Lecture 4: Gauges

Last time

We started with the hat matrix, introducing at least three ways to think about the concept of “leverage”

Next, we returned to the vulnerability data and examined the best way to include polynomial terms -- We got stopped there on the concept of an equivalent kernel, something I rushed...

Leverage

We noticed that if you duplicated the ith point in the design then, letting the ith diagonal value of the old hat matrix be $h_i = 1/k$, the (n+1)st diagonal element of the new hat matrix is 1/(k+1)

Given this kind of scaling, one interpretation of leverage is the number of equivalent observations that go into fitting $\widehat{\mu}_i$ -- If this number is small, 1 or 2 or even 5, then Huber says we should worry

Leverage

We also saw that the diagonal elements of the hat matrix can be interpreted as a kind of distance between the ith design point h_i (the vector of independent variables for the ith observation) and the rest of the data cloud

A poor man's version of this uses all n independent variables to estimate the mean and variance-covariance matrix of a multivariate normal distribution, and then evaluates the Mahalanobis distance between the ith data point and this fit -- The value turns out to be $(n-1)(h_i - 1/n)$

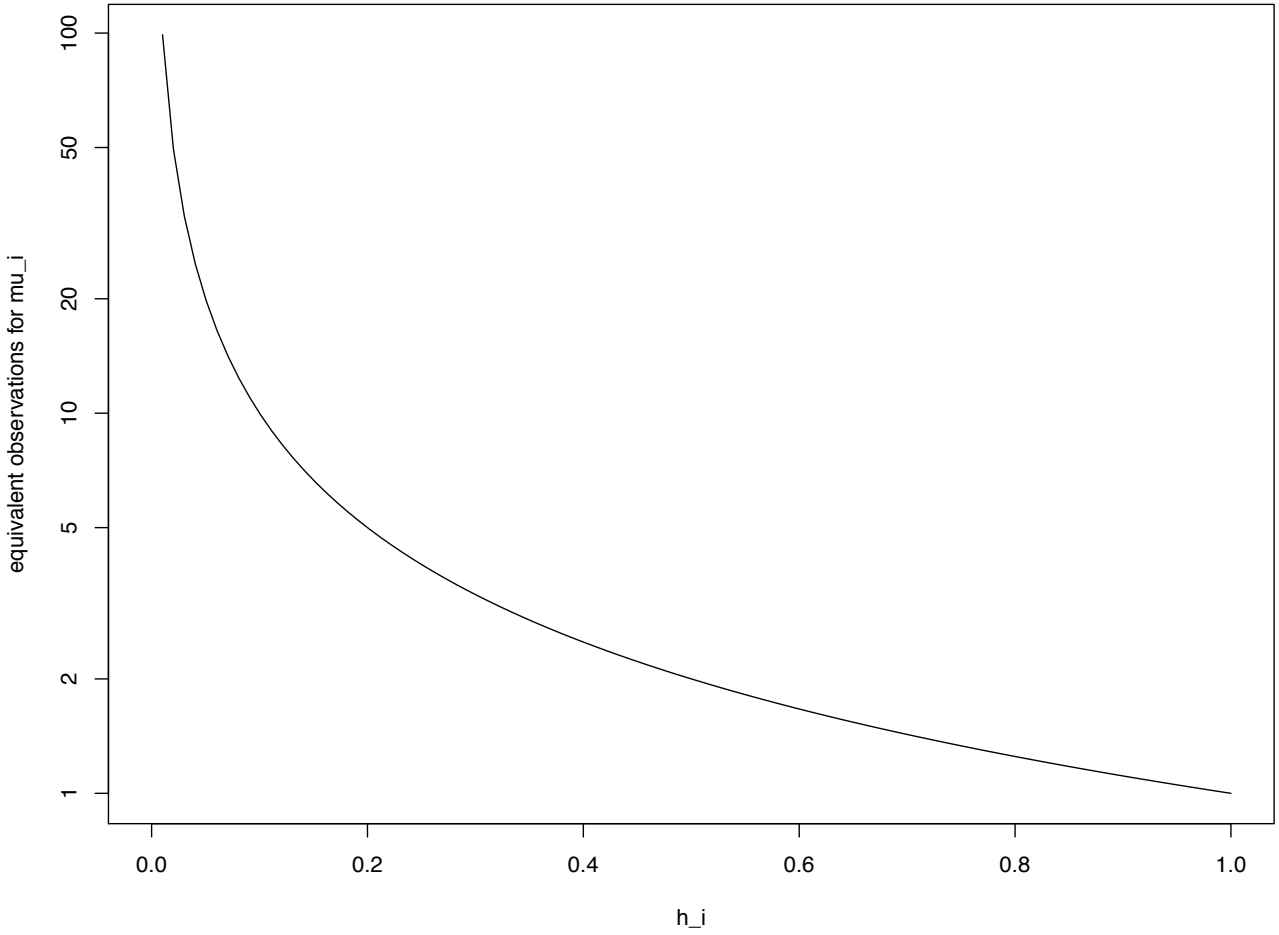
If we work a little harder, we can perform the fit leaving out the ith point -- This gives a value of

