Solution to Lab 2

Due on 09/23/2022 at 5:00 pm

Question 1 The 2014 and 2015 Royals surprised a lot of people when they seemingly came out of nowhere with back-to-back world series including a title in 2015. In this problem and in the next problem we will investigate aspects of weirdness surrounding these Royals teams. See this Foolish Baseball video, this Keith Law article, and this article about the failure of projection systems for background. In this problem you will construct a relevant dataset for analysis with the ultimate goal of describing just how unique these Royals were. Do the following:

- Construct a data frame which includes the following variables from the `Teams` data frame in the `Lahman` package: `yearID`, `teamID`, `AB`, `SO`, `H`, `HR`, `R`, `RA`, `W`, and `L`. Only keep seasons dating back to 1990, and remove the 1994, 1995, and 2020 seasons.

```
library(Lahman)
library(tidyverse)
library(doParallel)
data_1a <- Teams %>% filter(yearID >= 1990 & !(yearID %in% c(1994, 1995, 2020))) %>%
  select(yearID, teamID, AB, SO, H, HR, R, RA, W, L)

#team name agreement with baseball reference
data_1a$teamID = sub("CHN", "CHC", data_1a$teamID)
data_1a$teamID = sub("CHA", "CHW", data_1a$teamID)
data_1a$teamID = sub("KCA", "KCR", data_1a$teamID)
data_1a$teamID = sub("LAN", "LAD", data_1a$teamID)
data_1a$teamID = sub("ML4", "MIL", data_1a$teamID)
data_1a$teamID = sub("NYN", "NYM", data_1a$teamID)
data_1a$teamID = sub("NYA", "NYY", data_1a$teamID)
data_1a$teamID = sub("SDN", "SDP", data_1a$teamID)
data_1a$teamID = sub("SFN", "SFG", data_1a$teamID)
data_1a$teamID = sub("SLN", "STL", data_1a$teamID)
data_1a$teamID = sub("FLO", "FLA", data_1a$teamID)
data_1a$teamID = sub("WAS", "WSN", data_1a$teamID)
data_1a$teamID[data_1a$yearID >= 2008] = sub("TBA", "TBR", data_1a$teamID[data_1a$yearID >= 2008])
data_1a$teamID[data_1a$yearID < 2008] = sub("TBA", "TBD", data_1a$teamID[data_1a$yearID < 2008])

colnames(data_1a)[1:2] <- c('year_ID', 'team_ID')
```

- Run the code below to scrape data from baseball reference, and only keep seasons dating back to 1990, and remove the 1994, 1995, and 2020 seasons.

```
bwar_bat = readr::read_csv("https://www.baseball-reference.com/data/war_daily_bat.txt", na = "NULL")
bwar_pit = readr::read_csv("https://www.baseball-reference.com/data/war_daily_pitch.txt", na = "NULL")

data_1b_bat <- bwar_bat %>% filter(year_ID >= 1990 & !(year_ID %in% c(1994, 1995, 2020)))
data_1b_pit <- bwar_pit %>% filter(year_ID >= 1990 & !(year_ID %in% c(1994, 1995, 2020)))
```

- Obtain total team defensive WAR `WAR_def`, bullpen WAR, and base running runs `runs_br` for each year and add these quantities to the data frame that you previously constructed from the `Teams` data frame. Call these variables, respectively, `dWAR`, `penWAR`, `BRruns`.

```r
data_dWAR_BRruns <- data_1b_bat %>% group_by(year_ID, team_ID) %>%
  replace_na(list(WAR_def = 0, runs_br=0)) %>%
  summarise(dWAR = sum(WAR_def), BRruns = sum(runs_br))
```

```
## `summarise()` has grouped output by 'year_ID'. You can override using the
## `.groups` argument.
```

```r
data_penWAR <- data_1b_pit %>% mutate(bpWAR = IPouts_relief/IPouts*WAR) %>% group_by(year_ID, team_ID) %
  summarise(penWAR = sum(bpWAR))
```

```
## `summarise()` has grouped output by 'year_ID'. You can override using the
## `.groups` argument.
```

```r
data_1c <- merge(data_1a, merge(data_dWAR_BRruns, data_penWAR, by = c('year_ID', 'team_ID')), by = c('y
```

