Bootstrap Methods Using Baseball Salary Data

jdt

2/2/2022

Contents

Background

The Sashelp.Baseball data set contains salary and performance information for Major League Baseball players (excluding pitchers) who played at least one game in both the 1986 and 1987 seasons (Time Inc. 1987). The salaries are for the 1987 season, and the performance measures are from the 1986 season. The data set contains 322 observations.

In 1986 the minimum salary was $60.5k and the average salary was $412.5k. The maximum salary was $2,412.5k that was paid to Jim Rice of the Boston Red Sox in the American league. In 1976, Hank Aaron was highest paid player at $240k.

The following graph reflects the significant increase in salaries beginning in the 1980's.

Plan for the Analysis

The baseball data will be used to illustrate the bootstrap procedure using real data. The analysis that I will use is highly restrictive in terms of position players and their tenure in the major league. The variable of interest is player Salary. Salary are not normally distributed (the density is shaped like an exponential curve). Log salary is normally distributed and is often used in models that require normal data. The bootstrap procedure does not need normality.

In this document I illustrate the R and SAS code and output for several functions. The first three in the R output give an indication of the center of the Salary data and the last one, mad, gives an indication of the spread in the data.

Questions

This assignment is open ended as you get to run the R and SAS code with the baseball data of your choice. You might want to know something about length of careers, or American league outfielders salary or career hits.

Just tell me what you want to know and then do it.

If you don't know anything about baseball or R and SAS and you know someone in the class that does. Team up and have at it. I don't object to team work, I just can't assign teams as I don't know your schedules.

R

Needed Packages

```r
if (!require("boot")) install.packages("boot")
if (!require("MASS")) install.packages("MASS")
```

Read Baseball Salary data

```r
baseball <- read.table("baseball.csv", sep = ",", header = TRUE)
# summary(baseball)
```

Subset the data for catchers

```r
base_catcher = subset(baseball, Position == "C")
summary(base_catcher)
```

```
##      Name               Team               nAtBat          nHits
##  Length:40          Length:40          Min.   :127.0   Min.   : 31.00
##  Class :character   Class :character   1st Qu.:210.5   1st Qu.: 55.25
##  Mode  :character   Mode  :character   Median :308.5   Median : 69.50
##                                        Mean   :310.2   Mean   : 76.83
##                                        3rd Qu.:403.5   3rd Qu.: 98.75
##                                        Max.   :528.0   Max.   :147.00
##
##      nHome           nRuns            nRBI            nBB           YrMajor
##  Min.   : 1.0   Min.   :12.00   Min.   :  9.0   Min.   :12.00   Min.   : 1.00
##  1st Qu.: 5.0   1st Qu.:21.75   1st Qu.: 26.0   1st Qu.:17.50   1st Qu.: 3.75
##  Median : 8.0   Median :30.00   Median : 36.5   Median :28.50   Median : 7.00
##  Mean   : 9.2   Mean   :34.10   Mean   : 40.4   Mean   :32.38   Mean   : 7.60
##  3rd Qu.:13.0   3rd Qu.:45.25   3rd Qu.: 53.0   3rd Qu.:40.25   3rd Qu.:10.00
##  Max.   :24.0   Max.   :81.00   Max.   :105.0   Max.   :74.00   Max.   :18.00
##
##     CrAtBat           CrHits           CrHome           CrRuns
##  Min.   : 196.0   Min.   :  44.0   Min.   :  1.00   Min.   :  18.00
##  1st Qu.: 689.8   1st Qu.: 168.2   1st Qu.: 15.75   1st Qu.:  66.75
##  Median :1900.0   Median : 478.0   Median : 38.50   Median : 186.50
##  Mean   :2167.9   Mean   : 555.1   Mean   : 56.15   Mean   : 243.68
##  3rd Qu.:2997.2   3rd Qu.: 824.5   3rd Qu.: 69.00   3rd Qu.: 316.50
##  Max.   :6521.0   Max.   :1767.0   Max.   :281.00   Max.   :1003.00
##
##      CrRbi             CrBB           League            Division
##  Min.   : 10.00   Min.   : 14.00   Length:40          Length:40
##  1st Qu.: 80.25   1st Qu.: 55.75   Class :character   Class :character
##  Median :200.50   Median :163.50   Mode  :character   Mode  :character
##  Mean   :272.93   Mean   :208.07
##  3rd Qu.:380.75   3rd Qu.:277.50
##  Max.   :999.00   Max.   :680.00
##
##    Position             nOuts           nAssts           nError
##  Length:40          Min.   :202.0   Min.   :  9.00   Min.   : 2.00
```

```
##  Class :character   1st Qu.:330.2   1st Qu.: 30.00   1st Qu.: 4.00
##  Mode  :character   Median :444.0   Median : 42.00   Median : 6.00
##                     Mean   :495.3   Mean   : 44.92   Mean   : 7.05
##                     3rd Qu.:683.0   3rd Qu.: 56.00   3rd Qu.: 8.50
##                     Max.   :885.0   Max.   :105.00   Max.   :20.00
##
##      Salary           Div              logSalary       in_fielder  out_fielder
##  Min.   :  75.0   Length:40          Min.   :4.317   Min.   :0    Min.   :0
##  1st Qu.: 141.5   Class :character   1st Qu.:4.951   1st Qu.:0    1st Qu.:0
##  Median : 453.2   Mode  :character   Median :6.115   Median :0    Median :0
##  Mean   : 519.0                      Mean   :5.895   Mean   :0    Mean   :0
##  3rd Qu.: 787.5                      3rd Qu.:6.668   3rd Qu.:0    3rd Qu.:0
##  Max.   :1925.6                      Max.   :7.563   Max.   :0    Max.   :0
##  NA's   :10                          NA's   :10
##     catcher     CrHits2
##  Min.   :1   Min.   :   1936
##  1st Qu.:1   1st Qu.:  28317
##  Median :1   Median : 228605
##  Mean   :1   Mean   : 522720
##  3rd Qu.:1   3rd Qu.: 679837
##  Max.   :1   Max.   :3122289
##
```