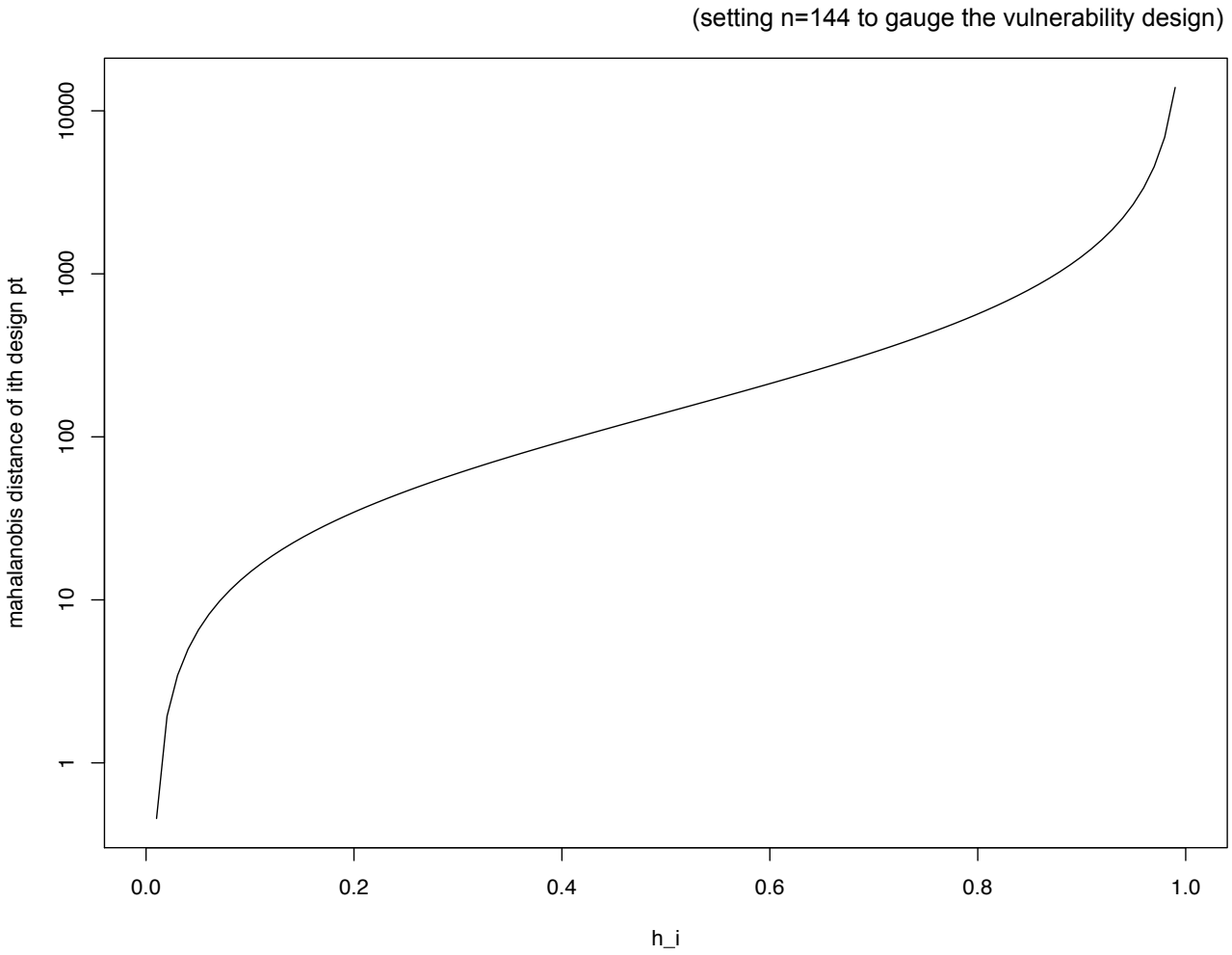
$$\left(\frac{h_i - 1/n}{1 - h_i}\right)\frac{n(n-2)}{n-1}$$

