Regression in Sports

Dr. Ekstrom

Purpose: To learn more about regression, getting and loading data, and using regression to identify key statistics in a variety of sports.

Sports Analytics: Try to explain how points (or goals) are scored in various sports. That is, to try to create a statistic that explains as much run scoring as possible.

Statistics: Discuss the assumptions in using regression for prediction, and a bit of multiple regression.

R: Downlaoding suitable data from the net, loading data into *R* and using multiple regression to create new statistics.

Model from Previous Activity

Most groups determined that the model:

$$\texttt{R} = -877.49 + 2175.73 \cdot \texttt{OPS}$$

had the highest R-squared and lowest residual standard error.

```r
bbr=data=read.csv("BaseballRuns.csv")
bbr$SLG=(bbr$X1B+2*bbr$X2B+3*bbr$X3B+4*bbr$HR)/(bbr$AB)
bbr$OBP=(bbr$X1B+bbr$X2B+bbr$X3B+bbr$HR+bbr$BB+bbr$HBP)/(bbr$AB+bbr$BB+
                                                          bbr$HBP+bbr$SF)
bbr$OPS=bbr$OBP+bbr$SLG
gos=lm(formula=bbr$R~bbr$OPS)
summary(gos)
```

```
##
## Call:
## lm(formula = bbr$R ~ bbr$OPS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -82.441 -16.071  -1.876  15.638  74.532
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -877.49      42.10  -20.84   <2e-16 ***
## bbr$OPS      2175.73      55.42   39.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.95 on 178 degrees of freedom
## Multiple R-squared:  0.8965, Adjusted R-squared:  0.8959
## F-statistic:  1541 on 1 and 178 DF,  p-value: < 2.2e-16
```

However, to use this model to predict `R` from a given value of `OPS`, there are a few assumptions that should be met:

Normally distributed variables: The dependent and independent variables should be normally distributed.

Linearity and additivity: The relationship between dependent and independent variables should be linear

Normally distributed residuals: The residuals should be normally distributed

Homoscedasticity: The residuals should have constant variance

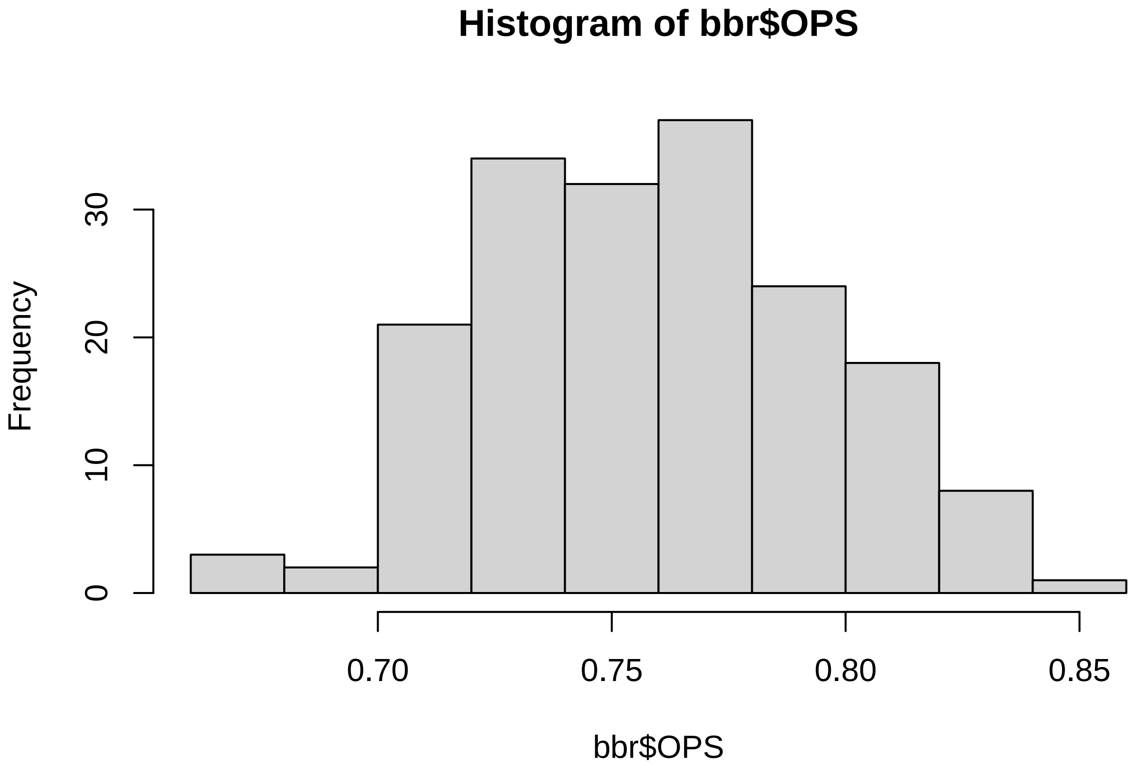
Free of Problem Points: The data should be free of outliers and leverage points. These points that have an undue influence on our model.

