Multiple Regression

Oliver

Get Some Data

We obtain the Student Survey and Golden State Warrior data from the web. These files are part of the Lock^5 3rd Ed. data files.

```
Survey = read.csv("http://facweb1.redlands.edu/fac/jim_bentley/Data/Lock5Ed3/Lock5Data3eCSV/StudentSu
names(Survey)
```

```
##  [1] "Year"       "Sex"        "Smoke"      "Award"      "HigherSAT"
##  [6] "Exercise"   "TV"         "Height"     "Weight"     "Siblings"
## [11] "BirthOrder" "VerbalSAT"  "MathSAT"    "SAT"        "GPA"
## [16] "Pulse"      "Piercings"
```

```
GSW = read.csv("http://facweb1.redlands.edu/fac/jim_bentley/Data/Lock5Ed3/Lock5Data3eCSV/GSWarriors20
names(GSW)
```

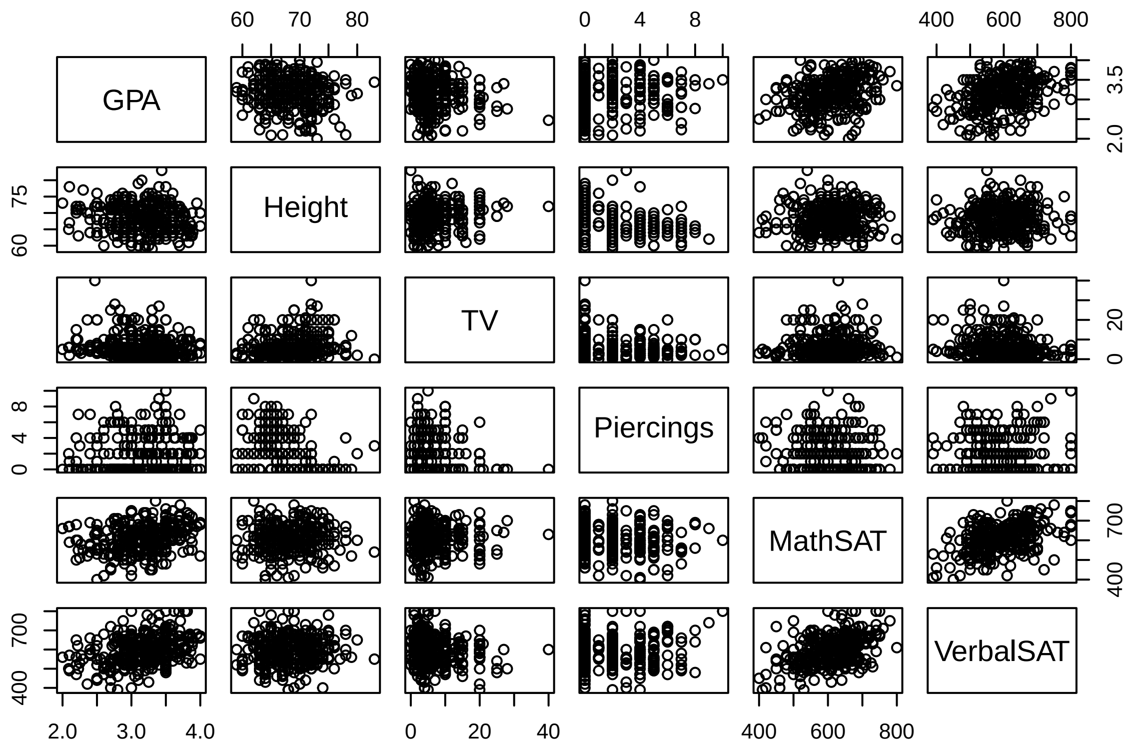
```
##  [1] "Game"         "Date"         "Location"     "Opp"          "Win"
##  [6] "Points"       "FG"           "FGA"          "FG3"          "FG3A"
## [11] "FT"           "FTA"          "Rebounds"     "OffReb"       "Assists"
## [16] "Steals"       "Blocks"       "Turnovers"    "Fouls"        "OppPoints"
## [21] "OppFG"        "OppFGA"       "OppFG3"       "OppFG3A"      "OppFT"
## [26] "OppFTA"       "OppRebounds"  "OppOffReb"    "OppAssists"   "OppSteals"
## [31] "OppBlocks"    "OppTurnovers" "OppFouls"
```

