Basketball Lab

Jenine K Harris

10/15/2019

# import basket data  
basketData <- read.csv("bball\_data.csv")  
  
# type in early data  
studentid <- c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17)  
basketMade <- c("Yes","No", "Yes","Yes", "Yes","No", "No",  
 "Yes","Yes","Yes","No", "No", "Yes", "Yes",  
 "Yes", "No", "No")  
distance <- c(3,9,3,1,4,10,11,8,12,10,9,12,11,8,3,11,7)  
wideBasket <- c('no','yes','no','yes','no',  
 'yes','no','yes','no','yes','no',  
 'yes','no','yes','no','yes','no')  
height <- c(62.5, 62, 64.5,65.5,64,68,62,68,64,68.5,60,69.3,63,62,6,66.5,64)  
  
#basketData <- data.frame(studentid, basketMade, distance,  
 # wideBasket, height)  
  
#write.csv(basketData, file = "basket-data-alm-early-section-2019.csv", row.names = FALSE)

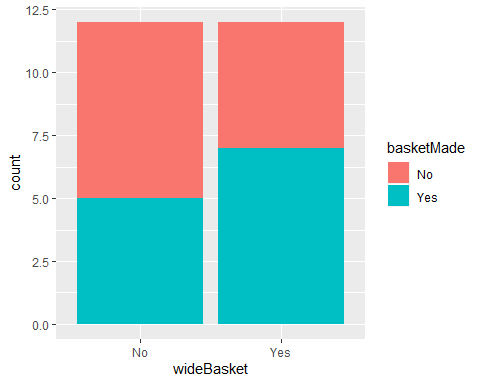
library(package = "tidyverse")

## -- Attaching packages --------------------------------------------------------- tidyverse 1.2.1 --

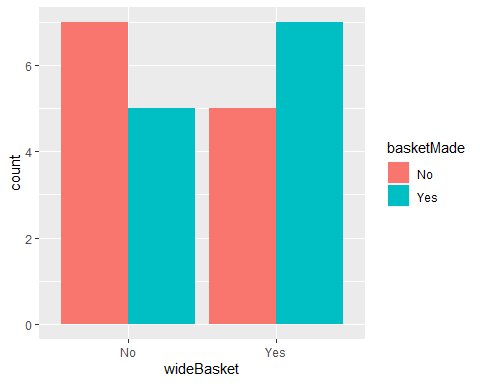
## v ggplot2 3.2.1 v purrr 0.3.2  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

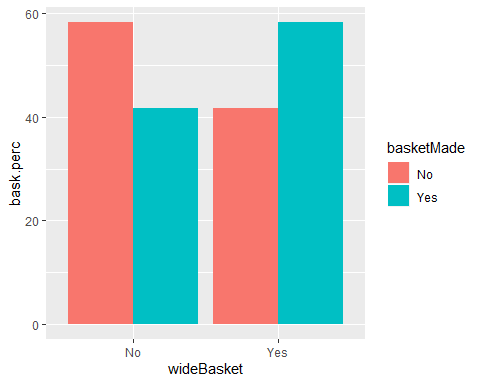
# stacked bar chart of basket   
basketData %>%  
 ggplot(aes(x = wideBasket, fill = basketMade)) +  
 geom\_bar()



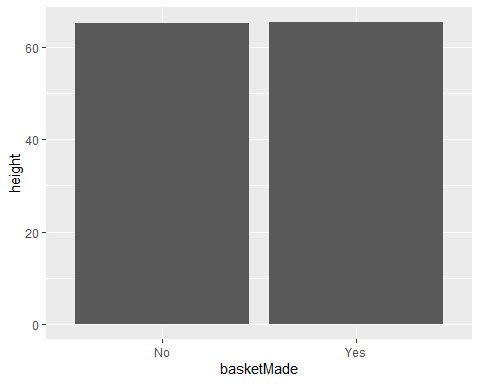
# grouped bar chart of basket   
basketData %>%  
 ggplot(aes(x = wideBasket, fill = basketMade)) +  
 geom\_bar(position = "dodge")



