R Code for Teaching Statistics Using Baseball

J. Bentley

This document contains R code to reproduce the plots, tables, and statistics presented in Jim Albert's ***Teaching Statistics Using Baseball***. Some of the data is taken from sources other than the author's web site.

## Chapter One

Lahman has created a "project" at R-Forge (<http://lahman.r-forge.r-project.org/>) that describes the data tables that he created. Documentation for the data can be found at <http://lahman.r-forge.r-project.org/doc/>.

### Rickey Henderson using Lahman's data

The first step is to install and load the ***Lahman*** package.

 #install.packages("Lahman", dependencies = TRUE)
 require(Lahman)

## Loading required package: Lahman

The Lahman package contains a number of tables and a few functions that help subset the data. For now we will work with Henderson's batting data.

### Get the Batting data.frame ###
data(Batting)
### Figure out what it looks like ###
head(Batting)

## playerID yearID stint teamID lgID G AB R H X2B X3B HR RBI SB CS BB
## 1 abercda01 1871 1 TRO NA 1 4 0 0 0 0 0 0 0 0 0
## 2 addybo01 1871 1 RC1 NA 25 118 30 32 6 0 0 13 8 1 4
## 3 allisar01 1871 1 CL1 NA 29 137 28 40 4 5 0 19 3 1 2
## 4 allisdo01 1871 1 WS3 NA 27 133 28 44 10 2 2 27 1 1 0
## 5 ansonca01 1871 1 RC1 NA 25 120 29 39 11 3 0 16 6 2 2
## 6 armstbo01 1871 1 FW1 NA 12 49 9 11 2 1 0 5 0 1 0
## SO IBB HBP SH SF GIDP
## 1 0 NA NA NA NA NA
## 2 0 NA NA NA NA NA
## 3 5 NA NA NA NA NA
## 4 2 NA NA NA NA NA
## 5 1 NA NA NA NA NA
## 6 1 NA NA NA NA NA

### Some of the Lahman functions require plyr ###
#install.packages("plyr", dependencies=TRUE)
require(plyr)

## Loading required package: plyr

Albert's text works on Rickey Henderson's first 23 seasons. It would be nice if we could pull this information out of the Lahman data to confirm the values and compute a few statistics.

As a first step, we need to find Henderson's player information.

 data(Master)
 playerInfo(nameLast = "Henderson")

## playerID nameFirst nameLast
## 7236 hendebe01 Bernie Henderson
## 7237 hendebi01 Bill Henderson
## 7238 hendebi99 Bill Henderson
## 7239 hendeda01 Dave Henderson
## 7240 hendeed01 Ed Henderson
## 7241 hendeha01 Hardie Henderson
## 7242 hendeji01 Jim Henderson
## 7243 hendejo01 Joe Henderson
## 7244 hendeke01 Ken Henderson
## 7245 henderi01 Rickey Henderson
## 7246 hendero01 Rod Henderson
## 7247 hendest01 Steve Henderson

 playerInfo(nameFirst = "Rickey")

## playerID nameFirst nameLast
## 3002 clarkri01 Rickey Clark
## 3511 cradlri01 Rickey Cradle
## 7245 henderi01 Rickey Henderson
## 8675 keetori01 Rickey Keeton

 playerInfo("henderi01")

## playerID nameFirst nameLast
## 7245 henderi01 Rickey Henderson

Knowing that Rickey Henderson is "henderi01" means that we can subset his information.

 batting = battingStats()
 Henderson = batting[batting$playerID=="henderi01",]
 head(Henderson)

## playerID yearID stint teamID lgID G AB R H X2B X3B HR RBI
## 56760 henderi01 1979 1 OAK AL 89 351 49 96 13 3 1 26
## 57716 henderi01 1980 1 OAK AL 158 591 111 179 22 4 9 53
## 58659 henderi01 1981 1 OAK AL 108 423 89 135 18 7 6 35
## 59615 henderi01 1982 1 OAK AL 149 536 119 143 24 4 10 51
## 60620 henderi01 1983 1 OAK AL 145 513 105 150 25 7 9 48
## 61615 henderi01 1984 1 OAK AL 142 502 113 147 27 4 16 58
## SB CS BB SO IBB HBP SH SF GIDP BA PA TB SlugPct OBP OPS
## 56760 33 11 34 39 0 2 8 3 4 0.274 398 118 0.336 0.338 0.674
## 57716 100 26 117 54 7 5 6 3 6 0.303 722 236 0.399 0.420 0.819
## 58659 56 22 64 68 4 2 0 4 7 0.319 493 185 0.437 0.408 0.845
## 59615 130 42 116 94 1 2 0 2 5 0.267 656 205 0.382 0.398 0.780
## 60620 108 19 103 80 8 4 1 1 11 0.292 622 216 0.421 0.414 0.835
## 61615 66 18 86 81 1 5 1 3 7 0.293 597 230 0.458 0.399 0.857
## BABIP
## 56760 0.303
## 57716 0.320
## 58659 0.365
## 59615 0.306
## 60620 0.332
## 61615 0.321

These values correspond to the yearly values in Table 1.1 of Albert's book. Through a little manipulation, we can confirm the totals in the last row of Table 1.1.

 ### Figure out how many variables/columns there are in the batting data
 n = ncol(batting)
 ### Get the variable names themselves
 batting.names = names(batting)
 ### Since the first five columns are not numeric, we exclude them and use
 ### columns 6 through n=22. The apply function works on the second
 ### and computes the ""sum" for each column.
 Henderson.total = apply(battingStats(Henderson[Henderson$yearID<=2001,6:n]),2,sum)
 ### Calling the variable lets us view its contents.
 Henderson.total

## G AB R H X2B X3B HR
## 2979.000 10710.000 2248.000 3000.000 503.000 65.000 290.000
## RBI SB CS BB SO IBB HBP
## 1094.000 1395.000 333.000 2141.000 1631.000 61.000 93.000
## SH SF GIDP BA PA TB SlugPct
## 30.000 66.000 169.000 7.373 13040.000 4503.000 10.942
## OBP OPS BABIP BA.1 PA.1 TB.1 SlugPct.1
## 10.785 21.727 8.093 7.373 13040.000 4503.000 10.942
## OBP.1 OPS.1 BABIP.1
## 10.785 21.727 8.093

It is clear that summing statistics like batting average over years (**BA** = 7.373 > 1.000) is not the correct way to compute career statistics. A better approach would be to compute the statistics based upon counts summed across years.

 Henderson.total = apply(Henderson[Henderson$yearID<=2001,6:n],2,sum)
 ### Calling the variable lets us view its contents.
 Henderson.total

## G AB R H X2B X3B HR
## 2979.000 10710.000 2248.000 3000.000 503.000 65.000 290.000
## RBI SB CS BB SO IBB HBP
## 1094.000 1395.000 333.000 2141.000 1631.000 61.000 93.000
## SH SF GIDP BA PA TB SlugPct
## 30.000 66.000 169.000 7.373 13040.000 4503.000 10.942
## OBP OPS BABIP
## 10.785 21.727 8.093

 ### Let temp be Henderson's ID information (cols 1:5) bound to the totals
 ### computed above.
 temp = cbind(Henderson[1,1:5],matrix(Henderson.total,ncol=n-5))
 ### We can pull the column names from the names of the original batting
 ### data that were stored in batting.names
 names(temp) = batting.names
 temp

## playerID yearID stint teamID lgID G AB R H X2B X3B HR
## 56760 henderi01 1979 1 OAK AL 2979 10710 2248 3000 503 65 290
## RBI SB CS BB SO IBB HBP SH SF GIDP BA PA TB SlugPct
## 56760 1094 1395 333 2141 1631 61 93 30 66 169 7.373 13040 4503 10.942
## OBP OPS BABIP
## 56760 10.785 21.727 8.093

 ### We now compute the career batting stats using the career totals
 Henderson.total = battingStats(temp)
 ### We set the ID variables to missing (NA)
 Henderson.total[,2:5] = NA
 Henderson.total

## playerID yearID stint teamID lgID G AB R H X2B X3B HR
## 56760 henderi01 NA NA NA NA 2979 10710 2248 3000 503 65 290
## RBI SB CS BB SO IBB HBP SH SF GIDP BA PA TB SlugPct
## 56760 1094 1395 333 2141 1631 61 93 30 66 169 7.373 13040 4503 10.942
## OBP OPS BABIP BA.1 PA.1 TB.1 SlugPct.1 OBP.1 OPS.1 BABIP.1
## 56760 10.785 21.727 8.093 0.28 13040 4503 0.42 0.402 0.822 0.306

These values compare well with those presented in Table 1.1 of the text.

### Rickey Henderson Using Albert's Site

Jim Albert's school website has the data that he uses in the book. The data is stored as text files at <http://www-math.bgsu.edu/~albert/tsub/datasets.htm>. Reading the data from the site can be accomplished using RStudio's GUI interface (Tools -> Import Dataset -> From Web URL -> <http://www-math.bgsu.edu/~albert/tsub/datasets/case_1_1.txt> - ) or commands.

 ### Since the text file is tab delimited, we use sep="\t"
 ### The file also contains the variable/column names as the first line so
 ### we include header=TRUE. R sees \_ as an assignment so we use . instead.
 case.1.1 = read.csv("http://www-math.bgsu.edu/~albert/tsub/datasets/case\_1\_1.txt",
 header=TRUE, sep="\t")
 names(case.1.1)

## [1] "Season" "TM" "G" "AB" "R" "H" "X2B"
## [8] "X3B" "HR" "RBI" "BB" "SO" "SB" "CS"
## [15] "AVG" "OBP" "SLG" "OPS"

 case.1.1

## Season TM G AB R H X2B X3B HR RBI BB SO SB CS AVG
## 1 1979 Oak 89 351 49 96 13 3 1 26 34 39 33 11 0.274
## 2 1980 Oak 158 591 111 179 22 4 9 53 117 54 100 26 0.303
## 3 1981 Oak 108 423 89 135 18 7 6 35 64 68 56 22 0.319
## 4 1982 Oak 149 536 119 143 24 4 10 51 116 94 130 42 0.267
## 5 1983 Oak 145 513 105 150 25 7 9 48 103 80 108 19 0.292
## 6 1984 Oak 142 502 113 147 27 4 16 58 86 81 66 18 0.293
## 7 1985 NYY 143 547 146 172 28 5 24 72 99 65 80 10 0.314
## 8 1986 NYY 153 608 130 160 31 5 28 74 89 81 87 18 0.263
## 9 1987 NYY 95 358 78 104 17 3 17 37 80 52 41 8 0.291
## 10 1988 NYY 140 554 118 169 30 2 6 50 82 54 93 13 0.305
## 11 1989 NYY/Oak 150 541 113 148 26 3 12 57 126 68 77 14 0.274
## 12 1990 Oak 136 489 119 159 33 3 28 61 97 60 65 10 0.325
## 13 1991 Oak 134 470 105 126 17 1 18 57 98 73 58 18 0.268
## 14 1992 Oak 117 396 77 112 18 3 15 46 95 56 48 11 0.283
## 15 1993 Oak/Tor 134 481 114 139 22 2 21 59 120 65 53 8 0.289
## 16 1994 Oak 87 296 66 77 13 0 6 20 72 45 22 7 0.260
## 17 1995 Oak 112 407 67 122 31 1 9 54 72 66 32 10 0.300
## 18 1996 SD 148 465 110 112 17 2 9 29 125 90 37 15 0.241
## 19 1997 Ana/SD 120 403 84 100 14 0 8 34 97 85 45 8 0.248
## 20 1998 Oak 152 542 101 128 16 1 14 57 118 114 66 13 0.236
## 21 1999 NYM 121 438 89 138 30 0 12 42 82 82 37 14 0.315
## 22 2000 Sea/NYM 123 420 75 98 14 2 4 32 88 75 36 11 0.233
## 23 2001 SD 123 379 70 86 17 3 8 42 81 84 25 7 0.227
## OBP SLG OPS
## 1 0.338 0.336 0.674
## 2 0.420 0.399 0.819
## 3 0.408 0.437 0.845
## 4 0.398 0.382 0.780
## 5 0.414 0.421 0.835
## 6 0.399 0.458 0.857
## 7 0.419 0.516 0.935
## 8 0.358 0.469 0.827
## 9 0.423 0.497 0.920
## 10 0.394 0.399 0.793
## 11 0.411 0.399 0.810
## 12 0.439 0.577 1.016
## 13 0.400 0.423 0.823
## 14 0.426 0.457 0.883
## 15 0.432 0.474 0.906
## 16 0.411 0.365 0.776
## 17 0.407 0.447 0.854
## 18 0.410 0.344 0.754
## 19 0.400 0.342 0.742
## 20 0.376 0.347 0.723
## 21 0.423 0.466 0.889
## 22 0.368 0.305 0.673
## 23 0.366 0.351 0.717

The imported data are for Henderson's first 23 seasons (1979 through 2001). The imported information does not include totals. We can compute totals for some of the variables as above, but will need to compute other statistics (**AVG** through **OPS**) from scratch. The formulas for these statistics will be covered later.

 ### -(1:2) drops the season (Season) and team (TM) variables/columns ###
 case.1.1.tot = apply(case.1.1[,-(1:2)],2,sum)
 case.1.1.tot

## G AB R H X2B X3B HR
## 2979.000 10710.000 2248.000 3000.000 503.000 65.000 290.000
## RBI BB SO SB CS AVG OBP
## 1094.000 2141.000 1631.000 1395.000 333.000 6.420 9.240
## SLG OPS
## 9.611 18.851

Note that the sum of the yearly batting averages here is 6.420 which is not the same as the Lahman value of 7.373. What could account for this difference? (Hint: How would you handle data from a player that is traded mid-season?)

### Henderson's On Base Percentage

***TSUB*** provides a dotplot of Henderson's OBP during his first 23 years. We can create this plot in a couple of ways.

 ### First use base graphics
 stripchart(case.1.1$OBP)
 ### Use the lattice library. cex is the character size multiplier
 ### Using pch=1 chooses open circles which better show overlapped data.
 require(lattice)

## Loading required package: lattice



 dotplot(~OBP, data=case.1.1, cex=1.25, pch=1)



Both of these plots are similar to Figure 1.1 in the text.

### Henderson's Doubles versus Home Runs

The text compares Henderson's home run production (HR) against his doubles (X2B) using a scatterplot. There are multiple methods for generatting scatterplots in R. We will look at three.

 ### Using base R we get ###
 plot(case.1.1$X2B, case.1.1$HR, xlab="Doubles", ylab="Home Runs")



 ### Using lattice graphics
 require(lattice)
 xyplot(HR~X2B, data=case.1.1, xlab="Doubles", ylab="Home Runs", main="Henderson")



 xyplot(HR~X2B, data=case.1.1, xlab="Doubles", ylab="Home Runs", xlim=c(0:35), ylim=c(0:35), aspect=1)



 ### Use ggplot ###
 #install.packages("ggplot2", dependencies = TRUE)
 require(ggplot2)

## Loading required package: ggplot2

 p = ggplot(case.1.1, aes(X2B, HR)) ### Define data.frame and variables
 p + geom\_point() ### Request a point plot



 p + geom\_point() + theme\_bw() + xlab("Doubles") + ylab("Home Runs")



### Simulating Henderson's Spinner Using R

Hendersons hitting statistics for 1990 were

 spinner = Henderson[Henderson$yearID==1990,c("AB","H","X2B","X3B","HR","BB","IBB","HBP")]
 ### Singles equal hits less doubles, tripples, homers
 spinner$X1B = spinner$H - (spinner$X2B + spinner$X3B + spinner$HR)
 ### Outs equal at bats less hits, walks, hit by pitch
 spinner$Out = spinner$AB - (spinner$H + spinner$BB + spinner$IBB + spinner$HBP)
 ### Combine "walks"
 spinner$Walk = spinner$BB + spinner$IBB + spinner$HBP
 ### Check the counts
 spinner

## AB H X2B X3B HR BB IBB HBP X1B Out Walk
## 67830 489 159 33 3 28 97 2 4 95 227 103

 ### Probabilities are counts diveded by at bats
 spinner.prob = spinner/spinner$AB
 ### Keep only those variables we care about
 spinner.prob = spinner.prob[c("X1B","X2B","X3B","HR","Walk","Out")]
 spinner.prob

## X1B X2B X3B HR Walk Out
## 67830 0.194274 0.06748466 0.006134969 0.05725971 0.2106339 0.4642127

 ### Turn the data frame into a vector
 spinner.prob = as.vector(t(spinner.prob))
 spinner.prob

## [1] 0.194274029 0.067484663 0.006134969 0.057259714 0.210633947 0.464212679

 ### Create Figure 1.3
 pie(spinner.prob,labels=c("X1B","X2B","X3B","HR","Walk","Out"))



 sum(spinner.prob)

## [1] 1

Now, Albert has created a nice spinner, but he hasn't used it. Let's see what we get for Henderson if we let him bat (randomly) 179 times --- the number of at bats that he had in 2002).

 outcomes = c("X1B","X2B","X3B","HR","Walk","Out")
 spin2001 = sample(outcomes, 179, replace=TRUE, prob=spinner.prob)
 spin2001 = factor(spin2001,levels=c("X1B","X2B","X3B","HR","Walk","Out"))
 table(spin2001)

## spin2001
## X1B X2B X3B HR Walk Out
## 31 11 1 19 38 79

 post2001 = Henderson[Henderson$yearID>2001,c("yearID","AB","H","X2B","X3B","HR","BB","IBB","HBP")]
 post2001$Walk = post2001$BB + post2001$IBB + post2001$HBP
 post2001$X1b = post2001$H - (post2001$X2B + post2001$X3B + post2001$HR)
 post2001$Out = post2001$AB - (post2001$H + post2001$Walk)
 post2001

## yearID AB H X2B X3B HR BB IBB HBP Walk X1b Out
## 82466 2002 179 40 6 1 5 38 0 4 42 28 97
## 83801 2003 72 15 1 0 2 11 0 1 12 12 45

### Computing Basic Measures of Performance

We have already computed a few measures of a batter's performance using functions defined by others. In the previous section, we computed a few of these by hand. We can confirm that the functions and raw code provide the same results.

