Learning R

Session 4: Creating a model with R using for loops and if-else statements

DR EDMUND RYAN

Overview of this R Course

1. Introduction to R (session 1)
2. Creating graphical plots with R (session 2)

Task 1

3. Correlation and Regression with R (session 3)
4. Creating a model with R using For loops and if-else statements (session 4)

Task 2

5. Hypothesis testing with R (session 5)
6. Using R packages to solve problems (session 6)

Task 3

Overview of this R Course

1. Introduction to R (session 1)
2. Creating graphical plots with R (session 2)

Task 1

3. Correlation and Regression with R (session 3)
4. Creating a model with R using For loops and if-else statements (session 4)

Task 2

5. Hypothesis testing with R (session 5)
6. Using R packages to solve problems (session 6)

Task 3

All R code & other files are on Github

https://github.com/edmundryan

In the previous session …

- In session 3:
  - We trained and carried out diagnostic checks for a linear regression model.
  - But I didn't show you how to use it.

- Second example from session 3
  - Predicting the miles per gallon of a car (mpg) from its cylinder displacement (disp), horsepower (hp) and weight (wt).
  - Model: $\text{mpg} = \beta_0 + \beta_1 disp + \beta_2 hp + \beta_3 wt$
  - Parameters or coefficients: $\beta_0,\ \beta_1,\ \beta_2,\ \beta_3$

Applying linear regression

```
#Training a linear on the 'mtcars' data and generating new input point:
LinearMod2 <- lm(mpg~disp+hp+wt, data = mtcars)
s1=summary(LinearMod2)
disp.newpoint=runif(1,min(mtcars$disp),max(mtcars$disp))
hp.newpoint=runif(1,min(mtcars$hp),max(mtcars$hp))
wt.newpoint=runif(1,min(mtcars$wt),max(mtcars$wt))

#Cylinder displacement in cubic inches (new point)
disp.newpoint

#horsepower (new point)
hp.newpoint

#weight of car in tons:
wt.newpoint

#Extracting the parameters of the linear model
names(s1)
s1$coefficients
param=as.vector(s1$coefficients[,1])

#Using the linear model to predict MPG using the values of the new input point:
mpg.pred=param[1]+(param[2]*disp.newpoint)+(param[3]*hp.newpoint)+(param[4]*wt.newpoint)
mpg.pred
```

In this session

- We will train and apply three multinomial logistic regression models
- These three models (or submodels) will join up to allow us to build a larger model that simulates the first batting innings of 20-20 cricket match.
- The model will also consist of various For loops and if-else statements, and will bring together other aspects of this R course (e.g. subsetting).

Multinomial logistic regression?

- Similar to linear regression except that what you're trying to predict only takes a fixed number of values.
- For example, if there are only two possible outcomes (such as $Y = 1$ or $Y = 2$) then:

$$\log\left(\frac{\Pr(Y=1)}{\Pr(Y=2)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_K X_K$$

- Then:

$$\Pr(Y = 1) = \frac{\exp(\beta_{0,m} + \beta_{1,m} X_1 + \cdots + \beta_{K,m} X_K)}{1 + \exp(\beta_{0,m} + \beta_{1,m} X_1 + \cdots + \beta_{K,m} X_K)}$$

- Probabilities add to 1, so:

$$\Pr(Y = 2) = 1 - \Pr(Y = 1)$$

Multinomial logistic regression?

- In general when there are M possible outcomes ($Y = 1, Y = 2, \ldots, Y = M$):

$$\log\left(\frac{\Pr(Y=1)}{\Pr(Y=M)}\right) = \beta_{0,1} + \beta_{1,1}X_1 + \beta_{2,1}X_2 + \cdots + \beta_{K,1}X_K$$

$$\log\left(\frac{\Pr(Y=2)}{\Pr(Y=M)}\right) = \beta_{0,2} + \beta_{1,2}X_1 + \beta_{2,2}X_2 + \cdots + \beta_{K,2}X_K$$

$$\vdots \qquad\qquad \vdots$$

$$\log\left(\frac{\Pr(Y=M-1)}{\Pr(Y=M)}\right) = \beta_{0,M-1} + \beta_{1,M-1}X_1 + \beta_{2,M-1}X_2 + \cdots + \beta_{K,M-1}X_K$$

