Lab 5

Kyle Mukire 505832075

2022-05-08

Contents


```
## Date last run: 2022-05-08

## Hello World!
```

Examples

Requires library xtable.

MLB 2021 Season Team-Game Data

```
## Read in our data
xdf <- read.csv("MLB_team_2021.csv", header=TRUE)

head(xdf, n=6)
```

```
##       date gameID                  team VorH bat_runs bat_homeRuns bat_strikeOuts
## 1 20210401 634615   Los Angeles Dodgers    V        5            0              6
## 2 20210401 634615      Colorado Rockies    H        8            0              4
## 3 20210401 634618  Arizona Diamondbacks    V        7            4             12
## 4 20210401 634618      San Diego Padres    H        8            2             10
## 5 20210401 634622        Atlanta Braves    V        2            1             10
## 6 20210401 634622 Philadelphia Phillies    H        3            0             13
##   bat_baseOnBalls pitch_runs pitch_homeRuns pitch_strikeOuts pitch_baseOnBalls
```

```
## 1              8          8          0           4           3
## 2              3          5          0           6           8
## 3              1          8          2          10           5
## 4              5          7          4          12           1
## 5              2          3          0          13           4
## 6              4          2          1          10           2
```

These included data sets were made by processing data obtained from MLB.

```
xbrks <- seq(-0.5, max(xdf[ , "bat_runs"])+0.5, by=1)
par(cex=0.65)
hist(xdf[ , "bat_runs"], breaks=xbrks, main="Team-Game Runs Scored, MLB 2021 Season")
```

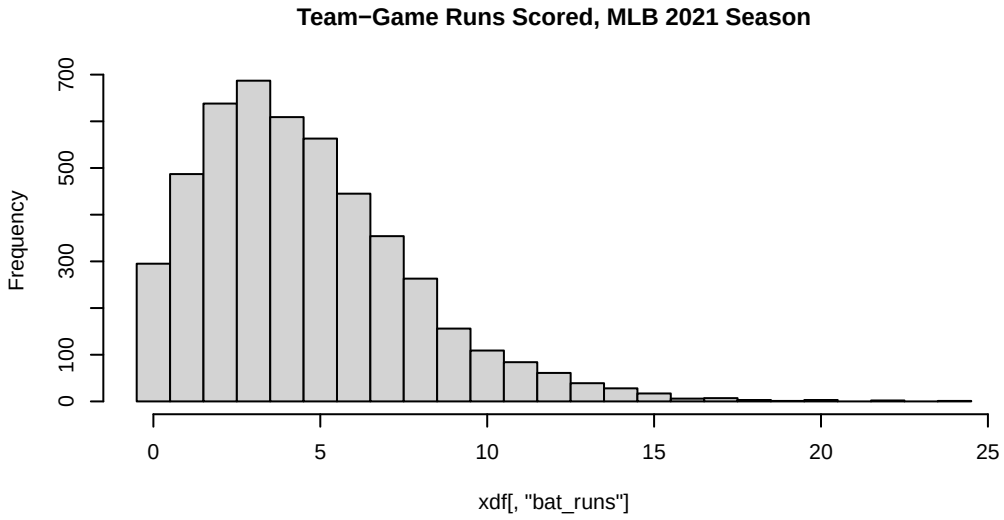

Figure 1: Distribution Team Runs scored by game.

Let's re-code runs scored (integer) into a two-level factor (a categorical attribute with two categories), "high" and "low".

If a team scores 5 or more runs, we'll call it "high"

```
xbruns <- xdf[ , "bat_runs"]

xcat_bruns <- c("low", "high")[ as.integer(xbruns >= 5) + 1 ]

xtbl <- table(xcat_bruns)[ c("low", "high") ]
xtbl
```

```
## xcat_bruns
##  low high
## 2716 2142
```

Do the same thing for game-team runs allowed:

```
xpruns <- xdf[ , "pitch_runs"]

xcat_pruns <- c("low", "high")[ as.integer(xpruns >= 5) + 1 ]

xtbl2 <- table(xcat_pruns)[ c("low", "high") ]
xtbl2
```

```
## xcat_pruns
##  low high
## 2716 2142
```

Of course, the distributions match.

Back to offense. Let's convert our frequency table into a proportions table.

```
xtbl_cellprops <- xtbl / sum(xtbl)

xtbl_cellprops
```

```
## xcat_bruns
##       low      high
## 0.5590778 0.4409222
```

Make a bar chart:

```
par(cex=0.65)
barplot(xtbl_cellprops, main="Dist. of Low or High Offensive Output")
```

Let's call the occurrence (event) that a team has high offensive output "A"

And let's call the occurrence (event) that a team allows low opponent offense "B"

$\mathsf{prop}[A] = 0.44092$

$\mathsf{prop}[B] = 0.55908$

Let's calculate $\mathsf{prop}[A \cap B]$ and $\mathsf{prop}[A|B]$.