- The 2014-2015 Royals were known for elite base running, an elite bullpen, and elite defense. They were also known for not striking out and not hitting home runs. Add the following scaled variables separately for each season to the data frame that you constructed in the previous step:
  - `scaledSO = scale(SO / AB)`,
  - `scaledBA = scale(H/AB)`,
  - `scaledABpHR = scale(AB/HR)`,
  - `scaledpenWAR = scale(penWAR)`,
  - `scaleddWAR = scale(dWAR)`,
  - `scaledBRruns = scale(BRruns)`

```r
data_1d <- do.call(rbind, mclapply(unique(data_1c$year_ID), mc.cores = 7, FUN = function(xx){
  data_1c %>% filter(year_ID == xx) %>%
  mutate(scaledSO = scale(SO / AB)[,1], scaledBA = scale(H/AB)[,1],
         scaledABpHR = scale(AB/HR)[,1], scaledpenWAR = scale(penWAR)[,1],
         scaleddWAR = scale(dWAR)[,1], scaledBRruns = scale(BRruns)[,1])
}))
```

- Compute and add winning percentage `Wpct` to your data frame. Use an equation in your notes and linear regression to compute the optimal k so that `Wpct` is well-explained by `Wpytk` $= R^k/(R^k + RA^k)$. Add `Wpytk` and `residuals_pytk = Wpct - Wpytk` to your data frame.

compute the k

```r
data_wl_rra <- data_1d %>% mutate(logWratio = log(W/L), logRratio = log(R/RA))

k <- lm(logWratio ~ logRratio - 1, data = data_wl_rra)$coefficients
k
```

```
## logRratio
##  1.857948
```

```r
data_1e <- data_1d %>%
  mutate(Wpct = W/(W+L), Wpytk = R^k/(R^k + RA^k), residuals_pytk = Wpct - Wpytk)
```

- Display the rows of this data frame corresponding to the 2014-2015 Royals seasons.

```r
data_1e %>% filter(year_ID %in% c(2014, 2015) & team_ID == 'KCR')
```

```
##   year_ID team_ID   AB  SO    H  HR   R  RA  W  L dWAR BRruns    penWAR
## 1    2014     KCR 5545 985 1456  95 651 624 89 73 4.95   6.88  8.196305
## 2    2015     KCR 5575 973 1497 139 724 641 95 67 5.22   8.14 10.283681
```

```
##    scaledSO scaledBA scaledABpHR scaledpenWAR scaleddWAR scaledBRruns      Wpct
## 1 -2.396136 1.050699   2.5666452     1.483794   1.566083    1.0828922 0.5493827
## 2 -2.681442 1.722058   0.7595895     2.242565   1.730843    0.9626888 0.5864198
##       Wpytk residuals_pytk
## 1 0.5196652     0.02971753
## 2 0.5563169     0.03010290
```

Question 2 In this problem we will perform analyses that investigate strengths and peculiarities of the 2014-2015 Royals. Do the following:

- Fit and analyze a regression model of `residuals_pytk` on `penWAR`. Determine how many wins one would expect the Royals to obtain above their Pythagorean expectations on the basis of their bullpen.

```r
mod_2a <- lm(residuals_pytk ~ penWAR, data = data_1e)

win_above <- predict(mod_2a, data_1e %>%
                       filter(year_ID %in% c(2014, 2015) & team_ID == 'KCR')) * 162

win_above
```

```
##         1         2
## 0.5443617 0.8541848
```

- Total bullpen WAR is just one aspect of what made the 2014-2015 Royals what they were. We will now use k-means clustering implemented via the kmeans function to determine whether or not teams similar to the 2014-2015 Royals beat their Pythagorean expectations. Do the following with the number of clusters ranging from $k = 30, ..., 50$: 1) run kmeans on a dataset containing the six scaled variables that you previously constructed with k centers; 2) add the cluster assignments to the original dataset; 3) extract the average of `residuals_pytk` for the clusters containing the 2014 or 2015 Royals after removing the Royals from consideration. When finished, compute the average `residuals_pytk` value for the 2014 and 2015 Royals and then multiply this number by 162. This is the number of expected wins above/below their Pythagorean expectations that similar teams produced. Report this value and compare it with the 2014-2015 Royals.

```r
set.seed(1)
Royals_vs_similar <- do.call(rbind, mclapply(c(30:50), mc.cores = 7, FUN = function(xx){
  data_kmeans <- data_1e %>% select(scaledSO, scaledBA, scaledABpHR, scaledpenWAR, scaledpenWAR, scaled
  m <- kmeans(data_kmeans, xx)
  data_2b <- cbind(data_1e, cluster = m$cluster)

## cluster that contain 2014 royals
  index_2014 <- (data_2b %>% filter(year_ID == 2014 & team_ID == 'KCR'))$cluster
## cluster that contain 2015 royals
  index_2015 <- (data_2b %>% filter(year_ID == 2015 & team_ID == 'KCR'))$cluster

  similar_team <- data_2b %>% filter(cluster %in% c(index_2014, index_2015)) %>%
  filter(!(year_ID %in% c(2014, 2015) & team_ID == 'KCR'))

  c(Similar_win = mean(similar_team$residuals_pytk)*162,
    Royals_win = mean((data_2b %>% filter(year_ID %in% c(2014,2015) & team_ID == 'KCR'))$residuals_pytk
}))
colMeans(Royals_vs_similar)
```

```
## Similar_win  Royals_win
##  -0.5912003   4.8454548
```

