Introduction to the Practice of Statistics using R: Chapter 11

Nicholas J. Horton* Ben Baumer

March 10, 2013

Contents

Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Sixth Edition of *Introduction to the Practice of Statistics* (2009) by David Moore, George McCabe and Bruce Craig. More information about the book can be found at `http://bcs.whfreeman.com/ips6e/`. This file as well as the associated `knitr` reproducible analysis source file can be found at `http://www.math.smith.edu/~nhorton/ips6e`.

This work leverages initiatives undertaken by Project MOSAIC (`http://www.mosaic-web.org`), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignette (`http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf`).

Additional examples of fitting multiple regression models can be found in the companion site which implements the examples within *The Statistical Sleuth* in R (`http://www.math.smith.edu/~nhorton/sleuth`).

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

```
> install.packages('mosaic')                # note the quotation marks
```

*Department of Mathematics and Statistics, Smith College, nhorton@smith.edu

The # character is a comment in R, and all text after that on the current line is ignored.
Once the package is installed (one time only), it can be loaded by running the command:

```
> require(mosaic)
```

This needs to be done once per session.
We also set some options to improve legibility of graphs and output.

```
> trellis.par.set(theme=col.mosaic())  # get a better color scheme for lattice
> options(digits=3)
```

The specific goal of this document is to demonstrate how to replicate the analysis described in Chapter 11: Multiple Regression.

1 Case study: GPA for computer science majors

1.1 Univariate analyses

As always, we begin with a description of the predictor variables and outcome (as displayed on pages 615 and 616).

```
> ds = read.csv("http://www.math.smith.edu/ips6e/appendix/csdata.csv")
> names(ds)

[1] "obs"  "gpa"  "hsm"  "hss"  "hse"  "satm" "satv" "sex"
```

```
> favstats(~ gpa, data=ds)

  min   Q1 median   Q3 max mean    sd   n missing
 0.12 2.17   2.74 3.21   4 2.64 0.779 224       0
```

```
> favstats(~ satm, data=ds)

 min  Q1 median  Q3 max mean   sd   n missing
 300 540    600 650 800  595 86.4 224       0
```

```
> favstats(~ satv, data=ds)

 min  Q1 median  Q3 max mean   sd   n missing
 285 440    490 570 760  505 92.6 224       0
```

```
> favstats(~ hsm, data=ds)

 min Q1 median Q3 max mean   sd   n missing
   2  7      9 10  10 8.32 1.64 224       0
```

```
> favstats(~ hss, data=ds)

 min Q1 median Q3 max mean  sd   n missing
   3  7      8 10  10 8.09 1.7 224       0
```

```
> favstats(~ hse, data=ds)

 min Q1 median Q3 max mean   sd   n missing
   3  7      8  9  10 8.09 1.51 224       0
```

```
> tally(~ sex, data=ds)


   1    2 Total
 145   79   224
```

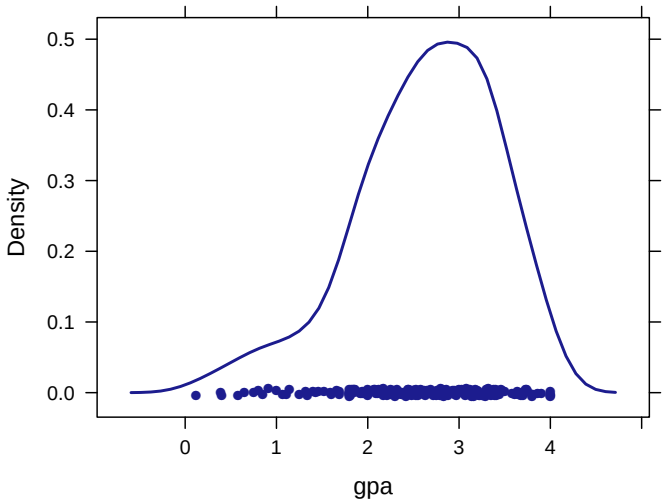
```
> tally(~ sex, format="percent", data=ds)


    1     2 Total
 64.7  35.3 100.0
```

```
> tally(~ hsm, data=ds)


   2    3    4    5    6    7    8    9   10 Total
   1    1    4    6   23   28   36   59   66   224
```