Define functions for the bootstrap with Specified Variables

```
# Median
fc_median <- function(d, i) {
    d2 <- d[i, ]
    return(median(d2$Salary, na.rm = TRUE))
}

# Midpoint
fc_mid <- function(d, i) {
    d2 <- d[i, ]
    mid <- (max(d2$Salary, na.rm = TRUE) + min(d2$Salary, na.rm = TRUE))/2
    return(mid)
}

# Trimmed Midpoint [10%]
fc_trimmid <- function(d, i) {
    d2 <- d[i, ]
    mid <- (quantile(d2$Salary, probs = 0.9, na.rm = TRUE) +
        quantile(d2$Salary, probs = 0.1, na.rm = TRUE))/2
    return(mid)
}

# Mean Absolute difference
```

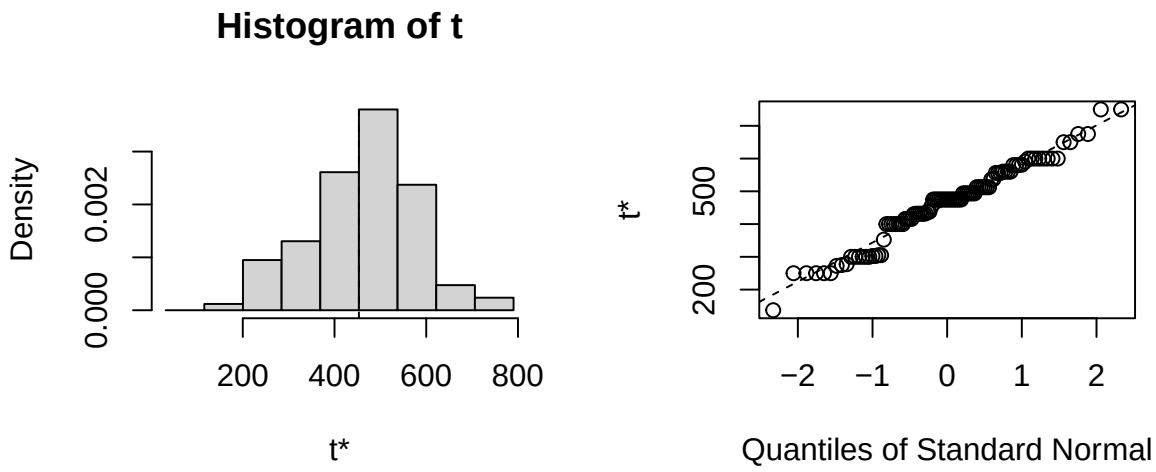
```r
fc_mad <- function(d, i) {
    d2 <- d[i, ]
    mid <- abs(d2$Salary - median(d2$Salary, na.rm = TRUE))
    return(sqrt((mean(mid, na.rm = TRUE))))
}
```