- Add the `OPSscale` and `WHIPscale` variables that you computed in Question 1 of Lab 1 to the data frame. Run a regression with Wpct as the response variable and all eight scaled variables as predictors (you can drop terms if you want to). Does this model over/under estimate the success of the 2014-2015 Royals?

```
dat <- Teams %>%
    select(yearID, teamID, franchID, W, L, AB, H, X2B, X3B, HR, BB, HBP, SF,
                 HA, HRA, BBA, SOA, IPouts, FP, R, RA, G) %>%
    filter(yearID >= 1990 & !(yearID %in% c(1994, 1995, 2020))) %>%
    replace_na(list(HBP = 0, SF = 0)) %>%
    mutate(RD = (R - RA) / (W + L), X1B = H - (X2B + X3B + HR)) %>%
    mutate(OBP = (H + BB + HBP)/(AB + BB + HBP + SF)) %>%
    mutate(SLG = (X1B + 2*X2B + 3*X3B + 4*HR)/AB) %>%
    mutate(OPS = OBP + SLG) %>%
    mutate(WHIP = 3*(HA + BBA)/IPouts) %>%
    mutate(FIP = 3*(13*HRA + 3*BBA - 2*SOA)/IPouts)
avg_data <- dat %>%
group_by(yearID) %>%
summarize(AB = sum(AB), H = sum(H), BB = sum(BB), HBP = sum(HBP), X2B = sum(X2B),
          X3B = sum(X3B),HR = sum(HR), SF = sum(SF), HA = sum(HA), BBA = sum(BBA),
          IPouts = sum(IPouts),avgFP = mean(FP), X1B = sum(X1B)) %>%
  mutate(OBP = (H + BB + HBP)/(AB + BB + HBP + SF)) %>%
    mutate(SLG = (X1B + 2*X2B + 3*X3B + 4*HR)/AB) %>%
    mutate(avgOPS = OBP + SLG) %>%
    mutate(avgWHIP = 3*(HA + BBA)/IPouts) %>% ungroup() %>%
  select(yearID, avgWHIP, avgOPS, avgFP)
scale_data <- merge(dat, avg_data, by="yearID")
scale_data <- scale_data %>%
  mutate(WHIPscale = avgWHIP/WHIP) %>%
  mutate(OPSscale = OPS/avgOPS) %>%
  mutate(FPscale = avgFP/FP)

#team name agreement with baseball reference
scale_data$teamID = sub("CHN", "CHC", scale_data$teamID)
scale_data$teamID = sub("CHA", "CHW", scale_data$teamID)
scale_data$teamID = sub("KCA", "KCR", scale_data$teamID)
scale_data$teamID = sub("LAN", "LAD", scale_data$teamID)
scale_data$teamID = sub("ML4", "MIL", scale_data$teamID)
scale_data$teamID = sub("NYN", "NYM", scale_data$teamID)
scale_data$teamID = sub("NYA", "NYY", scale_data$teamID)
scale_data$teamID = sub("SDN", "SDP", scale_data$teamID)
scale_data$teamID = sub("SFN", "SFG", scale_data$teamID)
scale_data$teamID = sub("SLN", "STL", scale_data$teamID)
scale_data$teamID = sub("FLO", "FLA", scale_data$teamID)
scale_data$teamID = sub("WAS", "WSN", scale_data$teamID)
scale_data$teamID[scale_data$yearID >= 2008] = sub("TBA", "TBR", scale_data$teamID[scale_data$yearID >=
scale_data$teamID[scale_data$yearID < 2008] = sub("TBA", "TBD", scale_data$teamID[scale_data$yearID < 20