Leverage

In the first case, we sense a difficulty because the fit is determined by just a small number of observations -- The point of regression, after all, is to borrow strength when estimating a conditional mean and having some fitted value depend only on one or two points is potentially problematic

Similarly, with the distance interpretation, we see that data points that live far from the rest of the data, we have the potential for damage -- If we really had normal data for our inputs, then the Mahalanobis distance measure comes with a Chisquare ruler for interpretation


equivalent observations for mu_i
100
50
20
10
5
2
1
0.0
0.2
0.4
0.6
0.8
1.0
h_i


(setting n=144 to gauge the vulnerability design)
mahalanobis distance of ith design pt
10000
1000
100
10
1
0.0
0.2
0.4
0.6
0.8
1.0
h_i

Fitting the right model

Let's now pick up where we left off last time, fitting with the author's final model -- That is, we include both HDI and the square of HDI for a quadratic fit

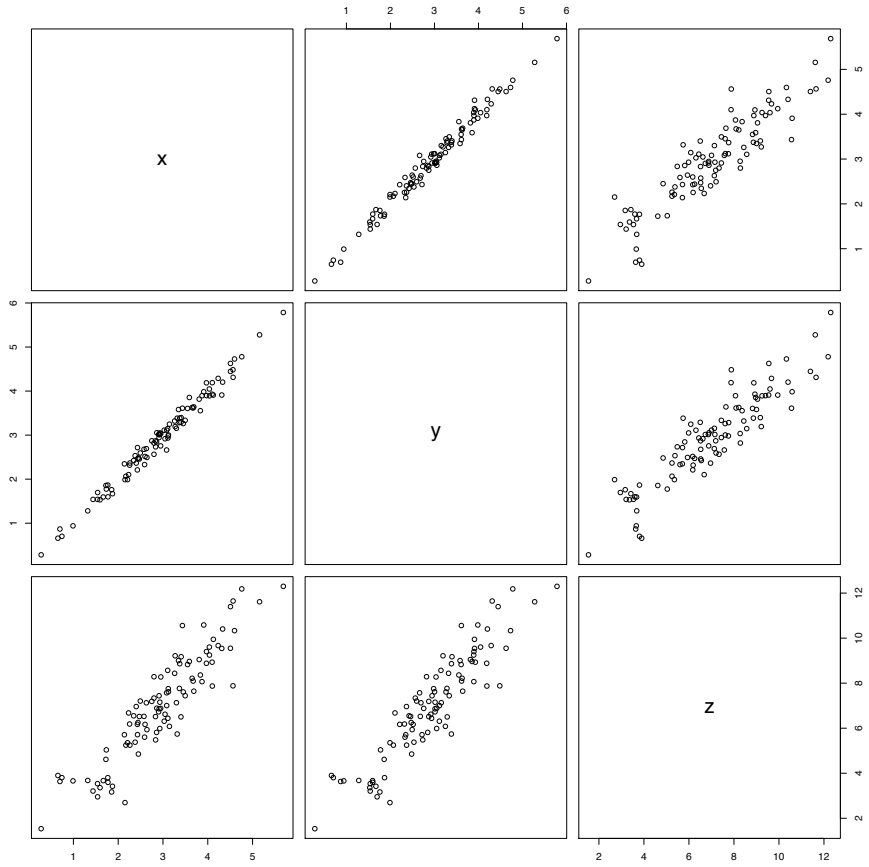
Recall we spent some time talking about instability that comes from having a set of collinear predictors -- This kind of instability seemed intuitive and we introduced the Variance Inflation Factor as a way to gauge it