Fit GPA

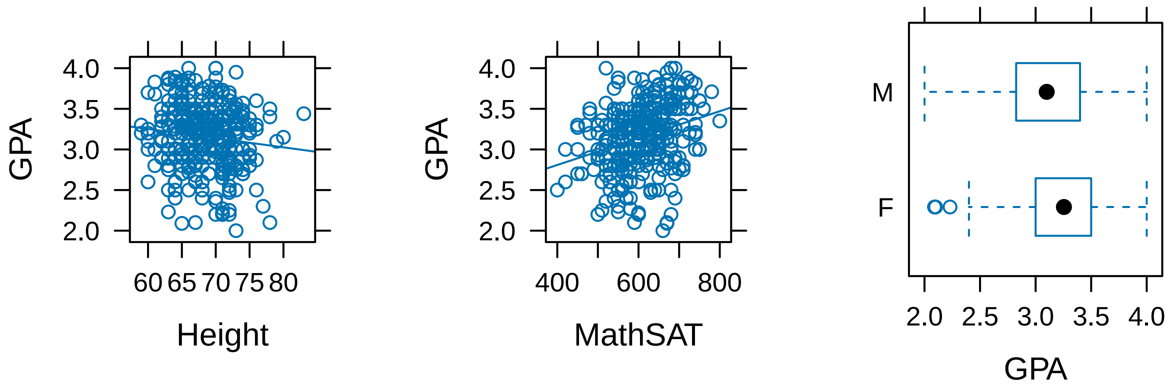
Suppose we want to figure out predictors of GPA. For college students from St. Lawrence University, what helps determine student success?

```
### Plot all of the variables against each other
pairs(Survey[,c("GPA","Height","TV","Piercings","MathSAT","VerbalSAT")])

### Load the lattice graphics package
p_load(lattice)
```




```
### Plot GPA as a function of a couple of variables.  Include the regression line.
  p1 = xyplot(GPA ~ Height, data=Survey, type=c("p","r"), aspect=1)
  p2 = xyplot(GPA ~ MathSAT, data=Survey, type=c("p","r"), aspect=1)
### A boxplot works well for categorical/factor variables
  p3 =bwplot(Sex~GPA, data=Survey, aspect=1)
### Plot the graphics in a sinle image to save space
  print(p1, split = c(1, 1, 3, 1), more = TRUE)
  print(p2, split = c(2, 1, 3, 1), more = TRUE)
  print(p3, split = c(3, 1, 3, 1), more = FALSE)
```




```r
### Fit a multiple linear regression model
GPA.lm = lm(GPA ~ Height + TV + Piercings + MathSAT + VerbalSAT, data=Survey)
### Get the parameter estimates, standard errors, t-stats, and p-vals
summary(GPA.lm)
```