colnames(scale_data)[1:2] <- c('year_ID', 'team_ID')


data_2c <- merge(data_1e, scale_data %>% select(year_ID, team_ID, OPSscale, WHIPscale),
                 by = c('year_ID', 'team_ID'))
```

```r
mod_2c <- lm(Wpct ~ scaledSO + scaledBA + scaledABpHR + scaledpenWAR + scaledpenWAR + scaledBRruns + OPS
summary(mod_2c)
```

```
##
## Call:
## lm(formula = Wpct ~ scaledSO + scaledBA + scaledABpHR + scaledpenWAR +
##     scaledpenWAR + scaledBRruns + OPSscale + WHIPscale, data = data_2c)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.103673 -0.019926 -0.001149  0.019711  0.095149
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.0405560  0.0777079 -13.391  < 2e-16 ***
## scaledSO      0.0001022  0.0014142   0.072    0.942
## scaledBA      0.0005886  0.0028942   0.203    0.839
## scaledABpHR  -0.0001342  0.0022231  -0.060    0.952
## scaledpenWAR  0.0094923  0.0011712   8.105 1.85e-15 ***
## scaledBRruns  0.0043087  0.0010919   3.946 8.61e-05 ***
## OPSscale      0.8162658  0.0786702  10.376  < 2e-16 ***
## WHIPscale     0.7221597  0.0189272  38.155  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03086 on 844 degrees of freedom
## Multiple R-squared:  0.8167, Adjusted R-squared:  0.8152
## F-statistic: 537.2 on 7 and 844 DF,  p-value: < 2.2e-16
```

```r
## remove scaledSO, scaledBA and scaledABpHR

mod_2c <- lm(Wpct ~ scaledpenWAR + scaledpenWAR + scaledBRruns + OPSscale + WHIPscale, data = data_2c)
summary(mod_2c)
```

```
##
## Call:
## lm(formula = Wpct ~ scaledpenWAR + scaledpenWAR + scaledBRruns +
##     OPSscale + WHIPscale, data = data_2c)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.104104 -0.019841 -0.001055  0.019688  0.094860
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.051386   0.029571 -35.555  < 2e-16 ***
## scaledpenWAR  0.009518   0.001153   8.254 5.84e-16 ***
## scaledBRruns  0.004314   0.001089   3.962 8.06e-05 ***
## OPSscale      0.827995   0.024085  34.378  < 2e-16 ***
## WHIPscale     0.721268   0.018438  39.119  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03081 on 847 degrees of freedom
```

```
## Multiple R-squared:  0.8167, Adjusted R-squared:  0.8158
## F-statistic: 943.4 on 4 and 847 DF,  p-value: < 2.2e-16
```

```
predict(mod_2c, (data_2c %>% filter(year_ID %in% c(2014,2015) & team_ID == 'KCR')))
```

```
##         1         2
## 0.5136626 0.5448905
```

```
(data_2c %>% filter(year_ID %in% c(2014,2015) & team_ID == 'KCR'))$Wpct
```

```
## [1] 0.5493827 0.5864198
```