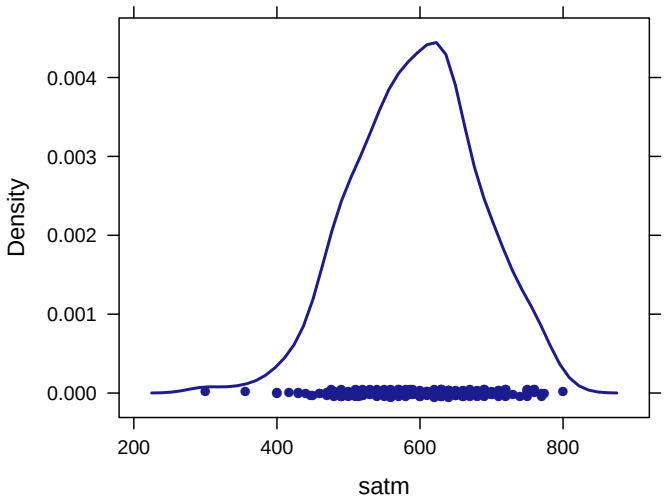
```
> tally(~ hss, data=ds)


   3    4    5    6    7    8    9   10 Total
   1    7    9   24   42   31   50   60   224
```

```
> tally(~ hse, data=ds)


  3   4   5   6   7   8   9  10 Total
  1   4   5  23  43  49  52  47   224
```

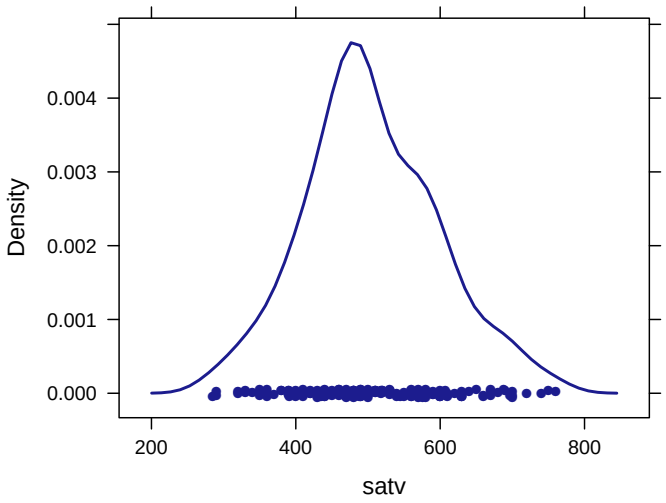
```
> densityplot(~ gpa, data=ds)
```




```
> densityplot(~ satm, data=ds)
```



```
> densityplot(~ satv, data=ds)
```



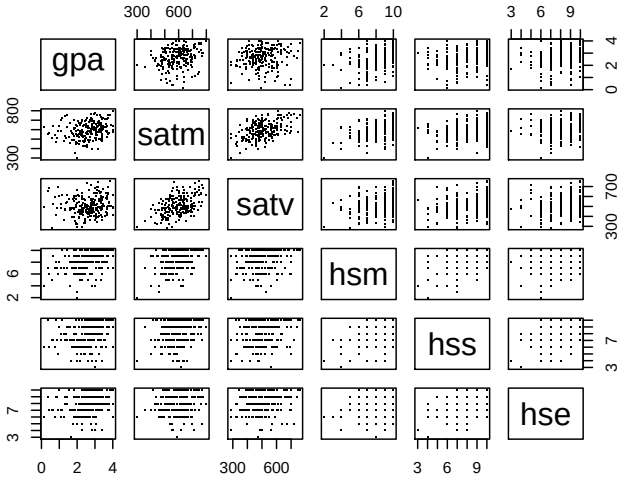
1.2 Bivariate comparisons

We can replicate the correlation matrix in Figure 11.3 (page 617).

```
> smallds = subset(ds, select=c("gpa", "satm", "satv", "hsm", "hss", "hse"))
> with(ds, cor(smallds))

       gpa  satm  satv   hsm   hss   hse
gpa  1.000 0.252 0.114 0.436 0.329 0.289
satm 0.252 1.000 0.464 0.454 0.240 0.108
satv 0.114 0.464 1.000 0.221 0.262 0.244
hsm  0.436 0.454 0.221 1.000 0.576 0.447
hss  0.329 0.240 0.262 0.576 1.000 0.579
hse  0.289 0.108 0.244 0.447 0.579 1.000
```