 ### Slugging percentage
 Hend7983 = Henderson[1:5,]
 Hend7983 = within.data.frame(Hend7983,{
 x1b = H - (X2B + X3B + HR)
 tb = 1\*x1b + 2\*X2B + 3\*X3B + 4\*HR
 slg = tb/AB
 })
 Hend7983[,c("x1b","TB","tb","SlugPct","slg")]

## x1b TB tb SlugPct slg
## 56760 79 118 118 0.336 0.3361823
## 57716 144 236 236 0.399 0.3993232
## 58659 104 185 185 0.437 0.4373522
## 59615 105 205 205 0.382 0.3824627
## 60620 109 216 216 0.421 0.4210526

Similar computations can be used to compute pitching statistics. We look at Orel Hershiser's AL years.

 ### Load the Lahan pitching data
 data(Pitching)
 ### Start a search for Hershier's player ID
 playerInfo("hers")

## playerID nameFirst nameLast
## 4958 etherse01 Seth Etherton
## 7396 hershea01 Earl Hersh
## 7397 hershfr01 Frank Hershey
## 7398 hershmi01 Mike Hershberger
## 7399 hershor01 Orel Hershiser
## 7400 hershwi01 Willard Hershberger

 ### Figure out what's in the pitching data.frame
 head(Pitching)

## playerID yearID stint teamID lgID W L G GS CG SHO SV IPouts H ER
## 1 bechtge01 1871 1 PH1 NA 1 2 3 3 2 0 0 78 43 23
## 2 brainas01 1871 1 WS3 NA 12 15 30 30 30 0 0 792 361 132
## 3 fergubo01 1871 1 NY2 NA 0 0 1 0 0 0 0 3 8 3
## 4 fishech01 1871 1 RC1 NA 4 16 24 24 22 1 0 639 295 103
## 5 fleetfr01 1871 1 NY2 NA 0 1 1 1 1 0 0 27 20 10
## 6 flowedi01 1871 1 TRO NA 0 0 1 0 0 0 0 3 1 0
## HR BB SO BAOpp ERA IBB WP HBP BK BFP GF R SH SF GIDP
## 1 0 11 1 NA 7.96 NA NA NA 0 NA NA 42 NA NA NA
## 2 4 37 13 NA 4.50 NA NA NA 0 NA NA 292 NA NA NA
## 3 0 0 0 NA 27.00 NA NA NA 0 NA NA 9 NA NA NA
## 4 3 31 15 NA 4.35 NA NA NA 0 NA NA 257 NA NA NA
## 5 0 3 0 NA 10.00 NA NA NA 0 NA NA 21 NA NA NA
## 6 0 0 0 NA 0.00 NA NA NA 0 NA NA 0 NA NA NA

 ### Subset out Hershiser's AL years
 HershAL = Pitching[Pitching$playerID == "hershor01" & Pitching$lgID == "AL",]
 ### Compute his effective "complete"" innings pitched based upon total outs
 HershAL$ip = HershAL$IPouts/3
 ### Convert innings to games
 HershAL$gp = HershAL$ip/9
 ### As is indicated in TSUB, ERA is earned runs over total games pitched
 HershAL$era = HershAL$ER/HershAL$gp
 ### Now compare
 HershAL[,c("yearID","teamID","IPouts","ER","ip","gp","ERA","era")]

## yearID teamID IPouts ER ip gp ERA era
## 30083 1995 CLE 502 72 167.3333 18.59259 3.87 3.872510
## 30673 1996 CLE 618 97 206.0000 22.88889 4.24 4.237864
## 31273 1997 CLE 586 97 195.3333 21.70370 4.47 4.469283

Not so amazingly, the internal value, **ERA**, is equal to our computed value, **era**.

## Chapter 2

The generation of the graphs and statistics that presented in the discussion of Case 2.1 is simplified by the use of R. The data from the 2001 MLB season is obtainable from Lahman.

 data(Teams)
 MLB2001 = subset.data.frame(Teams, subset=(yearID==2001 & lgID %in% c("AL","NL")))
 head(MLB2001)

## yearID lgID teamID franchID divID Rank G Ghome W L DivWin WCWin
## 2356 2001 AL ANA ANA W 3 162 81 75 87 N N
## 2357 2001 NL ARI ARI W 1 162 81 92 70 Y N
## 2358 2001 NL ATL ATL E 1 162 81 88 74 Y N
## 2359 2001 AL BAL BAL E 4 162 80 63 98 N N
## 2360 2001 AL BOS BOS E 2 161 81 82 79 N N
## 2361 2001 AL CHA CHW C 3 162 81 83 79 N N
## LgWin WSWin R AB H X2B X3B HR BB SO SB CS HBP SF RA ER
## 2356 N N 691 5551 1447 275 26 158 494 1001 116 52 77 53 730 671
## 2357 Y Y 818 5595 1494 284 35 208 587 1052 71 38 57 36 677 627
## 2358 N N 729 5498 1432 263 24 174 493 1039 85 46 45 52 643 578
## 2359 N N 687 5472 1359 262 24 136 514 989 133 53 77 49 829 744
## 2360 N N 772 5605 1493 316 29 198 520 1131 46 35 70 41 745 667
## 2361 N N 798 5464 1463 300 29 214 520 998 123 59 52 51 795 725
## ERA CG SHO SV IPouts HA HRA BBA SOA E DP FP
## 2356 4.20 6 1 43 4313 1452 168 525 947 103 142 0.983
## 2357 3.87 12 13 34 4379 1352 195 461 1297 84 148 0.986
## 2358 3.59 5 13 41 4342 1363 153 499 1133 103 133 0.983
## 2359 4.67 10 6 31 4297 1504 194 528 938 125 137 0.979
## 2360 4.15 3 9 48 4344 1412 146 544 1259 113 129 0.981
## 2361 4.55 8 9 51 4300 1465 181 500 921 118 149 0.981
## name park attendance BPF PPF
## 2356 Anaheim Angels Edison International Field 2000919 101 101
## 2357 Arizona Diamondbacks Bank One Ballpark 2736451 108 107
## 2358 Atlanta Braves Turner Field 2823530 103 102
## 2359 Baltimore Orioles Oriole Park at Camden Yards 3094841 95 96
## 2360 Boston Red Sox Fenway Park II 2625333 102 101
## 2361 Chicago White Sox Comiskey Park II 1766172 104 103
## teamIDBR teamIDlahman45 teamIDretro
## 2356 ANA ANA ANA
## 2357 ARI ARI ARI
## 2358 ATL ATL ATL
## 2359 BAL BAL BAL
## 2360 BOS BOS BOS
## 2361 CHW CHA CHA

 names(MLB2001)

## [1] "yearID" "lgID" "teamID" "franchID"
## [5] "divID" "Rank" "G" "Ghome"
## [9] "W" "L" "DivWin" "WCWin"
## [13] "LgWin" "WSWin" "R" "AB"
## [17] "H" "X2B" "X3B" "HR"
## [21] "BB" "SO" "SB" "CS"
## [25] "HBP" "SF" "RA" "ER"
## [29] "ERA" "CG" "SHO" "SV"
## [33] "IPouts" "HA" "HRA" "BBA"
## [37] "SOA" "E" "DP" "FP"
## [41] "name" "park" "attendance" "BPF"
## [45] "PPF" "teamIDBR" "teamIDlahman45" "teamIDretro"

### Case 2.1

While we could compute statistics like team batting average from hits and at bats, we can also import the data from the text.

 case.2.1 = read.csv("http://www-math.bgsu.edu/~albert/tsub/datasets/case\_2\_1.txt", header=TRUE, sep="\t")
 names(case.2.1)

## [1] "Team" "League" "G" "Avg" "OBP" "Slg" "AB"
## [8] "R" "H" "X2B" "X3B" "HR" "RBI" "BB"
## [15] "SO" "HBP"

 case.2.1

## Team League G Avg OBP Slg AB R H X2B X3B HR
## 1 Anaheim American 162 0.261 0.327 0.405 5551 691 1447 275 26 158
## 2 Baltimore American 161 0.248 0.319 0.380 5472 687 1359 262 24 136
## 3 Boston American 161 0.266 0.334 0.439 5605 772 1493 316 29 198
## 4 Chicago American 162 0.268 0.334 0.451 5464 798 1463 300 29 214
## 5 Cleveland American 162 0.278 0.350 0.458 5600 897 1559 294 37 212
## 6 Detroit American 162 0.260 0.320 0.409 5537 724 1439 291 60 139
## 7 Kansas City American 162 0.266 0.318 0.409 5643 729 1503 277 37 152
## 8 Minnesota American 162 0.272 0.337 0.433 5560 771 1514 328 38 164
## 9 New York American 161 0.267 0.334 0.435 5577 804 1488 289 20 203
## 10 Oakland American 162 0.264 0.345 0.439 5573 884 1469 334 22 199
## 11 Seattle American 162 0.288 0.360 0.445 5680 927 1637 310 38 169
## 12 Tampa Bay American 162 0.258 0.320 0.388 5524 672 1426 311 21 121
## 13 Texas American 162 0.275 0.344 0.471 5685 890 1566 326 23 246
## 14 Toronto American 162 0.263 0.325 0.430 5663 767 1489 287 36 195
## 15 Arizona National 162 0.267 0.341 0.442 5595 818 1494 284 35 208
## 16 Atlanta National 162 0.260 0.324 0.412 5498 729 1432 263 24 174
## 17 Chicago National 162 0.261 0.336 0.430 5406 777 1409 268 32 194
## 18 Cincinnati National 162 0.262 0.324 0.419 5583 735 1464 304 22 176
## 19 Colorado National 162 0.292 0.354 0.483 5690 923 1663 324 61 213
## 20 Florida National 162 0.264 0.326 0.423 5542 742 1461 325 30 166
## 21 Houston National 162 0.271 0.347 0.451 5528 847 1500 313 29 208
## 22 Los Angeles National 162 0.255 0.323 0.425 5493 758 1399 264 27 206
## 23 Milwaukee National 162 0.251 0.319 0.426 5488 740 1378 273 30 209
## 24 Montreal National 162 0.253 0.319 0.396 5379 670 1361 320 28 131
## 25 New York National 162 0.249 0.323 0.387 5459 642 1361 273 18 147
## 26 Philadelphia National 162 0.260 0.329 0.414 5497 746 1431 295 29 164
## 27 Pittsburgh National 162 0.247 0.313 0.393 5398 657 1333 256 25 161
## 28 San Diego National 162 0.252 0.336 0.399 5482 789 1379 273 26 161
## 29 San Francisco National 162 0.266 0.342 0.460 5612 799 1493 304 40 235
## 30 St. Louis National 162 0.270 0.339 0.441 5450 814 1469 274 32 199
## RBI BB SO HBP
## 1 662 494 1001 77
## 2 663 514 989 77
## 3 739 520 1131 70
## 4 770 520 998 52
## 5 868 577 1076 69
## 6 691 466 972 51
## 7 691 406 898 44
## 8 717 495 1083 64
## 9 774 519 1035 64
## 10 835 640 1021 88
## 11 881 614 989 62
## 12 645 456 1116 54
## 13 844 548 1093 75
## 14 728 470 1094 74
## 15 776 587 1052 57
## 16 696 493 1039 45
## 17 748 577 1077 66
## 18 690 468 1172 65
## 19 874 511 1027 61
## 20 713 470 1145 67
## 21 805 581 1119 89
## 22 714 519 1062 56
## 23 712 488 1399 72
## 24 622 478 1071 60
## 25 608 545 1062 65
## 26 708 551 1125 43
## 27 618 467 1106 67
## 28 753 678 1273 41
## 29 775 625 1090 50
## 30 768 529 1089 65

 MLB2001$avg = MLB2001$H/MLB2001$AB
 MLB2001 = MLB2001[order(MLB2001$lgID),]
 case.2.1[,c("Team","Avg")]

## Team Avg
## 1 Anaheim 0.261
## 2 Baltimore 0.248
## 3 Boston 0.266
## 4 Chicago 0.268
## 5 Cleveland 0.278
## 6 Detroit 0.260
## 7 Kansas City 0.266
## 8 Minnesota 0.272
## 9 New York 0.267
## 10 Oakland 0.264
## 11 Seattle 0.288
## 12 Tampa Bay 0.258
## 13 Texas 0.275
## 14 Toronto 0.263
## 15 Arizona 0.267
## 16 Atlanta 0.260
## 17 Chicago 0.261
## 18 Cincinnati 0.262
## 19 Colorado 0.292
## 20 Florida 0.264
## 21 Houston 0.271
## 22 Los Angeles 0.255
## 23 Milwaukee 0.251
## 24 Montreal 0.253
## 25 New York 0.249
## 26 Philadelphia 0.260
## 27 Pittsburgh 0.247
## 28 San Diego 0.252
## 29 San Francisco 0.266
## 30 St. Louis 0.270

 MLB2001[,c("teamID","avg")]

## teamID avg
## 2356 ANA 0.2606738
## 2359 BAL 0.2483553
## 2360 BOS 0.2663693
## 2361 CHA 0.2677526
## 2364 CLE 0.2783929
## 2366 DET 0.2598880
## 2369 KCA 0.2663477
## 2372 MIN 0.2723022
## 2374 NYA 0.2668101
## 2376 OAK 0.2635923
## 2380 SEA 0.2882042
## 2383 TBA 0.2581463
## 2384 TEX 0.2754617
## 2385 TOR 0.2629348
## 2357 ARI 0.2670241
## 2358 ATL 0.2604583
## 2362 CHN 0.2606363
## 2363 CIN 0.2622246
## 2365 COL 0.2922671
## 2367 FLO 0.2636232
## 2368 HOU 0.2713459
## 2370 LAN 0.2546878
## 2371 MIL 0.2510933
## 2373 MON 0.2530210
## 2375 NYN 0.2493131
## 2377 PHI 0.2603238
## 2378 PIT 0.2469433
## 2379 SDN 0.2515505
## 2381 SFN 0.2660371
## 2382 SLN 0.2695413

### Case 2.1

R makes it very easy to create a stemplot. A few options also make it easy to modify the stemplot so that it is easily interpretable.

 ### The default stem has too few stems
 stem(case.2.1$HR)

##
## The decimal point is 1 digit(s) to the right of the |
##
## 12 | 1169
## 14 | 728
## 16 | 11446946
## 18 | 45899
## 20 | 36889234
## 22 | 5
## 24 | 6

 ### We can split the stems in half to match the text
 stem(case.2.1$HR, 2)

##
## The decimal point is 1 digit(s) to the right of the |
##
## 12 | 1
## 13 | 169
## 14 | 7
## 15 | 28
## 16 | 114469
## 17 | 46
## 18 |
## 19 | 45899
## 20 | 36889
## 21 | 234
## 22 |
## 23 | 5
## 24 | 6

 ### Further division of stems is overkill
 stem(case.2.1$HR, 5)

##
## The decimal point is 1 digit(s) to the right of the |
##
## 12 | 1
## 12 |
## 13 | 1
## 13 | 69
## 14 |
## 14 | 7
## 15 | 2
## 15 | 8
## 16 | 1144
## 16 | 69
## 17 | 4
## 17 | 6
## 18 |
## 18 |
## 19 | 4
## 19 | 5899
## 20 | 3
## 20 | 6889
## 21 | 234
## 21 |
## 22 |
## 22 |
## 23 |
## 23 | 5
## 24 |
## 24 | 6

If we need a back-to-back stemplot we need to use the ***aplpack*** package.

 #install.packages("aplpack", dependencies = TRUE)
 require(aplpack)

## Loading required package: aplpack

## Loading required package: tcltk

 nlhr2001 = MLB2001$HR[MLB2001$lgID=="NL"]
 alhr2001 = MLB2001$HR[MLB2001$lgID=="AL"]
 stem.leaf.backback(nlhr2001,alhr2001)

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 1 | 2: represents 12, leaf unit: 1
## nlhr2001 alhr2001
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## | 12 |1 1
## 1 1| 13 |69 3
## 2 7| 14 |
## | 15 |28 5
## 6 6411| 16 |49 7
## 8 64| 17 |
## | 18 |
## 8 94| 19 |589 7
## 6 9886| 20 |3 4
## 2 3| 21 |24 3
## | 22 |
## 1 5| 23 |
## | 24 |6 1
## | 25 |
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## n: 16 14
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Team on base percentage is easily graphed. Computing statistics on the statistic is also simple.

 ### Using base R
 stem(case.2.1$OBP)

##
## The decimal point is 2 digit(s) to the left of the |
##
## 31 | 3
## 31 | 8999
## 32 | 003344
## 32 | 5679
## 33 | 444
## 33 | 6679
## 34 | 124
## 34 | 57
## 35 | 04
## 35 |
## 36 | 0

 ### Using aplpack
 stem.leaf(case.2.1$OBP)

## 1 | 2: represents 0.012
## leaf unit: 0.001
## n: 30
## 1 31\* | 3
## 5 31. | 8999
## 11 32\* | 003344
## 15 32. | 5679
## (3) 33\* | 444
## 12 33. | 6679
## 8 34\* | 124
## 5 34. | 57
## 3 35\* | 04
## 35. |
## 1 36\* | 0

 ### Find the median
 median(case.2.1$OBP)

## [1] 0.3315

 ### Find the quartiles
 quantile(case.2.1$OBP)[c(2,4)]

## 25% 75%
## 0.3230 0.3405

 ### Or...
 summary(case.2.1$OBP)

## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 0.3130 0.3230 0.3315 0.3321 0.3405 0.3600

### Case 2.2

The plots and statistics for Cal Ripken that are presented in Case 2.2 are variations on those already presented above. For this case we download the data from the author's site.