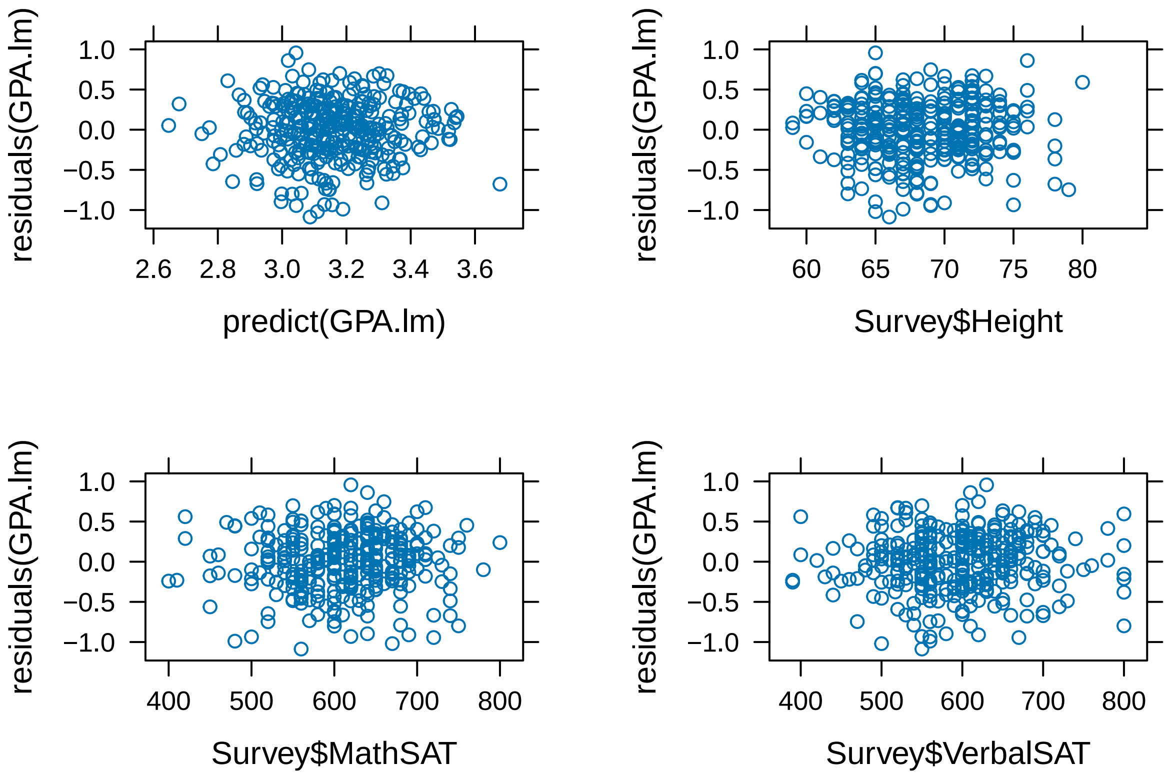
```
##
## Call:
## lm(formula = GPA ~ Height + TV + Piercings + MathSAT + VerbalSAT,
##     data = Survey)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.10364 -0.23038  0.02313  0.27887  0.92934
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.4380195  0.4605779   5.293 2.19e-07 ***
## Height      -0.0105225  0.0058703  -1.792  0.07397 .
## TV          -0.0046027  0.0036572  -1.259  0.20909
## Piercings    0.0064355  0.0113332   0.568  0.57053
## MathSAT      0.0009516  0.0003397   2.801  0.00539 **
## VerbalSAT    0.0014799  0.0003155   4.690 3.99e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3677 on 332 degrees of freedom
```

```
##   (24 observations deleted due to missingness)
## Multiple R-squared:  0.1606, Adjusted R-squared:  0.1479
## F-statistic:  12.7 on 5 and 332 DF,  p-value: 2.632e-11
```

