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 2nd Ed. data files.

Survey = read.csv("http://facweb1.redlands.edu/fac/jim\_bentley/Data/Lock5Ed2/Lock5Data2eCSV/StudentSurvey.csv")  
 names(Survey)

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

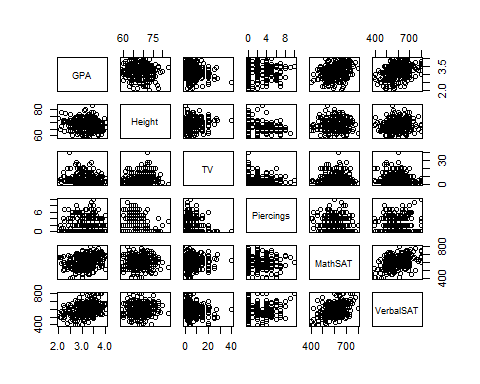
GSW = read.csv("http://facweb1.redlands.edu/fac/jim\_bentley/Data/Lock5Ed2/Lock5Data2eCSV/GSWarriors.csv")  
 names(GSW)

## [1] "Game" "Date" "Location" "Opp"   
## [5] "Win" "Points" "OppPoints" "FG"   
## [9] "FGA" "FG3" "FG3A" "FT"   
## [13] "FTA" "Rebounds" "OffReb" "Assists"   
## [17] "Steals" "Blocks" "Turnovers" "Fouls"   
## [21] "OppFG" "OppFGA" "OppFG3" "OppFG3A"   
## [25] "OppFT" "OppFTA" "OppRebounds" "OppOffReb"   
## [29] "OppAssists" "OppSteals" "OppBlocks" "OppTurnovers"  
## [33] "OppFouls"

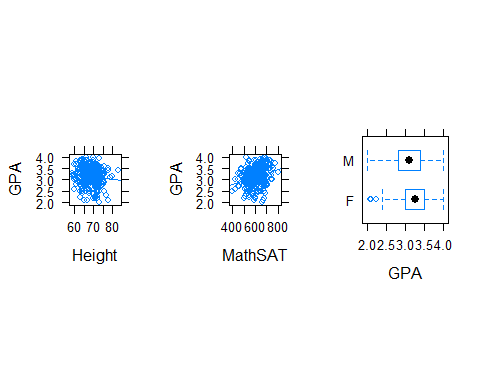
## Fit GPA

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(Gender~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)



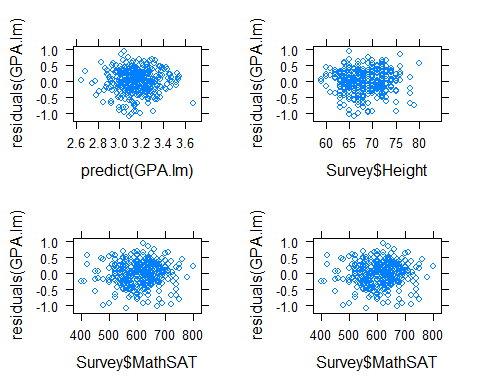
### Fit a multiple linear regression model  
 GPA.lm = lm(GPA ~ Height + TV + Piercings + MathSAT + VerbalSAT, data=Survey)  
 ### Get the parameter estimates, standard errors, t-stats, and p-vals  
 summary(GPA.lm)

##   
## 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

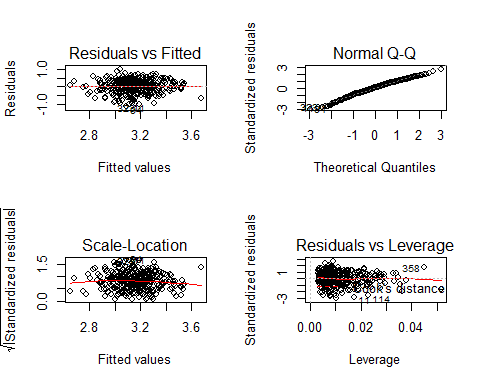
### Fit a reduced model  
 GPA.lm = lm(GPA ~ Height + MathSAT + VerbalSAT, data=Survey)  
 ### Get the parameter estimates, standard errors, t-stats, and p-vals  
 summary(GPA.lm)

##   
## 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

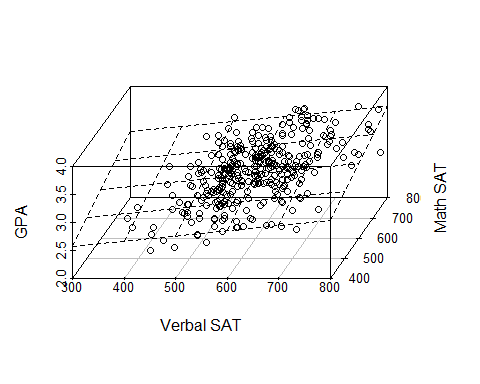
### Check residuals  
 p1 = xyplot(residuals(GPA.lm)~predict(GPA.lm))  
 p2 = xyplot(residuals(GPA.lm)~Survey$Height)  
 p3 = xyplot(residuals(GPA.lm)~Survey$MathSAT)  
 p4 = xyplot(residuals(GPA.lm)~Survey$MathSAT)  
 ### Plot using split to get four plots in a single image  
 print(p1, split = c(1, 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)