- Then for $Y = 1, Y = 2, \ldots, Y = M - 1$:

$$\Pr(Y = m) = \frac{\exp\left(\beta_{0,m} + \beta_{1,m}X_1 + \cdots + \beta_{K,m}X_K\right)}{1 + \sum_{m=1}^{M-1}\exp\left(\beta_{0,m} + \beta_{1,m}X_1 + \cdots + \beta_{K,m}X_K\right)}$$

- Probabilities add to 1, so:

$$\Pr(Y = M) = 1 - \sum_{m=1}^{M-1}\Pr(Y = m)$$

Simulating a cricket match

We will just use a 3 outcome multinomial logistic regression three times for each of the 120 balls:

- Submodel 1: Predicting ‘0 runs’, ‘1-6 runs’ or ‘Wicket’
- Submodel 2 (if 1-6 runs): Predicting ‘1-3 runs’, ‘4 runs’ or ‘6 runs’
- Submodel 3 (if 1-3 runs): Predicting ‘1 run’, ‘2 runs’ or ‘3 runs’

For each submodel we compare a random number (0-1) with the probabilities of each of the outcomes to determine which outcome to select.

This will involve two for loops and various if-else statements.

Simulating a cricket match

- Predictor variables for each of the submodels:
  - Batting average of batsman.
  - PowerPlay status (Powerplay = a certain fielding setup).
  - Type of bowler (spinner or pace bowler).
- Training data
  - 14 home matches (20-20) for the Durham county team
  - First innings only when Durham were batting.
- Validation data
  - A 15th 20-20 home match for Durham.
  - First innings only when Durham were batting.

Inputs

MatchNumber	Innings	Over	BallNumber	PowerPlay	SpinBowler	BattingAverage
1	1	1	1	1	0	18.9
1	1	1	2	1	0	18.9
1	1	1	3	1	0	18.9
1	1	1	4	1	0	18.9
1	1	1	5	1	0	47
1	1	1	6	1	0	47
1	1	2	1	1	0	24.72727273
1	1	2	2	1	0	24.72727273
1	1	2	3	1	0	47
1	1	2	4	1	0	47
1	1	2	5	1	0	47
1	1	2	6	1	0	47
1	1	3	1	1	0	24.72727273
1	1	3	2	1	0	24.72727273
1	1	3	3	1	0	24.72727273
1	1	3	4	1	0	24.72727273
1	1	3	5	1	0	47
1	1	3	6	1	0	47
1	1	4	1	1	0	47
1	1	4	2	1	0	47
1	1	4	3	1	0	47
1	1	4	4	1	0	47
1	1	4	5	1	0	47
1	1	4	6	1	0	47
1	1	5	1	1	0	24.72727273
1	1	5	2	1	0	24.72727273
1	1	5	3	1	0	24.72727273
1	1	5	4	1	0	47
1	1	5	5	1	0	47
1	1	5	6	1	0	47
1	1	6	1	1	0	47
1	1	6	2	1	0	47
1	1	6	3	1	0	47
1	1	6	4	1	0	47
1	1	6	5	1	0	47
1	1	6	6	1	0	47
1	1	7	1	0	0	24.72727273
1	1	7	2	0	0	24.72727273
1	1	7	3	0	0	47
1	1	7	4	0	0	24.72727273
1	1	7	5	0	0	47
1	1	7	6	0	0	47
1	1	8	1	0	1	47

Outputs

BatsmenRuns	runs_0	runs_1	runs_2	runs_3	runs_4	runs_6	Wicket
0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0
4	0	0	0	0	1	0	0
4	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0
2	0	0	1	0	0	0	0
4	0	0	0	0	1	0	0
4	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0
2	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0
1	0	1	0	0	0	0	0
4	0	0	0	0	1	0	0
2	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0
4	0	0	0	0	1	0	0
4	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0
4	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0
1	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0

Task 2

- Include another For Loop in order to be able to run the cricket model for 5000 repeats.
- Work out the mean total number of batsman runs in the match for each model repeat, averaging over the 5000 repeats.
- Create a histogram of the total number of batsman runs for all 5000 model repeats.