This will take some careful sub-setting in `R`.

```
totalAandB <- sum( xcat_bruns == "high" & xcat_pruns == "low" )
totalAandB
```

```
## [1] 1210
```

```
propAandB <- totalAandB / length(xcat_bruns)
propAandB
```

```
## [1] 0.2490737
```

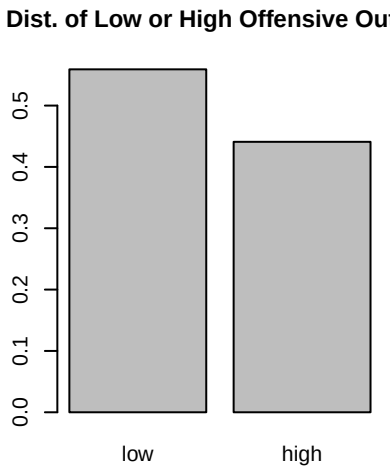

Figure 2: Distribution of Low or High Offensive Output.

```
#####

xmaskB <- xcat_pruns == "low"

xcat_bruns_givenB <- xcat_bruns[ xmaskB ]

propAgivenB <- sum( xcat_bruns_givenB == "high" ) / length(xcat_bruns_givenB)
propAgivenB
```

```
## [1] 0.4455081
```

$\mathsf{prop}[A \cap B] = 0.24907$

$\mathsf{prop}[A|B] = 0.44551$

Game-Level Independence Between Offensive & Defensive Performance

There is intuitive appeal that for most, if not all sports, at the game level, offensive and defensive performance are correlated.

To many fans, the success of the offense will be contagious and elevate the resolve and performance of the defense. And vice versa.

This question, that is, the dependence between game-level offensive and defensive performance is of considerable interest in sports analytics — especially handicapping. Yet this question, at least for baseball, is yet mostly unresolved.

Recall that if (and only if) event A and event B are independent, then $\mathsf{Pr}[A \cap B] = \mathsf{Pr}[A] \cdot \mathsf{Pr}[B]$

Empirically, within a data set, occurrence A and occurrence B are independent (uncorrelated) if and only if $\mathsf{prop}[A \cap B] = \mathsf{prop}[A] \cdot \mathsf{prop}[B]$

Within our 2021 MLB season team-game level data, we see that the joint occurrence of strong offense and strong defense is 0.24907.

The product of the two respective marginal proportions is 0.44092 times 0.55908 equals 0.24651.

We can see that the joint proportion is slightly greater than the product of the two marginal proportions. This, by the way, tells us the two are positively correlated.

```
xindicatorA <- c("low"=0, "high"=1)[ xcat_bruns ]

xindicatorB <- c("low"=1, "high"=0)[ xcat_pruns ]

cor(xindicatorA, xindicatorB)
```

```
## [1] 0.01040072
```

Of course our data is a subset of actual baseball games played; it is an infinitesimal subset of all possible baseball games that could be played.

We can use empirical probabilities to test whether this very small positive effect reveals a genuine pattern within "all" baseball games, or might just be the result of chance.

```
nn <- 2000 ### number of experiments

xsim_joint_prob <- numeric(nn)

for(i in 1:nn) {

  xsim_bruns <- sample(xcat_bruns) #### randomly shuffle team-game runs scored

  xsim_joint_prob[i] <- sum( xcat_pruns == "low"  &  xsim_bruns == "high" ) / length(xcat_bruns)

}
```

```
par(cex=0.65)
hist(xsim_joint_prob, main="Simulated Joint Proportions of Strong Offense & Defense")
abline(v=propAandB, lwd=2, col="009900")
```

