Logistic regression is a statistical approach used to describe the relationship of one or more independent variables to a dichotomous outcome. In this study you will be collecting data on three predictors of whether or not you and your classmates make a basket.

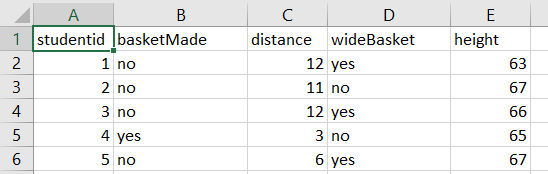
Step 1: Collect and enter data  
Use the table below to collect data on the basket-making skills of your class.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Outcome: Basket made?  (Yes, No) | Predictor 1: Distance from basket (feet) | Predictor 2: Wide  (Yes, No) | Predictor 3: Height (in inches) of the b-ball player |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |
| 11 |  |  |  |  |
| 12 |  |  |  |  |
| 13 |  |  |  |  |
| 14 |  |  |  |  |
| 15 |  |  |  |  |
| 16 |  |  |  |  |
| 17 |  |  |  |  |
| 18 |  |  |  |  |
| 19 |  |  |  |  |
| 20 |  |  |  |  |
| 21 |  |  |  |  |
| 22 |  |  |  |  |
| 23 |  |  |  |  |
| 24 |  |  |  |  |
| 25 |  |  |  |  |

**Follow the instructions below to answer the research question:**

*What predicts making a basket?*

**Enter the data into Excel**. Be sure to use easy-to-understand variable names in the first row of your Excel file, so it looks like this:



Save your Excel file as a **.csv** file in the folder you are using for class materials today. Use a file name with enough information in it to know what is in the file, like: **oct2019-baskets-data-alm.csv**

**If you do not have Excel,** follow the instructions below to open and save a new R Markdown document, thenenter the data directly into the R Markdown document in vectors and combine them using the data.frame() function, like this:

# enter the data

studentid <- c(1, 2, 3, 4, ....)

basketMade <- c("No", "No", "No", "Yes", ....)

distance <- c(12, 11, 12, 3, ....)

wideBasket <- c("Yes", "No", "Yes", "No", ....)

height <- c(63, 67, 66, 65, ....)

# combine into data frame

basketData <- data.frame(studentid, basketMade,

distance, wideBasket, height)

**Open R Studio and install the new package for today**

* odds.n.ends

**Open and save a new R Markdown document**

* Start a new R-Markdown file in RStudio by going to File 🡪 New File… 🡪 R Markdown
  + Give the new file a title and author, choose the Word file type (if you do not have Word installed, choose HTML since PDF requires other software installation to knit)
  + Click OK
* When the new file opens
  + Save it in the same folder as your data
  + Delete the default stuff underneath the second “---” in the new R Markdown file
* Add and run R code as needed to answer the questions below
  + R code chunks can be added by using “Insert” at the top of the code window and choosing **R** or by typing **```{r}** to start a chunk and **```** to end a chunk.

**Import and explore the data**

* Use **read.csv** to import data (you can skip this step if you typed your data directly into R Markdown)
* Try a stacked bar chart and a grouped bar chart to examine the relationship between making a basket and whether the basket was wide or narrow:

library(package = "tidyverse")

# stacked bar chart of basket

basketData %>%

ggplot(aes(x = wideBasket, fill = basketMade)) +

geom\_bar()

# grouped bar chart of basket

basketData %>%

ggplot(aes(x = wideBasket, fill = basketMade)) +

geom\_bar(position = "dodge")

Grouped bar plots tend to be easier to interpret than stacked. Copy/paste or sketch (loosely) the bar graph that appeared when you ran the grouped bar graph command. Does it look like it there were notably more baskets made for a wide or narrow basket?