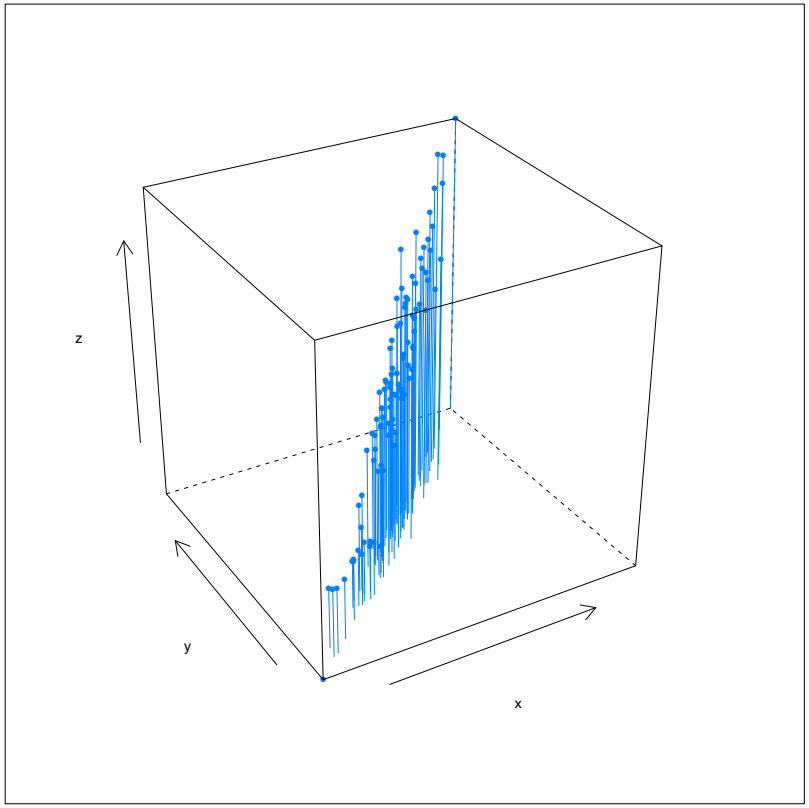
$$\mathrm{VIF}_i = \frac{1}{1 - R^2}$$

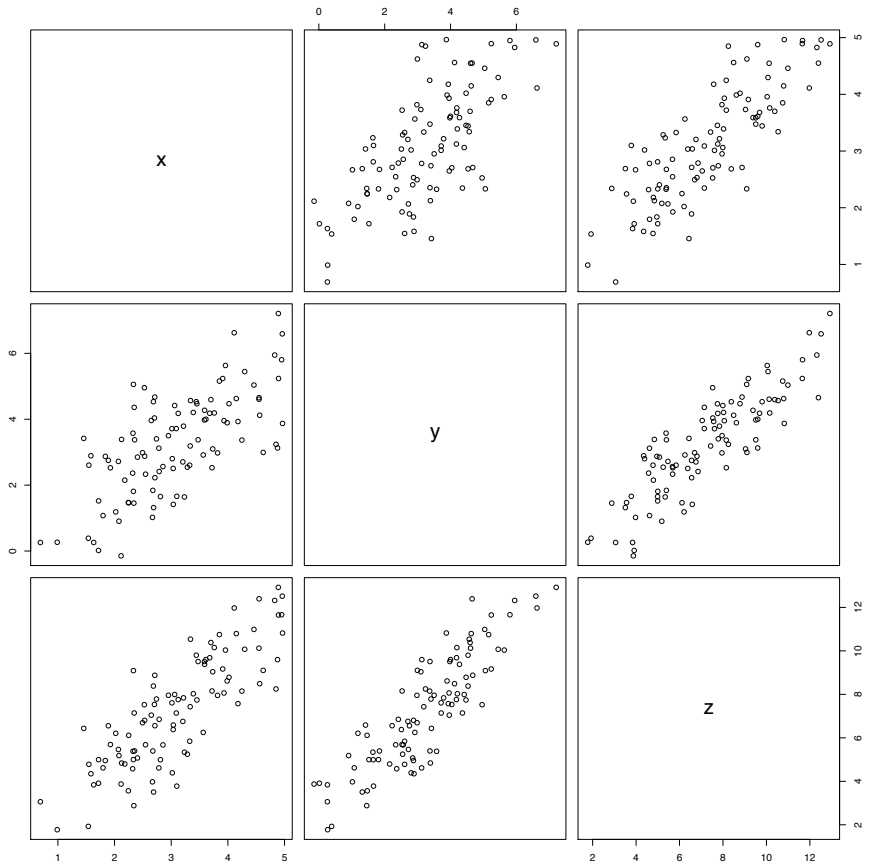
and we stated the relationship

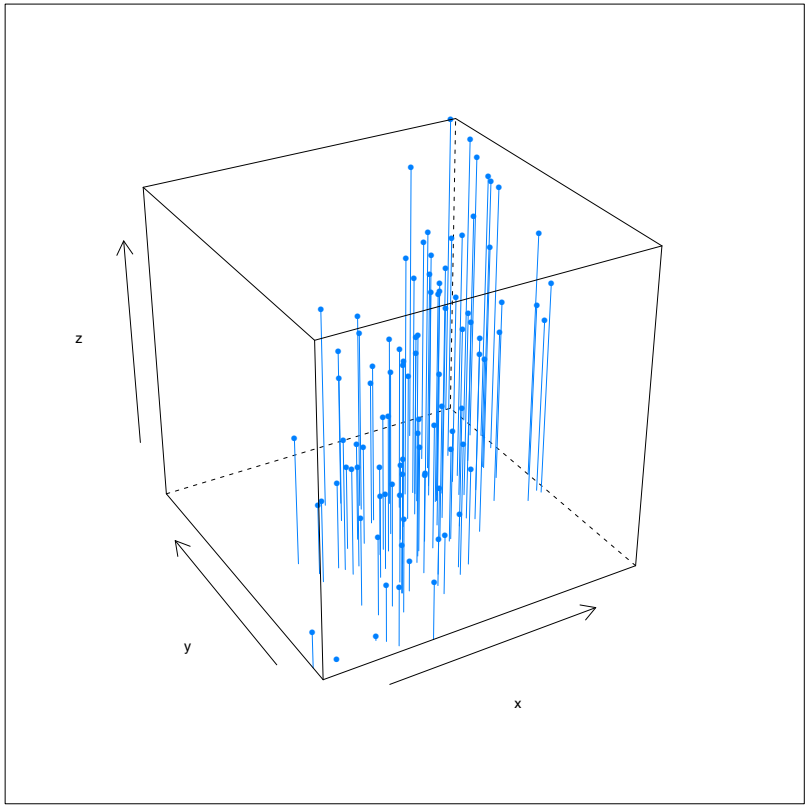
$$\mathrm{se}(\widehat{\beta}_j) = \widehat{\sigma}\sqrt{\frac{\mathrm{VIF}_j}{\sum_i (x_{ij} - \bar{x}_j)^2}}$$

where we recall that the variance-covariance matrix of the coefficients is just $\sigma^2(M^t M)^{-1}$


x
y
z
1
2
3
4
5
6
2
4
6
8
10
12


z
y
x


x
y
z


z
y
x

```
# fit without quadratic on hdi for the moment

fit <- lm(ln_death_risk~ln_events+ln_fert+ln_pop+hdi,data=vul)
summary(fit)

# Call:
# lm(formula = ln_death_risk ~ ln_events + ln_fert + ln_pop + hdi,
#    data = vul)
#
# Residuals:
#     Min      1Q  Median      3Q     Max
# -3.4518 -0.7673 -0.1513  0.5669  6.2271
#
# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)  -5.3485     1.5175  -3.524 0.000575 ***
# ln_events     1.3708     0.1792   7.649 3.04e-12 ***
# ln_fert       2.1961     0.4614   4.760 4.81e-06 ***
# ln_pop       -0.5672     0.1026  -5.529 1.54e-07 ***
# hdi           1.9922     1.2628   1.578 0.116928
# ---
# Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 #