 ### Get the data
 case.2.2 = read.table("http://www-math.bgsu.edu/~albert/tsub/datasets/case\_2\_2.txt", header=TRUE)
 ### See what's in it
 head(case.2.2)

## YR G AB R H X2B X3B HR RBI TB AVG OBP SLG OPS
## 1 1981 23 39 1 5 0 0 0 0 5 0.128 0.150 0.128 0.278
## 2 1982 160 598 90 158 32 5 28 93 284 0.264 0.317 0.475 0.792
## 3 1983 162 663 121 211 47 2 27 102 343 0.318 0.371 0.517 0.888
## 4 1984 162 641 103 195 37 7 27 86 327 0.304 0.374 0.510 0.884
## 5 1985 161 642 116 181 32 5 26 110 301 0.282 0.347 0.469 0.816
## 6 1986 162 627 98 177 35 1 25 81 289 0.282 0.355 0.461 0.816

 ### Albert tossed Ripken's rookie season (first row)
 case.2.2 = case.2.2[-1,]
 ### Create Figure 2.5
 require(lattice)
 dotplot(~HR, data=case.2.2, cex=1.25, pch=1, xlab="Home Runs")



 ### Create Figure 2.6
 plot(case.2.2$YR, case.2.2$HR, xlab="Year", ylab="Home Runs")
 ### Create Figure 2.7
 plot(case.2.2$YR, case.2.2$HR, xlab="Year", ylab="Home Runs")
 abline(a=1446, b=-0.715)



 require(lattice)
 xyplot(HR~YR, data=case.2.2,
 panel=function(x, y, ...){
 panel.xyplot(x, y, ...)
 panel.abline(a=1446, b=-0.715)
 }
 )



 require(ggplot2)
 ggplot(case.2.2, aes(YR, HR)) + geom\_point() + theme\_bw() + xlab("Year") + ylab("Home Runs") + stat\_smooth(method="lm", se=FALSE)



 ### Create Figure 2.8
 stem(case.2.2$OPS, 2)

##
## The decimal point is 1 digit(s) to the left of the |
##
## 6 | 4
## 6 | 9
## 7 | 223
## 7 | 556679
## 8 | 01222
## 8 | 89
## 9 | 4
## 9 | 5

 ### Oops, Albert's software truncated to two decimals rather than rounded
 stem(trunc(100\*case.2.2$OPS)/100, 2)

##
## The decimal point is 1 digit(s) to the left of the |
##
## 6 | 3
## 6 | 8
## 7 | 12344
## 7 | 5669
## 8 | 00112
## 8 | 88
## 9 | 4
## 9 | 5

 ### Create Figure 2.9
 ggplot(case.2.2, aes(YR, OPS)) + geom\_point() + theme\_bw() + xlab("Year")



### Case 2.3

This case provides an opportunity to create a new statistic and to use some new graphical features.

 ### Get the data
 case.2.3 = read.table("http://www-math.bgsu.edu/~albert/tsub/datasets/case\_2\_3.txt", header=TRUE)
 head(case.2.3)

## YR TEAM LG W L PCT G SV IP H R ER SO TBB IBB ERA
## 1 1984 Bos AL 9 4 0.692 21 0 133.1 146 67 64 126 29 3 4.32
## 2 1985 Bos AL 7 5 0.583 15 0 98.1 83 38 36 74 37 0 3.29
## 3 1986 Bos AL 24 4 0.857 33 0 254.0 179 77 70 238 67 0 2.48
## 4 1987 Bos AL 20 9 0.690 36 0 281.2 248 100 93 256 83 4 2.97
## 5 1988 Bos AL 18 12 0.600 35 0 264.0 217 93 86 291 62 4 2.93
## 6 1989 Bos AL 17 11 0.607 35 0 253.1 215 101 88 230 93 5 3.13

 ### Compute the magic strikeout rate, SOR
 case.2.3$sor = (case.2.3$SO/case.2.3$IP) \* 9
 case.2.3$sor

## [1] 8.519910 6.788991 8.433071 8.193457 9.920455 8.178586 8.246383
## [8] 8.000738 7.603574 7.531381 8.883666 8.485714 9.549959 9.954545
## [15] 10.414176 7.836538 8.290054 8.709677

 ### Create Figure 2.10
 stem(case.2.3$sor,2)

##
## The decimal point is at the |
##
## 6 | 8
## 7 |
## 7 | 568
## 8 | 022234
## 8 | 5579
## 9 |
## 9 | 59
## 10 | 04

 ### Again, the figures differ by rounding versus truncating
 ### Generate Figure 2.11
 plot(case.2.3$YR, case.2.3$sor, xlab="Year", ylab="Strikeout Rate")
 abline(h=9)



 require(ggplot2)
 ggplot(case.2.3, aes(YR, sor)) + geom\_point() + xlab("Year") + ylab("Strikeout Rate") + geom\_hline(aes(yintercept=9))



Of course, the data for Figures 2.12 and 2.13 are not in case\_2\_3.txt. So, we need to get a little creative.

 ### The pacman package loads multiple packages and doesn't get mad if the
 ### package is already loaded.
 #install.packages("pacman")
 require(pacman)

## Loading required package: pacman

 ### We'll use SQL through dplyr to pull Clemen's ID from Lahman's Master
 p\_load(Lahman, dplyr, sqldf)
 data(Master)
 clemens.id = as.character(select(filter(Master, nameFirst=="Roger",
 nameLast=="Clemens"), retroID))
 ### Starting pitchers and runs can be found in RetroSheet
 p\_load(retrosheet)
 game2001 = getRetrosheet("game", 2001)
 ### Now use SQL to subset the NYA games and order them by date
 sqlnya2001 = sqldf("SELECT \* FROM game2001 WHERE (HmTm='NYA' OR VisTm='NYA') ORDER BY Date")
 ### Now figure out which ones Clemen's started in and which were home
 sqlnya2001$RCstart = (sqlnya2001$VisStPchID==clemens.id | sqlnya2001$HmStPchID==clemens.id)
 sqlnya2001$AtHome = sqlnya2001$HmTm=="NYA"
 sqlnya2001[sqlnya2001$AtHome,"runs"] = sqlnya2001[sqlnya2001$AtHome, "HmRuns"]
 sqlnya2001[!sqlnya2001$AtHome,"runs"] = sqlnya2001[!sqlnya2001$AtHome, "VisRuns"]
 ### Subset the data for Clemen's starts
 runs.rc.start = sqlnya2001$runs[sqlnya2001$RCstart]
 ### Text "table" of runs when RC started. Note that the 26th game is different
 runs.rc.start

## [1] 7 16 3 6 5 3 6 2 2 2 12 9 4 9 10 2 7 4 5 8 7 12 5
## [24] 4 10 9 7 3 4 6 0 1 4

 ### Create Figure 2.12
 stem(runs.rc.start)

##
## The decimal point is at the |
##
## 0 | 00
## 2 | 0000000
## 4 | 00000000
## 6 | 0000000
## 8 | 0000
## 10 | 00
## 12 | 00
## 14 |
## 16 | 0

 ### While the shape is right, the leaves and stems aren't pretty. Try aplpack.
 p\_load(aplpack)
 stem.leaf(runs.rc.start, unit=1)

## 1 | 2: represents 12
## leaf unit: 1
## n: 33
## 2 0\* | 01
## 9 t | 2222333
## (8) f | 44444555
## 16 s | 6667777
## 9 0. | 8999
## 5 1\* | 00
## 3 t | 22
## f |
## 1 s | 6

 ### Create Figure 2.13
 runs.rc.nostart = sqlnya2001$runs[!sqlnya2001$RCstart]
 stem.leaf.backback(runs.rc.start,runs.rc.nostart)

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 1 | 2: represents 1.2, leaf unit: 0.1
## runs.rc.start runs.rc.nostart
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 1 0| 0 |0000 4
## 2 0| 1 |0000000000000 17
## 6 0000| 2 |0000000000000000000000 39
## 9 000| 3 |00000000000000 53
## 14 00000| 4 |00000000000000000 (17)
## (3) 000| 5 |000000000000 58
## 16 000| 6 |0000000000000 46
## 13 0000| 7 |000000000000 33
## 9 0| 8 |000000 21
## 8 000| 9 |000000 15
## 5 00| 10 |0 9
## | 11 |0 8
## 3 00| 12 |0 7
## | 13 |0 6
## | 14 |000 5
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## HI: 16 HI: 15 16
## n: 33 128
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Case 2.4

TSUB now looks at attendance. As seen above, this information is included in Lahman's data. However, we will use Albert's data for convenience.

 ### Of course, the data for Figures 2.12 and 2.13 is not in case\_2\_4.txt.
 ### So, we continue...

 ### Get the case\_2\_4.txt data
 case.2.4 = read.delim("http://www-math.bgsu.edu/~albert/tsub/datasets/case\_2\_4.txt", header=TRUE, sep="\t")
 head(case.2.4)

## TEAM LEAGUE HOME.ATTENDANCE
## 1 Anaheim American 24702
## 2 Baltimore American 38661
## 3 Boston American 32413
## 4 Chicago American 22077
## 5 Cleveland American 39694
## 6 Detroit American 24087

 ### Plot Figure 2.14 using base R
 hist(case.2.4$HOME.ATTENDANCE, xlab="Average Home Attendance", main="")



 ### and the lattice package
 histogram(~HOME.ATTENDANCE, data=case.2.4, xlab="Average Home Attendance", breaks=7000+5000\*(0:8))



 histogram(~HOME.ATTENDANCE/1000, data=case.2.4, xlab="Average Home Attendance (thousands)", type="count")



 histogram(~HOME.ATTENDANCE|LEAGUE, data=case.2.4, xlab="Average Home Attendance", type="count", layout=c(1,2))



 ggplot(case.2.4,aes(HOME.ATTENDANCE)) + geom\_histogram() + xlab("Average Home Attendance") + stat\_bin(breaks=7000+5000\*(0:8)) + theme\_bw()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



 ggplot(case.2.4,aes(HOME.ATTENDANCE)) + geom\_histogram() + xlab("Average Home Attendance") + theme\_bw() + facet\_grid(LEAGUE~.) + stat\_bin(binwidth=5000)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



 ### Create Figure 2.15
 stem(case.2.4$HOME.ATTENDANCE,5)

##
## The decimal point is 3 digit(s) to the right of the |
##
## 7 | 9
## 8 |
## 9 |
## 10 |
## 11 |
## 12 |
## 13 |
## 14 |
## 15 | 8
## 16 | 0
## 17 |
## 18 |
## 19 | 0
## 20 |
## 21 |
## 22 | 138
## 23 | 68
## 24 | 17
## 25 |
## 26 | 3
## 27 |
## 28 |
## 29 | 7
## 30 | 8
## 31 |
## 32 | 48
## 33 | 8
## 34 | 79
## 35 | 029
## 36 |
## 37 | 2
## 38 | 7
## 39 | 07
## 40 | 89
## 41 |
## 42 |
## 43 | 3

### Case 2.5

Finally, we look at the use of the sacrafice bunt during the 2000 season.

 ### Get the case\_2\_5.txt data
 case.2.5 = read.delim("http://www-math.bgsu.edu/~albert/tsub/datasets/case\_2\_5.txt", header=TRUE, sep="\t")
 head(case.2.5)

## LEAGUE MANAGER ATTEMPTS SUCCESS.RATE FAVORITE.INNING
## 1 American Fregosi 45 75.6 1
## 2 American Garner 58 79.3 9
## 3 American Hargrove 36 80.6 8
## 4 American Howe 40 77.5 6
## 5 American Kelly 37 75.7 3
## 6 American C. Manual 59 84.7 1

 ### Figure 2.16 is created as above
 dotplot(~ATTEMPTS, data=case.2.5, cex=1.25, pch=1)



 ### Compute simple statistics
 summary(case.2.5$ATTEMPTS)

## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 22.00 58.25 74.00 71.93 85.50 115.00

 ### Since we are using the lattice package, Figure 2.17 is an easy fix
 dotplot(LEAGUE~ATTEMPTS, data=case.2.5, cex=1.25, pch=1)



 ### Compute associated statistics
 by(case.2.5$ATTEMPTS,case.2.5$LEAGUE,summary)

## case.2.5$LEAGUE: American
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 22.00 41.25 58.50 54.86 70.50 75.00
## --------------------------------------------------------
## case.2.5$LEAGUE: National
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 52.00 78.00 85.00 86.88 100.80 115.00

## Chapter 3

Chapter 3 moves from describing one characteristic to comparing a couple. R's developers have made this transition easy.

### Case 3.1

The plots and statistics for comparing Barry Bonds and Junior Griffey that are presented in Case 3.1 are, for the most part, variations on those already presented above. For this case we download the data from the author's site.

 ### Get the data
 case.3.1 = read.table("http://www-math.bgsu.edu/~albert/tsub/datasets/case\_3\_1.txt", header=TRUE)
### Table 3.1 is just the data.frame
case.3.1

## Year Player Age OBP SLG OPS
## 1 1986 Bonds 22 0.330 0.416 0.746
## 2 1987 Bonds 23 0.329 0.492 0.821
## 3 1988 Bonds 24 0.368 0.491 0.859
## 4 1989 Bonds 25 0.351 0.426 0.777
## 5 1990 Bonds 26 0.406 0.565 0.971
## 6 1991 Bonds 27 0.410 0.514 0.924
## 7 1992 Bonds 28 0.456 0.624 1.080
## 8 1993 Bonds 29 0.458 0.677 1.135
## 9 1994 Bonds 30 0.426 0.647 1.073
## 10 1995 Bonds 31 0.431 0.577 1.008
## 11 1996 Bonds 32 0.461 0.615 1.076
## 12 1997 Bonds 33 0.446 0.585 1.031
## 13 1998 Bonds 34 0.438 0.609 1.047
## 14 1999 Bonds 35 0.389 0.617 1.006
## 15 2000 Bonds 36 0.450 0.707 1.157
## 16 2001 Bonds 37 0.515 0.863 1.378
## 17 1989 Griffey 19 0.329 0.420 0.749
## 18 1990 Griffey 20 0.366 0.481 0.847
## 19 1991 Griffey 21 0.399 0.527 0.926
## 20 1992 Griffey 22 0.361 0.535 0.896
## 21 1993 Griffey 23 0.408 0.617 1.025
## 22 1994 Griffey 24 0.402 0.674 1.076
## 23 1995 Griffey 25 0.379 0.481 0.860
## 24 1996 Griffey 26 0.392 0.628 1.020
## 25 1997 Griffey 27 0.382 0.646 1.028
## 26 1998 Griffey 28 0.365 0.611 0.976
## 27 1999 Griffey 29 0.384 0.576 0.960
## 28 2000 Griffey 30 0.385 0.549 0.934
## 29 2001 Griffey 31 0.365 0.533 0.898

### Albert next creates Figure 3.1 a back-to-back stemplot
require(pacman)
p\_load(aplpack)
bonds = subset(case.3.1, Player=="Bonds")
bonds

## Year Player Age OBP SLG OPS
## 1 1986 Bonds 22 0.330 0.416 0.746
## 2 1987 Bonds 23 0.329 0.492 0.821
## 3 1988 Bonds 24 0.368 0.491 0.859
## 4 1989 Bonds 25 0.351 0.426 0.777
## 5 1990 Bonds 26 0.406 0.565 0.971
## 6 1991 Bonds 27 0.410 0.514 0.924
## 7 1992 Bonds 28 0.456 0.624 1.080
## 8 1993 Bonds 29 0.458 0.677 1.135
## 9 1994 Bonds 30 0.426 0.647 1.073
## 10 1995 Bonds 31 0.431 0.577 1.008
## 11 1996 Bonds 32 0.461 0.615 1.076
## 12 1997 Bonds 33 0.446 0.585 1.031
## 13 1998 Bonds 34 0.438 0.609 1.047
## 14 1999 Bonds 35 0.389 0.617 1.006
## 15 2000 Bonds 36 0.450 0.707 1.157
## 16 2001 Bonds 37 0.515 0.863 1.378

griffey = subset(case.3.1, Player=="Griffey")
griffey

## Year Player Age OBP SLG OPS
## 17 1989 Griffey 19 0.329 0.420 0.749
## 18 1990 Griffey 20 0.366 0.481 0.847
## 19 1991 Griffey 21 0.399 0.527 0.926
## 20 1992 Griffey 22 0.361 0.535 0.896
## 21 1993 Griffey 23 0.408 0.617 1.025
## 22 1994 Griffey 24 0.402 0.674 1.076
## 23 1995 Griffey 25 0.379 0.481 0.860
## 24 1996 Griffey 26 0.392 0.628 1.020
## 25 1997 Griffey 27 0.382 0.646 1.028
## 26 1998 Griffey 28 0.365 0.611 0.976
## 27 1999 Griffey 29 0.384 0.576 0.960
## 28 2000 Griffey 30 0.385 0.549 0.934
## 29 2001 Griffey 31 0.365 0.533 0.898

stem.leaf.backback(bonds$OPS, griffey$OPS)

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 1 | 2: represents 0.12, leaf unit: 0.01
## bonds$OPS griffey$OPS
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 1 4| 7\* |4 1
## 2 7| 7. |
## 3 2| 8\* |4 2
## 4 5| 8. |699 5
## 5 2| 9\* |23 (2)
## 6 7| 9. |67 6
## (4) 4300| 10\* |222 4
## 6 877| 10. |7 1
## 3 3| 11\* |
## 2 5| 11. |
## | 12\* |
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## HI: 1.378
## n: 16 13
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Hang on. There are only 16 and 13 obs when there should be 17 and 14.
### Looking at case.3.1 and Table 3.1 we note that 2002 is not in the txt file.
names(case.3.1)

## [1] "Year" "Player" "Age" "OBP" "SLG" "OPS"

### Add year 2002 from Table 3.1
### How many obs are there?
dim(case.3.1)

## [1] 29 6

### Copy the first two lines to the bottom of the data.frame
case.3.1 = rbind(case.3.1, case.3.1[1:2,])
### Now there are the right number of obs
dim(case.3.1)

## [1] 31 6

### Now fill the new observations with the 2002 data
case.3.1[30,]$Year=2002
case.3.1[30,]$Player="Bonds"
case.3.1[30,]$Age=38
case.3.1[30,]$OBP=0.582
case.3.1[30,]$SLG=0.799
case.3.1[30,]$OPS=1.381
case.3.1[31,]$Year=2002
case.3.1[31,]$Player="Griffey"
case.3.1[31,]$Age=32
case.3.1[31,]$OBP=0.358
case.3.1[31,]$SLG=0.426
case.3.1[31,]$OPS=0.784
### Check to make sure the last two obs were corrected
case.3.1[28:31,]

## Year Player Age OBP SLG OPS
## 28 2000 Griffey 30 0.385 0.549 0.934
## 29 2001 Griffey 31 0.365 0.533 0.898
## 30 2002 Bonds 38 0.582 0.799 1.381
## 31 2002 Griffey 32 0.358 0.426 0.784