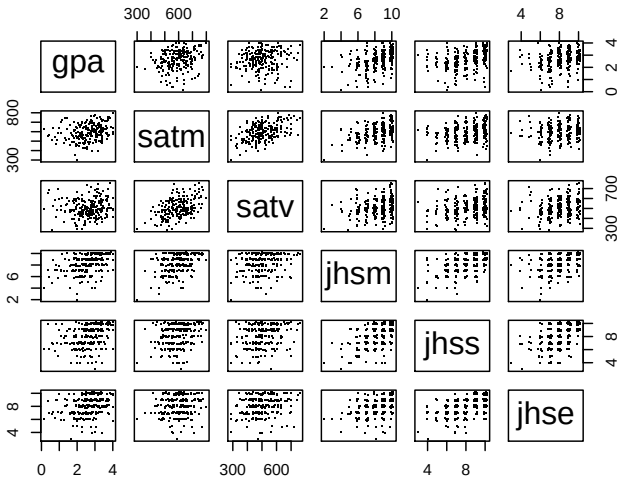
A graphical display may also be helpful:

```
> pairs(smallds, pch=".")
```



Note: jittering the categorical high school scores may improve the readability:

```
> ds = transform(ds, jhsm = jitter(hsm))
> ds = transform(ds, jhss = jitter(hss))
> ds = transform(ds, jhse = jitter(hse))
> smallds = subset(ds, select=c("gpa", "satm", "satv", "jhsm", "jhss", "jhse"))
> pairs(smallds, pch=".")
```



1.3 Multiple regression model

The output in Figure 11.4 (page 618) can be reproduced after fitting the model, which will be saved in the object called `lm1`.

```
> lm1 = lm(gpa ~ hsm + hss + hse, data=ds)
> coef(lm1)

(Intercept)         hsm         hss         hse
     0.5899      0.1686      0.0343      0.0451

> r.squared(lm1)

[1] 0.205

> summary(lm1)


Call:
lm(formula = gpa ~ hsm + hss + hse, data = ds)

Residuals:
    Min      1Q  Median      3Q     Max
-2.1289 -0.3407  0.0757  0.4744  1.7537

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.5899     0.2942    2.00    0.046 *
hsm           0.1686     0.0355    4.75  3.7e-06 ***
hss           0.0343     0.0376    0.91    0.362
hse           0.0451     0.0387    1.17    0.245
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7 on 220 degrees of freedom
Multiple R-squared: 0.205,Adjusted R-squared: 0.194
F-statistic: 18.9 on 3 and 220 DF,  p-value: 6.36e-11

> anova(lm1)

Analysis of Variance Table

Response: gpa
           Df Sum Sq Mean Sq F value  Pr(>F)
hsm         1   25.8   25.81   52.70 6.6e-12 ***
hss         1    1.2    1.24    2.53    0.11
hse         1    0.7    0.67    1.36    0.25
Residuals 220  107.8    0.49
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Predicted values can be calculated from this model.

```
> lm1fun = makeFun(lm1)
> lm1fun(hsm=9, hss=8, hse=7)

  1
2.7

> lm1fun(hsm=9, hss=8, hse=7:9)

   1    2    3
2.70 2.74 2.79
```

In Figure 11.6 (page 621), the HSS predictor is dropped from the model.

```
> lm2 = lm(gpa ~ hsm + hse, data=ds)
> summary(lm2)


Call:
lm(formula = gpa ~ hsm + hse, data = ds)

Residuals:
    Min      1Q  Median      3Q     Max
-2.0588 -0.3883  0.0695  0.4687  1.7332

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.6242     0.2917    2.14    0.033 *
hsm           0.1827     0.0320    5.72  3.5e-08 ***
hse           0.0607     0.0347    1.75    0.082 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7 on 221 degrees of freedom
Multiple R-squared: 0.202,Adjusted R-squared: 0.194
F-statistic: 27.9 on 2 and 221 DF,  p-value: 1.58e-11

> anova(lm2)

Analysis of Variance Table

Response: gpa
           Df Sum Sq Mean Sq F value  Pr(>F)
hsm         1   25.8   25.81   52.74 6.4e-12 ***
hse         1    1.5    1.49    3.05   0.082 .
```