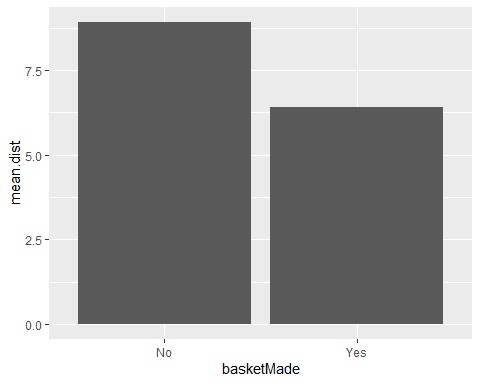
# grouped bar chart with percents   
basketData %>%   
 group\_by(basketMade, wideBasket) %>% # make groups of basket by wide  
 count() %>% # count how many in each group  
 group\_by(wideBasket) %>% # choose group to add to 100% in graph  
 mutate(bask.perc = 100\*(n/sum(n))) %>% # compute percents for chosen variable  
 ggplot(aes(x = wideBasket, y = bask.perc,   
 fill = basketMade)) +  
 geom\_col(position = "dodge")



basketData %>%  
 ggplot(aes(x = basketMade, y = height)) +  
 geom\_bar(stat = "summary", fun.y = mean)



basketData %>%  
 group\_by(basketMade) %>%  
 summarize(mean.dist = mean(x = distance)) %>%  
 ggplot(aes(x = basketMade, y = mean.dist)) +  
 geom\_col()



# table of descriptive statistics  
library(package = "tableone")  
basket.table <- CreateTableOne(data = basketData,   
 vars = c('height', 'distance',  
 'wideBasket'),  
 strata = 'basketMade')  
print(basket.table, showAllLevels = TRUE)

## Stratified by basketMade  
## level No Yes p test  
## n 12 12   
## height (mean (SD)) 65.28 (3.60) 65.42 (3.03) 0.918   
## distance (mean (SD)) 8.92 (2.02) 6.42 (3.70) 0.052   
## wideBasket (%) No 7 (58.3) 5 (41.7) 0.683   
## Yes 5 (41.7) 7 (58.3)

# recode to ensure outcome coded correctly  
basketData.clean <- basketData %>%  
 mutate(basketMadeNum = recode(basketMade,  
 `Yes` = 1,  
 `No` = 0))  
  
# check recoding  
table(basketData.clean$basketMadeNum,   
 basketData.clean$basketMade)

##   
## No Yes  
## 0 12 0  
## 1 0 12

#logistic regression model  
basketmodel <- glm(basketMadeNum ~ height + distance +   
 wideBasket,   
 data = basketData.clean,   
 family="binomial")  
summary(basketmodel)

##   
## Call:  
## glm(formula = basketMadeNum ~ height + distance + wideBasket,   
## family = "binomial", data = basketData.clean)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3824 -0.9770 -0.1575 0.8475 2.1231   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 6.71083 10.06755 0.667 0.5050   
## height -0.06863 0.15219 -0.451 0.6520   
## distance -0.37179 0.19592 -1.898 0.0577 .  
## wideBasketYes 1.36013 1.08001 1.259 0.2079   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 33.271 on 23 degrees of freedom  
## Residual deviance: 27.402 on 20 degrees of freedom  
## AIC: 35.402  
##   
## Number of Fisher Scoring iterations: 4

# predictions   
# wide from 5 feet  
1/(1+exp(-(6.71-.0686\*mean(basketData$height)-.372\*5+1.36\*1)))

## [1] 0.8490636

# narrow from 5 feet   
1/(1+exp(-(6.71-.0686\*mean(basketData$height)-.372\*5+1.36\*0)))

## [1] 0.5908006

# wide from 10 feet  
1/(1+exp(-(6.71-.0686\*mean(basketData$height)-.372\*10+1.36\*1)))

## [1] 0.4668676

# narrow from 10 feet  
1/(1+exp(-(6.71-.0686\*mean(basketData$height)-.372\*10+1.36\*0)))

## [1] 0.1835132

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

odds.n.ends(x = basketmodel)

## Waiting for profiling to be done...

## $`Logistic regression model significance`  
## Chi-squared d.f. p   
## 5.869 3.000 0.118   
##   
## $`Contingency tables (model fit): percent predicted`  
## Percent observed  
## Percent predicted 1 0 Sum  
## 1 0.37500000 0.04166667 0.41666667  
## 0 0.12500000 0.45833333 0.58333333  
## Sum 0.50000000 0.50000000 1.00000000  
##   
## $`Contingency tables (model fit): frequency predicted`  
## Number observed  
## Number predicted 1 0 Sum  
## 1 9 1 10  
## 0 3 11 14  
## Sum 12 12 24  
##   
## $`Predictor odds ratios and 95% CI`  
## OR 2.5 % 97.5 %  
## (Intercept) 821.2488642 2.385888e-06 2.183395e+12  
## height 0.9336755 6.727539e-01 1.263934e+00  
## distance 0.6894997 4.312339e-01 9.567461e-01  
## wideBasketYes 3.8966967 5.220473e-01 4.125429e+01  
##   
## $`Model sensitivity`  
## [1] 0.75  
##   
## $`Model specificity`  
## [1] 0.9166667