## Number of Fisher Scoring iterations: 4
```

R code for Multiple Logistic Models

- Suppose we fit a logistic model for `survived ~ age * sex`:

```
logreg2 <- glm(survived ~ age * sex, data = Titanic1_train, family = "binomial")
summary(logreg2)
```

```
##
## Call:
## glm(formula = survived ~ age * sex, family = "binomial", data = Titanic1_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1814  -0.7023  -0.4754   0.6428   2.2616
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.89363    0.43623   2.048   0.0405 *
## age          0.02204    0.01402   1.572   0.1159
## sexmale     -1.24793    0.55518  -2.248   0.0246 *
## age:sexmale -0.05741    0.01797  -3.195   0.0014 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 649.49  on 473  degrees of freedom
## Residual deviance: 446.95  on 470  degrees of freedom
## AIC: 454.95
##
## Number of Fisher Scoring iterations: 4
```

- What is the formula for the logistic model?

R code for Multiple Logistic Models

- Suppose we fit a logistic model for `survived ~ age * sex`:

```
logreg2 <- glm(survived ~ age * sex, data = Titanic1_train, family = "binomial")
summary(logreg2)
```

```
##
## Call:
## glm(formula = survived ~ age * sex, family = "binomial", data = Titanic1_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1814  -0.7023  -0.4754   0.6428   2.2616
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.89363    0.43623   2.048   0.0405 *
## age          0.02204    0.01402   1.572   0.1159
## sexmale     -1.24793    0.55518  -2.248   0.0246 *
## age:sexmale -0.05741    0.01797  -3.195   0.0014 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 649.49  on 473  degrees of freedom
## Residual deviance: 446.95  on 470  degrees of freedom
## AIC: 454.95
##
## Number of Fisher Scoring iterations: 4
```

- What is the formula for the logistic model?
- What is the survival odds for a male child of age 5? A female child of age 5?

R code for Multiple Logistic Models

- Suppose we fit a logistic model for `survived ~ age * sex`:

```
logreg2 <- glm(survived ~ age * sex, data = Titanic1_train, family = "binomial")
summary(logreg2)
```

```
##
## Call:
## glm(formula = survived ~ age * sex, family = "binomial", data = Titanic1_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1814  -0.7023  -0.4754   0.6428   2.2616
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.89363    0.43623   2.048   0.0405 *
## age          0.02204    0.01402   1.572   0.1159
## sexmale     -1.24793    0.55518  -2.248   0.0246 *
## age:sexmale -0.05741    0.01797  -3.195   0.0014 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 649.49  on 473  degrees of freedom
## Residual deviance: 446.95  on 470  degrees of freedom
## AIC: 454.95
##
## Number of Fisher Scoring iterations: 4
```

- What is the formula for the logistic model?
- What is the survival odds for a male child of age 5? A female child of age 5?
- What effect did male have on survival odds?

Classify Points

First, we obtain predicted probabilities for each of the 3 models:

Classify Points

First, we obtain predicted probabilities for each of the 3 models:

```
preds_simple <- predict(simple_logreg, newdata = Titanic1_test, type = "response")
preds_logreg <- predict(logreg, newdata = Titanic1_test, type = "response")
preds_logreg2 <- predict(logreg2, newdata = Titanic1_test, type = "response")
```

Classify Points

First, we obtain predicted probabilities for each of the 3 models:

```
preds_simple <- predict(simple_logreg, newdata = Titanic1_test, type = "response")
preds_logreg <- predict(logreg, newdata = Titanic1_test, type = "response")
preds_logreg2 <- predict(logreg2, newdata = Titanic1_test, type = "response")
```

- Now we assemble information into a long data frame:

Classify Points

First, we obtain predicted probabilities for each of the 3 models:

```
preds_simple <- predict(simple_logreg, newdata = Titanic1_test, type = "response")
preds_logreg <- predict(logreg, newdata = Titanic1_test, type = "response")
preds_logreg2 <- predict(logreg2, newdata = Titanic1_test, type = "response")
```

- Now we assemble information into a long data frame:

```
my_results <- data.frame(
  passenger_id = rep(1:159, times = 3),
  prob = c(preds_simple, preds_logreg, preds_logreg2),
  model = rep(c("simple", "logreg1", "logreg2"), each = 159),
  obs = rep(as.factor(Titanic1_test$survived), times = 3))
```