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$GPA, xlab="Verbal SAT", ylab="Math SAT", zlab="GPA", angle=65)  
 s3d$plane3d(lm(GPA ~ MathSAT + VerbalSAT, data=Survey))

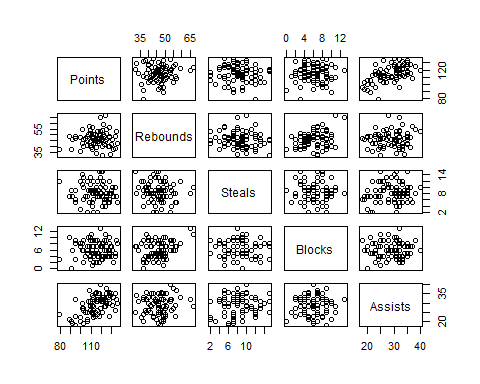


par(mfrow=c(1,1))

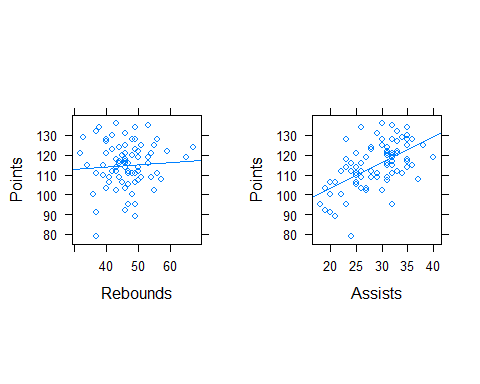
## Fit Points

We can estimate the number of points scored by the Golden State Warriors (2015-2016 season) using a multiple linear regression model.

### 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 get them in a single graphic.  
 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)



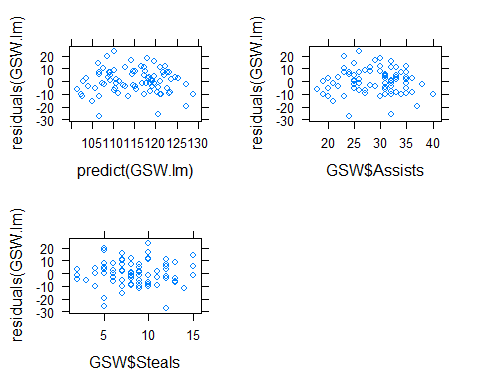
### Fit a multiple linear regression model  
 GSW.lm = lm(Points ~ Rebounds + Steals + Blocks + Assists, data=GSW)  
 ### Get the parameter estimates, standard errors, t-stats, and p-vals  
 summary(GSW.lm)

##   
## Call:  
## lm(formula = Points ~ Rebounds + Steals + Blocks + Assists, data = GSW)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -27.4781 -6.3809 -0.1247 6.6939 23.8040   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 81.11009 10.10065 8.030 8.90e-12 \*\*\*  
## Rebounds -0.02799 0.17779 -0.157 0.875   
## Steals -0.48586 0.36143 -1.344 0.183   
## Blocks 0.11728 0.44345 0.264 0.792   
## Assists 1.32842 0.21489 6.182 2.81e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 9.704 on 77 degrees of freedom  
## Multiple R-squared: 0.3379, Adjusted R-squared: 0.3035   
## F-statistic: 9.824 on 4 and 77 DF, p-value: 1.788e-06

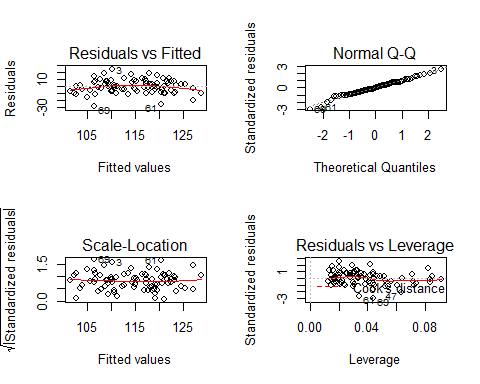
### Try a reduced model  
 GSW.lm = lm(Points~Steals + Assists, data=GSW)  
 summary(GSW.lm)

##   
## Call:  
## lm(formula = Points ~ Steals + Assists, data = GSW)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -27.6121 -6.2238 -0.1036 6.8414 23.7751   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 80.5363 6.6426 12.124 < 2e-16 \*\*\*  
## Steals -0.4800 0.3532 -1.359 0.178   
## Assists 1.3265 0.2110 6.286 1.67e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 9.585 on 79 degrees of freedom  
## Multiple R-squared: 0.3372, Adjusted R-squared: 0.3205   
## F-statistic: 20.1 on 2 and 79 DF, p-value: 8.784e-08

### 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$Points, xlab="Steals", ylab="Assists", zlab="Points", angle=60)  
 s3d$plane3d(lm(Points ~ Steals + Assists, data=GSW))