### Now we can continue
bonds = subset(case.3.1, Player=="Bonds")
bonds

## Year Player Age OBP SLG OPS
## 1 1986 Bonds 22 0.330 0.416 0.746
## 2 1987 Bonds 23 0.329 0.492 0.821
## 3 1988 Bonds 24 0.368 0.491 0.859
## 4 1989 Bonds 25 0.351 0.426 0.777
## 5 1990 Bonds 26 0.406 0.565 0.971
## 6 1991 Bonds 27 0.410 0.514 0.924
## 7 1992 Bonds 28 0.456 0.624 1.080
## 8 1993 Bonds 29 0.458 0.677 1.135
## 9 1994 Bonds 30 0.426 0.647 1.073
## 10 1995 Bonds 31 0.431 0.577 1.008
## 11 1996 Bonds 32 0.461 0.615 1.076
## 12 1997 Bonds 33 0.446 0.585 1.031
## 13 1998 Bonds 34 0.438 0.609 1.047
## 14 1999 Bonds 35 0.389 0.617 1.006
## 15 2000 Bonds 36 0.450 0.707 1.157
## 16 2001 Bonds 37 0.515 0.863 1.378
## 30 2002 Bonds 38 0.582 0.799 1.381

griffey = subset(case.3.1, Player=="Griffey")
griffey

## Year Player Age OBP SLG OPS
## 17 1989 Griffey 19 0.329 0.420 0.749
## 18 1990 Griffey 20 0.366 0.481 0.847
## 19 1991 Griffey 21 0.399 0.527 0.926
## 20 1992 Griffey 22 0.361 0.535 0.896
## 21 1993 Griffey 23 0.408 0.617 1.025
## 22 1994 Griffey 24 0.402 0.674 1.076
## 23 1995 Griffey 25 0.379 0.481 0.860
## 24 1996 Griffey 26 0.392 0.628 1.020
## 25 1997 Griffey 27 0.382 0.646 1.028
## 26 1998 Griffey 28 0.365 0.611 0.976
## 27 1999 Griffey 29 0.384 0.576 0.960
## 28 2000 Griffey 30 0.385 0.549 0.934
## 29 2001 Griffey 31 0.365 0.533 0.898
## 31 2002 Griffey 32 0.358 0.426 0.784

stem.leaf.backback(bonds$OPS, griffey$OPS)

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 1 | 2: represents 0.12, leaf unit: 0.01
## bonds$OPS griffey$OPS
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 1 4| 7\* |4 1
## 2 7| 7. |8 2
## 3 2| 8\* |4 3
## 4 5| 8. |699 6
## 5 2| 9\* |23 (2)
## 6 7| 9. |67 6
## (4) 4300| 10\* |222 4
## 7 877| 10. |7 1
## 4 3| 11\* |
## 3 5| 11. |
## | 12\* |
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## HI: 1.378
## 1.381
## n: 17 14
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Five number summaries can be computed in a number of ways
### Median
m = sort(bonds$OPS)[(17+1)/2]
m

## [1] 1.031

median(bonds$OPS)

## [1] 1.031

### Quartiles
sort(bonds$OPS[bonds$OPS < m])

## [1] 0.746 0.777 0.821 0.859 0.924 0.971 1.006 1.008

ql = median(bonds$OPS[bonds$OPS < m])
ql

## [1] 0.8915

qu = median(bonds$OPS[bonds$OPS > m])
qu

## [1] 1.1075

quantile(bonds$OPS)[c(2,4)]

## 25% 75%
## 0.924 1.080

### Note that these values are not the same as those computed above.
### R does not go half way between the values when finding the medians of
### the values below/above the median. The median is 1.031 and there are
### eight values less than 1.031. The median of these values is somewhere
### between 0.859 and 0.924. Both methods give values in this interval.
sort(bonds$OPS)

## [1] 0.746 0.777 0.821 0.859 0.924 0.971 1.006 1.008 1.031 1.047 1.073
## [12] 1.076 1.080 1.135 1.157 1.378 1.381

### Min and max are easy
min(bonds$OPS)

## [1] 0.746

max(bonds$OPS)

## [1] 1.381

range(bonds$OPS)

## [1] 0.746 1.381

### Together, the statistics from above are the five-number-summary
quantile(bonds$OPS)

## 0% 25% 50% 75% 100%
## 0.746 0.924 1.031 1.080 1.381

summary(bonds$OPS) # Ignore the mean for now.

## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 0.746 0.924 1.031 1.028 1.080 1.381

###
### Albert computes a "step" that is then used to compute the u/l fences
### Note that the difference between steps found using Albert's method and
### the internal function is minimal. We'll use Albert's value.
step = 1.5\*(qu-ql)
step

## [1] 0.324

1.5 \* c(-1,1) %\*% quantile(bonds$OPS)[c(2,4)]

## [,1]
## [1,] 0.234

fu = qu + step
fu

## [1] 1.4315

fl = ql - step
fl

## [1] 0.5675

### So values greater than fu and less than fl are possible outliers
bonds$OPS[bonds$OPS > fu | bonds$OPS < fl]

## numeric(0)

sort(bonds$OPS)

## [1] 0.746 0.777 0.821 0.859 0.924 0.971 1.006 1.008 1.031 1.047 1.073
## [12] 1.076 1.080 1.135 1.157 1.378 1.381

### Now we generate boxplots
boxplot(OPS~Player, data=case.3.1, ylab="OPS")



p\_load(lattice)
bwplot(Player~OPS, data=case.3.1)



p\_load(ggplot2)
ggplot(case.3.1,aes(Player, OPS)) + geom\_boxplot()



ggplot(case.3.1,aes(Player, OPS)) + geom\_boxplot() + coord\_flip()



### Compare means
median(bonds$OPS)-median(griffey$OPS)

## [1] 0.101

### Create Figure 3.4
plot(case.3.1$Age, case.3.1$OPS, pch=as.numeric(case.3.1$Player))



xyplot(OPS~Age, data=case.3.1, group=Player,
 type=c("p","smooth"), span=0.5, pch=c(1,3),
 key=list(text=list(c("Bonds", "Griffey")),
 points=list(pch=c(1,3), col=c(13,14)),
 columns=2)
 )



### Case 3.2

The plots and statistics for comparing the pitchers Robin Roberts and Whitey Ford that are presented in Case 3.2 are, for the most part, variations on those already presented above. For this case we again download the data from the author's site.

 ### Get the data
 case.3.2 = read.table("http://www-math.bgsu.edu/~albert/tsub/datasets/case\_3\_2.txt",
 header=TRUE)
 ### Albert next creates Figure 3.5, a back-to-back stemplot
 require(pacman)
 p\_load(aplpack)
 roberts = subset(case.3.2, Player=="Roberts")
 roberts

## Year Player Age Tm W L G IP H ER HR BB SO ERA X.lgERA
## 1 1948 Roberts 21 PHI 7 9 20 146.7 148 52 10 61 84 3.19 3.96
## 2 1949 Roberts 22 PHI 15 15 43 226.7 229 93 15 75 95 3.69 3.96
## 3 1950 Roberts 23 PHI 20 11 40 304.3 282 102 29 77 146 3.02 4.06
## 4 1951 Roberts 24 PHI 21 15 44 315.0 284 106 20 64 127 3.03 3.84
## 5 1952 Roberts 25 PHI 28 7 39 330.0 292 95 22 45 148 2.59 3.66
## 6 1953 Roberts 26 PHI 23 16 44 346.7 324 106 30 61 198 2.75 4.20
## 7 1954 Roberts 27 PHI 23 15 45 336.7 289 111 35 56 185 2.97 4.03
## 8 1955 Roberts 28 PHI 23 14 41 305.0 292 111 41 53 160 3.28 3.96
## 9 1956 Roberts 29 PHI 19 18 43 297.3 328 147 46 40 157 4.45 3.73
## 10 1957 Roberts 30 PHI 10 22 39 249.7 246 113 40 43 128 4.07 3.80
## 11 1958 Roberts 31 PHI 17 14 35 269.7 270 97 30 51 130 3.24 3.95
## 12 1959 Roberts 32 PHI 15 17 35 257.3 267 122 34 35 137 4.27 4.11
## 13 1960 Roberts 33 PHI 12 16 35 237.3 256 106 31 34 122 4.02 3.88
## 14 1961 Roberts 34 PHI 1 10 26 117.0 154 76 19 23 54 5.85 4.07
## 15 1962 Roberts 35 BAL 10 9 27 191.3 176 59 17 41 102 2.78 3.77
## 16 1963 Roberts 36 BAL 14 13 35 251.3 230 93 35 40 124 3.33 3.52
## 17 1964 Roberts 37 BAL 13 7 31 204.0 203 66 18 52 109 2.91 3.59
## 18 1965 Roberts 38 BAL/HOU 10 9 30 190.7 171 59 18 30 97 2.78 3.42
## 19 1966 Roberts 39 CHC/HOU 5 8 24 112.0 141 60 15 21 54 4.82 3.54
## X.ERA.
## 1 124
## 2 107
## 3 135
## 4 127
## 5 141
## 6 152
## 7 136
## 8 121
## 9 84
## 10 93
## 11 122
## 12 96
## 13 96
## 14 70
## 15 136
## 16 106
## 17 123
## 18 123
## 19 73

 ford = subset(case.3.2, Player=="Ford")
 ford

## Year Player Age Tm W L G IP H ER HR BB SO ERA X.lgERA
## 20 1950 Ford 21 NYY 9 1 20 112.0 87 35 7 52 59 2.81 4.30
## 21 1953 Ford 24 NYY 18 6 32 207.0 187 69 13 110 110 3.00 3.68
## 22 1954 Ford 25 NYY 16 8 34 210.7 170 66 10 101 125 2.82 3.42
## 23 1955 Ford 26 NYY 18 7 39 253.7 188 74 20 113 137 2.63 3.76
## 24 1956 Ford 27 NYY 19 6 31 225.7 187 62 13 84 141 2.47 3.87
## 25 1957 Ford 28 NYY 11 5 24 129.3 114 37 10 53 84 2.57 3.60
## 26 1958 Ford 29 NYY 14 7 30 219.3 174 49 14 62 145 2.01 3.54
## 27 1959 Ford 30 NYY 16 10 35 204.0 194 69 13 89 114 3.04 3.63
## 28 1960 Ford 31 NYY 12 9 33 192.7 168 66 15 65 85 3.08 3.60
## 29 1961 Ford 32 NYY 25 4 39 283.0 242 101 23 92 209 3.21 3.70
## 30 1962 Ford 33 NYY 17 8 38 257.7 243 83 22 69 160 2.90 3.73
## 31 1963 Ford 34 NYY 24 7 38 269.3 240 82 26 56 189 2.74 3.52
## 32 1964 Ford 35 NYY 17 6 39 244.7 212 58 10 57 172 2.13 3.62
## 33 1965 Ford 36 NYY 16 13 37 244.3 241 88 22 50 162 3.24 3.39
## 34 1966 Ford 37 NYY 2 5 22 73.0 79 20 8 24 43 2.47 3.33
## 35 1967 Ford 38 NYY 2 4 7 44.0 40 8 2 9 21 1.64 3.13
## X.ERA.
## 20 153
## 21 123
## 22 121
## 23 143
## 24 156
## 25 140
## 26 176
## 27 119
## 28 117
## 29 115
## 30 129
## 31 128
## 32 170
## 33 105
## 34 135
## 35 191

 stem.leaf.backback(ford$ERA, roberts$ERA, m=2, Min=1.5, Max=6)

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 1 | 2: represents 1.2, leaf unit: 0.1
## ford$ERA roberts$ERA
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## | 1\* |
## 1 6| 1. |
## 5 4410| 2\* |
## (6) 988765| 2. |577799 6
## 5 22000| 3\* |001223 (6)
## | 3. |6 7
## | 4\* |0024 6
## | 4. |8 2
## | 5\* |
## | 5. |8 1
## | 6\* |
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## n: 16 19
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 ###
 ### Make Figure 3.6
 ### ALbert forgot to make league average pitcher information available.
 ### The data can be found at http://www.baseball-reference.com/leagues/NL/pitch.shtml#teams\_standard\_pitching::none
 ### and http://www.baseball-reference.com/leagues/AL/pitch.shtml#teams\_standard\_pitching::none
 ### Click on CSV and then EXPORT to save them to your machine.
 ### The files are also available at the URLs given below.
 nlpitch = read.csv("http://bulldog2.redlands.edu/fac/jim\_bentley/downloads/fs21/nlpitchingaverages.txt", header=TRUE)
 alpitch = read.csv("http://bulldog2.redlands.edu/fac/jim\_bentley/downloads/fs21/alpitchingaverages.txt", header=TRUE)
 nlpitch$lgID = "NL"
 alpitch$lgID = "AL"
 pitch = rbind(alpitch, nlpitch)
 plot(alpitch$Year,alpitch$ERA,type="l", lwd=2, xlab="Year", ylab="Mean ERA")
 lines(nlpitch$Year,nlpitch$ERA, lwd=1)
 ### Note that Albert only used years where both the AL and NL had data -- 1901 on.
 ### Albert didn't have data beyond 2002 but we have through 2016.
 abline(v=2002,lty=3)
 abline(v=1901, lty=3)
 ### Add a legend.
 legend(1978,3.4,legend=c("AL","NL"),lwd=c(2,1))



 ###
 ### The above figure can be created using data scraping.
 ### Install and load a few support packages.
 p\_load(rvest) # Scrapes
 p\_load(stringr) # String tools
 p\_load(tidyr) # Data frame manipulation
 ### Get the NL pitching page
 url = 'http://www.baseball-reference.com/leagues/NL/pitch.shtml#teams\_standard\_pitching::none'
 webpage = read\_html(url)
 ### Use the rvest functions html\_nodes and html\_table to extract what we want
 ### Grab the tables on the page
 nl\_table = html\_nodes(webpage, 'table')
 nl\_table

## {xml\_nodeset (2)}
## [1] <table class="sortable stats\_table" id="teams\_standard\_pitching">\n ...
## [2] <table class="sortable stats\_table" id="teams\_standard\_pitching\_tot ...

 ### Assign the tables to separate data frames
 nl1 = html\_table(nl\_table)[[1]]
 nl2 = html\_table(nl\_table)[[2]]
 head(nl1)

## Year Tms #P PAge R/G ERA G GF CG SHO tSho SV IP H R
## 1 2016 15 366 28.2 4.51 4.20 1877 0.98 0.02 0.01 0.06 0.26 8.95 8.66 4.51
## 2 2015 15 393 28.4 4.21 3.91 2430 0.98 0.02 0.01 0.07 0.27 8.94 8.68 4.21
## 3 2014 15 347 28.5 4.00 3.66 2430 0.98 0.02 0.01 0.07 0.27 8.98 8.49 4.00
## 4 2013 15 352 28.1 4.04 3.74 2430 0.98 0.02 0.01 0.07 0.26 8.99 8.55 4.04
## 5 2012 16 374 28.5 4.26 3.95 2592 0.98 0.02 0.01 0.06 0.26 8.90 8.62 4.26
## 6 2011 16 357 28.5 4.16 3.82 2590 0.97 0.03 0.01 0.07 0.27 8.97 8.63 4.16
## ER HR BB IBB SO HBP BK WP BF ERA+ WHIP BAbip H9 HR9
## 1 4.18 1.11 3.23 0.24 8.09 0.36 0.03 0.37 38.22 101 1.328 .301 8.7 1.1
## 2 3.88 0.97 2.93 0.23 7.83 0.32 0.03 0.34 37.83 101 1.299 .303 8.7 1.0
## 3 3.65 0.84 2.88 0.22 7.76 0.34 0.02 0.34 37.77 101 1.267 .299 8.5 0.8
## 4 3.73 0.89 2.97 0.23 7.48 0.33 0.03 0.34 37.94 101 1.282 .296 8.6 0.9
## 5 3.91 0.96 3.05 0.24 7.61 0.29 0.03 0.32 37.89 101 1.311 .300 8.7 1.0
## 6 3.81 0.90 3.12 0.30 7.27 0.31 0.04 0.30 38.15 101 1.309 .296 8.7 0.9
## BB9 SO9 SO/W E Attendance Attend/G
## 1 3.2 8.1 2.51 0.61 29,677,540 31,639
## 2 2.9 7.9 2.67 0.59 38,862,357 31,985
## 3 2.9 7.8 2.69 0.60 39,248,477 32,303
## 4 3.0 7.5 2.52 0.58 39,436,049 32,457
## 5 3.1 7.7 2.50 0.64 41,475,495 32,002
## 6 3.1 7.3 2.33 0.63 40,752,289 31,468

 head(nl2)

## Year Tms #P PAge R/G ERA G GF CG SHO tSho SV IP H
## 1 2016 15 366 28.2 4.51 4.20 1877 1848 29 16 113 491 16799.2 16258
## 2 2015 15 393 28.4 4.21 3.91 2430 2392 38 21 181 653 21713.1 21091
## 3 2014 15 347 28.5 4.00 3.66 2430 2373 57 34 180 645 21815.0 20633
## 4 2013 15 352 28.1 4.04 3.74 2430 2370 60 27 173 629 21847.1 20781
## 5 2012 16 374 28.5 4.26 3.95 2592 2533 59 35 168 671 23070.1 22355
## 6 2011 16 357 28.5 4.16 3.82 2590 2510 80 36 174 707 23244.0 22348
## R ER HR BB IBB SO HBP BK WP BF ERA+ WHIP BAbip H9
## 1 8460 7839 2075 6058 448 15190 678 61 702 71733 101 1.328 .301 8.7
## 2 10225 9436 2355 7112 560 19023 782 79 820 91923 101 1.299 .303 8.7
## 3 9709 8878 2035 7003 523 18853 820 54 833 91773 101 1.267 .299 8.5
## 4 9819 9075 2152 7219 566 18174 800 67 832 92192 101 1.282 .296 8.6
## 5 11037 10128 2481 7894 613 19715 759 77 818 98206 101 1.311 .300 8.7
## 6 10772 9865 2328 8069 766 18833 796 106 782 98820 101 1.309 .296 8.7
## HR9 BB9 SO9 SO/W E Attendance Attend/G
## 1 1.1 3.2 8.1 2.51 1153 29,677,540 31,639
## 2 1.0 2.9 7.9 2.67 1423 38,862,357 31,985
## 3 0.8 2.9 7.8 2.69 1455 39,248,477 32,303
## 4 0.9 3.0 7.5 2.52 1405 39,436,049 32,457
## 5 1.0 3.1 7.7 2.50 1661 41,475,495 32,002
## 6 0.9 3.1 7.3 2.33 1621 40,752,289 31,468

