

Chapter 9

Nick Lauerman

Contents

| | |
|-------------------------|---|
| setup | 1 |
| Section 9.2 | 2 |
| Section 9.2.2 | 2 |
| Section 9.2.3 | 3 |
| Section 9.2.4 | 3 |
| Section 9.2.5 | 5 |
| Section 9.2.6 | 6 |
| Section 9.3 | 7 |
| Section 9.3.2 | 7 |
| Section 9.3.3 | 8 |
| Section 9.3.4 | 8 |
| Section 9.3.5 | 9 |
| Section 9.3.6 | 9 |

Graphs are included in a separate code block, when needed, to allow caching of data within the markdown environment allowing to reduced time to execute and compile the page.

setup

Libraries used:

```
# library(plotrix)
# library(ggplot2)
# library(lattice)
library(car)
library(plyr)
```

Setting the seed for random numbers. The authors didn't set a seed or share it if they had set on in the book, this has rendered most to the results in the simulations chapter non-reproducible. A seed is set here to for self consistency, allowing the results to be reproduced internally.

```
set.seed(894619)
```

Section 9.2

Section 9.2.2

```
data2011 <- read.csv("E:/Baseball/data/retro/all2011.csv",
                    header = FALSE)
fields <- read.csv("E:/Baseball/data/book/fields.csv")
names(data2011) <- fields[, "Header"]

data2011$HALF_INNING <- with(data2011,
                             paste(GAME_ID, INN_CT, BAT_HOME_ID))
data2011$RUNS.SCORED <- with(data2011,
                             (BAT_DEST_ID > 3) +
                             (RUN1_DEST_ID > 3) +
                             (RUN2_DEST_ID > 3) +
                             (RUN3_DEST_ID > 3))

get.state <- function(runner1, runner2, runner3, out){
  runners <- paste0(runner1, runner2, runner3)
  paste(runners, out)
}

RUNNER1 <- ifelse(as.character(data2011[, "BASE1_RUN_ID"]) == "", 0, 1)
RUNNER2 <- ifelse(as.character(data2011[, "BASE2_RUN_ID"]) == "", 0, 1)
RUNNER3 <- ifelse(as.character(data2011[, "BASE3_RUN_ID"]) == "", 0, 1)
data2011$STATE <- get.state(RUNNER1, RUNNER2, RUNNER3, data2011$OUTS_CT)

NRUNNER1 <- with(data2011,
                 as.numeric(RUN1_DEST_ID == 1 |
                             BAT_DEST_ID == 1))
NRUNNER2 <- with(data2011,
                 as.numeric(RUN1_DEST_ID == 2 |
                             RUN1_DEST_ID == 2 |
                             BAT_DEST_ID == 2))
NRUNNER3 <- with(data2011,
                 as.numeric(RUN1_DEST_ID == 3 |
                             RUN2_DEST_ID == 3 |
                             RUN3_DEST_ID == 3 |
                             BAT_DEST_ID == 3))
NOUTS <- with(data2011, OUTS_CT + EVENT_OUTS_CT)
data2011$NEW.STATE <- get.state(NRUNNER1, NRUNNER2, NRUNNER3, NOUTS)

data2011 <- subset(data2011, (STATE != NEW.STATE) | (RUNS.SCORED > 0))

require(plyr)
data.outs <- ddply(data2011, .(HALF_INNING),
                  summarize,
                  Outs.Inning = sum(EVENT_OUTS_CT))
data2011 <- merge(data2011, data.outs)
```

*# NOTE: the second subset command is modified from the one given in the book to
align with the goals of the two subset commands given the the text preceding*

```

# the code fragment.
data2011C <- subset(data2011, Outs.Inning == 3)
data2011C <- subset(data2011C, BAT_EVENT_FL == TRUE)

require(car)
data2011C$NEW.STATE <- recode(data2011C$NEW.STATE,
                             "c('000 3', '100 3','010 3','001 3',
                                '110 3','101 3','011 3','111 3') = '3' ")

```

Section 9.2.3

```

T.matrix <- with(data2011C,
                 table(STATE, NEW.STATE))
P.matrix <- prop.table(T.matrix, 1)

P.matrix <- rbind(P.matrix, c(rep(0, 24), 1))

P1 <- round(P.matrix["000 0", ], 3)
data.frame(Prob=P1[P1 > 0])

```

```

##          Prob
## 000 0 0.027
## 000 1 0.679
## 001 0 0.006
## 010 0 0.050
## 100 0 0.239

```

```

P2 <- round(P.matrix["010 2",], 3)
data.frame(Prob = P2[P2 > 0])