```
Residuals 221  108.2    0.49
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In Figure 11.7 (page 622), the model is fit using SAT scores as explanatory variables.

```
> lm3 = lm(gpa ~ satm + satv, data=ds)
> summary(lm3)


Call:
lm(formula = gpa ~ satm + satv, data = ds)

Residuals:
    Min      1Q  Median      3Q     Max
-2.5948 -0.3792  0.0826  0.5573  1.3993

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.29e+00   3.76e-01    3.43  0.00073 ***
satm         2.28e-03   6.63e-04    3.44  0.00069 ***
satv        -2.46e-05   6.19e-04   -0.04  0.96836
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.758 on 221 degrees of freedom
Multiple R-squared: 0.0634,Adjusted R-squared: 0.0549
F-statistic: 7.48 on 2 and 221 DF,  p-value: 0.000722

> anova(lm3)

Analysis of Variance Table

Response: gpa
           Df Sum Sq Mean Sq F value  Pr(>F)
satm        1    8.6    8.58    14.9 0.00015 ***
satv        1    0.0    0.00     0.0 0.96836
Residuals 221  126.9    0.57
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Finally, in Figure 11.8 (page 624), all possible explanatory variables are used.

```
> lm.full = lm(gpa ~ satm + satv + hsm + hss + hse, data=ds)
> summary(lm.full)
```

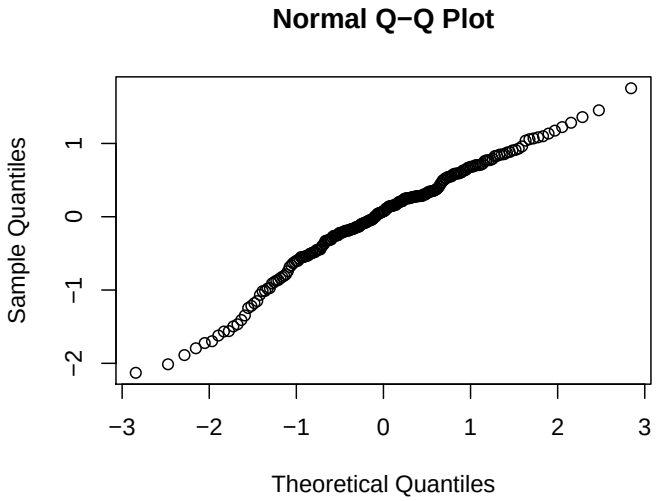
```
Call:
lm(formula = gpa ~ satm + satv + hsm + hss + hse, data = ds)

Residuals:
    Min      1Q  Median      3Q     Max
-2.0649 -0.3084  0.0689  0.4876  1.7054

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.326719   0.399996    0.82  0.41493
satm         0.000944   0.000686    1.38  0.17018
satv        -0.000408   0.000592   -0.69  0.49152
hsm          0.145961   0.039261    3.72  0.00026 ***
hss          0.035905   0.037798    0.95  0.34321
hse          0.055293   0.039569    1.40  0.16372
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7 on 218 degrees of freedom
Multiple R-squared: 0.211,Adjusted R-squared: 0.193
F-statistic: 11.7 on 5 and 218 DF,  p-value: 5.06e-10

> anova(lm.full)

Analysis of Variance Table

Response: gpa
           Df Sum Sq Mean Sq F value  Pr(>F)
satm        1    8.6    8.58   17.52 4.1e-05 ***
satv        1    0.0    0.00    0.00   0.966
hsm         1   17.7   17.73   36.18 7.5e-09 ***
hss         1    1.4    1.38    2.81   0.095 .
hse         1    1.0    0.96    1.95   0.164
Residuals 218  106.8    0.49
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

1.4 Regression diagnostics

As always, we want to assess the fit of the model, and the assumptions needed for it.

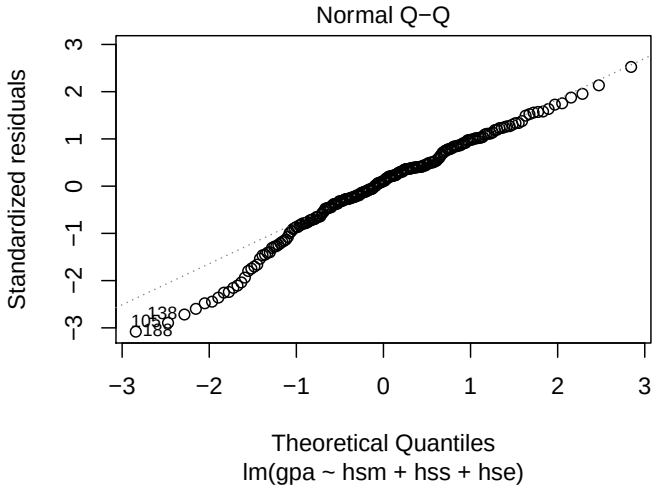
We begin by considering the distribution of the residuals. Figure 11.5 (page 620) displays the normal quantile plot, which can be generated using the `qqnorm()` function.

```
> qqnorm(residuals(lm1))
```