 ### Since the tables appear to be identical, get rid of the second
 nl = nl1
 rm(nl1, nl2)
 head(nl)

## Year Tms #P PAge R/G ERA G GF CG SHO tSho SV IP H R
## 1 2016 15 366 28.2 4.51 4.20 1877 0.98 0.02 0.01 0.06 0.26 8.95 8.66 4.51
## 2 2015 15 393 28.4 4.21 3.91 2430 0.98 0.02 0.01 0.07 0.27 8.94 8.68 4.21
## 3 2014 15 347 28.5 4.00 3.66 2430 0.98 0.02 0.01 0.07 0.27 8.98 8.49 4.00
## 4 2013 15 352 28.1 4.04 3.74 2430 0.98 0.02 0.01 0.07 0.26 8.99 8.55 4.04
## 5 2012 16 374 28.5 4.26 3.95 2592 0.98 0.02 0.01 0.06 0.26 8.90 8.62 4.26
## 6 2011 16 357 28.5 4.16 3.82 2590 0.97 0.03 0.01 0.07 0.27 8.97 8.63 4.16
## ER HR BB IBB SO HBP BK WP BF ERA+ WHIP BAbip H9 HR9
## 1 4.18 1.11 3.23 0.24 8.09 0.36 0.03 0.37 38.22 101 1.328 .301 8.7 1.1
## 2 3.88 0.97 2.93 0.23 7.83 0.32 0.03 0.34 37.83 101 1.299 .303 8.7 1.0
## 3 3.65 0.84 2.88 0.22 7.76 0.34 0.02 0.34 37.77 101 1.267 .299 8.5 0.8
## 4 3.73 0.89 2.97 0.23 7.48 0.33 0.03 0.34 37.94 101 1.282 .296 8.6 0.9
## 5 3.91 0.96 3.05 0.24 7.61 0.29 0.03 0.32 37.89 101 1.311 .300 8.7 1.0
## 6 3.81 0.90 3.12 0.30 7.27 0.31 0.04 0.30 38.15 101 1.309 .296 8.7 0.9
## BB9 SO9 SO/W E Attendance Attend/G
## 1 3.2 8.1 2.51 0.61 29,677,540 31,639
## 2 2.9 7.9 2.67 0.59 38,862,357 31,985
## 3 2.9 7.8 2.69 0.60 39,248,477 32,303
## 4 3.0 7.5 2.52 0.58 39,436,049 32,457
## 5 3.1 7.7 2.50 0.64 41,475,495 32,002
## 6 3.1 7.3 2.33 0.63 40,752,289 31,468

 ### Repeat the above for the AL
 ### Get the AL pitching page
 url = 'http://www.baseball-reference.com/leagues/AL/pitch.shtml#teams\_standard\_pitching::none'
 webpage = read\_html(url)
 ### Use the rvest functions html\_nodes and html\_table to extract what we want
 ### Grab the tables on the page
 al\_table = html\_nodes(webpage, 'table')
 al\_table

## {xml\_nodeset (2)}
## [1] <table class="sortable stats\_table" id="teams\_standard\_pitching">\n ...
## [2] <table class="sortable stats\_table" id="teams\_standard\_pitching\_tot ...

 ### Assign the first table to the data frame
 al = html\_table(al\_table)[[1]]
 head(al)

## Year Tms #P PAge R/G ERA G GF CG SHO tSho SV IP H R
## 1 2016 15 358 28.7 4.47 4.22 1875 0.98 0.02 0.01 0.05 0.27 8.91 8.81 4.47
## 2 2015 15 383 28.3 4.29 4.01 2428 0.97 0.03 0.01 0.07 0.26 8.94 8.66 4.29
## 3 2014 15 370 28.5 4.14 3.82 2430 0.97 0.03 0.01 0.07 0.25 8.97 8.63 4.14
## 4 2013 15 349 28.7 4.29 3.99 2432 0.97 0.03 0.01 0.06 0.26 8.97 8.76 4.29
## 5 2012 14 325 28.2 4.40 4.09 2268 0.97 0.03 0.01 0.06 0.26 8.94 8.69 4.40
## 6 2011 14 325 28.2 4.43 4.08 2268 0.96 0.04 0.02 0.07 0.24 8.94 8.78 4.43
## ER HR BB IBB SO HBP BK WP BF ERA+ WHIP BAbip H9 HR9
## 1 4.17 1.22 2.96 0.14 7.88 0.31 0.03 0.36 37.87 101 1.322 .299 8.9 1.2
## 2 3.98 1.05 2.87 0.16 7.59 0.34 0.03 0.39 37.77 101 1.290 .296 8.7 1.1
## 3 3.81 0.89 2.89 0.19 7.65 0.34 0.03 0.36 37.92 101 1.284 .298 8.7 0.9
## 4 3.98 1.03 3.05 0.19 7.62 0.30 0.03 0.37 38.11 101 1.318 .298 8.8 1.0
## 5 4.06 1.08 3.00 0.19 7.37 0.32 0.04 0.32 37.91 101 1.308 .293 8.7 1.1
## 6 4.06 0.98 3.06 0.21 6.90 0.33 0.03 0.34 38.11 101 1.325 .293 8.8 1.0
## BB9 SO9 SO/W E Attendance Attend/G
## 1 3.0 8.0 2.66 0.56 27,398,174 29,209
## 2 2.9 7.6 2.65 0.58 34,856,983 28,712
## 3 2.9 7.7 2.65 0.60 34,491,145 28,387
## 4 3.1 7.7 2.50 0.55 34,590,988 28,446
## 5 3.0 7.4 2.45 0.59 33,383,773 29,438
## 6 3.1 6.9 2.25 0.63 32,673,378 28,812

 ### Did we get the right years for the leagues?
 range(al$Year)

## [1] "1901" "Year"

 range(nl$Year)

## [1] "1876" "Year"

 # The mins look good but the "Year" values are strange. It turns out that
 # the tables had multiple "headers" and only the first is used. We need to
 # get rid of the observations with Year == "Year".
 nl[nl$Year=="Year",]

## Year Tms #P PAge R/G ERA G GF CG SHO tSho SV IP H R ER HR BB IBB SO
## 26 Year Tms #P PAge R/G ERA G GF CG SHO tSho SV IP H R ER HR BB IBB SO
## 52 Year Tms #P PAge R/G ERA G GF CG SHO tSho SV IP H R ER HR BB IBB SO
## 78 Year Tms #P PAge R/G ERA G GF CG SHO tSho SV IP H R ER HR BB IBB SO
## 104 Year Tms #P PAge R/G ERA G GF CG SHO tSho SV IP H R ER HR BB IBB SO
## 130 Year Tms #P PAge R/G ERA G GF CG SHO tSho SV IP H R ER HR BB IBB SO
## HBP BK WP BF ERA+ WHIP BAbip H9 HR9 BB9 SO9 SO/W E Attendance Attend/G
## 26 HBP BK WP BF ERA+ WHIP BAbip H9 HR9 BB9 SO9 SO/W E Attendance Attend/G
## 52 HBP BK WP BF ERA+ WHIP BAbip H9 HR9 BB9 SO9 SO/W E Attendance Attend/G
## 78 HBP BK WP BF ERA+ WHIP BAbip H9 HR9 BB9 SO9 SO/W E Attendance Attend/G
## 104 HBP BK WP BF ERA+ WHIP BAbip H9 HR9 BB9 SO9 SO/W E Attendance Attend/G
## 130 HBP BK WP BF ERA+ WHIP BAbip H9 HR9 BB9 SO9 SO/W E Attendance Attend/G

 al[al$Year=="Year",]

## Year Tms #P PAge R/G ERA G GF CG SHO tSho SV IP H R ER HR BB IBB SO
## 26 Year Tms #P PAge R/G ERA G GF CG SHO tSho SV IP H R ER HR BB IBB SO
## 52 Year Tms #P PAge R/G ERA G GF CG SHO tSho SV IP H R ER HR BB IBB SO
## 78 Year Tms #P PAge R/G ERA G GF CG SHO tSho SV IP H R ER HR BB IBB SO
## 104 Year Tms #P PAge R/G ERA G GF CG SHO tSho SV IP H R ER HR BB IBB SO
## HBP BK WP BF ERA+ WHIP BAbip H9 HR9 BB9 SO9 SO/W E Attendance Attend/G
## 26 HBP BK WP BF ERA+ WHIP BAbip H9 HR9 BB9 SO9 SO/W E Attendance Attend/G
## 52 HBP BK WP BF ERA+ WHIP BAbip H9 HR9 BB9 SO9 SO/W E Attendance Attend/G
## 78 HBP BK WP BF ERA+ WHIP BAbip H9 HR9 BB9 SO9 SO/W E Attendance Attend/G
## 104 HBP BK WP BF ERA+ WHIP BAbip H9 HR9 BB9 SO9 SO/W E Attendance Attend/G

 nl = nl[!(nl$Year=="Year"),] # ! is read as "not"
 al = al[!(al$Year=="Year"),] # ! is read as "not"
 range(al$Year)

## [1] "1901" "2016"

 range(nl$Year)

## [1] "1876" "2016"

 # That's better. Now compare ERAs.
 plot(al$Year,al$ERA,type="l", lwd=2, xlab="Year", ylab="Mean ERA")
 lines(nl$Year,nl$ERA, lwd=1)
 abline(v=2002,lty=3)
 abline(v=1901, lty=3)
 legend(1978,3.4,legend=c("AL","NL"),lwd=c(2,1))



 ###
 ### Create Figure 3.7
 ### We can use the data already imported for Figure 3.5
 head(roberts)

## Year Player Age Tm W L G IP H ER HR BB SO ERA X.lgERA
## 1 1948 Roberts 21 PHI 7 9 20 146.7 148 52 10 61 84 3.19 3.96
## 2 1949 Roberts 22 PHI 15 15 43 226.7 229 93 15 75 95 3.69 3.96
## 3 1950 Roberts 23 PHI 20 11 40 304.3 282 102 29 77 146 3.02 4.06
## 4 1951 Roberts 24 PHI 21 15 44 315.0 284 106 20 64 127 3.03 3.84
## 5 1952 Roberts 25 PHI 28 7 39 330.0 292 95 22 45 148 2.59 3.66
## 6 1953 Roberts 26 PHI 23 16 44 346.7 324 106 30 61 198 2.75 4.20
## X.ERA.
## 1 124
## 2 107
## 3 135
## 4 127
## 5 141
## 6 152

 ### Note that x.ERA. (\*ERA) is equal to the formula in the book (within rounding)
 cbind(roberts$X.lgERA/roberts$ERA\*100, roberts$X.ERA.)[1:5,]

## [,1] [,2]
## [1,] 124.1379 124
## [2,] 107.3171 107
## [3,] 134.4371 135
## [4,] 126.7327 127
## [5,] 141.3127 141

 stem.leaf.backback(ford$X.ERA.,roberts$X.ERA., Max=190)

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 1 | 2: represents 12, leaf unit: 1
## ford$X.ERA. roberts$X.ERA.
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## | 7 |03 2
## | 8 |4 3
## | 9 |366 6
## 1 5| 10 |67 8
## 4 975| 11 |
## 8 9831| 12 |123347 11
## (1) 5| 13 |566 5
## 7 30| 14 |1 2
## 5 63| 15 |2 1
## | 16 |
## 3 60| 17 |
## | 18 |
## 1 1| 19 |
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## n: 16 19
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Case 3.3

Albert now looks at home runs and the rate at which they occur. For this case we download the data from the author's site.

 ### Grab the data from ALbert's site
 case.3.3 = read.table("http://www-math.bgsu.edu/~albert/tsub/datasets/case\_3\_3.txt",
 header=TRUE)
 ### R doesn't generate parallel stemplots so for Figure 3.8 we create
 ### multiple back to back stemplots
 homers.1927 = case.3.3[case.3.3$YEAR==1927,]
 homers.1961 = case.3.3[case.3.3$YEAR==1961,]
 homers.1998 = case.3.3[case.3.3$YEAR==1998,]
 homers.2001 = case.3.3[case.3.3$YEAR==2001,]
 p\_load(aplpack)
 stem.leaf.backback(homers.1927$RATE,homers.1961$RATE, Min=.1, Max=1.6)

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 1 | 2: represents 0.12, leaf unit: 0.01
## homers.1927$RATE
## homers.1961$RATE
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 4 9887| 1 |
## 7 544| 2 |
## (5) 76553| 3 |
## 4 8| 4 |
## 3 5| 5 |6 1
## | 6 |669 4
## 2 01| 7 |4 5
## | 8 |35 7
## | 9 |13 (2)
## 1 2| 10 |234 9
## | 11 |0378 6
## | 12 |1 2
## | 13 |
## | 14 |7 1
## | 15 |
## | 16 |
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## n: 16 18
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 stem.leaf.backback(homers.1927$RATE,homers.1998$RATE, Min=.1, Max=1.6)

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 1 | 2: represents 0.12, leaf unit: 0.01
## homers.1927$RATE
## homers.1998$RATE
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 4 9887| 1 |
## 7 544| 2 |
## (5) 76553| 3 |
## 4 8| 4 |
## 3 5| 5 |
## | 6 |68 2
## 2 01| 7 |1710 5
## | 8 |245 8
## | 9 |1124889 15
## 1 2| 10 |223 (3)
## | 11 |3 12
## | 12 |12478 11
## | 13 |02367 6
## | 14 |5 1
## | 15 |
## | 16 |
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## n: 16 30
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 stem.leaf.backback(homers.1927$RATE,homers.2001$RATE, Min=.1, Max=1.6)

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 1 | 2: represents 0.12, leaf unit: 0.01
## homers.1927$RATE homers.2001$RATE
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 4 9887| 1 |
## 7 544| 2 |
## (5) 76553| 3 |
## 4 8| 4 |
## 3 5| 5 |
## | 6 |
## 2 01| 7 |5 1
## | 8 |146 4
## | 9 |14899 9
## 1 2| 10 |112479 15
## | 11 |
## | 12 |333678891010 15
## | 13 |112 5
## | 14 |5 2
## | 15 |2 1
## | 16 |
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## n: 16 30
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 stem.leaf.backback(homers.1961$RATE,homers.1998$RATE, Min=.1, Max=1.6)

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 1 | 2: represents 0.12, leaf unit: 0.01
## homers.1961$RATE
## homers.1998$RATE
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## | 1 |
## | 2 |
## | 3 |
## | 4 |
## 1 6| 5 |
## 4 966| 6 |68 2
## 5 4| 7 |1710 5
## 7 53| 8 |245 8
## (2) 31| 9 |1124889 15
## 9 432| 10 |223 (3)
## 6 8730| 11 |3 12
## 2 1| 12 |12478 11
## | 13 |02367 6
## 1 7| 14 |5 1
## | 15 |
## | 16 |
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## n: 18 30
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 stem.leaf.backback(homers.1961$RATE,homers.2001$RATE, Min=.1, Max=1.6)

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 1 | 2: represents 0.12, leaf unit: 0.01
## homers.1961$RATE homers.2001$RATE
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## | 1 |
## | 2 |
## | 3 |
## | 4 |
## 1 6| 5 |
## 4 966| 6 |
## 5 4| 7 |5 1
## 7 53| 8 |146 4
## (2) 31| 9 |14899 9
## 9 432| 10 |112479 15
## 6 8730| 11 |
## 2 1| 12 |333678891010 15
## | 13 |112 5
## 1 7| 14 |5 2
## | 15 |2 1
## | 16 |
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## n: 18 30
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 stem.leaf.backback(homers.1998$RATE,homers.2001$RATE, Min=.1, Max=1.6)

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 1 | 2: represents 0.12, leaf unit: 0.01
## homers.1998$RATE homers.2001$RATE
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## | 1 |
## | 2 |
## | 3 |
## | 4 |
## | 5 |
## 2 86| 6 |
## 5 0171| 7 |5 1
## 8 542| 8 |146 4
## 15 9884211| 9 |14899 9
## (3) 322| 10 |112479 15
## 12 3| 11 |
## 11 87421| 12 |333678891010 15
## 6 76320| 13 |112 5
## 1 5| 14 |5 2
## | 15 |2 1
## | 16 |
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## n: 30 30
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 ### Tough to interpret? Yep. Maybe a boxplot would be better.
 # Create 5-number summaries
 by(case.3.3[,"RATE"],case.3.3[,"YEAR"], summary)

## case.3.3[, "YEAR"]: 1927
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 0.1700 0.2275 0.3400 0.3725 0.3975 1.0200
## --------------------------------------------------------
## case.3.3[, "YEAR"]: 1961
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 0.5600 0.7625 0.9750 0.9544 1.1220 1.4700
## --------------------------------------------------------
## case.3.3[, "YEAR"]: 1998
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 0.660 0.865 1.005 1.040 1.262 1.450
## --------------------------------------------------------
## case.3.3[, "YEAR"]: 2001
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 0.750 0.990 1.145 1.123 1.277 1.520

 # and figure out how many teams there were in each year
 cnts = table(case.3.3$YEAR)
 cnts

##
## 1927 1961 1998 2001
## 16 18 30 30

 # The Figure 3.9 boxplots look like
 boxplot(RATE~YEAR, data=case.3.3, ylab="Home Run Rate (HR/G)", varwidth=TRUE)



 p\_load(lattice)
 bwplot(YEAR~RATE, data=case.3.3, ylab="Home Run Rate (HR/G)", varwidth=TRUE)



 bwplot(RATE~YEAR, data=case.3.3, ylab="Home Run Rate (HR/G)", varwidth=TRUE, horizontal = FALSE)



 p\_load(ggplot2)
 ggplot(case.3.3, aes(factor(YEAR), RATE)) + geom\_boxplot(varwidth=TRUE) + xlab("Year") + ylab("Home Run Rate (HR/G)")



 ggplot(case.3.3,aes(factor(YEAR), RATE)) + geom\_boxplot(varwidth=TRUE) + coord\_flip() + xlab("Year") + ylab("Home Run Rate (HR/G)")



### Case 3.4

The text now discusses the idea of the normal distribution. We will suplement the discussion with a couple of "modern" techniques. For this case we download the data from the author's site. Note that many of the values disagree with those in the text. Please let me know if you figure out why.