Checking the Assumptions

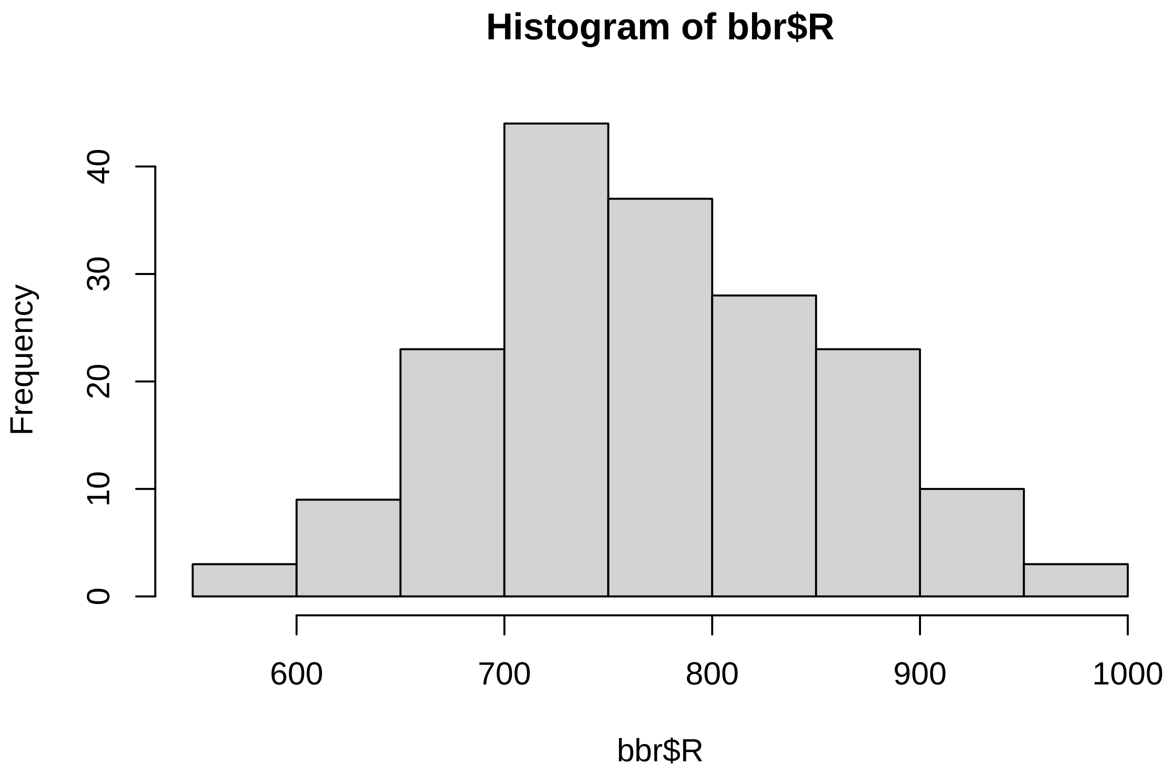
Normally distributed variables:

Plot histograms and look for the classic bell-shape:

```
hist(bbr$OPS)
```

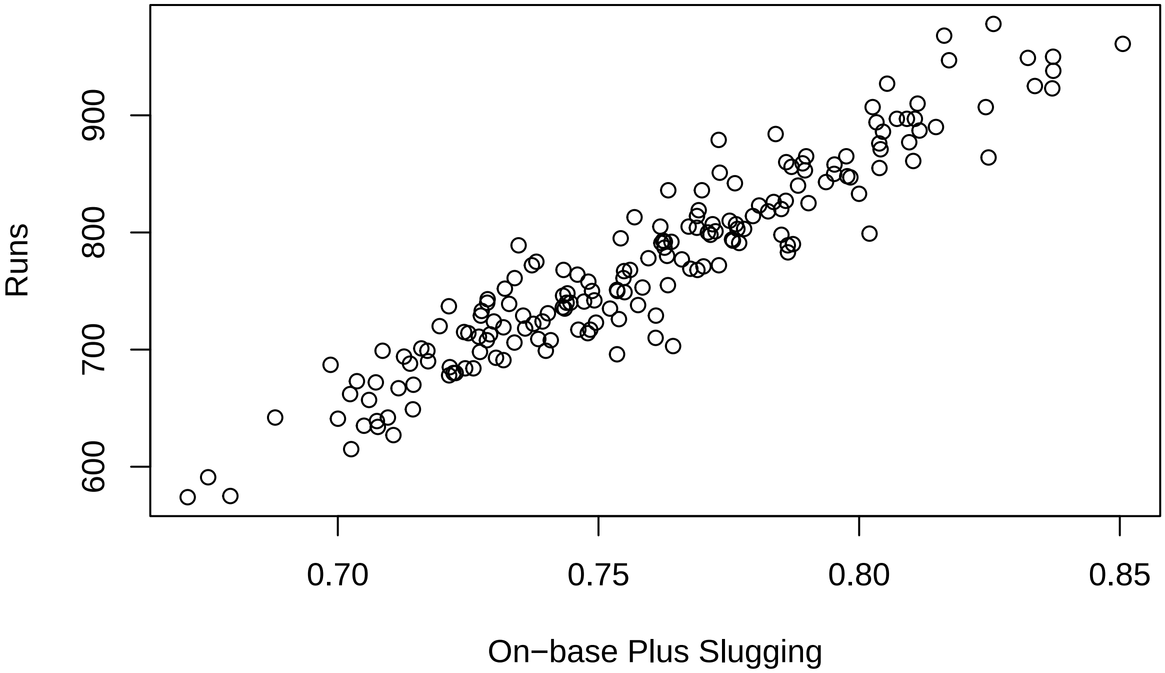


```
hist(bbr$R)
```


Histogram of bbr$R
Frequency
0
10
20
30
40
600
700
800
900
1000
bbr$R

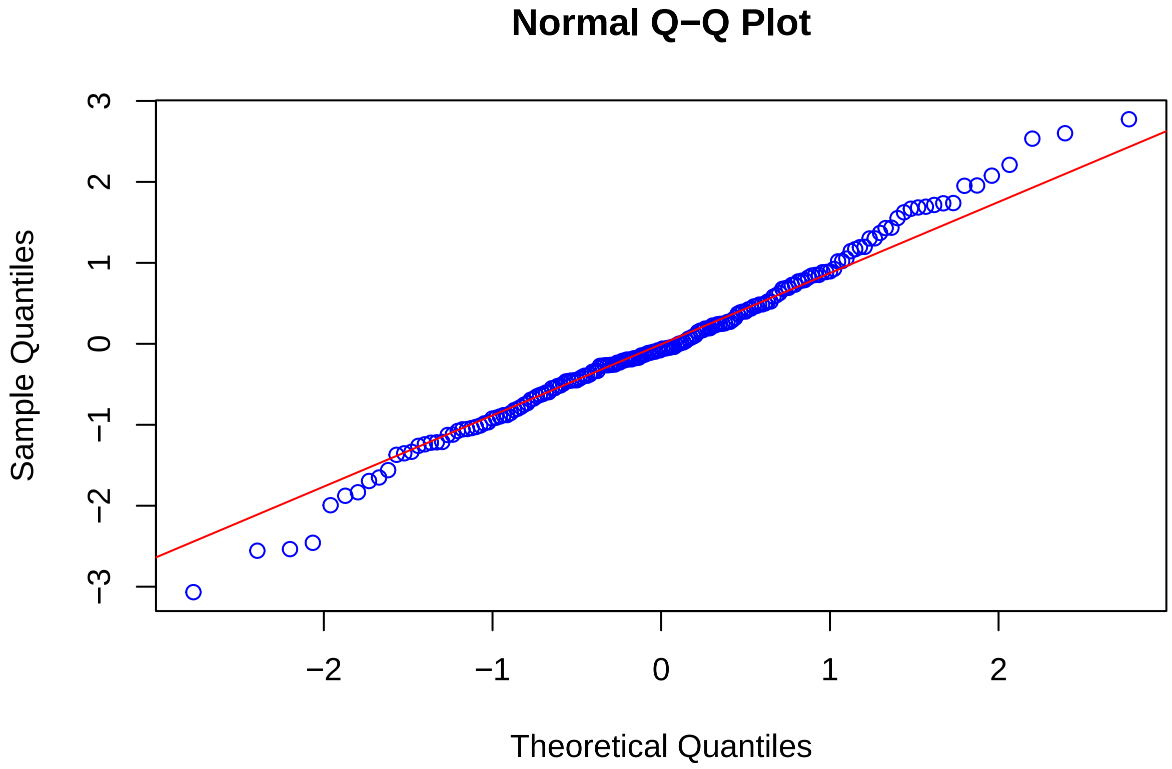
Linearity and additivity: Plot scatterplot and check that the main “cloud’’ has a linear shape:

```
plot(bbr$OPS,bbr$R,xlab="On-base Plus Slugging",ylab="Runs")
```



