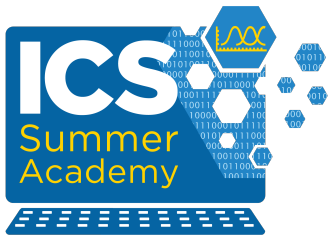
Linear and Logistic Regression

Babak Shahbaba and Sam Behseta
UC Irvine and CSU Fullerton

July and August, 2022



Linear Regression

- Now that you have learned about the intricacies of linear models, we will build a multiple regression model for predicting the left hippocampus volume of the brain, labeled as *lhippo*, through two predictors, namely *age* and *educ*.

Linear Regression

- Now that you have learned about the intricacies of linear models, we will build a multiple regression model for predicting the left hippocampus volume of the brain, labeled as *lhippo*, through two predictors, namely *age* and *educ*.
- Remember that in the general, the linear model can be written as follows:

$$Y = \beta_0 + \beta_1 X_1 + ... + \beta_k X_k.$$

- the left hand side of this model is the response variable, a numerical continuous variable.

Linear Regression

- Now that you have learned about the intricacies of linear models, we will build a multiple regression model for predicting the left hippocampus volume of the brain, labeled as *lhippo*, through two predictors, namely *age* and *educ*.
- Remember that in the general, the linear model can be written as follows:

$$Y = \beta_0 + \beta_1 X_1 + ... + \beta_k X_k.$$

- the left hand side of this model is the response variable, a numerical continuous variable.
- Thereby:

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + ... + \hat{\beta}_k X_k + \epsilon,$$

where ϵ refers to the error of the estimating the model parameters with the existing set of data.

Linear Regression for Predicting Left Hippocampus Volume

- Recall the left hippocampus volume *lhippo* is likely to shrink as Alzheimer's severs. Also, from Yueqi's introduction, while the progress of the disease is a function of age, it is possible that *education* can have a reverse effect on the progress of the disease.

Linear Regression for Predicting Left Hippocampus Volume

- Recall the left hippocampus volume *lhippo* is likely to shrink as Alzheimer's severs. Also, from Yueqi's introduction, while the progress of the disease is a function of age, it is possible that *education* can have a reverse effect on the progress of the disease.
- To fit linear models all we need to do is to apply the *lm* command in R. We begin with plotting the response versus each predictor, separately.

```
plot(age,lhippo,pch=16,col="red")
plot(educ,lhippo,pch=16,col="blue")
```

Linear Regression for Predicting Left Hippocampus Volume

- Here is the regression of *lhippo* versus *age*:

```
> lm.age<-lm(lhippo~age)
> summary(lm.age)

Call:
lm(formula = lhippo ~ age)

Residuals:
     Min       1Q   Median       3Q      Max
-2.58855 -0.28598  0.01999  0.31504  1.58641

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.0639626  0.0543632   74.76   <2e-16 ***
age         -0.0149051  0.0007657  -19.46   <2e-16 ***
---
Multiple R-squared:  0.1231,Adjusted R-squared:  0.1228
```

Linear Regression for Predicting Left Hippocampus Volume

- Here is the regression of *lhippo* versus *age*:

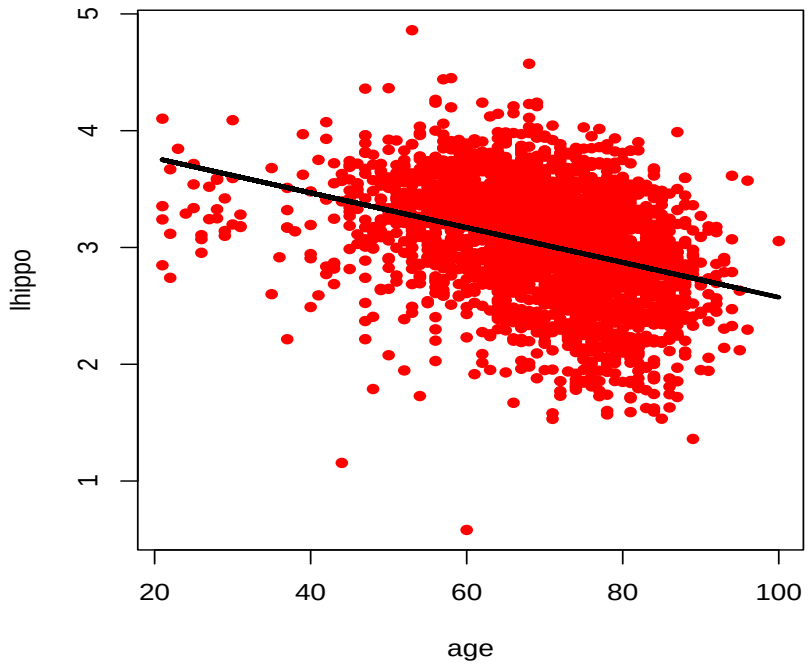
```
lm.age<-lm(lhippo~age)
summary(lm.age)
plot(age,lhippo,pch=16,col="red")
pred.age<-predict(lm.age)
lines(age,pred.age,lwd=3)
```

Linear Regression for Predicting Left Hippocampus Volume

- Here is the regression of *lhippo* versus *age*:

```
lm.age<-lm(lhippo~age)
summary(lm.age)
plot(age,lhippo,pch=16,col="red")
pred.age<-predict(lm.age)
lines(age,pred.age,lwd=3)
```

- Let's see the fitted line:



Linear Regression for Predicting Left Hippocampus Volume

- Here is the regression of *lhippo* versus *educ*:

```
> lm.age<-lm(lhippo~educ)
> summary(lm.educ)

Call:
lm(formula = lhippo ~ educ)

Residuals:
     Min       1Q   Median       3Q      Max
-2.45617 -0.30433  0.01738  0.33743  1.77443

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.647647   0.042951  61.644   <2e-16 ***
educ        0.024351   0.002743   8.877   <2e-16 ***
---
 Multiple R-squared:  0.02838,Adjusted R-squared:  0.02802
F-statistic:  78.8 on 1 and 2698 DF,  p-value: < 2.2e-16
```

Linear Regression for Predicting Left Hippocampus Volume

- Here is the regression of *lhippo* versus *education*:

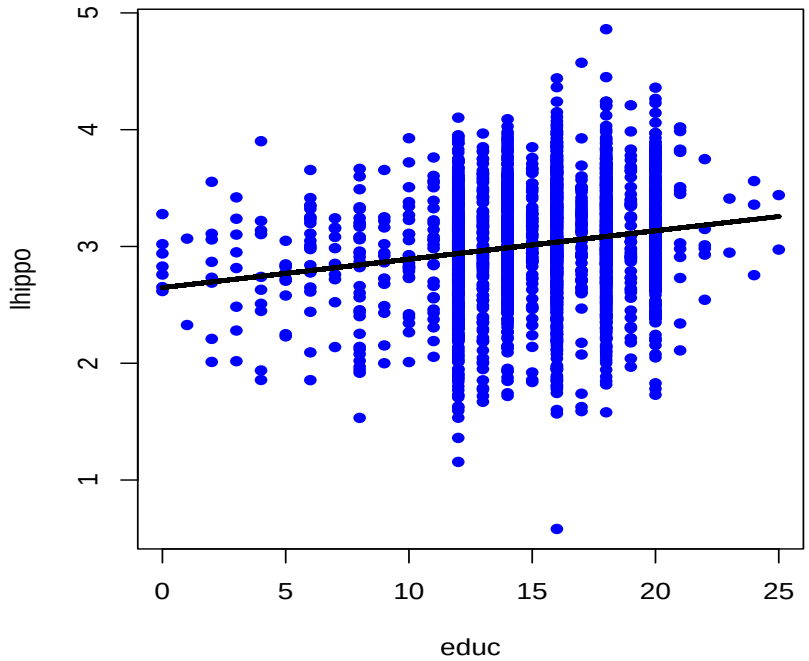
```
lm.educ<-lm(lhippo~educ)
summary(lm.educ)
plot(educ,lhippo,pch=16,col="blue")
pred.educ<-predict(lm.educ)
lines(educ,pred.educ,lwd=3)
```