Perform Bootstrap

```r
set.seed(321)  #start the Bootstrap at the same location
# Median
b.median = boot(base_catcher, fc_median, R = 100)
b.median
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = base_catcher, statistic = fc_median, R = 100)
##
##
## Bootstrap Statistics :
##     original  bias    std. error
## t1*   453.25    9.54    119.2608
```

```r
plot(b.median)
```

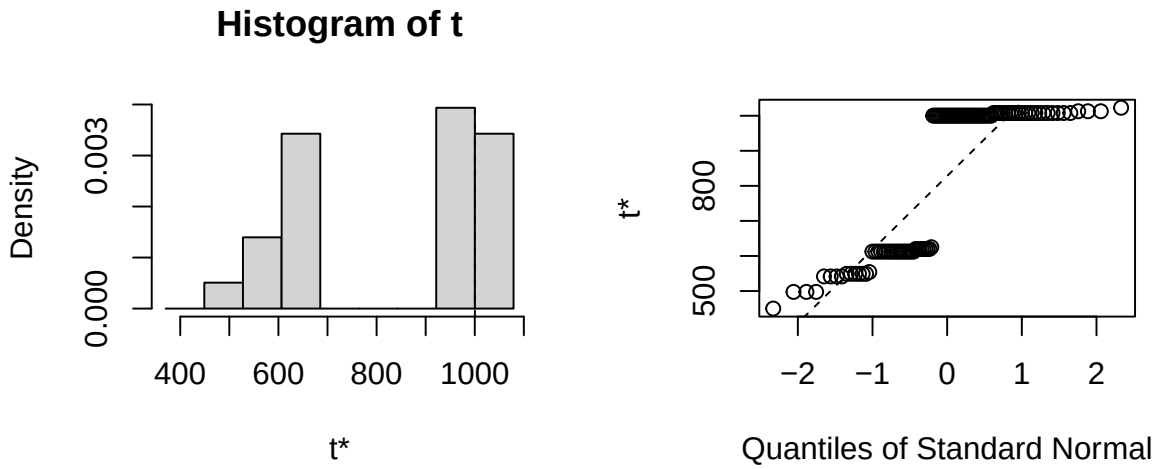



```r
set.seed(321)
# Midpoint
b.mid = boot(base_catcher, fc_mid, R = 100)
b.mid
```

```
##
```

```
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = base_catcher, statistic = fc_mid, R = 100)
##
##
## Bootstrap Statistics :
##     original    bias    std. error
## t1* 1000.285 -172.1116    210.0311
```