**[Optional bonus]** Create a grouped bar chart with percentages *by group* instead of frequencies. Sketch the chart loosely and note any differences between this chart and the frequency version. The easiest (but-still-not-easy) way to do this is to summarize the data by group and use **geom\_col()** instead of the **geom\_bar()** geometry, like this:

# grouped bar chart with percents

basketData %>%

group\_by(basketMade, wideBasket) %>% # make groups of basket by wide

count() %>% # count how many in each group

group\_by(wideBasket) %>% # choose group to add to 100% in graph

mutate(group.perc = 100\*(n/sum(n))) %>% # compute percents for chosen variable

ggplot(aes(x = wideBasket, y = group.perc,

fill = basketMade)) +

geom\_col(position = "dodge")

To examine the relationship between making a basket and the distance and height variables, the bar graph should show the mean or median distance/height for baskets made or missed. There are two options for this:

1. Compute the mean within the geom\_bar() layer of the plot using the mean function for the variable on the y-axis, like this:

# barplot of average distance for baskets

basketData %>%

ggplot(aes(x = basketMade, y = distance)) +

geom\_bar(stat = "summary", fun.y = mean)

1. Compute the mean first for each group and then use geom\_col() instead of geom\_bar(); geom\_col() makes columns/bars based on summary data like percentages and means instead of raw data, like this:

# barplot of average distance for baskets

basketData %>%

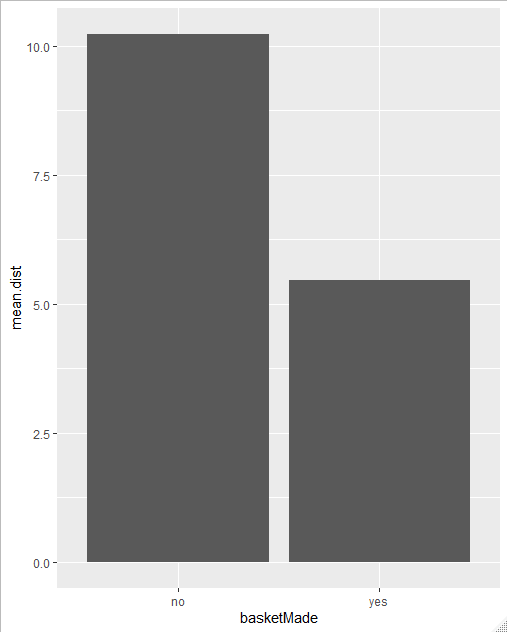
group\_by(basketMade) %>%

summarize(mean.dist = mean(x = distance)) %>%

ggplot(aes(x = basketMade, y = mean.dist)) +

geom\_col()

You should get something like this for each predictor:



Sketch the two graphs here and briefly describe any general patterns you found (e.g., taller people made fewer baskets).

**Exploring using numbers.** Often researchers will examine bivariate relationships in their data to understand and select variables to enter into their models. Examine bivariate relationships between making a basket and each of the predictors. Use **tableone** and add a **strata =**  argument with the basketMade variable so that you get separate columns for baskets made and missed:

# table of descriptive statistics

library(package = "tableone")

basket.table <- CreateTableOne(data = basketData,

vars = c('height', 'distance',

'wideBasket'),

strata = 'basketMade')

print(basket.table, showAllLevels = TRUE)

When **strata** = is used in **CreateTableOne**, the output has a p-value for each variable. For categorical variables, this p-value comes from a chi-squared test of the **strata** variable that is in the table columns and the variable shown in the row of the table. For continuous variables this p-value comes from a t-test or an alternate test if the data were non-normal.

Table 1. The association between making a basket, basket shape, distance from basket, and player height in a sample of \_\_\_ students at the Brown School (2019).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Basket made | Basket missed |  |
|  | **n (%)** | **n (%)** | **p** |
| Basket shape |  |  |  |
| Wide |  |  |  |
| Narrow |  |  |  |
|  | **m (s.d.)** | **m (s.d.)** | **p** |
| Distance from basket (feet) |  |  |  |
| Height of b-ball player (inches) |  |  |  |

Describe what you found in the bivariate analysis. How does it consistent with (or not consistent with) the graphs you used above to explore these relationships?

Next, write your null and alternate hypotheses for the relationships between each predictor and the outcome. Like in linear regression, a **null** hypothesis in logistic regression predicts that there is No Relationship or No Association between the predictor variable and the outcome. For example, a null hypothesis in a study where smoking (yes/no) is predicted by years of education might be:

H0: There is no relationship between education and smoking status (beduc=0).  
HA: There is a relationship between education and smoking status (beduc≠0).

Write your H0 and HA for the relationships between the outcome and each independent predictor.

Wide:

H0:

HA:

Distance:

H0:

HA:

Height:

H0:

HA:

Step 3: Running your regression & reporting results

Logistic regression is similar conceptually to linear regression, but instead of a continuous outcome, the outcome is binary (e.g., yes/no, alive/dead, smokes/doesn’t smoke). Since you can’t do anything linear with a binary outcome, the entire regression equation is transformed so that the left side of the regression equation represents the **probability** of the outcome, like this:

Here you would say: **the *probability* of y happening is predicted by the logistic model**. Notice you have all the same players as with linear regression (b0, b1, x, y) you’ve just transformed them to account for the binary distribution of the outcome.