Linear Regression for Predicting Left Hippocampus Volume

- Here is the regression of *lhippo* versus *education*:

```
lm.educ<-lm(lhippo~educ)
summary(lm.educ)
plot(educ,lhippo,pch=16,col="blue")
pred.educ<-predict(lm.educ)
lines(educ,pred.educ,lwd=3)
```

- Let's see the fitted line:



Linear Regression for Predicting Left Hippocampus Volume

- Here is the regression of *lhippo* versus *age* and*educ*:

```
> lm.AgeEduc<-lm(lhippo~age+educ)
> summary(lm.AgeEduc)

Call:
lm(formula = lhippo ~ age + educ)

Residuals:
     Min       1Q   Median       3Q      Max
-2.59525 -0.28746  0.01681  0.31416  1.54719

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.7348265  0.0709453  52.644  < 2e-16 ***
age         -0.0142527  0.0007643 -18.649  < 2e-16 ***
educ         0.0185428  0.0026010   7.129 1.29e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4551 on 2697 degrees of freedom
Multiple R-squared:  0.1394,Adjusted R-squared:  0.1387
F-statistic: 218.4 on 2 and 2697 DF,  p-value: < 2.2e-16
```

Logistic Regression

- Remember from the lecture that we are fitting a regression model with a binary outcome.

Logistic Regression

- Remember from the lecture that we are fitting a regression model with a binary outcome.
- As such, the model is as follows:

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 X_1 + ... + \beta_k X_k$$

- the left hand side of this model is the logarithm of the odds of success.

Logistic Regression

- Remember from the lecture that we are fitting a regression model with a binary outcome.
- As such, the model is as follows:

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 X_1 + ... + \beta_k X_k$$

- the left hand side of this model is the logarithm of the odds of success.
- Thereby, the probability of success of π can be written as follows:

$$\pi = \frac{\exp(\beta_0 + \beta_1 X_1 + ... + \beta_k X_k)}{1 + \exp(\beta_0 + \beta_1 X_1 + ... + \beta_k X_k)}$$

Logistic Regression

- Remember from the lecture that we are fitting a regression model with a binary outcome.
- As such, the model is as follows:

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 X_1 + ... + \beta_k X_k$$

- the left hand side of this model is the logarithm of the odds of success.
- Thereby, the probability of success of π can be written as follows:

$$\pi = \frac{\exp(\beta_0 + \beta_1 X_1 + ... + \beta_k X_k)}{1 + \exp(\beta_0 + \beta_1 X_1 + ... + \beta_k X_k)}$$

- The above means once we estimate the coefficients of the model (β's), we can estimate the probability of success of the outcome of interest.

Logistic Regression for *diagnosis* in R

- Let's revisit Alzheimer's data set, and consider the task of building a logistic regression model with *diagnosis* as its response variable and variables *age*, *education*, *naccicv*, and *female* as its predictors.

Logistic Regression for *diagnosis* in R

- Let's revisit Alzheimer's data set, and consider the task of building a logistic regression model with *diagnosis* as its response variable and variables *age*, *education*, *naccicv*, and *female* as its predictors.
- Let's begin by transforming the response to a new feature with two categories: no symptoms (0) versus mild or strong symptoms (1). There are a number of ways to achieve that in R. Below is a simple solution via the package *car*, and the command *recode* in that package.

```
library(car)
diagnosis.new<-recode(diagnosis,"c(1,2)='1';else='0'")
diag.new<-as.factor(diagnosis.new)
summary(diag.new)
```

Logistic Regression for *diagnosis* in R

- Running a logistic regression model in R is pretty straightforward. Before we do that, we should notice *female* is a binary variable as well. As such, we should make sure R recognizes that feature as a factor variable.

```
fem<-as.factor(female)

diag.logistic<-glm(diag.new~educ+age+naccicv+fem, family=binomial)
summary(diag.logistic)
```