```

```

##          Prob
## 000 2 0.020
## 001 2 0.006
## 010 2 0.055
## 100 2 0.243
## 101 2 0.034
## 3    0.642

```

Section 9.2.4

```

count.runners.outs <- function(s){
  sum(as.numeric(strsplit(s,"")[[1]]),
      na.rm = TRUE)
}

runners.outs <- sapply(dimnames(T.matrix)[[1]],
                      count.runners.outs)[-25]
R <- outer(runners.outs + 1,
           runners.outs,

```

```

FUN = "-")
dimnames(R)[[1]] <- dimnames(T.matrix)[[1]][-25]
dimnames(R)[[2]] <- dimnames(T.matrix)[[1]][-25]

R <- cbind(R, rep(0,24))

simulate.half.inning <- function(P, R, start = 1){
  s <- start
  path <- NULL
  runs <- 0
  while(s < 25){
    s.new <- sample(1:25, 1, prob = P[s, ])
    path <- c(path, s.new)
    runs <- runs + R[s, s.new]
    s <- s.new
  }
  runs
}

RUNS <- replicate(10000,
                  simulate.half.inning(T.matrix, R))
table(RUNS)

```

```

## RUNS
##      0      1      2      3      4      5      6      7      8
## 6604 1917  801  396  164   75   31    9    3

```

```
sum(RUNS[RUNS >= 5]) / 10000
```

```
## [1] 0.0648
```

```
mean(RUNS)
```

```
## [1] 0.6011
```

```

RUNS.J <- function(j){
  mean(replicate(10000,
                simulate.half.inning(T.matrix, R, j)))
}

Runs.Expectancy <- sapply(1:24, RUNS.J)
Runs.Expectancy <- t(round(matrix(Runs.Expectancy, 3, 8), 2))
dimnames(Runs.Expectancy)[[2]] <- c("0 Outs", "1 Outs", "2 Outs")
dimnames(Runs.Expectancy)[[1]] <- c("000", "001", "010", "011", "100",
                                   "101", "110", "111")

Runs.Expectancy

```

```

##      0 Outs 1 Outs 2 Outs
## 000    0.60   0.31   0.10
## 001    1.50   0.97   0.32
## 010    1.46   1.12   0.43
## 011    2.42   1.84   0.66

```

```
## 100    1.05    0.61    0.21
## 101    1.98    1.27    0.47
## 110    1.90    1.26    0.41
## 111    2.72    1.88    0.68
```

```
Runs <- matrix(c(0.47, 0.25, 0.10, 1.45, 0.94, 0.32,
                 1.06, 0.65, 0.31, 1.93, 1.34, 0.54,
                 0.84, 0.50, 0.22, 1.75, 1.15, 0.49,
                 1.41, 0.87, 0.42, 2.17, 1.47, 0.76),
               8, 3, byrow = TRUE)
Runs - Runs.Expectancy
```

```
##      0 Outs 1 Outs 2 Outs
## 000 -0.13 -0.06  0.00
## 001 -0.05 -0.03  0.00
## 010 -0.40 -0.47 -0.12
## 011 -0.49 -0.50 -0.12
## 100 -0.21 -0.11  0.01
## 101 -0.23 -0.12  0.02
## 110 -0.49 -0.39  0.01
## 111 -0.55 -0.41  0.08
```

Section 9.2.5

```
P.matrix.3 <- P.matrix %*% P.matrix %*% P.matrix

sorted.P <- sort(round(P.matrix.3["000 0", ], 3),
                 decreasing = TRUE)
head(data.frame(Prob = sorted.P))
```

```
##      Prob
## 3      0.371
## 100 2 0.241
## 000 2 0.084
## 110 1 0.057
## 100 1 0.053
## 010 2 0.048
```

```
Q <- P.matrix[-25, -25]
N <- solve(diag(rep(1, 24)) - Q)

N.0000 <- round(N["000 0", ], 2)
head(data.frame(N = N.0000))
```

```
##      N
## 000 0 1.04
## 000 1 0.77
## 000 2 0.64
## 001 0 0.01
## 001 1 0.04
## 001 2 0.05
```

```
sum(N.0000)
```

```
## [1] 4.28
```

```
Length <- round(t(N %*% rep(1, 24)), 2)
data.frame(L= Length[1, 1:8])
```

```
##           L
## 000 0 4.27
## 000 1 2.87
## 000 2 1.47
## 001 0 4.36
## 001 1 2.96
## 001 2 1.51
## 010 0 4.29
## 010 1 2.92
```