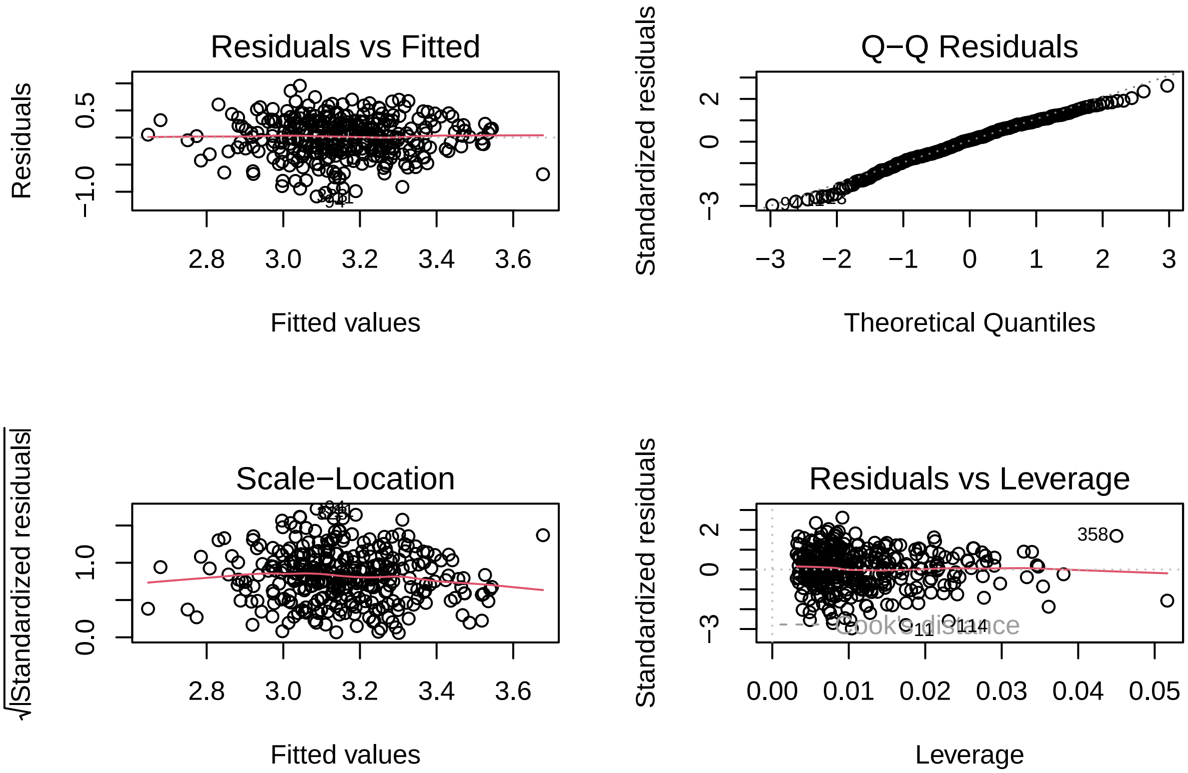
```r
### Fit a reduced model
GPA.lm = lm(GPA ~ Height + MathSAT + VerbalSAT, data=Survey)
### Get the parameter estimates, standard errors, t-stats, and p-vals
summary(GPA.lm)
```

```
##
## Call:
## lm(formula = GPA ~ Height + MathSAT + VerbalSAT, data = Survey)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.08701 -0.23130  0.02617  0.26942  0.95687
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.6159121  0.3818689   6.850 3.56e-11 ***
## Height      -0.0135121  0.0048959  -2.760  0.00610 **
## MathSAT      0.0008769  0.0003314   2.646  0.00852 **
## VerbalSAT    0.0015691  0.0003091   5.076 6.42e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3678 on 334 degrees of freedom
##   (24 observations deleted due to missingness)
## Multiple R-squared:  0.1554, Adjusted R-squared:  0.1479
## F-statistic: 20.49 on 3 and 334 DF,  p-value: 3.27e-12
```