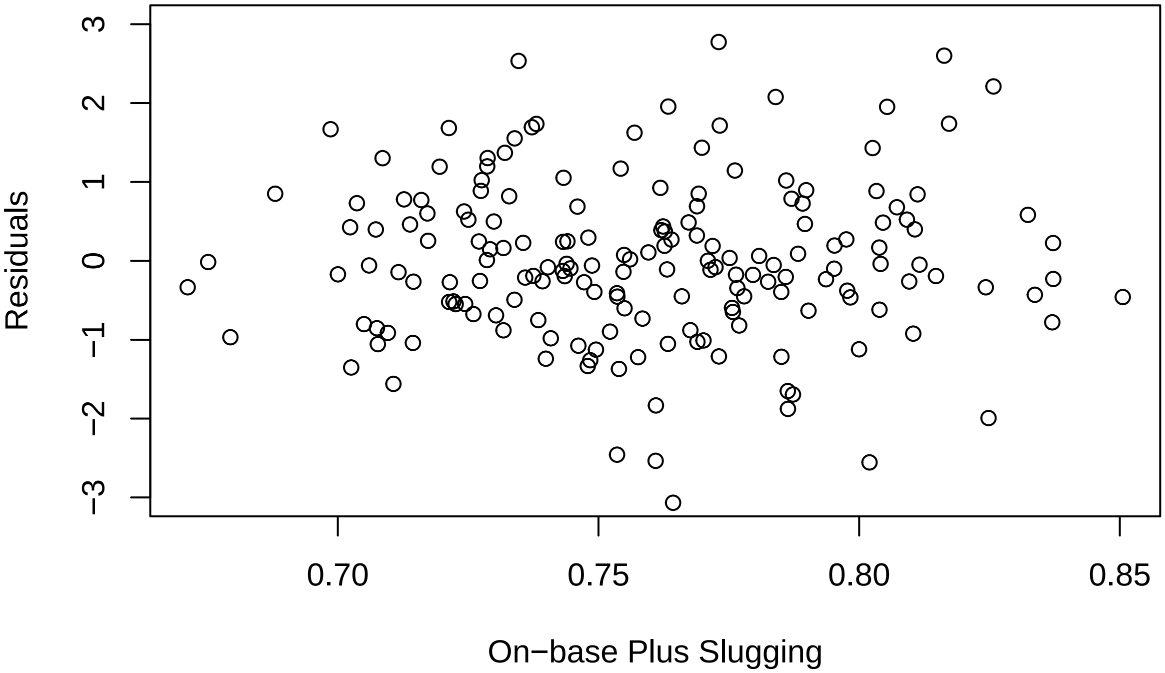
Normally distributed residuals: Make a QQ-Plot of residuals (and put in the QQ-line) and see if the line is a good fit (which would mean the residuals are at least approximately normal). Note that `rstandard()` computes the standardized residuals - which are generally more useful for our purposes:

```
res=rstandard(gos)
qqnorm(res,col="blue")
qqline(res,col="red")
```