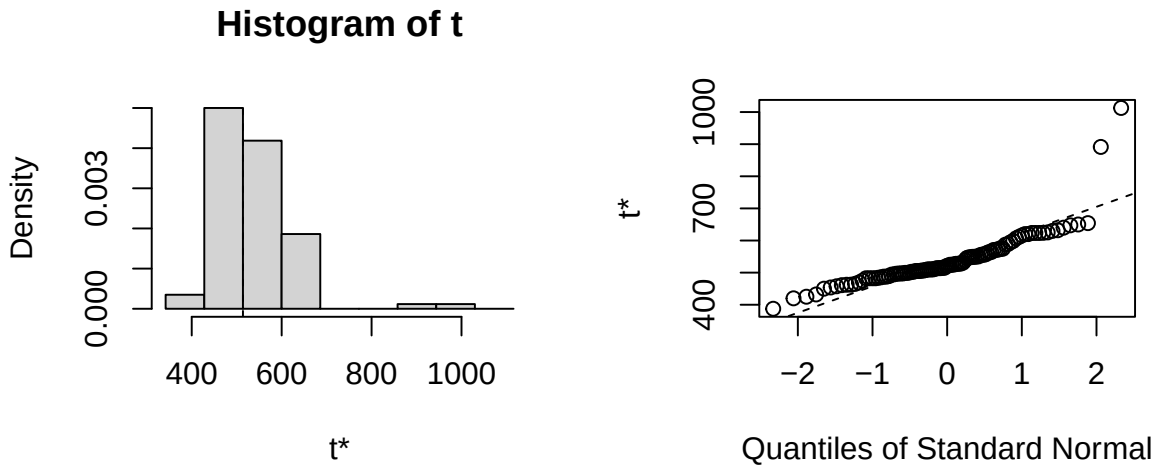
```
plot(b.mid)
```




```
set.seed(321)
# Trimmed Midpoint Using 10% on both sides
b.trimmid = boot(base_catcher, fc_trimmid, R = 100)
b.trimmid
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = base_catcher, statistic = fc_trimmid, R = 100)
##
##
## Bootstrap Statistics :
##     original   bias    std. error
## t1* 513.9167 26.04812    82.50774
```

```
plot(b.trimmid)
```



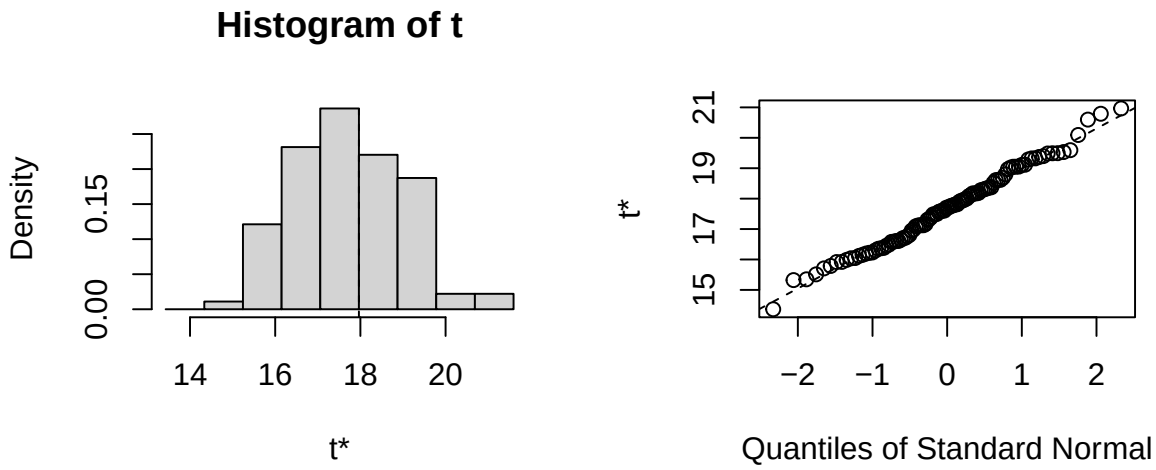

```
set.seed(321)
# MAD #Measure of the sd in Salary

b.mad = boot(base_catcher, fc_mad, R = 100)
b.mad
```

This value is

$$\{\frac{1}{N}\sum_{i=1}^{N} \mid x_i - median(X) \mid\}^{1/2}$$

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = base_catcher, statistic = fc_mad, R = 100)
##
##
## Bootstrap Statistics :
##     original     bias    std. error
## t1* 17.96562 -0.2890404    1.314317
```