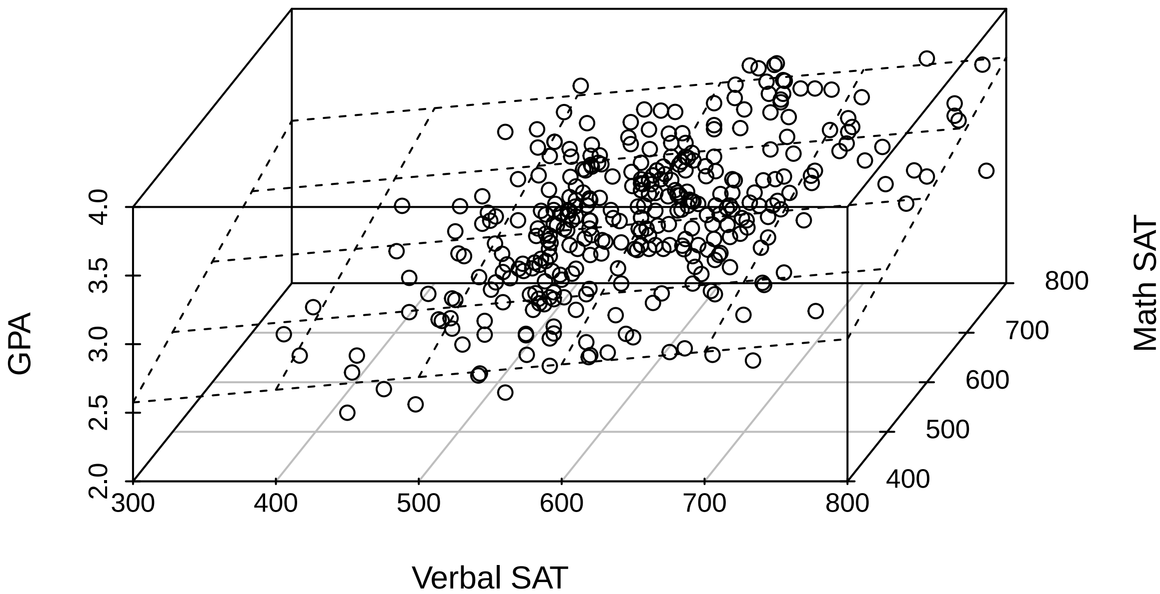
```r
### Check residuals
  p1 = xyplot(residuals(GPA.lm)~predict(GPA.lm))
  p2 = xyplot(residuals(GPA.lm)~Survey$Height)
  p3 = xyplot(residuals(GPA.lm)~Survey$MathSAT)
  p4 = xyplot(residuals(GPA.lm)~Survey$VerbalSAT)
### Plot using split to get four plots in a single image
  print(p1, split = c(1, 1, 2, 2), more = TRUE)
  print(p2, split = c(2, 1, 2, 2), more = TRUE)
  print(p3, split = c(1, 2, 2, 2), more = TRUE)
  print(p4, split = c(2, 2, 2, 2), more = FALSE)  # more = FALSE is redundant
```




```
### Plot the default residual plots in a single image
par(mfrow=c(2,2))
plot(GPA.lm)
```




```r
par(mfrow=c(1,1))
### For fun, plot the plane of estimates determined by Math and Verbal SATs
p_load(scatterplot3d)
s3d = scatterplot3d(Survey$VerbalSAT, Survey$MathSAT, Survey$GPA, xlab="Verbal SAT", ylab="Math SAT",
s3d$plane3d(lm(GPA ~ MathSAT + VerbalSAT, data=Survey))
```