 ### Grab the data from ALbert's site
 case.3.4 = read.delim("http://www-math.bgsu.edu/~albert/tsub/datasets/case\_3\_4.txt", header=TRUE, comment.char="#")
 head(case.3.4)

## Lastname Firstname Year AB H X2B X3B HR X1B SLG
## 1 Abreu Bobby 1999 546 183 35 11 20 117 0.5494505
## 2 Alfonzo Edgardo 1999 628 191 41 1 27 122 0.5015924
## 3 Allen Chad 1999 481 133 21 3 10 99 0.3950104
## 4 Alomar Roberto 1999 563 182 40 3 24 115 0.5328597
## 5 Anderson Brady 1999 564 159 28 5 24 102 0.4769504
## 6 Anderson Garret 1999 620 188 36 2 21 129 0.4693548

 case.3.4$SLG

## [1] 0.5494505 0.5015924 0.3950104 0.5328597 0.4769504 0.4693548 0.3606195
## [8] 0.4444444 0.4148472 0.5907473 0.4364896 0.4321608 0.3497053 0.5409836
## [15] 0.4288840 0.5568761 0.4275093 0.4539970 0.4572954 0.5413153 0.4569640
## [22] 0.4449153 0.4161184 0.4025357 0.4539580 0.4143763 0.5610278 0.4206186
## [29] 0.3677419 0.4686347 0.5627907 0.2967864 0.5387205 0.3655031 0.4780488
## [36] 0.4083885 0.4612850 0.5074627 0.4451613 0.4527027 0.4239829 0.4635294
## [43] 0.4266409 0.4768439 0.4453782 0.4351852 0.5706806 0.4346405 0.5263158
## [50] 0.4335155 0.4499018 0.3743590 0.5711207 0.4105960 0.4494845 0.5254237
## [57] 0.4147982 0.4854369 0.4302326 0.6033835 0.5530435 0.6142035 0.4570064
## [64] 0.4500907 0.4303571 0.6014235 0.5488599 0.3407407 0.4806071 0.5879479
## [71] 0.4928058 0.4814815 0.5759076 0.4145937 0.4364754 0.6000000 0.2761905
## [78] 0.4768856 0.5865052 0.4657534 0.2995868 0.5557987 0.5637584 0.5518341
## [85] 0.4131455 0.4831081 0.6331570 0.4652778 0.4755245 0.4471154 0.5501730
## [92] 0.5107632 0.5321782 0.4543947 0.5107212 0.4020202 0.4928910 0.4202401
## [99] 0.3546798 0.4634146 0.3085106 0.5509804 0.4322581 0.3901192 0.5064695
## [106] 0.3871595 0.5537849 0.4584041 0.3976608 0.5519849 0.6967370 0.3657244
## [113] 0.4825291 0.3289474 0.4266145 0.3623188 0.4243119 0.4351536 0.4949664
## [120] 0.4629948 0.4589615 0.5096154 0.3173077 0.3909091 0.5178571 0.6300885
## [127] 0.4028986 0.4334975 0.5749064 0.6628352 0.4729299 0.4170940 0.4000000
## [134] 0.4063260 0.5856574 0.5436242 0.5583333 0.5249406 0.3695198 0.5271967
## [141] 0.5135699 0.5227687 0.4899194 0.4508772 0.6352000 0.4653061 0.5329567
## [148] 0.4660422 0.4854932 0.4111842 0.4918276 0.5200000 0.5530726 0.5212264
## [155] 0.4266442 0.4711934 0.5404858 0.3977778 0.4824017 0.5345455 0.5076336
## [162] 0.5289116 0.3789474 0.4757085 0.4355401 0.7100457 0.3973635 0.4071730
## [169] 0.5046382 0.5363790 0.4573460 0.5358852 0.3816425 0.5020747 0.3697068
## [176] 0.3552632 0.5222603 0.4880952

 ### Create Figure 3.10
 stem(case.3.4$SLG)

##
## The decimal point is 2 digit(s) to the left of the |
##
## 26 | 6
## 28 | 7
## 30 | 097
## 32 | 9
## 34 | 1055
## 36 | 126680049
## 38 | 27015788
## 40 | 0233678113455567
## 42 | 014477789002234555666
## 44 | 4555790013444777789
## 46 | 133455669913667778
## 48 | 1123355802335
## 50 | 22567801148
## 52 | 0123556792335669
## 54 | 0114990122334678
## 56 | 1341156
## 58 | 6781
## 60 | 0134
## 62 | 035
## 64 |
## 66 | 3
## 68 | 7
## 70 | 0

 ### Compute the statistics that Albert presents
 n = length(case.3.4$SLG)
 n # The number of observations

## [1] 178

 xbar = mean(case.3.4$SLG)
 xbar # The sample mean

## [1] 0.4731246

 median(case.3.4$SLG)

## [1] 0.4655298

 mean(case.3.4$SLG, trim=0.05) # A 10% trimmed mean

## [1] 0.472301

 variance = var(case.3.4$SLG)
 sd = sqrt(variance)
 sd # Standard deviation

## [1] 0.07758348

 sqrt(variance/n)

## [1] 0.005815128

 summary(case.3.4$SLG)

## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 0.2762 0.4215 0.4655 0.4731 0.5285 0.7100

 ### Computing the mean plus/minus k standard deviations is "easy"
 xbar = .473
 sd = 0.078
 xbar + c(-1,1)%\*%t(1:3)\*sd

## [,1] [,2] [,3]
## [1,] 0.395 0.317 0.239
## [2,] 0.551 0.629 0.707

 # Compare with actual values
 length(case.3.4$SLG[.395 < case.3.4$SLG & case.3.4$SLG < .551])

## [1] 126

 length(case.3.4$SLG[.317 < case.3.4$SLG & case.3.4$SLG < .629])

## [1] 168

 length(case.3.4$SLG[.239 < case.3.4$SLG & case.3.4$SLG < .707])

## [1] 177

 c(126,168,177)/178

## [1] 0.7078652 0.9438202 0.9943820

 ### A graphical method for checking normality is the normal probability
 ### or normal quantile plot. If the data are normally distributed, what
 ### we observe and what we expect to see should be the same. Hence, the
 ### plotted values should fall on the diagonal line.
 qqnorm(case.3.4$SLG)
 qqline(case.3.4$SLG)



 ### There are also some "tests" for normality.
 p\_load(nortest)
 ad.test(case.3.4$SLG)

##
## Anderson-Darling normality test
##
## data: case.3.4$SLG
## A = 0.3625, p-value = 0.4391

 cvm.test(case.3.4$SLG)

##
## Cramer-von Mises normality test
##
## data: case.3.4$SLG
## W = 0.066683, p-value = 0.3074

 lillie.test(case.3.4$SLG)

##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: case.3.4$SLG
## D = 0.047604, p-value = 0.4178

 pearson.test(case.3.4$SLG)

##
## Pearson chi-square normality test
##
## data: case.3.4$SLG
## P = 8.9663, p-value = 0.7755

 sf.test(case.3.4$SLG)

##
## Shapiro-Francia normality test
##
## data: case.3.4$SLG
## W = 0.99328, p-value = 0.5081

 shapiro.test(case.3.4$SLG)

##
## Shapiro-Wilk normality test
##
## data: case.3.4$SLG
## W = 0.99338, p-value = 0.603

### Case 3.5

The last case in this chapter discusses battting averages. For this case we download the data from the author's site.

 ### Grab the data from ALbert's site
 case.3.5 = read.delim("http://www-math.bgsu.edu/~albert/tsub/datasets/case\_3\_5.txt", header=TRUE, comment.char="#")
 head(case.3.5)

## Lastname Firstname Year AB H X2B X3B HR
## 1 Appling Luke 1941 592 186 26 8 1
## 2 Benjamin Stan 1941 480 113 20 7 3
## 3 Berardino Johnny 1941 469 127 30 4 5
## 4 Bloodworth Jimmy 1941 506 124 24 3 7
## 5 Boudreau Lou 1941 579 149 45 8 10
## 6 Bragan Bobby 1941 557 140 19 3 4

 dim(case.3.5)

## [1] 98 8

 table(case.3.5$Year)

##
## 1941
## 98

 ### Unfortunately, the online dataset is for 1941 and not 1977. We go to
 ### Lahman to get the 1977 data.
 p\_load(Lahman)
 data(Batting)
 head(Batting)

## playerID yearID stint teamID lgID G AB R H X2B X3B HR RBI SB CS BB
## 1 abercda01 1871 1 TRO NA 1 4 0 0 0 0 0 0 0 0 0
## 2 addybo01 1871 1 RC1 NA 25 118 30 32 6 0 0 13 8 1 4
## 3 allisar01 1871 1 CL1 NA 29 137 28 40 4 5 0 19 3 1 2
## 4 allisdo01 1871 1 WS3 NA 27 133 28 44 10 2 2 27 1 1 0
## 5 ansonca01 1871 1 RC1 NA 25 120 29 39 11 3 0 16 6 2 2
## 6 armstbo01 1871 1 FW1 NA 12 49 9 11 2 1 0 5 0 1 0
## SO IBB HBP SH SF GIDP
## 1 0 NA NA NA NA NA
## 2 0 NA NA NA NA NA
## 3 5 NA NA NA NA NA
## 4 2 NA NA NA NA NA
## 5 1 NA NA NA NA NA
## 6 1 NA NA NA NA NA

 batting = battingStats()
 dim(batting)

## [1] 99846 29

 batting = batting[batting$yearID==1977,]
 dim(batting)

## [1] 984 29

 batting = batting[batting$lgID %in% c("AL","NL"),]
 dim(batting)

## [1] 984 29

 batting = batting[batting$AB >= 400, ]
 dim(batting)

## [1] 360 29

 ### Well, there appear to have been 360 MLB batters with at least 400
 ### appearances in 1977. We can analyze them as Albert did to see if
 ### our conclusions are the same.
 # Figure 3.11
 stem(batting$BA)

##
## The decimal point is 2 digit(s) to the left of the |
##
## 20 | 06
## 21 | 6
## 22 | 9
## 23 | 0133559
## 24 | 001135566778
## 25 | 00111222344567799
## 26 | 001111223334444556677899
## 27 | 001122233445555557788999
## 28 | 00000122223333444446777788999
## 29 | 000111122336778899
## 30 | 00022445778899
## 31 | 011225788
## 32 | 002568
## 33 | 668
## 34 |
## 35 |
## 36 |
## 37 |
## 38 | 8

 summary(batting$BA)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
## 0.2000 0.2608 0.2780 0.2774 0.2922 0.3880 192

 xbar = mean(batting$BA)
 xbar

## [1] NA

 sd=sqrt(var(batting$BA))
 sd

## [1] NA

 ### The standard deviation is NA? How many are missing?
 table(is.na(batting$BA))

##
## FALSE TRUE
## 168 192

 # Get rid of the missings and try again
 batting = batting[!is.na(batting$BA),]
 stem(batting$BA)

##
## The decimal point is 2 digit(s) to the left of the |
##
## 20 | 06
## 21 | 6
## 22 | 9
## 23 | 0133559
## 24 | 001135566778
## 25 | 00111222344567799
## 26 | 001111223334444556677899
## 27 | 001122233445555557788999
## 28 | 00000122223333444446777788999
## 29 | 000111122336778899
## 30 | 00022445778899
## 31 | 011225788
## 32 | 002568
## 33 | 668
## 34 |
## 35 |
## 36 |
## 37 |
## 38 | 8

 xbar = mean(batting$BA)
 xbar

## [1] 0.2774226

 sd = sqrt(var(batting$BA))
 sd

## [1] 0.02717155

 batting$BA.z = (batting$BA - xbar)/sd
 stem(batting$BA.z)

##
## The decimal point is at the |
##
## -2 | 86
## -2 | 3
## -1 | 8776666
## -1 | 444333222211100000
## -0 | 99999988887766666666555555555
## -0 | 44443333322222221111111100
## 0 | 0011111111122222222222223444444444
## 0 | 55555555566777888888899
## 1 | 000111122222334
## 1 | 555666889
## 2 | 222
## 2 |
## 3 |
## 3 |
## 4 | 1

 summary(batting$BA.z)

## Min. 1st Qu. Median Mean 3rd Qu. Max.
## -2.84900 -0.61360 0.02125 0.00000 0.54570 4.07000

## Chapter 4

Chapter 4 moves from looking at a single variable in isolation to looking at how variables behave together. Changing one may cause the other to change, or they may just change together. We discuss both causality and association.

### Case 4.1

In this section, Albert looks at team offensive stats. We will use the data from Albert's site.

 ### Grab the data from ALbert's site
 case.4.1 = read.delim("http://www-math.bgsu.edu/~albert/tsub/datasets/case\_4\_1.txt", header=TRUE, comment.char="#")
 head(case.4.1)

## Team League Avg OBP Slg AB R H X2B X3B HR RBI BB
## 1 Anaheim American 0.280 0.352 0.472 5628 864 1574 309 34 236 837 608
## 2 Baltimore American 0.272 0.341 0.435 5549 794 1508 310 22 184 750 558
## 3 Boston American 0.267 0.341 0.423 5630 792 1503 316 32 167 755 611
## 4 Chicago American 0.286 0.356 0.470 5646 978 1615 325 33 216 926 591
## 5 Cleveland American 0.288 0.367 0.470 5683 950 1639 310 30 221 889 685
## 6 Detroit American 0.275 0.343 0.438 5644 823 1553 307 41 177 785 562
## SO HBP
## 1 1024 47
## 2 900 49
## 3 1019 42
## 4 960 53
## 5 1057 51
## 6 982 43

 dim(case.4.1)

## [1] 30 15

 ### Figure 4.1 is just a scatter plot
 plot(case.4.1$Slg, case.4.1$OBP, xlab="SLG", ylab="OBP")



 plot(OBP~Slg, data=case.4.1, xlab="SLG", ylab="OBP")



 p\_load(lattice)
 xyplot(OBP~Slg, data=case.4.1, xlab="SLG", ylab="OBP")



 p\_load(ggplot2)
 ggplot(case.4.1, aes(Slg, OBP)) + geom\_point() + xlab("SLG") + ylab("OBP") + theme\_bw()



 ### Figure 4.2 uses three letter team designators
 case.4.1$Team3 = substr(case.4.1$Team,1,3)
 head(case.4.1)

## Team League Avg OBP Slg AB R H X2B X3B HR RBI BB
## 1 Anaheim American 0.280 0.352 0.472 5628 864 1574 309 34 236 837 608
## 2 Baltimore American 0.272 0.341 0.435 5549 794 1508 310 22 184 750 558
## 3 Boston American 0.267 0.341 0.423 5630 792 1503 316 32 167 755 611
## 4 Chicago American 0.286 0.356 0.470 5646 978 1615 325 33 216 926 591
## 5 Cleveland American 0.288 0.367 0.470 5683 950 1639 310 30 221 889 685
## 6 Detroit American 0.275 0.343 0.438 5644 823 1553 307 41 177 785 562
## SO HBP Team3
## 1 1024 47 Ana
## 2 900 49 Bal
## 3 1019 42 Bos
## 4 960 53 Chi
## 5 1057 51 Cle
## 6 982 43 Det

 plot(case.4.1$X3B, case.4.1$X2B, xlab="Triples", ylab="Doubles")
 text(case.4.1$X3B+1, case.4.1$X2B+5, case.4.1$Team3, cex=0.75)



 ###
 ### Analysis of Table 4.2 data
 # Since Albert doesn't give us the data, we recreate it.
 # I created a file that contains the team and year for each observation.
 # tmyr = read.csv("d://fall16/fys/TSUBAlbert/TSUBData/Table.4.2.Team.Year.csv", header=TRUE)
 tmyr = read.csv("http://bulldog2.redlands.edu/fac/jim\_bentley/downloads/fs21/Table.4.2.Team.Year.csv", header=TRUE)
 # Now we suplement using Lahman and a little SQL
 p\_load(Lahman, sqldf)
 data(Teams)
 # We "select"" from the Teams data certain variables to keep.
 # An "inner join" keeps only those observations that are in both frames
 # The observations are matched "on"" team and year
 table.4.2 = sqldf("SELECT Teams.yearID, Teams.teamID, Teams.G, Teams.HR, Teams.X3B FROM Teams INNER JOIN tmyr ON Teams.teamID=tmyr.Team AND Teams.yearID=tmyr.Year")
 # Now create the variables that we are going to analyze
 table.4.2$HRG = table.4.2$HR/table.4.2$G
 table.4.2$X3BG = table.4.2$X3B/table.4.2$G
 head(table.4.2)

## yearID teamID G HR X3B HRG X3BG
## 1 1901 DET 136 29 80 0.21323529 0.5882353
## 2 1906 NY1 153 15 53 0.09803922 0.3464052
## 3 1907 DET 153 11 75 0.07189542 0.4901961
## 4 1908 WS1 155 8 74 0.05161290 0.4774194
## 5 1913 PHI 159 73 78 0.45911950 0.4905660
## 6 1914 KCF 154 39 77 0.25324675 0.5000000

 # It's time to make Figure 4.3
 p\_load(lattice)
 xyplot(X3BG~HRG, data=table.4.2, xlab="Home Runs per Game", ylab="Triples per Game")



 # Note that we could have computed the rates as part of the plot call
 xyplot(X3B/G ~ HR/G, data=table.4.2)



 ### Create Figure 4.4
 # To create the factor variable that contains the year groups we use "cut"
 table.4.2$yearGRP = cut(table.4.2$yearID,c(1901,1926,1951,1976,2001),include.lowest=TRUE,labels=c("1901-1925","1926-1950","1951-1975","1976-2000"))
 p\_load(ggplot2)
 ggplot(table.4.2, aes(x=HRG, y=X3BG, shape=yearGRP, col=yearGRP)) + geom\_point() + xlab("Home Runs per Game") + ylab("Triples per Game")



### Case 4.2

Albert now turns to looking at offensive statistics that are associated with scoring runs. We will use the data from Albert's site.