This can also be generated using a built-in plot option:

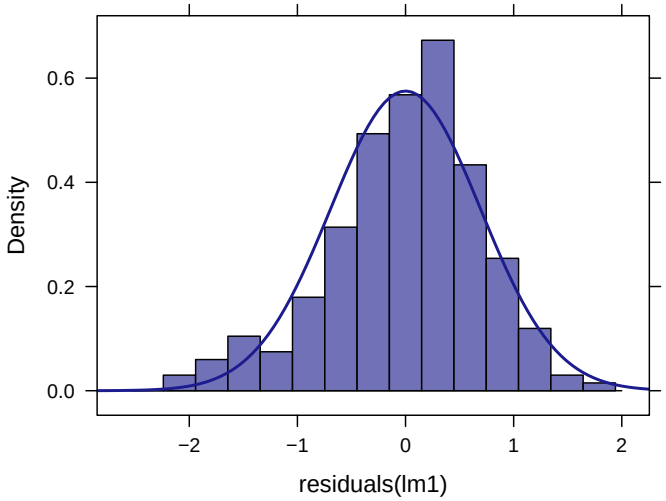
```
> plot(lm1, which=2)
```



Both displays indicate that the distribution of the residuals is approximately normal (with some evidence for a slightly heavy left tail).

We could also generate a histogram with overlaid normal density (mean 0 and standard deviation equal to the root MSE from the model).

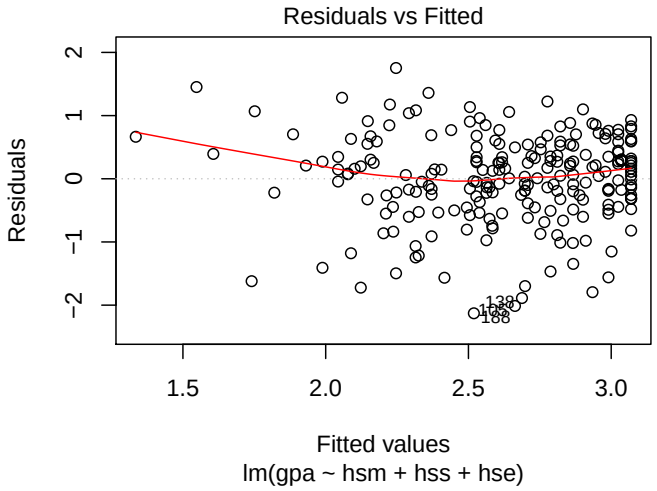
```
> xhistogram(~ residuals(lm1), fit="normal")

Loading required package:  MASS
```



Next we want to consider the distribution of the residuals as a function of the fitted (predicted) values, as we don't want to see a systematic pattern in the relationship between these quantities.

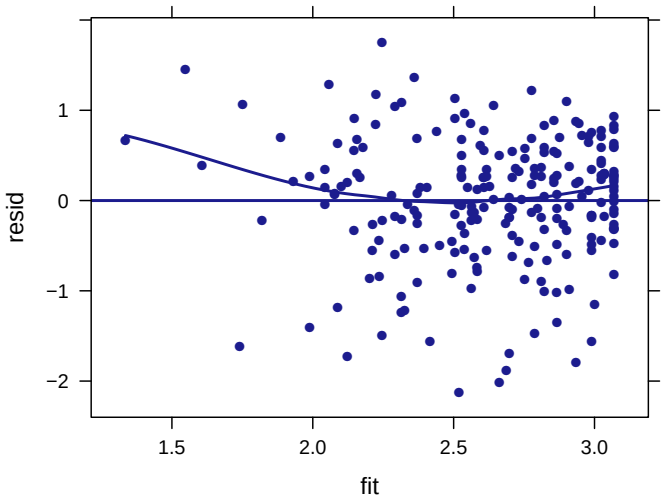
As is often the case, there are multiple ways to generate these plots within R.

```
> plot(lm1, which=1)
```