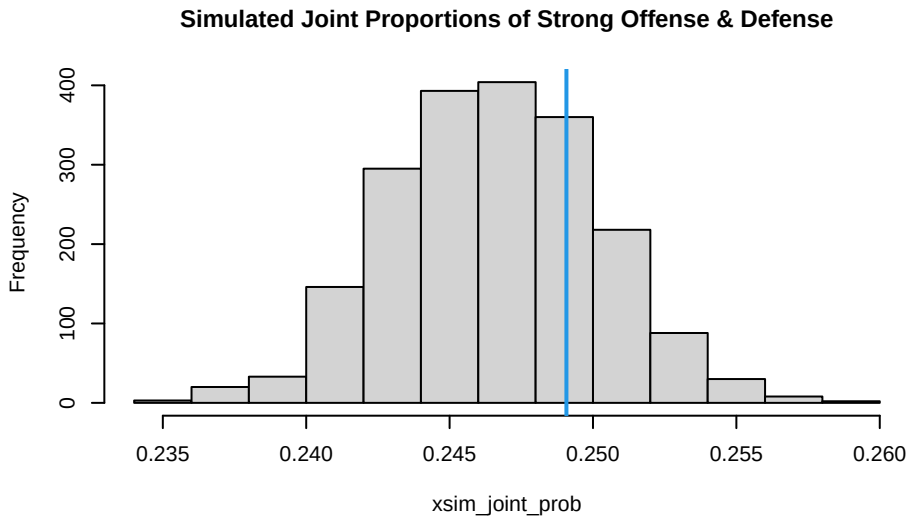

Figure 3: 2000 Simulations. Proportions of joint occurrence of Strong Offense & Defense under the assumption of independence. The blue line segment marks our actual observation.

Your Work

Make sure to edit the "author" information in the YAML header near the top to include your name and UID.

Complete/answer the following.

1 — Does our MLB team-game data represent "stacked" data? Why or why not? Yes, because each row is an observation and each column is a variable

2 — Read in the NHL data and create a new numeric variable shots allowed. Now, re-code team shots into a 2-level categorical attribute. A value of 33 or more is 'high', otherwise, 'low'. Do the same thing for shots allowed.

```
##### here's a head start. Un-comment the following lines
nhldf <- read.table("NHL_20202021_teamGame.tsv", sep="\t", header=TRUE)
head(nhldf, n=6)
```

```
##       date   season                team VorH team_goals team_pim team_shots
## 1 20210113 20202021 Pittsburgh Penguins   VT          3        6         34
## 2 20210113 20202021 Philadelphia Flyers   HT          6        6         27
## 3 20210113 20202021  Chicago Blackhawks   VT          1        8         23
## 4 20210113 20202021 Tampa Bay Lightning   HT          5        6         33
## 5 20210113 20202021  Montréal Canadiens   VT          4       13         32
## 6 20210113 20202021 Toronto Maple Leafs   HT          5       11         34
##   team_powerPlayGoals team_powerPlayOpportunities team_blocked team_takeaways
## 1                   1                           3           11              8
## 2                   2                           3           13              6
## 3                   1                           3            7              4
## 4                   2                           4           12              4
## 5                   2                           3           22              6
## 6                   2                           4           17              5
##   team_giveaways team_hits assists goals shots powerPlayGoals powerPlayAssists
## 1             10        23       3     3    34              1                 0
```

```
## 2             10        31      11     6    27              2                 4
## 3              1        14       2     1    23              1                 2
## 4              1        16      10     5    33              2                 4
## 5             13        32       8     4    32              2                 4
## 6              9        14       8     5    34              2                 3
##   penaltyMinutes faceOffWins faceoffTaken shortHandedGoals shortHandedAssists
## 1              4          21           50                0                 0
## 2              6          29           50                0                 0
## 3              8          28           57                0                 0
## 4              6          29           57                0                 0
## 5             13          29           63                0                 0
## 6              9          34           63                0                 0
```

```
mean(nhldf[ , "shots"])
```

```
## [1] 29.95507
```

```
N <- nrow(nhldf)
cross_ndx <- 1:N + rep( c(1, -1), N/2 )

xshots_allowed <- nhldf[ cross_ndx, "shots"]
mean(xshots_allowed)
```

```
## [1] 29.95507
```

```
xshots <- nhldf[ , "shots"]

xcat_shots <- c("low", "high")[ as.integer(xshots >= 33) + 1 ]

xcat_shots_allowed <- c("low", "high")[ as.integer(xshots_allowed >= 33) + 1 ]
```

3 — Call high shots occurrence C, and low shots allowed occurrence D. Calculate the following:

a: Proportion of C

b: Proportion of D

c: Proportion of C and D

d: Proportion of C given D

e: Confirm that the proportion of C given D times the proportion of D equals the proportion of C and D.

f: Write these results in-line and in context, and make sure to comment on the independence/dependence of C and D.