 ### Albert starts off by atlking about some variables. He then generates
 ### Figure 4.5 from the 2000 team data that was used in Case 4.1
 head(case.4.1)

## Team League Avg OBP Slg AB R H X2B X3B HR RBI BB
## 1 Anaheim American 0.280 0.352 0.472 5628 864 1574 309 34 236 837 608
## 2 Baltimore American 0.272 0.341 0.435 5549 794 1508 310 22 184 750 558
## 3 Boston American 0.267 0.341 0.423 5630 792 1503 316 32 167 755 611
## 4 Chicago American 0.286 0.356 0.470 5646 978 1615 325 33 216 926 591
## 5 Cleveland American 0.288 0.367 0.470 5683 950 1639 310 30 221 889 685
## 6 Detroit American 0.275 0.343 0.438 5644 823 1553 307 41 177 785 562
## SO HBP Team3
## 1 1024 47 Ana
## 2 900 49 Bal
## 3 1019 42 Bos
## 4 960 53 Chi
## 5 1057 51 Cle
## 6 982 43 Det

 pairs(case.4.1[,c("R","OBP","Slg","X2B","X3B","HR")])



 ### The correlations presented in Table 4.3 are easily computed
 case.4.1$TB = case.4.1$AB \* case.4.1$Slg
 round(cor(case.4.1[,c("R","H","X2B","X3B","HR","RBI","Avg","TB","Slg","OBP")]),3)

## R H X2B X3B HR RBI Avg TB Slg OBP
## R 1.000 0.749 0.233 0.080 0.638 0.995 0.798 0.872 0.872 0.941
## H 0.749 1.000 0.546 0.293 0.225 0.735 0.983 0.795 0.697 0.696
## X2B 0.233 0.546 1.000 0.420 -0.063 0.243 0.467 0.442 0.335 0.186
## X3B 0.080 0.293 0.420 1.000 -0.338 0.081 0.278 0.107 0.062 0.064
## HR 0.638 0.225 -0.063 -0.338 1.000 0.673 0.288 0.744 0.822 0.592
## RBI 0.995 0.735 0.243 0.081 0.673 1.000 0.783 0.889 0.892 0.933
## Avg 0.798 0.983 0.467 0.278 0.288 0.783 1.000 0.816 0.751 0.768
## TB 0.872 0.795 0.442 0.107 0.744 0.889 0.816 1.000 0.980 0.804
## Slg 0.872 0.697 0.335 0.062 0.822 0.892 0.751 0.980 1.000 0.826
## OBP 0.941 0.696 0.186 0.064 0.592 0.933 0.768 0.804 0.826 1.000

 # Oddly, these values are not the same as Albert's
 ### The data in case\_4\_2.text on ALbert's site are not used.
 case.4.2 = read.delim("http://www-math.bgsu.edu/~albert/tsub/datasets/case\_4\_2.txt", header=TRUE, comment.char="#")
 head(case.4.2)

## TEAM YEAR HR.G X3.G
## 1 DET 1901 0.21 0.59
## 2 NY1 1906 0.10 0.35
## 3 DET 1907 0.07 0.49
## 4 WS1 1908 0.05 0.48
## 5 CHN 1911 0.34 0.64
## 6 PHI 1913 0.46 0.49

 dim(case.4.2)

## [1] 30 4

### Case 4.3

We now look at the most valuable hitting statistics. In this section we get to play with linear regression and prediction. We will use the data from Albert's site.

 ### We get the Table 4.4 data from ALbert's site.
 case.4.3 = read.delim("http://www-math.bgsu.edu/~albert/tsub/datasets/case\_4\_3.txt", header=TRUE, comment.char="#")
 head(case.4.3)

## Team League Avg OBP Slg AB R H X2B X3B HR RBI BB
## 1 Anaheim American 0.280 0.352 0.472 5628 864 1574 309 34 236 837 608
## 2 Baltimore American 0.272 0.341 0.435 5549 794 1508 310 22 184 750 558
## 3 Boston American 0.267 0.341 0.423 5630 792 1503 316 32 167 755 611
## 4 Chicago American 0.286 0.356 0.470 5646 978 1615 325 33 216 926 591
## 5 Cleveland American 0.288 0.367 0.470 5683 950 1639 310 30 221 889 685
## 6 Detroit American 0.275 0.343 0.438 5644 823 1553 307 41 177 785 562
## SO HBP
## 1 1024 47
## 2 900 49
## 3 1019 42
## 4 960 53
## 5 1057 51
## 6 982 43

 # Oops. Albert forgot games and runs per game
 # Calculate runs per game
 case.4.3$G = rep(162, nrow(case.4.3))
 case.4.3$G[c(9,10,12)] = 161
 case.4.3$rpg = case.4.3$R/case.4.3$G
 head(case.4.3)

## Team League Avg OBP Slg AB R H X2B X3B HR RBI BB
## 1 Anaheim American 0.280 0.352 0.472 5628 864 1574 309 34 236 837 608
## 2 Baltimore American 0.272 0.341 0.435 5549 794 1508 310 22 184 750 558
## 3 Boston American 0.267 0.341 0.423 5630 792 1503 316 32 167 755 611
## 4 Chicago American 0.286 0.356 0.470 5646 978 1615 325 33 216 926 591
## 5 Cleveland American 0.288 0.367 0.470 5683 950 1639 310 30 221 889 685
## 6 Detroit American 0.275 0.343 0.438 5644 823 1553 307 41 177 785 562
## SO HBP G rpg
## 1 1024 47 162 5.333333
## 2 900 49 162 4.901235
## 3 1019 42 162 4.888889
## 4 960 53 162 6.037037
## 5 1057 51 162 5.864198
## 6 982 43 162 5.080247

 # Figure 4.6 looks like
 plot(rpg~Avg, data=case.4.3, ylab="Runs per Game")



 # We compute a linear regression line
 summary(lm(rpg~Avg, data=case.4.3))

##
## Call:
## lm(formula = rpg ~ Avg, data = case.4.3)
##
## Residuals:
## Min 1Q Median 3Q Max
## -0.4993 -0.2577 -0.1195 0.1371 0.7654
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.683 3.161 -1.165 0.2666
## Avg 32.591 11.468 2.842 0.0148 \*
## ---
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3683 on 12 degrees of freedom
## Multiple R-squared: 0.4023, Adjusted R-squared: 0.3525
## F-statistic: 8.076 on 1 and 12 DF, p-value: 0.01485

 coef(lm(rpg~Avg, data=case.4.3))

## (Intercept) Avg
## -3.683006 32.590901

 # Albert starts looking at goodness of fit. R helps us with this.
 rpg.lm.avg = lm(rpg~Avg, data=case.4.3)
 case.4.3$predrpg = predict(rpg.lm.avg)
 case.4.3$residrpg = resid(rpg.lm.avg)
 # Figure 4.7
 plot(rpg~Avg, data=case.4.3)
 abline(rpg.lm.avg)
 segments(case.4.3$Avg, case.4.3$rpg, case.4.3$Avg, case.4.3$predrpg, col="red")
 text(case.4.3$Avg[c(8,10)]+0.0025, case.4.3$rpg[c(8,10)] , case.4.1$Team[c(8,10)], cex=0.75)



 # We now look at the Mean Squared Error (MSE)
 case.4.3$sqresidrpg = case.4.3$residrpg^2
 n = nrow(case.4.3)
 mse = sum(case.4.3$sqresidrpg)/n
 mse

## [1] 0.1162806

 rmse = sqrt(mse)
 rmse

## [1] 0.3409995

 sort(case.4.3$residrpg)

## [1] -0.49925301 -0.30565077 -0.28048419 -0.27724725 -0.19924455
## [6] -0.14006022 -0.12987537 -0.10911263 0.03532335 0.06526462
## [11] 0.16102436 0.39904567 0.51481937 0.76545062

 # Albert suggests that the residuals are normally distributed. R helps us
 # check this.
 plot(rpg.lm.avg)



 # Or on one page...
 par(mfrow=c(2,2))
 plot(rpg.lm.avg)



 par(mfrow=c(1,1))
 # Next we try OBP
 rpg.lm.obp = lm(rpg~OBP, data=case.4.3)
 coef(rpg.lm.obp)

## (Intercept) OBP
## -8.248087 38.839457

 # Table 4.6
 cbind(case.4.3$Team, round(case.4.3[,c("OBP","rpg")],3), pred=round(predict(rpg.lm.obp),3), res=round(resid(rpg.lm.obp),3), sqrres=round(resid(rpg.lm.obp)^2,3))

## case.4.3$Team OBP rpg pred res sqrres
## 1 Anaheim 0.352 5.333 5.423 -0.090 0.008
## 2 Baltimore 0.341 4.901 4.996 -0.095 0.009
## 3 Boston 0.341 4.889 4.996 -0.107 0.012
## 4 Chicago 0.356 6.037 5.579 0.458 0.210
## 5 Cleveland 0.367 5.864 6.006 -0.142 0.020
## 6 Detroit 0.343 5.080 5.074 0.006 0.000
## 7 Kansas City 0.348 5.426 5.268 0.158 0.025
## 8 Minnesota 0.337 4.617 4.841 -0.224 0.050
## 9 New York 0.354 5.410 5.501 -0.091 0.008
## 10 Oakland 0.360 5.882 5.734 0.148 0.022
## 11 Seattle 0.361 5.599 5.773 -0.174 0.030
## 12 Tampa Bay 0.329 4.553 4.530 0.023 0.001
## 13 Texas 0.352 5.235 5.423 -0.189 0.036
## 14 Toronto 0.341 5.315 4.996 0.319 0.102

 mse = sum(mean(resid(rpg.lm.obp)^2))
 rmse = sqrt(mse)
 c(mse, rmse)

## [1] 0.03799386 0.19492014

### Case 4.4

We now switch to looking at multiple predictor variables by using multiple linear regression.

 ### The initial model that Albert fits uses the 2000 data from Case 4.1
 head(case.4.1)

## Team League Avg OBP Slg AB R H X2B X3B HR RBI BB
## 1 Anaheim American 0.280 0.352 0.472 5628 864 1574 309 34 236 837 608
## 2 Baltimore American 0.272 0.341 0.435 5549 794 1508 310 22 184 750 558
## 3 Boston American 0.267 0.341 0.423 5630 792 1503 316 32 167 755 611
## 4 Chicago American 0.286 0.356 0.470 5646 978 1615 325 33 216 926 591
## 5 Cleveland American 0.288 0.367 0.470 5683 950 1639 310 30 221 889 685
## 6 Detroit American 0.275 0.343 0.438 5644 823 1553 307 41 177 785 562
## SO HBP Team3 TB
## 1 1024 47 Ana 2656.416
## 2 900 49 Bal 2413.815
## 3 1019 42 Bos 2381.490
## 4 960 53 Chi 2653.620
## 5 1057 51 Cle 2671.010
## 6 982 43 Det 2472.072

 ### Unfortunately the Runs per Game variable we need is not included.
 ### Since Games are also not included, we get the data from Lahman.
 require(pacman)
 p\_load(Lahman, sqldf)
 data(Teams)
 MLB2000 = sqldf("SELECT \* FROM Teams WHERE (yearID=2000 AND (lgID='AL' OR lgID='NL'))")
 MLB2000$X1B = MLB2000$H - (MLB2000$X2B + MLB2000$X3B + MLB2000$HR)
 MLB2000$rpg = MLB2000$R/MLB2000$G
 head(MLB2000)

## yearID lgID teamID franchID divID Rank G Ghome W L DivWin WCWin
## 1 2000 AL ANA ANA W 3 162 81 82 80 N N
## 2 2000 NL ARI ARI W 3 162 81 85 77 N N
## 3 2000 NL ATL ATL E 1 162 81 95 67 Y N
## 4 2000 AL BAL BAL E 4 162 81 74 88 N N
## 5 2000 AL BOS BOS E 2 162 81 85 77 N N
## 6 2000 AL CHA CHW C 1 162 81 95 67 Y N
## LgWin WSWin R AB H X2B X3B HR BB SO SB CS HBP SF RA ER
## 1 N N 864 5628 1574 309 34 236 608 1024 93 52 47 43 869 805
## 2 N N 792 5527 1466 282 44 179 535 975 97 44 59 58 754 698
## 3 N N 810 5489 1490 274 26 179 595 1010 148 56 59 45 714 648
## 4 N N 794 5549 1508 310 22 184 558 900 126 65 49 54 913 855
## 5 N N 792 5630 1503 316 32 167 611 1019 43 30 42 48 745 683
## 6 N N 978 5646 1615 325 33 216 591 960 119 42 53 61 839 751
## ERA CG SHO SV IPouts HA HRA BBA SOA E DP FP
## 1 5.00 5 3 46 4344 1534 228 662 846 134 182 0.978
## 2 4.35 16 8 38 4331 1441 190 500 1220 107 138 0.982
## 3 4.05 13 9 53 4321 1428 165 484 1093 129 138 0.979
## 4 5.37 14 6 33 4300 1547 202 665 1017 116 151 0.981
## 5 4.23 7 12 46 4358 1433 173 498 1121 109 120 0.982
## 6 4.66 5 7 43 4351 1509 195 614 1037 133 190 0.978
## name park attendance BPF PPF
## 1 Anaheim Angels Edison International Field 2066982 102 103
## 2 Arizona Diamondbacks Bank One Ballpark 2942251 105 103
## 3 Atlanta Braves Turner Field 3234304 101 99
## 4 Baltimore Orioles Oriole Park at Camden Yards 3297031 95 96
## 5 Boston Red Sox Fenway Park II 2585895 104 103
## 6 Chicago White Sox Comiskey Park II 1947799 102 102
## teamIDBR teamIDlahman45 teamIDretro X1B rpg
## 1 ANA ANA ANA 995 5.333333
## 2 ARI ARI ARI 961 4.888889
## 3 ATL ATL ATL 1011 5.000000
## 4 BAL BAL BAL 992 4.901235
## 5 BOS BOS BOS 988 4.888889
## 6 CHW CHA CHA 1041 6.037037

 ### We can now fit the optimized average
 rpg.lm.optavg = lm(rpg ~ X1B + X2B + X3B + HR + BB + HBP, data=MLB2000)
 coef(rpg.lm.optavg)

## (Intercept) X1B X2B X3B HR
## -4.212230095 0.004834378 0.002695029 0.008346529 0.008300588
## BB HBP
## 0.002961090 0.002417944

 ### It is likely that the differences that we are seeing here are the result
 ### of Albert having rounded his data.
 # The RMSE is
 sqrt(sum(mean(resid(rpg.lm.optavg)^2)))

## [1] 0.1267053

 # Albert fits other single variable models to fill Table 4.9
 MLB2000$avg = MLB2000$H/MLB2000$AB
 rpg.lm.avg = lm(rpg ~ avg, data=MLB2000)
 sqrt(sum(mean(resid(rpg.lm.avg)^2)))

## [1] 0.2902596

 MLB2000$tb = 1\*MLB2000$X1B + 2\*MLB2000$X2B + 3\*MLB2000$X3B + 4\*MLB2000$HR
 MLB2000$slg = MLB2000$tb/MLB2000$AB
 rpg.lm.slg = lm(rpg ~ slg, data=MLB2000)
 sqrt(sum(mean(resid(rpg.lm.slg)^2)))

## [1] 0.2376677

 MLB2000$obp = (MLB2000$H + MLB2000$BB + MLB2000$HBP)/(MLB2000$AB + MLB2000$BB + MLB2000$HBP + MLB2000$SF)
 rpg.lm.obp = lm(rpg ~ obp, data=MLB2000)
 sqrt(sum(mean(resid(rpg.lm.obp)^2)))

## [1] 0.1630679

 MLB2000$rc = (MLB2000$H + MLB2000$BB)\*MLB2000$tb/(MLB2000$AB + MLB2000$BB)
 rpg.lm.rc = lm(rpg ~ rc, data=MLB2000)
 sqrt(sum(mean(resid(rpg.lm.rc)^2)))

## [1] 0.1621795

 MLB2000$ops = MLB2000$obp + MLB2000$slg
 rpg.lm.ops = lm(rpg ~ ops, data=MLB2000)
 sqrt(sum(mean(resid(rpg.lm.ops)^2)))

## [1] 0.1772865

 ### Table 4.10 is easy to create using information from above
 round(rbind(coef(rpg.lm.optavg)[-1],coef(rpg.lm.optavg)[-1]/coef(rpg.lm.optavg)[2]),5)

## X1B X2B X3B HR BB HBP
## [1,] 0.00483 0.00270 0.00835 0.00830 0.00296 0.00242
## [2,] 1.00000 0.55747 1.72649 1.71699 0.61251 0.50016

### Case 4.5

We now switch to looking at multiple predictor variables by using non-linear (in the coefficients) regression.