Logistic Regression for *diagnosis* in R

- Let's try to carefully analyze the output of the model:

```
> summary(diag.logistic)

Call:
glm(formula = diag.new ~ educ + age + naccicv + fem, family = binomial,
    data = train.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0318  -1.0049  -0.6775   1.1187   2.2714

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.9063629  0.6766066  -1.340    0.180
educ        -0.0719999  0.0132779  -5.423 5.88e-08 ***
age          0.0425461  0.0041366  10.285  < 2e-16 ***
naccicv     -0.0005368  0.0003891  -1.380    0.168
fem1        -0.9422589  0.1030502  -9.144  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

Cross-Validation for the Logistic Regression in R

- How good is our model?

Cross-Validation for the Logistic Regression in R

- How good is our model?
- Let's try to calculate its *accuracy* or alternatively its *misclassification rate*. The higher the accuracy of a model, the better.

Cross-Validation for the Logistic Regression in R

- How good is our model?
- Let's try to calculate its *accuracy* or alternatively its *misclassification rate*. The higher the accuracy of a model, the better.
- We can achieve the above with a simple technique called *cross validation*.

Cross-Validation for the Logistic Regression in R

- How good is our model?
- Let's try to calculate its *accuracy* or alternatively its *misclassification rate*. The higher the accuracy of a model, the better.
- We can achieve the above with a simple technique called *cross validation*.
- This is an old approach, devised by the statisticians Fred Mosteller and John Tukey (1968).

Cross-Validation for the Logistic Regression in R

- How good is our model?
- Let's try to calculate its *accuracy* or alternatively its *misclassification rate*. The higher the accuracy of a model, the better.
- We can achieve the above with a simple technique called *cross validation*.
- This is an old approach, devised by the statisticians Fred Mosteller and John Tukey (1968).
- We split the data into *training* and *validation* or *test* sets. We fit or train the model using the training portion of the data set, and gauge its accuracy using the validation set.

Cross-Validation for the Logistic Regression in R

- Let's split the data set into training and validation sets. We let 2500 subjects to form our training set, and will hold the rest for validation purposes.

```
set.seed(1234)
a=seq(1,2700,1)
b=sample(a,2500,replace = F)

alz.logistic<-alz[,c("educ","age","naccicv")]
alz.logistic<-cbind(diag.new,alz.logistic,fem)
train.data<-alz.logistic[b,]
test.data<-alz.logistic[-b,]
```

Cross-Validation for the Logistic Regression in R

- Next, we train the model:

```
library(tidyverse)
library(caret)
diag.logistic<-glm(diag.new~educ+age+naccicv+fem,
family=binomial,data=train.data)
```

Cross-Validation for the Logistic Regression in R

- Next, we train the model:

```
library(tidyverse)
library(caret)
diag.logistic<-glm(diag.new~educ+age+naccicv+fem,
family=binomial,data=train.data)
```

- Followed, by testing it via the validation set. This means to calculate the probability of success for each subject in the validation set:

```
probability<-diag.logistic %>% predict(test.data,type="response")
```

```
> head(probability)
       14        18        49        52        66        67
0.2869525 0.3989560 0.4289635 0.5610842 0.4180050 0.6017610
```

Cross-Validation for the Logistic Regression in R

- We are now ready to calculate the accuracy of our trained model. To accomplish this, we translate all probabilities of success above to 0.5 to a 1 and otherwise to a 0, followed by tracking the number of correct predictions (1's correctly predicted as 1's and 0's correctly predicted as 0's).

```
predicted.classes <- ifelse(probability > 0.5, "1", "0")
```

```
> mean(predicted.classes == diag.new[-b])
[1] 0.65
```

Cross-Validation for the Logistic Regression in R

- We are now ready to calculate the accuracy of our trained model. To accomplish this, we translate all probabilities of success above to 0.5 to a 1 and otherwise to a 0, followed by tracking the number of correct predictions (1's correctly predicted as 1's and 0's correctly predicted as 0's).

```
predicted.classes <- ifelse(probability > 0.5, "1", "0")
```

```
> mean(predicted.classes == diag.new[-b])
[1] 0.65
```

- This model yields a 65% accuracy rate!