So, the Royals overperformed in each season from 2014 to 2015.

Question 3 Do the following:

- Select a period of your choice (at least 20 years) and fit the Pythagorean formula model (after finding the optimal exponent) to the run-differential, win-loss data.

```
data_3a <- Teams %>% filter(yearID >= 1990 & yearID <= 2009) %>%
  group_by(teamID) %>%
  summarize(Wpct = sum(W)/(sum(W)+sum(L)),
         logWratio = log(sum(W)/sum(L)),
         logRratio = log(sum(R)/sum(RA)), R = sum(R), RA = sum(RA))
mod_3a <- lm(logWratio ~ logRratio-1, data = data_3a)
k <- mod_3a$coefficients
data_3a <- data_3a %>% mutate(Wpct_pytk = R^k / (R^k + RA^k)) %>%
  mutate(residuals_pytk = Wpct - Wpct_pytk)
```

- On the basis of your fit in the previous part and the list of managers obtained from Retrosheet, compile a top 10 list of managers who most overperformed their Pythagorean winning percentage and a top 10 list of managers who most underperformed their Pythagorean winning percentage.

```
library(retrosheet)
#Getting all games from 1990-2009
data_3b = getRetrosheet(type = "game", year = 1990)
for(i in 1991:2009) {
  gm = getRetrosheet(type = "game", year = i)

  data_3b = rbind(data_3b, gm)
}

#Getting all manaager names and wins from 1990-2009
hm_gms <- data_3b %>%
      mutate(hW = ifelse(HmRuns > VisRuns, 1, 0), hL = ifelse(HmRuns < VisRuns, 1, 0)) %>% group_by(HmMg
      summarize(hW = sum(hW), hL = sum(hL), hR = sum(HmRuns), hRA = sum(VisRuns))

vis_gms <- data_3b %>%
      mutate(vW = ifelse(HmRuns < VisRuns, 1, 0), vL = ifelse(HmRuns > VisRuns, 1, 0)) %>% group_by(Vis
      summarize(vW = sum(vW), vL = sum(vL), vR = sum(VisRuns), vRA = sum(HmRuns))

colnames(hm_gms)[1] = "mgr"
colnames(vis_gms)[1] = "mgr"

hm_vis = merge(hm_gms, vis_gms, by = "mgr")

#get managers pythagorean winning percentages based on k calculated for this problem
top_mgrs<- hm_vis %>%
```

```r
    mutate(W = hW + vW,
           L = hL + vL,
           G = W + L,
           R = hR + vR,
           RA = hRA + vRA,
           Wpct = W / (W + L),
           Wpct_pytk = R^k / (R^k + RA^k),
           residuals_pytk = Wpct - Wpct_pytk) %>%
    select(mgr, W, L, G, R, RA, Wpct, Wpct_pytk, residuals_pytk)
```

The top 10 managers overperforming their Pythagorean Winning Percentage were:

```r
top_mgrs %>% arrange(desc(residuals_pytk)) %>% head(10)
```

```
##                  mgr   W   L   G    R   RA      Wpct Wpct_pytk residuals_pytk
## 1        Jamie Quirk   4   5   9   36   49 0.4444444 0.3501452     0.09429923
## 2         Dave Clark   4   9  13   41   78 0.3076923 0.2158486     0.09184369
## 3         Bill Doran   4   6  10   49   71 0.4000000 0.3221558     0.07784422
## 4      Don Wakamatsu  88  80 168  656  715 0.5238095 0.4569206     0.06688891
## 5   Red Schoendienst  13  11  24   85   88 0.5416667 0.4826137     0.05905301
## 6       Cecil Cooper 171 170 341 1456 1582 0.5014663 0.4584764     0.04298990
## 7         Dave Miley 125 164 289 1287 1594 0.4325260 0.3943381     0.03818782
## 8        Luis Pujols  55 100 155  562  824 0.3548387 0.3170051     0.03783357
## 9         Joe Nossek   3   5   8   25   35 0.3750000 0.3373989     0.03760105
## 10        Russ Nixon  25  40  65  267  365 0.3846154 0.3481636     0.03645175
```

The top 10 managers underperforming their Pythagorean Winning Percentage were:

```r
top_mgrs %>% arrange((residuals_pytk)) %>% head(10)
```

```
##               mgr  W  L   G   R  RA      Wpct Wpct_pytk residuals_pytk
## 1   Don Mattingly  1  1   2   8   4 0.5000000 0.8006461    -0.30064609
## 2     Gary Varsho  1  2   3  17  15 0.3333333 0.5624365    -0.22910315
## 3    Mike Cubbage  3  4   7  30  23 0.4285714 0.6301723    -0.20160088
## 4      Duffy Dyer  1  7   8  37  63 0.1250000 0.2558716    -0.13087159
## 5     Ken Griffey  2  2   4  10   8 0.5000000 0.6100658    -0.11006576
## 6   John Mizerock  5  8  13  49  51 0.3846154 0.4799498    -0.09533437
## 7    Cookie Rojas  2  2   4  13  11 0.5000000 0.5829955    -0.08299552
## 8      Bruce Kimm 33 45  78 357 361 0.4230769 0.4944129    -0.07133597
## 9      Bucky Dent 18 31  49 188 215 0.3673469 0.4331097    -0.06576274
## 10     Phil Regan 71 73 144 704 640 0.4930556 0.5476490    -0.05459348
```