Section 9.2.6

```
data2011C$HOME_TEAM_ID <- with(data2011C,
                                substr(GAME_ID, 1, 3))
data2011C$BATTING.TEAM <- with(data2011C,
                                ifelse(BAT_HOME_ID == 0,
                                        as.character(AWAY_TEAM_ID),
                                        as.character(HOME_TEAM_ID)))

Team.T <- with(data2011C,
                table(BATTING.TEAM,
                      STATE,
                      NEW.STATE))
d.state <- subset(data2011C, STATE == "100 2")
Team.T.S <- with(d.state,
                  table(BATTING.TEAM,
                        NEW.STATE))

Team.T.S
```

```
##           NEW.STATE
## BATTING.TEAM 000 2 001 2 010 2 011 2 100 2 101 2 110 2 3
##           ANA    11    3    7    8    0    16    56 253
##           ARI    11    4   13    2    0    14    73 240
##           ATL     7    2    4    7    0    23    68 273
##           BAL    13    2   10    7    0    24    67 278
##           BOS    10    4   14   14    1   21    77 251
##           CHA     8    0    6   10    0   18    80 238
##           CHN     9    4    9    6    0   10    70 275
##           CIN    14    0    6    9    0   23    76 273
##           CLE    11    4   12    6    0   25    68 233
##           COL     7    3    7    6    1   21    77 243
##           DET    12    4   12    4    0   14    86 271
##           FLO     7    1   11    8    0   19    85 261
```

| | | | | | | | | | |
|----|-----|----|---|----|----|---|----|----|-----|
| ## | HOU | 11 | 2 | 13 | 8 | 0 | 12 | 70 | 257 |
| ## | KCA | 2 | 7 | 12 | 17 | 0 | 16 | 78 | 251 |
| ## | LAN | 12 | 5 | 10 | 11 | 0 | 25 | 74 | 249 |
| ## | MIL | 10 | 2 | 10 | 7 | 0 | 24 | 67 | 285 |
| ## | MIN | 5 | 2 | 5 | 6 | 1 | 16 | 64 | 254 |
| ## | NYA | 13 | 2 | 6 | 3 | 0 | 13 | 71 | 271 |
| ## | NYN | 8 | 4 | 6 | 10 | 0 | 24 | 81 | 269 |
| ## | OAK | 10 | 0 | 7 | 7 | 0 | 14 | 75 | 291 |
| ## | PHI | 18 | 2 | 13 | 5 | 0 | 27 | 76 | 256 |
| ## | PIT | 6 | 5 | 8 | 9 | 0 | 14 | 70 | 255 |
| ## | SDN | 7 | 5 | 10 | 8 | 0 | 13 | 58 | 243 |
| ## | SEA | 10 | 1 | 5 | 8 | 0 | 22 | 69 | 255 |
| ## | SFN | 7 | 4 | 6 | 12 | 0 | 18 | 83 | 255 |
| ## | SLN | 6 | 0 | 7 | 4 | 0 | 20 | 75 | 282 |
| ## | TBA | 13 | 2 | 13 | 8 | 0 | 19 | 63 | 211 |
| ## | TEX | 11 | 5 | 16 | 6 | 0 | 20 | 67 | 268 |
| ## | TOR | 7 | 3 | 8 | 10 | 1 | 18 | 50 | 269 |
| ## | WAS | 9 | 1 | 5 | 10 | 0 | 25 | 60 | 243 |

Section 9.3

Section 9.3.2

```
make.schedule <- function(teams, k){
  n.teams <- length(teams)
  Home <- rep(gl(n.teams,
                n.teams,
                length = n.teams^2,
                labels = teams),
             k)
  Visitor <- rep(gl(n.teams,
                   1,
                   length = n.teams^2,
                   labels = teams),
                k)
  schedule <- data.frame(Home = Home,
                        Visitor = Visitor)
  subset(schedule, Home != Visitor)
}

NL <- c("ATL",
        "CHN",
        "CIN",
        "HOU",
        "LAN",
        "NYN",
        "PHI",
        "PIT",
        "SFN",
        "SLN")
AL <- c("BAL",
        "BOS",
```

```

      "CAL",
      "CHA",
      "CLE",
      "DET",
      "MIN",
      "NYA",
      "OAK",
      "WS2")
teams <- c(NL,AL)
league <- c(rep(1,10),
            rep(2,10))
schedule <- rbind(make.schedule(NL, 9),
                  make.schedule(AL, 9))