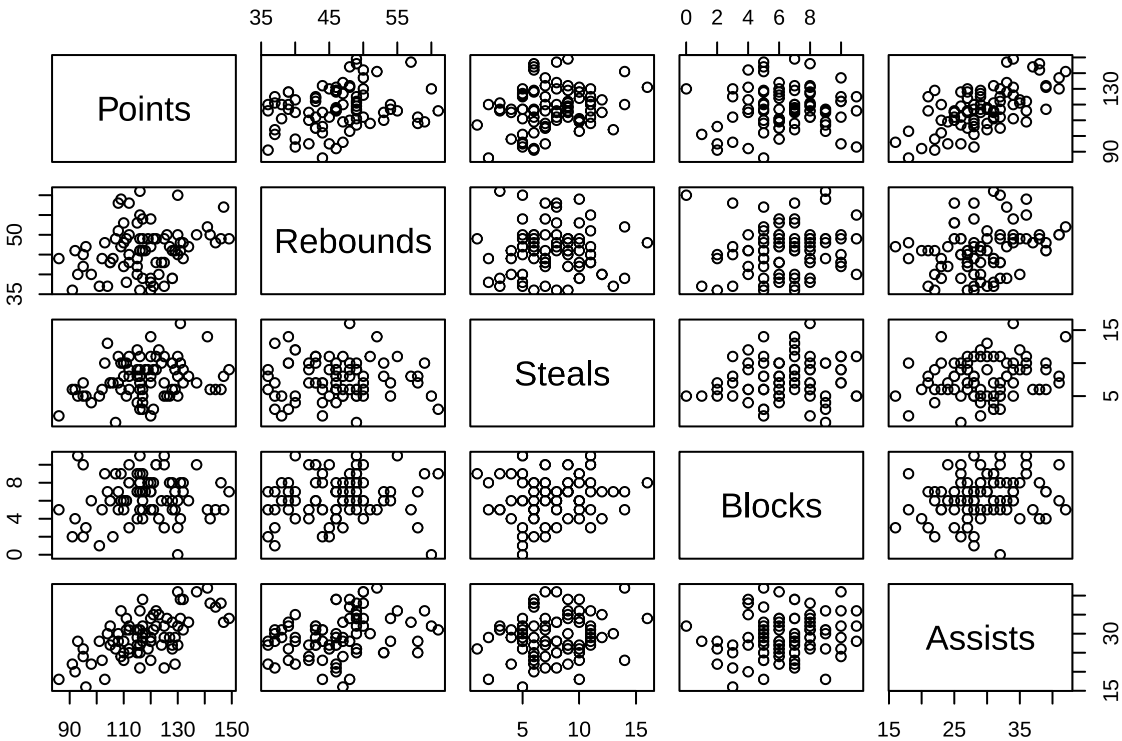



```r
par(mfrow=c(1,1))
```

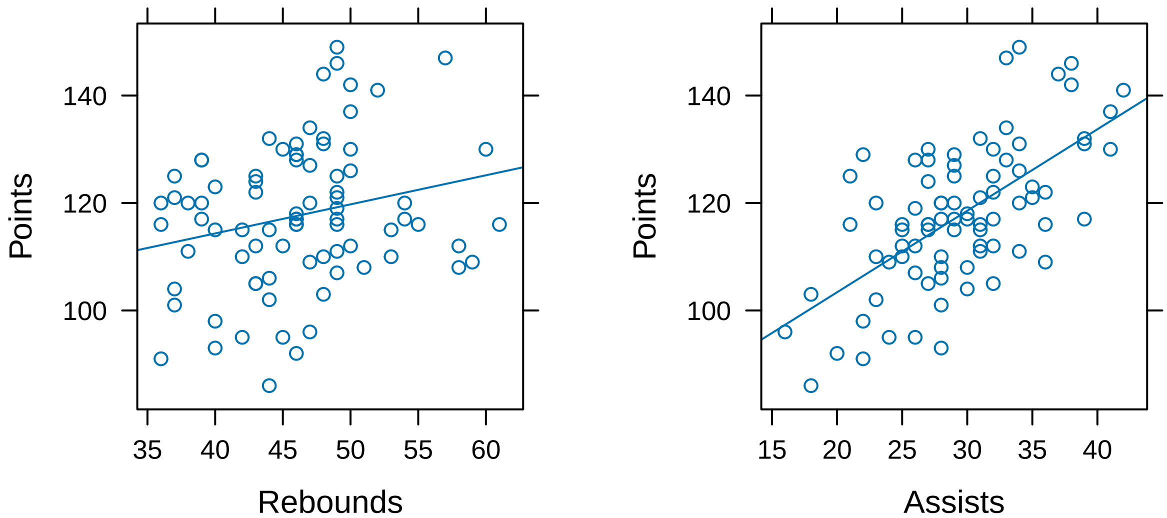
Fit Points

We can estimate the number of points scored by the Golden State Warriors (2018-2019 regular season) using a multiple linear regression model.

```r
### Plot all of the variables against each other
pairs(GSW[,c("Points","Rebounds","Steals","Blocks","Assists")])
```




```
### Load the lattice graphics package
p_load(lattice)
### Plot Points as a function of a couple of variables.  Include the regression line. Use split to ge
p1 = xyplot(Points ~ Rebounds, data=GSW, type=c("p","r"), aspect=1)
p2 = xyplot(Points ~ Assists, data=GSW, type=c("p","r"), aspect=1)
print(p1, split = c(1, 1, 2, 1), more = TRUE)
print(p2, split = c(2, 1, 2, 1), more = FALSE)
```



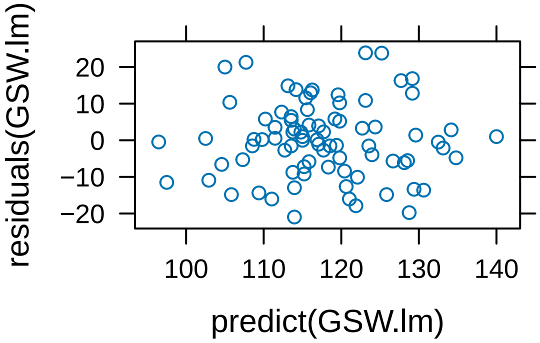

```r
### Fit a multiple linear regression model
GSW.lm = lm(Points ~ Rebounds + Steals + Blocks + Assists, data=GSW)
### Get the parameter estimates, standard errors, t-stats, and p-vals
summary(GSW.lm)
```