The top 10 managers overperforming their Pythagorean Winning Percentage were:

```r
top_mgrs %>% arrange(desc(residuals_pytk)) %>% filter(G >= 300) %>% head(10)
```

```
##                  mgr    W    L    G    R   RA      Wpct Wpct_pytk residuals_pytk
## 1       Cecil Cooper  171  170  341 1456 1582 0.5014663 0.4584764     0.04298990
## 2  Marcel Lachemann  163  171  334 1681 1794 0.4880240 0.4674219     0.02060206
## 3     Fredi Gonzalez  242  245  487 2340 2444 0.4969199 0.4782078     0.01871207
## 4      Kevin Kennedy  309  273  582 3167 3067 0.5309278 0.5160837     0.01484416
## 5        Felipe Alou 1031 1018 2049 9153 9341 0.5031723 0.4898060     0.01336632
## 6       Trey Hillman  139  182  321 1364 1602 0.4330218 0.4200421     0.01297973
## 7          Ken Macha  448  362  810 3889 3588 0.5530864 0.5403084     0.01277806
## 8       Greg Riddoch  200  194  394 1543 1558 0.5076142 0.4951489     0.01246536
## 9    Bobby Valentine  747  664 1411 6482 6266 0.5294118 0.5169883     0.01242343
## 10         Hal McRae  399  473  872 3749 4180 0.4575688 0.4456459     0.01192290
```

The top 10 managers underperforming their Pythagorean Winning Percentage were:

```
top_mgrs %>% arrange((residuals_pytk)) %>% filter(G >= 300) %>% head(10)
```

```
##                mgr   W   L    G    R   RA      Wpct Wpct_pytk residuals_pytk
## 1     Dallas Green 229 283  512 2297 2360 0.4472656 0.4864350    -0.03916940
## 2       Ray Miller 157 167  324 1668 1600 0.4845679 0.5208594    -0.03629151
## 3    Alan Trammell 189 301  490 2169 2571 0.3857143 0.4155547    -0.02984037
## 4     John Gibbons 309 307  616 2896 2743 0.5016234 0.5271914    -0.02556805
## 5       Eric Wedge 559 573 1132 5593 5382 0.4938163 0.5192744    -0.02545815
## 6    Larry Dierker 435 348  783 4128 3533 0.5555556 0.5774220    -0.02186640
## 7  Tom Trebelhorn 203 229  432 1984 2021 0.4699074 0.4907354    -0.02082799
## 8      Davey Lopes 144 195  339 1529 1709 0.4247788 0.4444211    -0.01964234
## 9        Bob Geren 226 259  485 2146 2209 0.4659794 0.4854947    -0.01951536
## 10      Buddy Bell 514 715 1229 6038 6864 0.4182262 0.4360563    -0.01783013
```

Question 4 The first question on page 21 in Section 1.4.3 of Analyzing Baseball Data with R.

```
devtools::install_github("daviddalpiaz/bbd")
mlb_1998 = bbd::statcast(
    start = "1998-01-01",
    end = "1998-12-31",
    process = TRUE,
    names = TRUE,
    verbose = TRUE
)
```

```
#get Mark Mcgwire HR and opportunities with men on base
mcg <- mlb_1998 %>% filter(batter_name == "Mark McGwire")
mcg_HR <- mcg %>%
  filter(!is.na(on_1b) | !is.na(on_2b) | !is.na(on_3b)) %>%
  filter(events == "home_run") %>% nrow()
mcg_opp <- mcg %>%
  filter(!is.na(on_1b) | !is.na(on_2b) | !is.na(on_3b)) %>%
  filter(!is.na(events), events != "caught_stealing_2b") %>% nrow()

#get Sammy Sosa HR and opportunities with men on base
sosa <- mlb_1998 %>% filter(batter_name == "Sammy Sosa")
sosa_HR <- sosa %>%
  filter(!is.na(on_1b) | !is.na(on_2b) | !is.na(on_3b)) %>%
  filter(events == "home_run") %>% nrow()
sosa_opp <- sosa %>%
  filter(!is.na(on_1b) | !is.na(on_2b) | !is.na(on_3b)) %>%
  filter(!is.na(events), events != "caught_pstealing_2b") %>% nrow()
#data frame with both players' HR and opportunities
sosa_mcg <- data.frame("Opportunities" = c(sosa_opp, mcg_opp), "Home Runs" = c(sosa_HR, mcg_HR), row.na
sosa_mcg
```

```
##              Opportunities Home.Runs
## Sammy Sosa             371        29
## Mark McGwire           313        37
```