```

Section 9.3.3

```

s.talent <- 0.20
talents <- rnorm(20, 0, s.talent)
TAL <- data.frame(Team = teams,
                  League = league,
                  Talent = talents)
SCH <- merge(schedule,
             TAL,
             by.x = "Home",
             by.y = "Team")
names(SCH)[4] <- "Talent.Home"
SCH <- merge(SCH,
             TAL,
             by.x = "Visitor",
             by.y = "Team")
names(SCH)[6] <- "Talent.Visitor"
SCH$prob.Home <- with(SCH,
                      exp(Talent.Home)/(exp(Talent.Home) + exp(Talent.Visitor)))
head(SCH)

```

```

##   Visitor Home League.x Talent.Home League.y Talent.Visitor prob.Home
## 1    ATL  PHI         1  0.15497356         1    -0.178516  0.5826082
## 2    ATL  PIT         1 -0.22599570         1    -0.178516  0.4881323
## 3    ATL  SFN         1 -0.27148211         1    -0.178516  0.4767752
## 4    ATL  NYN         1  0.47231639         1    -0.178516  0.6571980
## 5    ATL  HOU         1  0.08080374         1    -0.178516  0.5644691
## 6    ATL  SLN         1 -0.07209257         1    -0.178516  0.5265808

```

Section 9.3.4

```

SCH$outcome <- with(SCH,
                   rbinom(nrow(SCH), 1, prob.Home))
SCH$winner <- with(SCH,
                   ifelse(outcome,

```



```

                                as.character(Home),
                                as.character(Visitor)))
head(SCH[ , c("Visitor",
              "Home",
              "prob.Home",
              "outcome",
              "winner")])

```

```

##   Visitor Home prob.Home outcome winner
## 1    ATL  PHI 0.5826082      1    PHI
## 2    ATL  PIT 0.4881323      1    PIT
## 3    ATL  SFN 0.4767752      1    SFN
## 4    ATL  NYN 0.6571980      1    NYN
## 5    ATL  HOU 0.5644691      1    HOU
## 6    ATL  SLN 0.5265808      0    ATL

```

```

wins <- table(SCH$winner)
WIN <- data.frame(Team = names(wins),
                  Wins = as.numeric(wins))
RESULTS <- merge(TAL, WIN)

```

Section 9.3.5

```

win.league <- function(RR, league){
  wins <- RR$Wins * (RR$League == league)
  MAX <- max(wins)
  if(sum(wins == MAX) > 1){
    prob <- exp(RR$Talent) * (wins == MAX)
    outcome <- c(rmultinom(1, 1, prob))
    RR$Winner.Lg <- RR$Winner.Lg + outcome
  }
  if(sum(wins == MAX) == 1){
    RR$Winner.Lg <- RR$Winner.Lg + as.numeric(wins == MAX)
  }
  RR
}
RESULTS$Winner.Lg <- 0
RESULTS$Winner.WS <- 0
for(j in 1:2){
  RESULTS <- win.league(RESULTS, j)
}
teams <- (1:20)[RESULTS$Winner.Lg == 1]
outcome <- c(rmultinom(1, 7, exp(RESULTS$Talent)[teams]))
winner <- teams[1] * (diff(outcome) < 0) + teams[2] * (diff(outcome) > 0)
RESULTS$Winner.WS[winner] <- 1

```

Section 9.3.6

```

one.simulation.68 <- function(s.talent){
  make.schedule <- function(teams, k){
    n.teams <- length(teams)
    Home <- rep(gl(n.teams,
                  n.teams,
                  length = n.teams^2,
                  labels = teams),
               k)
    Visitor <- rep(gl(n.teams,
                     1,
                     length = n.teams^2,
                     labels = teams),
                  k)
    schedule <- data.frame(Home = Home,
                          Visitor = Visitor)
    subset(schedule, Home != Visitor)
  }

  NL <- c("ATL",
          "CHN",
          "CIN",
          "HOU",
          "LAN",
          "NYN",
          "PHI",
          "PIT",
          "SFN",
          "SLN")
  AL <- c("BAL",
          "BOS",
          "CAL",
          "CHA",
          "CLE",
          "DET",
          "MIN",
          "NYA",
          "OAK",
          "WS2")
  teams <- c(NL,AL)
  league <- c(rep(1,10),
             rep(2,10))
  schedule <- rbind(make.schedule(NL, 9),
                   make.schedule(AL, 9))
  talents <- rnorm(20, 0, s.talent)
  TAL <- data.frame(Team = teams,
                   League = league,
                   Talent = talents)
  SCH <- merge(schedule,
              TAL,
              by.x = "Home",
              by.y = "Team")
  names(SCH)[4] <- "Talent.Home"
  SCH <- merge(SCH,