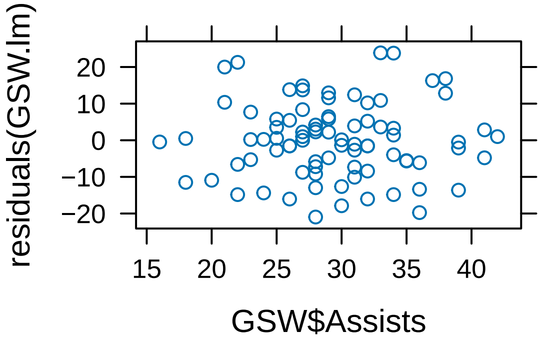
```
##
## Call:
## lm(formula = Points ~ Rebounds + Steals + Blocks + Assists, data = GSW)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.9152  -5.8121  -0.2544   5.4964  23.5621
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  64.7579    10.2811   6.299 1.71e-08 ***
## Rebounds      0.1766     0.2070   0.853   0.3962
## Steals        0.6831     0.4055   1.684   0.0961 .
## Blocks       -0.4222     0.5037  -0.838   0.4046
## Assists       1.4363     0.2237   6.421 1.01e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.41 on 77 degrees of freedom
## Multiple R-squared:  0.426,  Adjusted R-squared:  0.3962
## F-statistic: 14.29 on 4 and 77 DF,  p-value: 9.094e-09
```

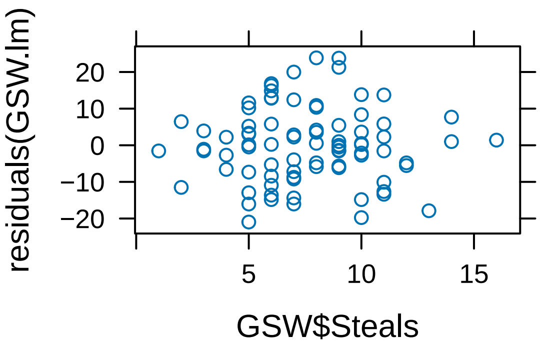
```r
### Try a reduced model
GSW.lm = lm(Points~Steals + Assists, data=GSW)
summary(GSW.lm)
```

```
##
## Call:
## lm(formula = Points ~ Steals + Assists, data = GSW)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.9611  -6.4733   0.0576   5.7018  23.8716
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  70.0135     6.4874  10.792   <2e-16 ***
## Steals        0.6262     0.4008   1.562    0.122
## Assists       1.4577     0.2102   6.933    1e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.36 on 79 degrees of freedom
## Multiple R-squared:  0.416,  Adjusted R-squared:  0.4012
## F-statistic: 28.13 on 2 and 79 DF,  p-value: 5.941e-10
```

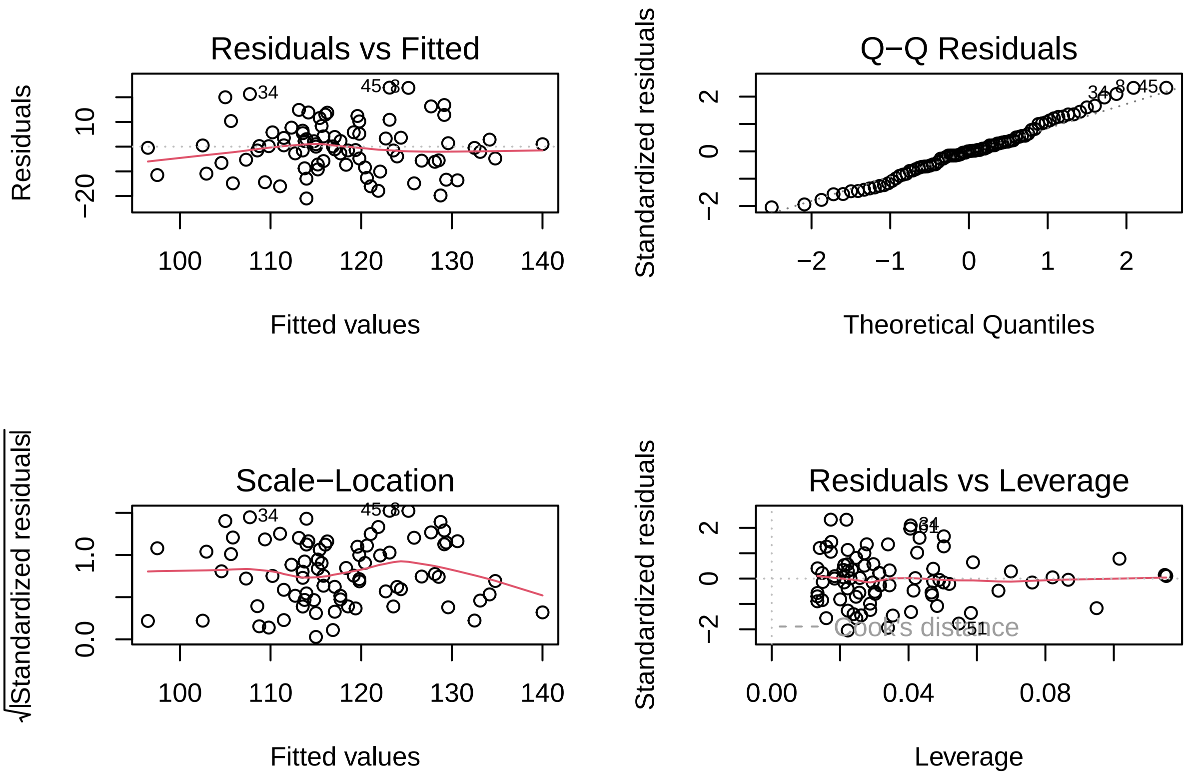
```r
### Check residuals
p1 = xyplot(residuals(GSW.lm)~predict(GSW.lm))
p2 = xyplot(residuals(GSW.lm)~GSW$Assists)
p3 = xyplot(residuals(GSW.lm)~GSW$Steals)
### Plot lattice plots in single graphic image
  print(p1, split = c(1, 1, 2, 2), more = TRUE)
  print(p2, split = c(2, 1, 2, 2), more = TRUE)
  print(p3, split = c(1, 2, 2, 2), more = FALSE)
```