In linear and logistic regression we often want to predict outcomes using more than one variable. For example, we might want to predict smoking status (yes/no) using education level and gender since we know both of these things might influence smoking. In our study we collected three variables, **distance, height,** and **wide**.

Before we conduct the analyses, it is important to know how the outcome variable is coded. The logistic regression function, **glm()**, in R will predict or explain the probability of one of the two categories of the outcome. We are usually interested in predicting the “Yes” category. To ensure that the model is predicting the category of interest, recode the variable so that it has the values of 1 and 0, where 1 is the category you are interested in predicting and 0 is the other category.

# recode to ensure outcome coded correctly

basketData.clean <- basketData %>%

mutate(basketMadeNum = recode(basketMade,

`Yes` = 1,

`No` = 0))

**CHECK YOUR RECODING!**

A quick table comparing your recoded variable with your original variable can ensure that you recoded the way you thought you were recoding:

# check recoding

table(basketData.clean$basketMadeNum,

basketData.clean$basketMade)

**Use logistic regression to see if these variables together predict the outcome:**

# logistic regression model

basketmodel <- glm(basketMadeNum ~ height + distance +

wideBasket,

data = basketData.clean,

family="binomial")

summary(basketmodel)

**This set of commands will result in output that looks something like this:**

Call:

glm(formula = basketMadeNum ~ height + distance + wideBasket,

family = "binomial", data = basketData.clean)

Deviance Residuals:

Min 1Q Median 3Q Max

-1.56707 -0.35190 0.02924 0.22013 1.99541

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 9.890230 14.552121 0.680 0.497

height -0.004696 0.202246 -0.023 0.981

distance -0.911541 0.450391 -2.024 0.043 \*

wideBasketyes -3.225124 2.218003 -1.454 0.146

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 27.526 on 19 degrees of freedom

Residual deviance: 10.999 on 16 degrees of freedom

AIC: 18.999

Number of Fisher Scoring iterations: 7

**The output will help you fill in the equation that looks like this:**

Where x1 , x2, x3 are your independent variables (**distance, height,** and **wide**).

In this case, the table indicates:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 9.890230 14.552121 0.680 0.497

height -0.004696 0.202246 -0.023 0.981

distance -0.911541 0.450391 -2.024 0.043 \*

wideBasketyes -3.225124 2.218003 -1.454 0.146

Where:

* **Estimate**: the coefficient, like b in linear regression
* **Std. Error**: standard error of estimate/coefficient/b
* **z value**: test statistic to determine if the coefficient is statistically significantly different from zero (b/se), like the t-statistic in linear
* **Pr(>|z|):** p-value for the z-statistic that signifies coefficient significance

Once you have this equation you can substitute in values for the predictors to find out the probability of a basket based on **distance, height,** and **wideBasket**. When the basket is wide, substitute a 1 for wideBasket; when the basket is narrow, substitute in a 0.

The table shows that distance has a significant relationship with the outcome (p = .04) but height does not (p = .98) and wide does not (p = .15). We would conclude that distance has a statistically significant relationship with making a basket, while height and wideBasket do not have statistically significant relationships with making a basket.

Write the equation predicting basket made from the numbers in your output:

Using R as a calculator, substitute in values for distance, height, and wide, to determine the predicted probabilities of making a basket for the different combinations below. In R, use “exp()” for the “e”. For example, e-2 would go into R as exp(-2). *Hold height constant at the mean height for the class.*

Wide basket from 5 feet:

Narrow basket from 5 feet:

Wide basket from 10 feet:

Narrow basket from 10 feet:

Look back at your bar graphs. Do the predicted probabilities you just calculated seem consistent with the patterns in the graphs? Why or why not?

Based on your output, would you reject your null hypotheses about the predictors? Why? Write a statement that rejects or fails to reject one of your null hypotheses (e.g., “I reject the null hypothesis that bdistance=0. The distance a person stands from the basket is negatively and significantly associated with making a basket (b = -.91; p = .01).”)