This can also be created in other ways:

```
> ds = transform(ds, fit=fitted(lm1))
> ds = transform(ds, resid=residuals(lm1))
> xyplot(resid ~ fit, type=c("p", "r", "smooth"), data=ds)
```



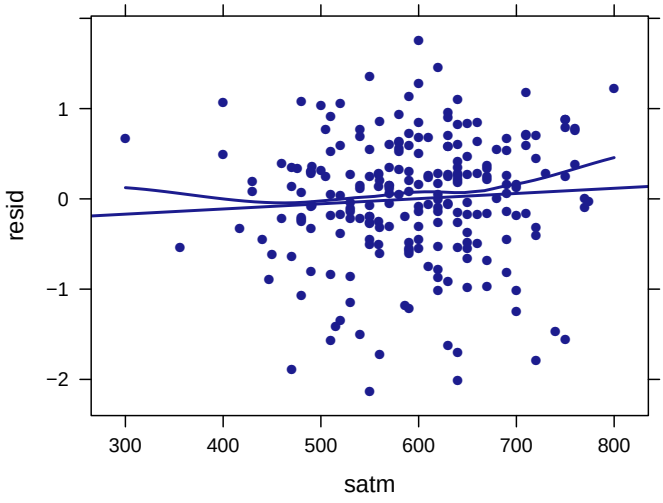
Here the `"r"` option for `type` specifies a regression (which will be a straight line), and the `"smooth"` option adds a lowess (smooth line). There is some indication of non-linearity, particularly in the tails (but that's exactly where the lowess isn't to be trusted, as there's little data).

Subject 188 may bear some additional scrutiny (as it has a very large negative residual):

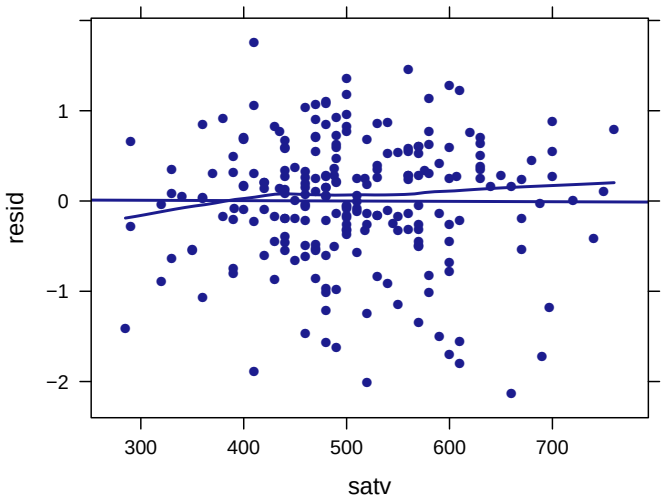
```
> ds[188,]

    obs  gpa hsm hss hse satm satv sex jhsm jhss jhse  fit resid
188 188 0.39   7  10   9  550  660   2 7.16 9.97 8.85 2.52 -2.13
```

We also want to display the residuals against each of the continuous predictors in the model.

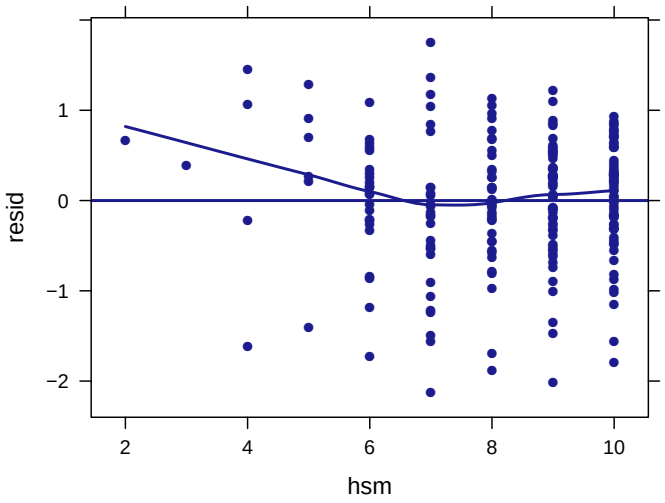
```
> xyplot(resid ~ satm, type=c("p", "r", "smooth"), data=ds)
```



```
> xyplot(resid ~ satv, type=c("p", "r", "smooth"), data=ds)
```