```

```

        TAL,
        by.x = "Visitor",
        by.y = "Team")
names(SCH)[6] <- "Talent.Visitor"
SCH$prob.Home <- with(SCH,
                      exp(Talent.Home)/(exp(Talent.Home) + exp(Talent.Visitor)))
SCH$outcome <- with(SCH,
                   rbinom(nrow(SCH), 1, prob.Home))
SCH$winner <- with(SCH,
                  ifelse(outcome,
                          as.character(Home),
                          as.character(Visitor)))
head(SCH[ , c("Visitor",
              "Home",
              "prob.Home",
              "outcome",
              "winner")])

wins <- table(SCH$winner)
WIN <- data.frame(Team = names(wins),
                  Wins = as.numeric(wins))
RESULTS <- merge(TAL, WIN)
win.league <- function(RR, league){
  wins <- RR$Wins * (RR$League == league)
  MAX <- max(wins)
  if(sum(wins == MAX) > 1){
    prob <- exp(RR$Talent) * (wins == MAX)
    outcome <- c(rmultinom(1, 1, prob))
    RR$Winner.Lg <- RR$Winner.Lg + outcome
  }
  if(sum(wins == MAX) == 1){
    RR$Winner.Lg <- RR$Winner.Lg + as.numeric(wins == MAX)
  }
  RR
}
RESULTS$Winner.Lg <- 0
RESULTS$Winner.WS <- 0
for(j in 1:2){
  RESULTS <- win.league(RESULTS, j)
}
teams <- (1:20)[RESULTS$Winner.Lg == 1]
outcome <- c(rmultinom(1, 7, exp(RESULTS$Talent)[teams]))
winner <- teams[1] * (diff(outcome) < 0) + teams[2] * (diff(outcome) > 0)
RESULTS$Winner.WS[winner] <- 1
return(RESULTS)
}
RESULTS <- one.simulation.68(0.20)
RESULTS

```

| ## | Team | League | Talent | Wins | Winner.Lg | Winner.WS |
|------|------|--------|---------------|------|-----------|-----------|
| ## 1 | ATL | 1 | -0.0313325935 | 64 | 0 | 0 |
| ## 2 | BAL | 2 | -0.0481668689 | 92 | 0 | 0 |
| ## 3 | BOS | 2 | -0.0573066080 | 81 | 0 | 0 |
| ## 4 | CAL | 2 | 0.0697354324 | 90 | 0 | 0 |

```
## 5   CHA      2  0.1260102394  98      1      1
## 6   CHN      1  0.1050667450  84      0      0
## 7   CIN      1 -0.1055061474  76      0      0
## 8   CLE      2 -0.2642395301  69      0      0
## 9   DET      2  0.0000797262  76      0      0
## 10  HOU      1 -0.0568444753  83      0      0
## 11  LAN      1  0.0003622449  87      0      0
## 12  MIN      2 -0.0076010452  78      0      0
## 13  NYA      2  0.1646464993  83      0      0
## 14  NYN      1  0.2379460148  90      0      0
## 15  OAK      2 -0.1433051785  76      0      0
## 16  PHI      1  0.3614376500  91      1      0
## 17  PIT      1 -0.1192595870  73      0      0
## 18  SFN      1  0.1274553732  77      0      0
## 19  SLN      1 -0.0469721817  85      0      0
## 20  WS2      2 -0.1546618505  67      0      0
```

```
display.standing <- function(RESULTS, league){
  Standings <- subset(RESULTS,
                     League == league,
                     select = c("Team",
                                "Wins"))
  Standings$Losses <- 162 - Standings$Wins
  Standings[order(Standings$Wins, decreasing = TRUE), ]
}
cbind(display.standing(RESULTS, 1),
      display.standing(RESULTS, 2))
```

```
##      Team Wins Losses Team Wins Losses
## 16  PHI   91    71  CHA   98    64
## 14  NYN   90    72  BAL   92    70
## 11  LAN   87    75  CAL   90    72
## 19  SLN   85    77  NYA   83    79
## 6   CHN   84    78  BOS   81    81
## 10  HOU   83    79  MIN   78    84
## 18  SFN   77    85  DET   76    86
## 7   CIN   76    86  OAK   76    86
## 17  PIT   73    89  CLE   69    93
## 1   ATL   64    98  WS2   67    95
```

```
with(RESULTS, as.character(Team[Winner.Lg == 1]))
```

```
## [1] "CHA" "PHI"
```

```
with(RESULTS, as.character(Team[Winner.WS == 1]))
```

```
## [1] "CHA"
```