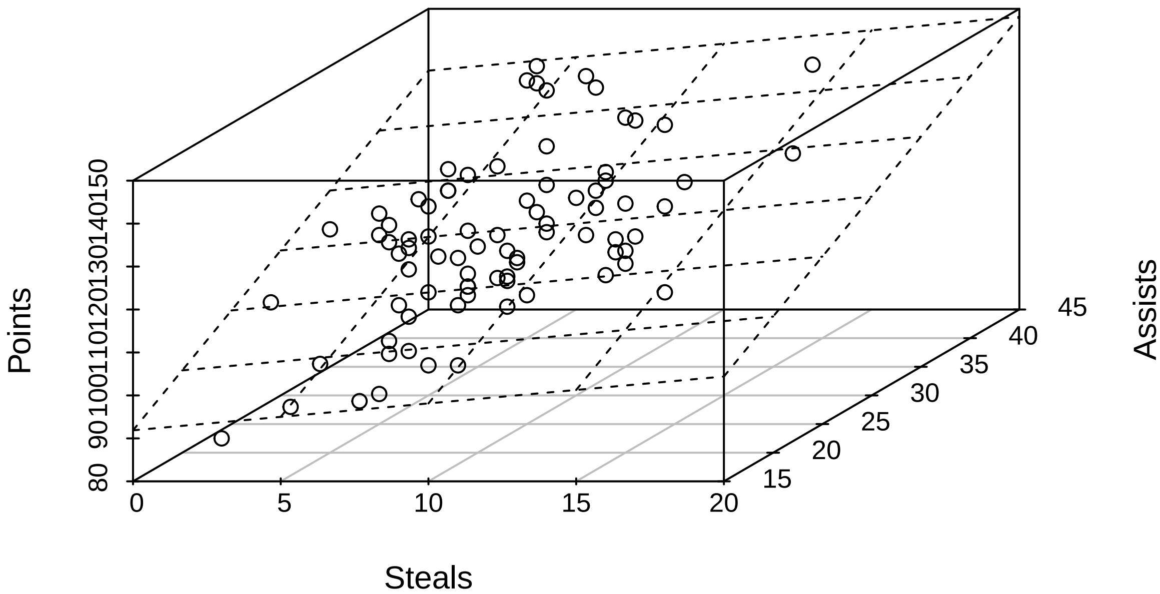





```
### Use base plot to get default residual plots in a single graphic
par(mfrow=c(2,2))
plot(GSW.lm)
```




```
par(mfrow=c(1,1))
### For fun, plot the plane of estimates determined by Rebounds and Assists
p_load(scatterplot3d)
s3d = scatterplot3d(GSW$Steals, GSW$Assists, GSW$Points, xlab="Steals", ylab="Assists", zlab="Points"
s3d$plane3d(lm(Points ~ Steals + Assists, data=GSW))
```


Points
Steals
Assists
80 90 100 110 120 130 140 150
0 5 10 15 20
15 20 25 30 35 40 45