Homoscedasticity: Make a scatter plot of residuals vs independent variable and look to see if the "cloud' ' is roughly rectangular. If the cloud has a megaphone shape, it is a problem that needs to be addressed.

```
plot(bbr$OPS,res,ylim=c(-3,3),xlab="On-base Plus Slugging",ylab="Residuals")
```



Free of Problem Points: In the plot above, look for standardized residuals that are greater than 3 (or less than -3) - these could be outliers. Also look for independent variable values that are well outside the “normal range’ ’ of independent variable values - these could be leverage points. If there are candidates for either type of problem points, try removing them from the data and recalculating the linear model to see if it is much different from the original. You may prefer to use the model that is free from problem points, but it is a judgement call.

Just a Bit of Multiple Regression If you compute the statistic

$$\mathtt{OOPS} = 2 \cdot \mathtt{OBP} + \mathtt{SLG}$$

it does a better job than `OPS` in explaining `R`.

```
bbr$OOPS=2*(bbr$OBP)+bbr$SLG
goos=lm(formula=bbr$R~bbr$OOPS)
summary(goos)
```

```
##
## Call:
## lm(formula = bbr$R ~ bbr$OOPS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -75.039 -14.917  -1.613  15.926  60.233
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1016.86      43.22  -23.53   <2e-16 ***
## bbr$OOPS     1638.01      39.50   41.47   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.66 on 178 degrees of freedom
## Multiple R-squared:  0.9062, Adjusted R-squared:  0.9057
## F-statistic:  1720 on 1 and 178 DF,  p-value: < 2.2e-16
```

How do we know that combining twice as much `OBP` as `SLG` would give a better predictor? Endless trial and error?

A bit of multiple linear regression is helpful. You can use the same `lm()` command with more variables:

```
gm=lm(formula=bbr$R~bbr$OBP+bbr$SLG)
summary(gm)
```

```
##
## Call:
## lm(formula = bbr$R ~ bbr$OBP + bbr$SLG)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -74.280 -15.232  -1.345  15.434  61.780
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1003.65      49.63  -20.22   <2e-16 ***
## bbr$OBP      3156.71     232.93   13.55   <2e-16 ***
## bbr$SLG      1700.80     121.88   13.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.71 on 177 degrees of freedom
## Multiple R-squared:  0.9064, Adjusted R-squared:  0.9053
## F-statistic: 856.7 on 2 and 177 DF,  p-value: < 2.2e-16
```

Note that the coefficient for `OBP` is roughly twice the size of `SLG`. Also note that the significance of each of the variables. Only include variables that are significant (small probabilities, such as less than 0.05).

If we add `SB` (stolen bases) to the mix, we see that it is not signficant enough to be included.

```
gmb=lm(formula=bbr$R~bbr$OBP+bbr$SLG+bbr$SB)
summary(gmb)
```

```
##
## Call:
## lm(formula = bbr$R ~ bbr$OBP + bbr$SLG + bbr$SB)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -75.402 -15.693  -2.276  15.655  58.930
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.012e+03  4.950e+01 -20.451   <2e-16 ***
## bbr$OBP      3.103e+03  2.330e+02  13.318   <2e-16 ***
## bbr$SLG      1.737e+03  1.226e+02  14.173   <2e-16 ***
## bbr$SB       1.206e-01  6.423e-02   1.878   0.0621 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.52 on 176 degrees of freedom
## Multiple R-squared:  0.9082, Adjusted R-squared:  0.9066
## F-statistic: 580.5 on 3 and 176 DF,  p-value: < 2.2e-16
```

Class Activity

Our goal is to perform an analysis similar to the one we performed in *Baseball runs part 1* on another sport. Can we identify a statistic that is the best predictor of scoring? (or best explains scoring? or winning percentage? or wins?)

- Choose a sport among: College Basketball, NBA, College Football, NFL, NHL, or Soccer.
- Find data for your sport. A great reference is: **https://www.sports-reference.com/**
- Download the appropriate data. (CSVs work well.)
- Load the data into *R*.
- Perform your analysis on the data.

When your group is finished, turn in a copy of the groups R-notebook and data to the appropriate Gradescope assignment.