Independence: P(C)*P(D)=P(C and D)

a*b=c

```
sum(xcat_shots =='high')/nrow(nhldf)
```

```
## [1] 0.3335253
```

```
sum(xcat_shots_allowed =='low')/nrow(nhldf)
```

```
## [1] 0.6664747
```

```
sum(xcat_shots =='high'& xcat_shots_allowed =='low')/nrow(nhldf)
```

```
## [1] 0.2482719
```

```
sum(xcat_shots =='high'& xcat_shots_allowed =='low')/sum(xcat_shots_allowed =='low')
```

```
## [1] 0.3725151
```

The proportion of C is sum(xcat_shots == 'high')nrow(nldf)

The proportion of D is sum(xcat_shots_allowed == 'low')/nrow(nldf)

The proportion of C and D is sum(xcat_shots == 'high' & xcat_shots_allowed == 'low')/nrow(nldf)

The proportion of C givn D is sum(xcat_shots == 'high' & xcat_shots_allowed == 'low')/sum(xcat_shots_allowed == 'low'

4 — Repeat 2 and 3 above for penalty minutes. Define high as 8 or more. Call team high penalty minutes occurrence G, and low opponent penalty minutes occurrence H.

```
nhldf <- read.table("NHL_20202021_teamGame.tsv", sep="\t", header=TRUE)
head(nhldf, n=6)
```

```
##       date   season                team VorH team_goals team_pim team_shots
## 1 20210113 20202021 Pittsburgh Penguins   VT          3        6         34
## 2 20210113 20202021 Philadelphia Flyers   HT          6        6         27
## 3 20210113 20202021  Chicago Blackhawks   VT          1        8         23
## 4 20210113 20202021 Tampa Bay Lightning   HT          5        6         33
## 5 20210113 20202021  Montréal Canadiens   VT          4       13         32
## 6 20210113 20202021 Toronto Maple Leafs   HT          5       11         34
##   team_powerPlayGoals team_powerPlayOpportunities team_blocked team_takeaways
## 1                   1                           3           11              8
## 2                   2                           3           13              6
## 3                   1                           3            7              4
## 4                   2                           4           12              4
## 5                   2                           3           22              6
## 6                   2                           4           17              5
##   team_giveaways team_hits assists goals shots powerPlayGoals powerPlayAssists
## 1             10        23       3     3    34              1                 0
## 2             10        31      11     6    27              2                 4
## 3              1        14       2     1    23              1                 2
## 4              1        16      10     5    33              2                 4
## 5             13        32       8     4    32              2                 4
## 6              9        14       8     5    34              2                 3
##   penaltyMinutes faceOffWins faceoffTaken shortHandedGoals shortHandedAssists
## 1              4          21           50                0                 0
## 2              6          29           50                0                 0
## 3              8          28           57                0                 0
## 4              6          29           57                0                 0
## 5             13          29           63                0                 0
## 6              9          34           63                0                 0
```

```
mean(nhldf[ , "penaltyMinutes"])
```

```
## [1] 7.876152
```

```
N <- nrow(nhldf)
cross_ndx <- 1:N + rep( c(1, -1), N/2 )

xpenaltyMinutes <- nhldf[ cross_ndx, "penaltyMinutes"]
mean(xpenaltyMinutes)
```

```
## [1] 7.876152
```

```
xpenaltyMinutes <- nhldf[ , "penaltyMinutes"]

xcat_penaltyMinutes <- c("low", "high")[ as.integer(xpenaltyMinutes >= 8) + 1 ]
```

a: Proportion of G

b: Proportion of H

c: Proportion of G and H

d: Proportion of G given H

e: Confirm that the proportion of G given H times the proportion of H equals the proportion of G and H.

f: Write these results in-line and in context, and make sure to comment on the independence/dependence of C and D.

Independence: P(C)*P(D)=P(C and D)

a*b=c

```
sum(xcat_penaltyMinutes =='high')/nrow(nhldf)
```

```
## [1] 0.468318
```

```
sum(xcat_penaltyMinutes =='low')/nrow(nhldf)
```

```
## [1] 0.531682
```

```
sum(xcat_penaltyMinutes =='high'& xpenaltyMinutes =='low')/nrow(nhldf)
```

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

```
sum(xcat_penaltyMinutes =='high'& xpenaltyMinutes =='low')/sum(xcat_penaltyMinutes =='low')
```

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

The proportion of G is sum(xcat_penaltyMinutes == 'high)/nrow(nldf)

The proportion of H is sum(xcatpenaltyMinutes == 'low')/nrow(nldf)

The proportion of G and H is sum(xcat_penaltyMinutes == 'high' & xcat_shots_allowed == 'low')/nrow(nldf)

The proportion of G givn H is sum(xcat_penaltyMinutes == 'high' & xcatpenaltyMinutes == 'low')/sum(xcatpenaltyMinutes == 'low'

Extra Credit:

5 — Examine our simulation in the above Examples Section — in particular, the code sample(xcat_bruns). Explain in two or three sentences why this simulation assumes independence. This generation of data has no correlation to bruns. Although they are usually related because they are shots and total points allowed, here we are generating a sample from the same data as bruns. The simulation has nothing to do with, thus making it independent.