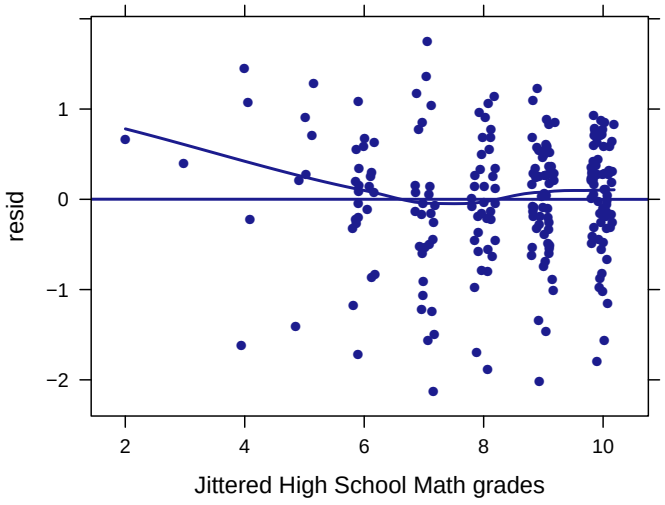



```
> xyplot(resid ~ hsm, type=c("p", "r", "smooth"), data=ds)

Warning:  pseudoinverse used at 9
Warning:  neighborhood radius 1
Warning:  reciprocal condition number  0
Warning:  pseudoinverse used at 9
Warning:  neighborhood radius 1
Warning:  reciprocal condition number  0
Warning:  pseudoinverse used at 9
Warning:  neighborhood radius 1
Warning:  reciprocal condition number  0
Warning:  pseudoinverse used at 9
Warning:  neighborhood radius 1
Warning:  reciprocal condition number  0
Warning:  pseudoinverse used at 9
Warning:  neighborhood radius 1
Warning:  reciprocal condition number  0
```

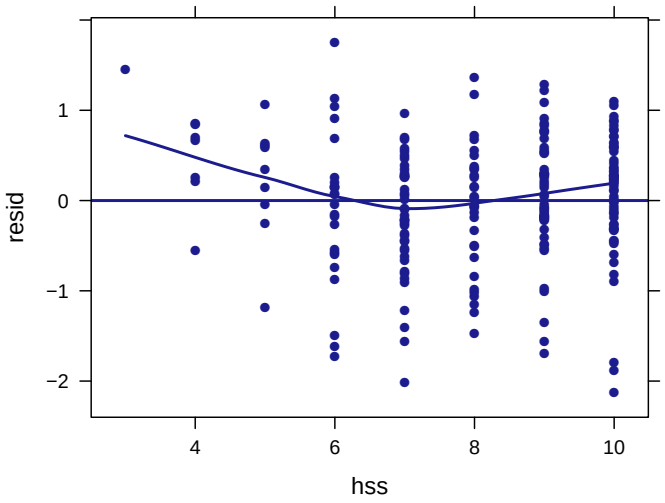


The warnings are due to the discrete nature of the high school math variable, which takes on relatively few values. Jittering will help here:

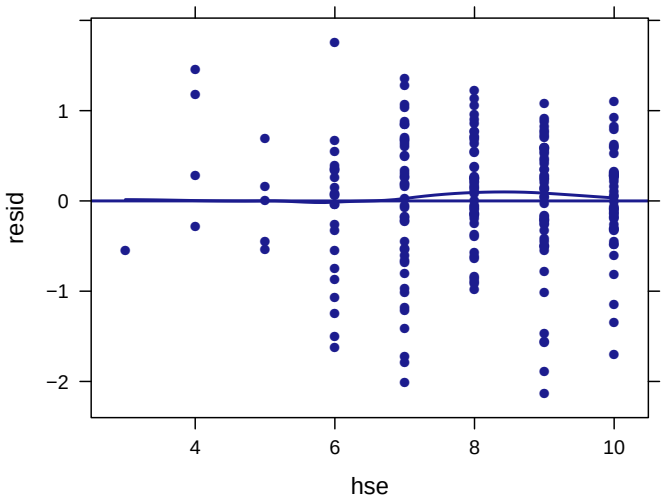
```
> xyplot(resid ~ jhsm, type=c("p", "r", "smooth"),
   xlab="Jittered High School Math grades", data=ds)
```




```
> xyplot(resid ~ hss, type=c("p", "r", "smooth"), data=ds)
```




```
> xyplot(resid ~ hse, type=c("p", "r", "smooth"), data=ds)
```