**Reporting regression results**

Like linear regression, you will report 3 things:

|  |  |  |
| --- | --- | --- |
|  | Linear regression | Logistic regression |
| 1. Predictor significance, magnitude, direction | b, t, p | OR, 95% CI |
| 1. Model significance | F, df, p | Chi-squared, df, p |
| 1. Model fit | Adj R-squared | % correctly predicted |

Unfortunately…most of the things you report with logistic are not part of the model summary from the **glm()** function.

Compute odds ratios, model significance, and model fit for your model using the **odds.n.ends** package.

1. **Predictor significance, magnitude, direction**

* Coefficients (b) are less useful in logistic regression compared to linear regression because they are difficult to interpret in light of the transformed outcome. Instead, odds ratios and confidence intervals are reported. If the odds ratio is greater than 1 for a predictor, this indicates the odds of the outcome increase with an increase in the predictor. If an odds ratio is below 1, the odds of the outcome decrease with an increase in the predictor.
* For example, say an odds ratio for a predictor is 1.5, we would interpret this as: For every one unit increase in predictor 1, the odds of the outcome increase by 50%. --- OR --- For every one unit increase in predictor 1, the odds of the outcome increase 1.5 times.
* Or, if an odds ratio is 0.6, we would interpret this as: For every one unit increase in the predictor, there is a 40% decrease in the odds of the outcome. (…because .6 is .4 or 40% below 1)

For binary predictors, odds ratios are very clear to interpret. For example, say that the variable with the 1.5 odds ratio was gender, with single = 0 and married = 1. A one unit increase in this type of variable is just a shift from single (0) to married (1), so the interpretation becomes more straightforward: The odds of the outcome is 50% higher in married people than single people.

Confidence intervals for odds ratios show the likely value of the odds ratio in the population. **Confidence intervals that cross 1 indicate non-significant odds ratios** since an odds ratio of 1 would be interpreted as the odds of the outcome increasing by 1 times. Non-significant odds ratios are not typically interpreted!

Use the **odds.n.ends** package to get all the information you need by entering the name of your logistic model into the **odds.n.ends()** function.

# compute odds ratios, model significance, model fit

library(package = "odds.n.ends")

odds.n.ends(x = basketmodel)

Interpret any significant odds ratios for **distance, height,** and **wideBasket**.

Distance:

Height:

Wide basket:

1. **Model significance**

In addition to the coefficient/odds ratio size and significance for the predictors, the **model significance** (like the F statistic) and model fit (like the R-squared) are reported. Model significance indicates whether the model is better than the baseline at explaining the outcome. The baseline in linear regression is the mean of the outcome, while the baseline in logistic is the probability of the outcome, which is making a basket (baskets made/total baskets).

Model significance is determined using the chi-squared from the **odds.n.ends(x = basketmodel)** output.

Circle “was” or “was not” and fill in the appropriate numbers in the statement of model significance:

My model **was/was not** statistically significantly better than the baseline percentage of baskets made at explaining the likelihood of making a basket (χ2( ) = ; p ).

1. **Model fit**

Finally, to measure **model fit**, you can examine how many of the baskets were correctly predicted by your model from the contingency tables in the **odds.n.ends(x = basketmodel)** output. They look like this:

$`Contingency tables (model fit): percent predicted`

Percent observed

Percent predicted 1 0 Sum

1 0.50 0.05 0.55

0 0.05 0.40 0.45

Sum 0.55 0.45 1.00

$`Contingency tables (model fit): frequency predicted`

Number observed

Number predicted 1 0 Sum

1 10 1 11

0 1 8 9

Sum 11 9 20

The columns show the observed data, the rows show what your model predicted. The percent that was correctly predicted by the model is known as the *Count R2.*

How many of the people in **your data set** did your model *correctly* predict *made a basket*? What percentage of those who made a basket were correctly predicted?

How many of the people in **your data set** did your model *correctly* predict *did NOT make a basket*? What percentage of those who did not make a basket were correctly predicted?

What percent of the people in your data set did your model correctly predict overall? This is the Count R2, one of the model fit options available for logistic regression.

A logistic model that is better at correctly identifying the negatives (0s) is said to be **specific**, while a model that is better at correctly identifying the positives (1s) is termed **sensitive**. Is your model more sensitive or more specific? What do you think might be the benefits or challenges related to this in your field? (e.g., if your model were predicting likelihood of Ebola given certain symptoms, what would you want your logistic model to be better at predicting, the 0s or the 1s?)

The end.

Make sure you save your R code so you can use it again!