```
head(my_results, 5)
```

```
##   passenger_id      prob  model obs
## 1            1 0.4426271 simple   1
## 2            2 0.5190708 simple   1
## 3            3 0.3538913 simple   1
## 4            4 0.3664795 simple   1
## 5            5 0.3948028 simple   0
```

```
tail(my_results, 5)
```

```
##     passenger_id      prob   model obs
## 473          155 0.2246623 logreg2   0
## 474          156 0.1792104 logreg2   0
## 475          157 0.7528725 logreg2   0
## 476          158 0.1456248 logreg2   0
## 477          159 0.1844731 logreg2   0
```

Classify Points

First, we obtain predicted probabilities for each of the 3 models:

```
preds_simple <- predict(simple_logreg, newdata = Titanic1_test, type = "response")
preds_logreg <- predict(logreg, newdata = Titanic1_test, type = "response")
preds_logreg2 <- predict(logreg2, newdata = Titanic1_test, type = "response")
```

- Now we assemble information into a long data frame:

```
my_results <- data.frame(
  passenger_id = rep(1:159, times = 3),
  prob = c(preds_simple, preds_logreg, preds_logreg2),
  model = rep(c("simple", "logreg1", "logreg2"), each = 159),
  obs = rep(as.factor(Titanic1_test$survived), times = 3))
```

```
head(my_results, 5)
```

```
##   passenger_id      prob  model obs
## 1            1 0.4426271 simple   1
## 2            2 0.5190708 simple   1
## 3            3 0.3538913 simple   1
## 4            4 0.3664795 simple   1
## 5            5 0.3948028 simple   0
```

```
tail(my_results, 5)
```

```
##     passenger_id      prob   model obs
## 473          155 0.2246623 logreg2   0
## 474          156 0.1792104 logreg2   0
## 475          157 0.7528725 logreg2   0
## 476          158 0.1456248 logreg2   0
## 477          159 0.1844731 logreg2   0
```

- Finally, we classify points

Classify Points

First, we obtain predicted probabilities for each of the 3 models:

```
preds_simple <- predict(simple_logreg, newdata = Titanic1_test, type = "response")
preds_logreg <- predict(logreg, newdata = Titanic1_test, type = "response")
preds_logreg2 <- predict(logreg2, newdata = Titanic1_test, type = "response")
```

- Now we assemble information into a long data frame:

```
my_results <- data.frame(
  passenger_id = rep(1:159, times = 3),
  prob = c(preds_simple, preds_logreg, preds_logreg2),
  model = rep(c("simple", "logreg1", "logreg2"), each = 159),
  obs = rep(as.factor(Titanic1_test$survived), times = 3))
```

```
head(my_results, 5)
```

```
##   passenger_id      prob  model obs
## 1            1 0.4426271 simple   1
## 2            2 0.5190708 simple   1
## 3            3 0.3538913 simple   1
## 4            4 0.3664795 simple   1
## 5            5 0.3948028 simple   0
```

```
tail(my_results, 5)
```

```
##     passenger_id      prob   model obs
## 473          155 0.2246623 logreg2   0
## 474          156 0.1792104 logreg2   0
## 475          157 0.7528725 logreg2   0
## 476          158 0.1456248 logreg2   0
## 477          159 0.1844731 logreg2   0
```

- Finally, we classify points

```
my_results <- my_results %>% mutate(pred = as.factor(ifelse(prob > 0.5, 1, 0)))
```

Assessing Accuracy

Since all predictions and observations for the 3 models are in the same data frame, we can use group_by to simultaneously assess:

```
library(yardstick)
my_results %>% group_by(model) %>%
  accuracy(truth = obs, estimate = pred)
```

```
## # A tibble: 3 x 4
##   model   .metric  .estimator .estimate
##   <chr>   <chr>    <chr>          <dbl>
## 1 logreg1 accuracy binary         0.774
## 2 logreg2 accuracy binary         0.774
## 3 simple  accuracy binary         0.572
```

```
my_results %>% group_by(model) %>%
  roc_auc(truth = obs, prob, event_level = "second")
```

```
## # A tibble: 3 x 4
##   model   .metric .estimator .estimate
##   <chr>   <chr>   <chr>          <dbl>
## 1 logreg1 roc_auc binary         0.783
## 2 logreg2 roc_auc binary         0.806
## 3 simple  roc_auc binary         0.534
```

ROC Curve

```
r<- my_results %>% group_by(model) %>%
  roc_curve(truth = obs, prob, event_level = "second")

autoplot(r)
```