```
plot(b.mad)
```



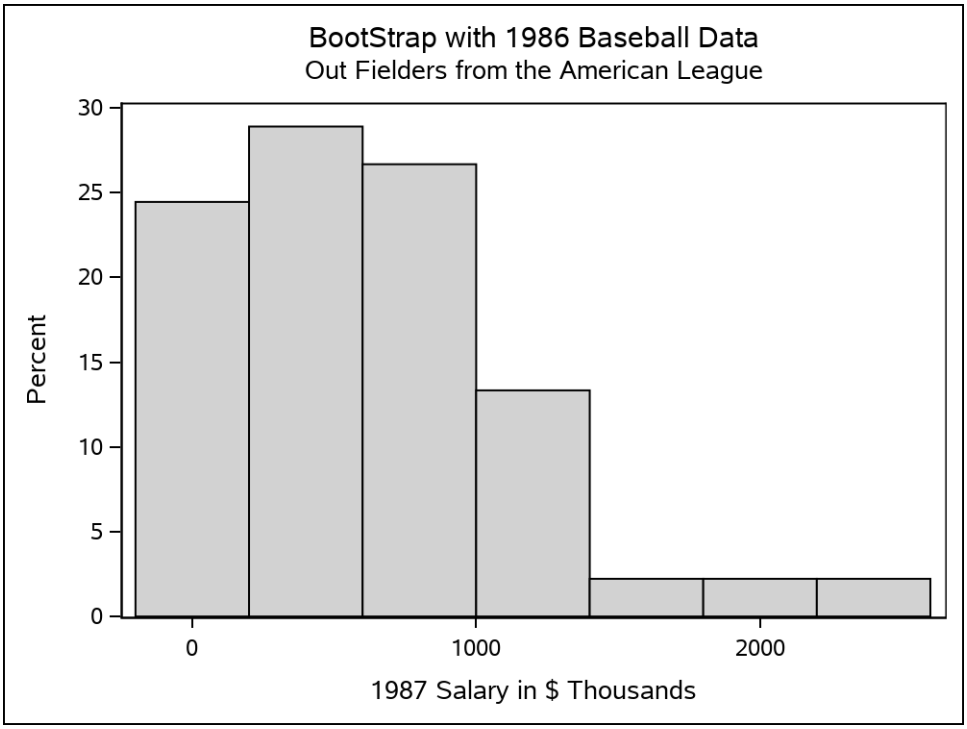
SAS

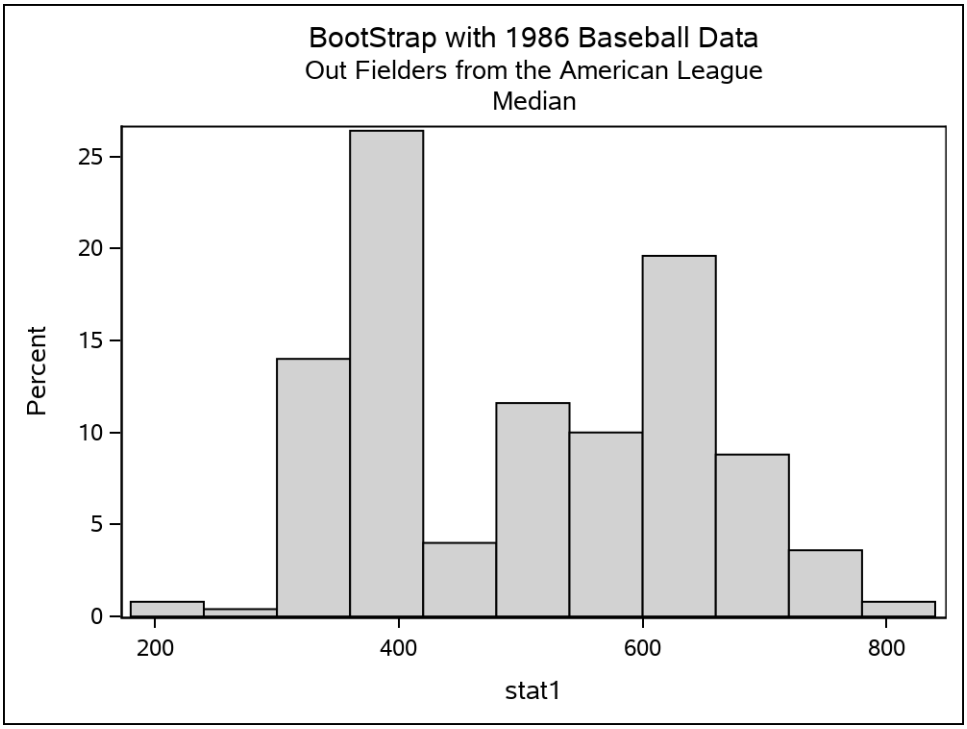
Code

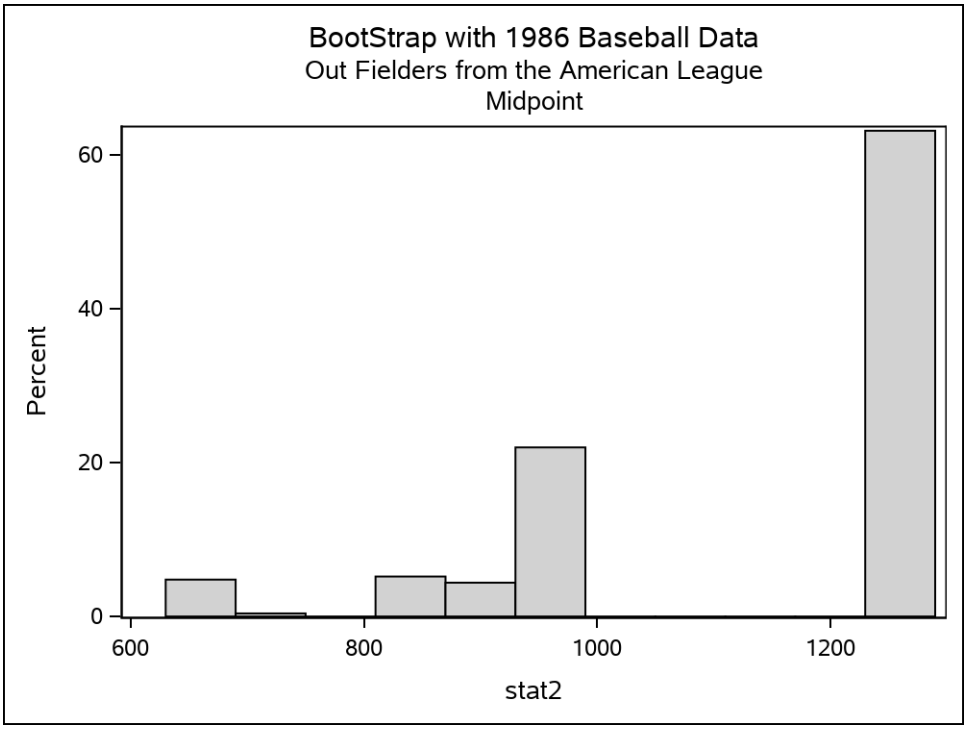
```
options center nodate pagesize=80 ls=70;
libname ldata '/home/jacktubbs/my_shared_file_links/jacktubbs/LaTeX/Class';