 ### We get the Table 4.11 data from ALbert's site.
 case.4.5 = read.delim("http://www-math.bgsu.edu/~albert/tsub/datasets/case\_4\_5.txt", header=TRUE, comment.char="#")
 head(case.4.5)

## Team W L R RA X
## 1 Anaheim 82 80 864 869 NA
## 2 Arizona 85 77 792 754 NA
## 3 Atlanta 95 67 810 714 NA
## 4 Baltimore 74 88 794 913 NA
## 5 Boston 85 77 792 745 NA
## 6 Chicago\_AL 95 67 978 839 NA

 # Get rid of the "X" column
 case.4.5 = case.4.5[,-6]
 head(case.4.5)

## Team W L R RA
## 1 Anaheim 82 80 864 869
## 2 Arizona 85 77 792 754
## 3 Atlanta 95 67 810 714
## 4 Baltimore 74 88 794 913
## 5 Boston 85 77 792 745
## 6 Chicago\_AL 95 67 978 839

 ### Alternatively, we could pull the table from Lahman's data.
 require(pacman)
 p\_load(Lahman, sqldf)
 data(Teams)
 tbl.4.11 = sqldf("SELECT teamId, W, L, R, RA FROM Teams WHERE (yearID=2000 AND (lgID='AL' OR lgID='NL')) ORDER BY teamID")
 head(tbl.4.11)

## teamID W L R RA
## 1 ANA 82 80 864 869
## 2 ARI 85 77 792 754
## 3 ATL 95 67 810 714
## 4 BAL 74 88 794 913
## 5 BOS 85 77 792 745
## 6 CHA 95 67 978 839

 # Create Figure 4.8
 tbl.4.11$lnwl = log(tbl.4.11$W/tbl.4.11$L)
 tbl.4.11$lnrra = log(tbl.4.11$R/tbl.4.11$RA)
 plot(lnwl~lnrra, data=tbl.4.11, xlab="log(R/RA)", ylab="log(W/L)")



 # Fit the linearized model through the point (0,0)---i.e. without intercept
 tbl.4.11.lm = lm(lnwl ~ lnrra - 1, data=tbl.4.11)
 summary(tbl.4.11.lm)

##
## Call:
## lm(formula = lnwl ~ lnrra - 1, data = tbl.4.11)
##
## Residuals:
## Min 1Q Median 3Q Max
## -0.210901 -0.044967 0.006942 0.036033 0.152170
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## lnrra 1.9201 0.1237 15.52 1.39e-15 \*\*\*
## ---
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08163 on 29 degrees of freedom
## Multiple R-squared: 0.8926, Adjusted R-squared: 0.8888
## F-statistic: 240.9 on 1 and 29 DF, p-value: 1.388e-15

 # Try again using data from Albert's site
 # Create Figure 4.8
 case.4.5$lnwl = log(case.4.5$W/case.4.5$L)
 case.4.5$lnrra = log(case.4.5$R/case.4.5$RA)
 plot(lnwl~lnrra, data=case.4.5, xlab="log(R/RA)", ylab="log(W/L)")



 # Fit the linearized model through the point (0,0)---i.e. without intercept
 case.4.5.lm = lm(lnwl ~ lnrra - 1, data=case.4.5)
 summary(case.4.5.lm)

##
## Call:
## lm(formula = lnwl ~ lnrra - 1, data = case.4.5)
##
## Residuals:
## Min 1Q Median 3Q Max
## -0.210901 -0.044967 0.006942 0.036033 0.152170
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## lnrra 1.9201 0.1237 15.52 1.39e-15 \*\*\*
## ---
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08163 on 29 degrees of freedom
## Multiple R-squared: 0.8926, Adjusted R-squared: 0.8888
## F-statistic: 240.9 on 1 and 29 DF, p-value: 1.388e-15

 # As found in the text, the coefficient is approximately 2
 coef(tbl.4.11.lm)

## lnrra
## 1.920082

 # Figure 4.9 can be plotted as
 par(mfrow=c(2,1))
 plot(lnwl ~ lnrra, data=tbl.4.11, xlab="log(R/RA)", ylab="log(W/L)")
 abline(tbl.4.11.lm)
 plot(resid(tbl.4.11.lm)~tbl.4.11$lnrra, xlab="log(R/RA)", ylab="Residual")
 abline(h=0, lty=2)



 par(mfrow=c(1,1))
 # Note that R has the default diagnostic plots
 par(mfrow=c(2,2))
 plot(tbl.4.11.lm)



 par(mfrow=c(1,1))
 # Table 4.12 is generated using R functions "pred" and "resid"
 tbl.4.11$predw = tbl.4.11$L\*(tbl.4.11$R/tbl.4.11$RA)^1.91 #coef(tbl.4.11.lm)
 tbl.4.11$residw = tbl.4.11$W - tbl.4.11$predw
 cbind(tbl.4.11[,c("teamID","W","L")], Predicted=round(tbl.4.11$predw,1), Residual=round(tbl.4.11$residw,3))

## teamID W L Predicted Residual
## 1 ANA 82 80 79.1 2.877
## 2 ARI 85 77 84.6 0.418
## 3 ATL 95 67 85.3 9.745
## 4 BAL 74 88 67.4 6.603
## 5 BOS 85 77 86.5 -1.544
## 6 CHA 95 67 89.8 5.208
## 7 CHN 65 97 70.3 -5.339
## 8 CIN 85 77 88.9 -3.946
## 9 CLE 90 72 96.3 -6.262
## 10 COL 82 80 92.5 -10.529
## 11 DET 79 83 82.2 -3.235
## 12 FLO 79 82 69.5 9.480
## 13 HOU 72 90 88.9 -16.911
## 14 KCA 77 85 76.3 0.681
## 15 LAN 86 76 90.3 -4.329
## 16 MIL 73 89 72.1 0.858
## 17 MIN 69 93 68.2 0.817
## 18 MON 67 95 64.8 2.246
## 19 NYA 87 74 84.2 2.788
## 20 NYN 94 68 80.7 13.342
## 21 OAK 91 70 93.7 -2.681
## 22 PHI 65 97 71.6 -6.597
## 23 PIT 69 93 74.9 -5.925
## 24 SDN 76 86 73.8 2.250
## 25 SEA 91 71 94.7 -3.708
## 26 SFN 97 65 97.8 -0.769
## 27 SLN 95 67 87.6 7.434
## 28 TBA 69 92 70.6 -1.598
## 29 TEX 71 91 69.8 1.156
## 30 TOR 83 79 71.4 11.626

 # These values do not match Albert's. We solve for k using his values.
 log(80.6/80)/log(864/869) # Angels

## [1] -1.294897

 log(88/76)/log(798/729) # Dodgers

## [1] 1.621095

 # Both should have been 1.91. Try the same for our predicted values.
 log(79.12313/80)/log(864/869) # Angels

## [1] 1.910001

 log(90.32949/76)/log(798/729) # Dodgers

## [1] 1.910001

 # It looks like Albert made an error
 #
 # Figure 4.10 is easy although, based on our results, Albert's contention that
 # we hit 80% prediction within 4 runs is a joke. It's more like 17/30 or 23%.
 p\_load(aplpack)
 stem.leaf(abs(tbl.4.11$residw))

## 1 | 2: represents 1.2
## leaf unit: 0.1
## n: 30
## 5 0 | 46788
## 8 1 | 155
## 13 2 | 22678
## (3) 3 | 279
## 14 4 | 3
## 13 5 | 239
## 10 6 | 256
## 7 7 | 4
## 8 |
## 6 9 | 47
## 4 10 | 5
## 3 11 | 6
## 12 |
## 2 13 | 3
## HI: 16.9105755572982

### Case 4.6

Albert now brings up the concept of regression to the mean --- those things that were big will get smaller and those that were small will get bigger. He focuses on OBP, but could have used any baseball stat.

 ### We get the Table 4.13 data from ALbert's site.
 case.4.6 = read.delim("http://www-math.bgsu.edu/~albert/tsub/datasets/case\_4\_6.txt", header=TRUE, comment.char="#")
 head(case.4.6)

## X OBP\_1999 OBP\_2000 IMPROVEMENT
## 1 Jeter 438 416 -22
## 2 O'Neill 353 336 -17
## 3 Justice 413 377 -36
## 4 Williams 435 391 -44
## 5 Brosius 307 299 -8
## 6 Knoblauch 393 366 -27

 ### Figure 4.11 and the regression line are computed as above.
 plot(OBP\_2000 ~ OBP\_1999, data=case.4.6, xlab="OBP 1999", ylab="OBP 2000")
 case.4.6.lm1 = lm(OBP\_2000 ~ OBP\_1999, case.4.6)
 abline(case.4.6.lm1)



 coef(case.4.6.lm1)

## (Intercept) OBP\_1999
## 138.7129210 0.6146784

 summary(case.4.6)

## X OBP\_1999 OBP\_2000 IMPROVEMENT
## Agbayani: 1 Min. :307.0 Min. :299.0 Min. :-44.000
## Alfonzo : 1 1st Qu.:341.0 1st Qu.:337.5 1st Qu.:-27.250
## Bordick : 1 Median :362.0 Median :362.0 Median :-13.000
## Brosius : 1 Mean :369.6 Mean :365.9 Mean : -3.688
## Hamilton: 1 3rd Qu.:387.8 3rd Qu.:392.8 3rd Qu.: 12.250
## Harris : 1 Max. :438.0 Max. :425.0 Max. : 76.000
## (Other) :10

 # Compute William's values
 c(1,435)%\*%coef(case.4.6.lm1)

## [,1]
## [1,] 406.098

 c(1,435)%\*%coef(case.4.6.lm1)-mean(case.4.6$OBP\_2000)

## [,1]
## [1,] 40.22302

 ### Now look at improvement in Figure 4.12 and the regression model
 plot(IMPROVEMENT ~ OBP\_1999, data=case.4.6, xlab="OBP 1999", ylab="Improvement in OBP")
 case.4.6.lm2 = lm(IMPROVEMENT ~ OBP\_1999, case.4.6)
 abline(case.4.6.lm2)



 cor(case.4.6[,-1])

## OBP\_1999 OBP\_2000 IMPROVEMENT
## OBP\_1999 1.0000000 0.6036335 -0.4287781
## OBP\_2000 0.6036335 1.0000000 0.4614295
## IMPROVEMENT -0.4287781 0.4614295 1.0000000

 coef(case.4.6.lm2)

## (Intercept) OBP\_1999
## 138.7129210 -0.3853216

 ### Albert doesn't look at residuals, but you ought to.
 par(mfrow=c(2,2))
 plot(case.4.6.lm1)



 plot(case.4.6.lm2)



 par(mfrow=c(1,1))

### Case 4.7

Albert's last case study in this section looks at home run production during consecutive years (1999 and 2000). This is an analysis of the so-called "Dinger Drop-off". We'll use the data from the web site.

 ### We pull down team data for MLB teams in 2000 and 2001.
 p\_load(Lahman, sqldf, plyr)
 data(Teams)
 MLB2000 = sqldf("SELECT \* FROM Teams WHERE (yearID=2000 AND (lgID='AL' OR lgID='NL'))")
 head(MLB2000)

## yearID lgID teamID franchID divID Rank G Ghome W L DivWin WCWin
## 1 2000 AL ANA ANA W 3 162 81 82 80 N N
## 2 2000 NL ARI ARI W 3 162 81 85 77 N N
## 3 2000 NL ATL ATL E 1 162 81 95 67 Y N
## 4 2000 AL BAL BAL E 4 162 81 74 88 N N
## 5 2000 AL BOS BOS E 2 162 81 85 77 N N
## 6 2000 AL CHA CHW C 1 162 81 95 67 Y N
## LgWin WSWin R AB H X2B X3B HR BB SO SB CS HBP SF RA ER
## 1 N N 864 5628 1574 309 34 236 608 1024 93 52 47 43 869 805
## 2 N N 792 5527 1466 282 44 179 535 975 97 44 59 58 754 698
## 3 N N 810 5489 1490 274 26 179 595 1010 148 56 59 45 714 648
## 4 N N 794 5549 1508 310 22 184 558 900 126 65 49 54 913 855
## 5 N N 792 5630 1503 316 32 167 611 1019 43 30 42 48 745 683
## 6 N N 978 5646 1615 325 33 216 591 960 119 42 53 61 839 751
## ERA CG SHO SV IPouts HA HRA BBA SOA E DP FP
## 1 5.00 5 3 46 4344 1534 228 662 846 134 182 0.978
## 2 4.35 16 8 38 4331 1441 190 500 1220 107 138 0.982
## 3 4.05 13 9 53 4321 1428 165 484 1093 129 138 0.979
## 4 5.37 14 6 33 4300 1547 202 665 1017 116 151 0.981
## 5 4.23 7 12 46 4358 1433 173 498 1121 109 120 0.982
## 6 4.66 5 7 43 4351 1509 195 614 1037 133 190 0.978
## name park attendance BPF PPF
## 1 Anaheim Angels Edison International Field 2066982 102 103
## 2 Arizona Diamondbacks Bank One Ballpark 2942251 105 103
## 3 Atlanta Braves Turner Field 3234304 101 99
## 4 Baltimore Orioles Oriole Park at Camden Yards 3297031 95 96
## 5 Boston Red Sox Fenway Park II 2585895 104 103
## 6 Chicago White Sox Comiskey Park II 1947799 102 102
## teamIDBR teamIDlahman45 teamIDretro
## 1 ANA ANA ANA
## 2 ARI ARI ARI
## 3 ATL ATL ATL
## 4 BAL BAL BAL
## 5 BOS BOS BOS
## 6 CHW CHA CHA

 MLB1999 = sqldf("SELECT \* FROM Teams WHERE (yearID=1999 AND (lgID='AL' OR lgID='NL'))")
 head(MLB1999)

## yearID lgID teamID franchID divID Rank G Ghome W L DivWin WCWin
## 1 1999 AL ANA ANA W 4 162 81 70 92 N N
## 2 1999 NL ARI ARI W 1 162 81 100 62 Y N
## 3 1999 NL ATL ATL E 1 162 81 103 59 Y N
## 4 1999 AL BAL BAL E 4 162 81 78 84 N N
## 5 1999 AL BOS BOS E 2 162 81 94 68 N Y
## 6 1999 AL CHA CHW C 2 162 81 75 86 N N
## LgWin WSWin R AB H X2B X3B HR BB SO SB CS HBP SF RA ER
## 1 N N 711 5494 1404 248 22 158 511 1022 71 45 NA NA 826 762
## 2 N N 908 5658 1566 289 46 216 588 1045 137 39 NA NA 676 615
## 3 Y N 840 5569 1481 309 23 197 608 962 148 66 NA NA 661 593
## 4 N N 851 5637 1572 299 21 203 615 890 107 46 NA NA 815 761
## 5 N N 836 5579 1551 334 42 176 597 928 67 39 NA NA 718 638
## 6 N N 777 5644 1563 298 37 162 499 810 110 50 NA NA 870 786
## ERA CG SHO SV IPouts HA HRA BBA SOA E DP FP
## 1 4.79 4 7 37 4293 1472 177 624 877 106 156 0.98
## 2 3.77 16 9 42 4401 1387 176 543 1198 104 129 0.98
## 3 3.63 9 9 45 4413 1398 142 507 1197 111 122 0.98
## 4 4.77 17 11 33 4305 1468 198 647 982 89 192 0.98
## 5 4.00 6 12 50 4309 1396 160 469 1131 127 131 0.97
## 6 4.92 6 3 39 4314 1608 210 596 968 136 148 0.97
## name park attendance BPF PPF
## 1 Anaheim Angels Edison International Field 2253123 99 100
## 2 Arizona Diamondbacks Bank One Ballpark 3019654 101 101
## 3 Atlanta Braves Turner Field 3284897 100 98
## 4 Baltimore Orioles Oriole Park at Camden Yards 3433150 96 96
## 5 Boston Red Sox Fenway Park II 2446162 104 103
## 6 Chicago White Sox Comiskey Park II 1338851 101 101
## teamIDBR teamIDlahman45 teamIDretro
## 1 ANA ANA ANA
## 2 ARI ARI ARI
## 3 ATL ATL ATL
## 4 BAL BAL BAL
## 5 BOS BOS BOS
## 6 CHW CHA CHA

 ### Figure 4.13
 p\_load(aplpack)
 stem.leaf.backback(MLB1999$HR, MLB2000$HR)

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 1 | 2: represents 12, leaf unit: 1
## MLB1999$HR MLB2000$HR
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## 1 5| 10 |
## | 11 |6 1
## 2 8| 12 |
## | 13 |
## 3 5| 14 |4 2
## 6 831| 15 |07 4
## 11 85321| 16 |01278 9
## 13 61| 17 |377899 15
## (4) 9871| 18 |34 (2)
## 13 743| 19 |88 13
## 10 993| 20 |05 11
## 7 622| 21 |16 9
## 4 3| 22 |16 7
## 3 50| 23 |569 5
## 1 4| 24 |49 2
## | 25 |
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
## n: 30 30
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 ### Five number summaries (with means)
 summary(MLB1999$HR)

## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 105.0 162.2 187.5 184.3 209.0 244.0

 summary(MLB2000$HR)

## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 116.0 167.2 181.0 189.8 214.8 249.0

 sort(MLB2000$HR)

## [1] 116 144 150 157 160 161 162 167 168 173 177 177 178 179 179 183 184
## [18] 198 198 200 205 211 216 221 226 235 236 239 244 249

 sort(MLB2000$HR)[c(15,16)]

## [1] 179 183

 # Apparently, Albert miscounted when computing his median
 ### Compute total HRs
 mean(MLB1999$HR)\*30

## [1] 5528

 mean(MLB2000$HR)\*30

## [1] 5693

 sum(MLB1999$HR)

## [1] 5528

 sum(MLB2000$HR)

## [1] 5693

 sum(MLB2000$HR)-sum(MLB1999$HR)

## [1] 165

 ### We get the Pre/Post data from ALbert's site.
 case.4.7 = read.delim("http://www-math.bgsu.edu/~albert/tsub/datasets/case\_4\_7.txt", header=TRUE, comment.char="#")
 head(case.4.7)

## TEAM PRE\_HR PRE\_AB PRE\_RATE POST\_HR POST\_AB POST\_RATE
## 1 Anaheim 141 3059 0.046 95 2569 0.037
## 2 Baltimore 117 2969 0.039 67 2580 0.026
## 3 Boston 98 2937 0.033 69 2693 0.026
## 4 Chi\_AL 128 3057 0.042 88 2589 0.034
## 5 Cleveland 132 3017 0.044 89 2666 0.033
## 6 Detroit 103 2946 0.035 74 2698 0.027

 ### Figure 4.14
 plot(POST\_RATE ~ PRE\_RATE, data=case.4.7, xlab="Pre Home Run Rate", ylab="Post Home Run Rate")
 abline(a=0, b=1) # Intercept=0, Slope=1



 # The correlation is computed as above
 cor(case.4.7$PRE\_RATE,case.4.7$POST\_RATE)

## [1] 0.6789298

## Note

The above code is intended to show one or two ways of computing some of the statistics that might be used in looking at sports related data. In a few cases, our results differed from Albert's. While this is unfortunate, the process of identifying possible reasons for these differences are very informative. In the real world it is important to verify results. Having experience in debugging analyses is a good thing. While I am pretty sure that Albert did not intend to include the errors that we found, we should be glad that we had the opportunity to find them.