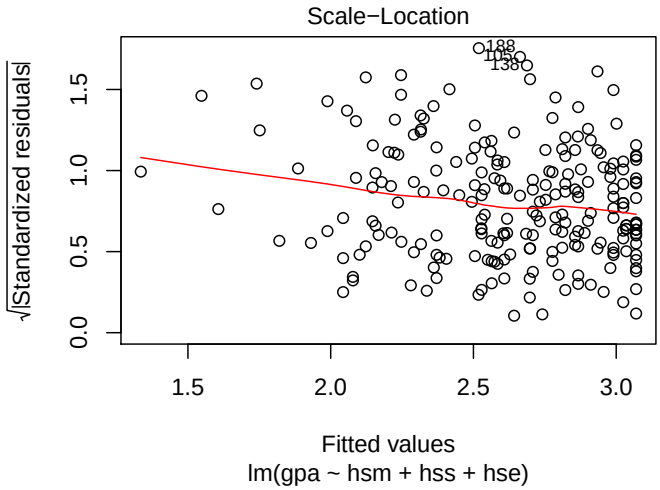
Overall, we see reasonable linearity in the relationships, though for some subjects with low high school math or science grades, the regression model tends to systematically underpredict.

```
> subset(ds, hsm < 4 | hss < 4)

    obs gpa hsm hss hse satm satv sex jhsm jhss jhse  fit resid
8     8   2   3   7   6  460  530   1 2.98 7.03 6.03 1.61 0.394
84   84   3   4   3   4  620  560   1 4.00 2.97 4.09 1.55 1.453
183 183   2   2   4   6  300  290   2 2.00 3.97 6.02 1.33 0.665
```

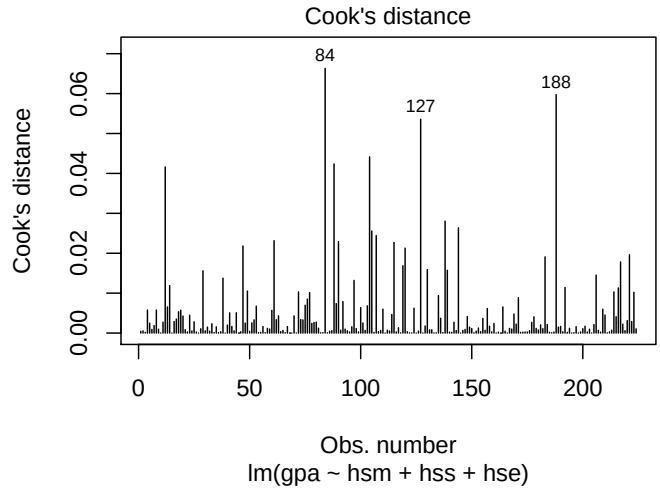
1.5 More advanced residual analysis and regression diagnostics

Additional built-in residual plots can be requested (but we won't be using these much: check out *The Statistical Sleuth* for more information).

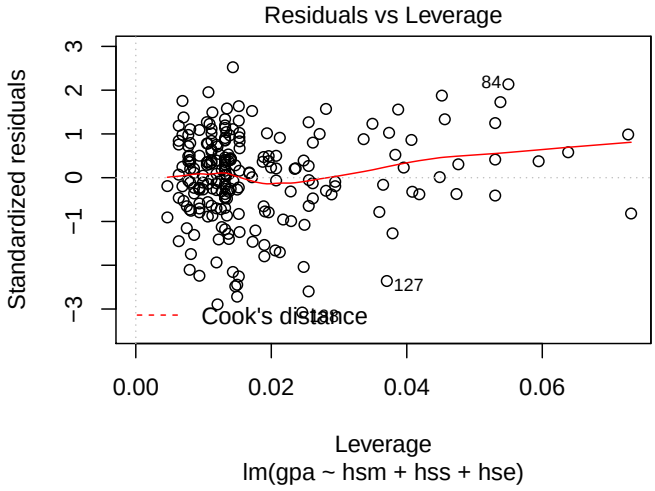
```
> plot(lm1, which=3)
```




```
> plot(lm1, which=4)
```




```
> plot(lm1, which=5)
```




```
> plot(lm1, which=6)
```