title "BootStrap with 1986 Baseball Data";
data baseball; set sashelp.baseball;
run;


data baseball; set baseball;
in_fielder = (position in ('1B' '2B' 'SS' '3B'));
out_fielder = (position in ('CF' 'RF' 'LF' 'OF'));
catcher = (position = 'C');
CrHits2 = CrHits*CrHits;
run;

Title2 'Out Fielders from the American League';
data amer_of; set baseball;
if out_fielder = 1 and league = 'American';
run;

proc sgplot data=amer_of;
histogram salary;
run;

*******************************************************;
* Using PROC SURVEYSELECT;
*******************************************************;
```

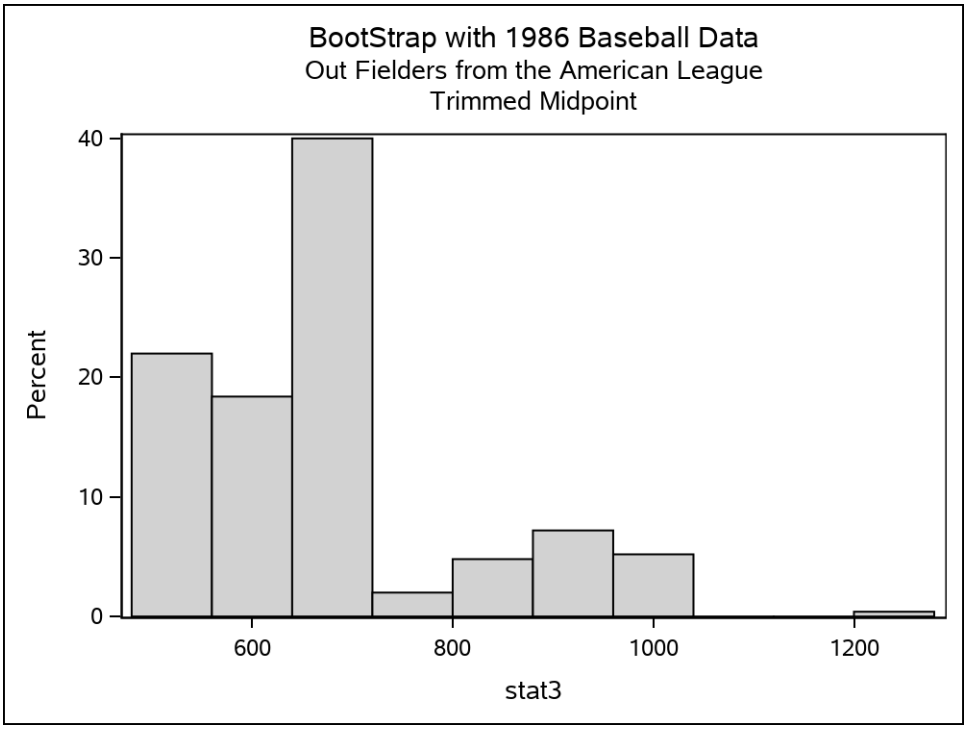
```
proc surveyselect data=amer_of NOPRINT seed=123
     out=BootSS
     method=balboot
     reps=250;
run;

proc summary data=BootSS;by replicate;
var salary;
   output out=Bootdist (drop=_freq_ _type_) median=med p90=p90 p10=p10 min=min max=max;
run;

data OutStatsUni; set Bootdist; stat1 = med; stat2=(min+max)/2; stat3=(p10+p90)/2;
run;

title3 'Median';
proc sgplot data=OutStatsUni;
   histogram stat1;
run;

title3 'Midpoint';
proc sgplot data=OutStatsUni;
   histogram stat2;
run;

title3 'Trimmed Midpoint';
proc sgplot data=OutStatsUni;
   histogram stat3;
run;

title3 'Average over bootstrap samples';
proc means data=OutStatsUni mean std min max; var stat1 stat2 stat3;
output out=Outstats2;
run;
```

Output


BootStrap with 1986 Baseball Data
Out Fielders from the American League
Percent
30
25
20
15
10
5
0
0
1000
2000
1987 Salary in $ Thousands


BootStrap with 1986 Baseball Data
Out Fielders from the American League
Median
Percent
25
20
15
10
5
0
200
400
600
800
stat1


BootStrap with 1986 Baseball Data
Out Fielders from the American League
Midpoint
Percent
60
40
20
0
600
800
1000
1200
stat2


BootStrap with 1986 Baseball Data
Out Fielders from the American League
Trimmed Midpoint
Percent
40
30
20
10
0
600
800
1000
1200
stat3

BootStrap with 1986 Baseball Data
Out Fielders from the American League
Average over bootstrap samples

The MEANS Procedure

Variable	Mean	Std Dev	Minimum	Maximum
stat1	500.065	129.819	207.500	832.500
stat2	1115.720	175.290	652.750	1256.250
stat3	680.652	127.002	485.000	1256.250