# Residual standard error: 1.35 on 139 degrees of freedom
# Multiple R-squared: 0.4221,!Adjusted R-squared: 0.4055
# F-statistic: 25.38 on 4 and 139 DF,  p-value: 8.522e-16
```

```
fit <- lm(ln_death_risk~ln_events+ln_fert+ln_pop+hdi+I(hdi^2),data=vul)
summary(fit)

# Call:
# lm(formula = ln_death_risk ~ ln_events + ln_fert + ln_pop + hdi +
#     I(hdi^2), data = vul)
#
# Residuals:
#      Min       1Q   Median       3Q      Max
# -3.81655 -0.80298 -0.04575  0.63866  5.60679
#
# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept) -10.92243    1.81119  -6.031 1.42e-08 ***
# ln_events     1.42774    0.16648   8.576 1.79e-14 ***
# ln_fert       1.47558    0.45232   3.262  0.00139 **
# ln_pop       -0.56450    0.09507  -5.938 2.22e-08 ***
# hdi          25.06179    4.86656   5.150 8.81e-07 ***
# I(hdi^2)    -18.89905    3.86980  -4.884 2.84e-06 ***
# ---
# Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 #
# Residual standard error: 1.251 on 138 degrees of freedom
# Multiple R-squared: 0.5073,!Adjusted R-squared: 0.4894
# F-statistic: 28.41 on 5 and 138 DF,  p-value: < 2.2e-16
```

```
> vif(simu1[,-3])
       x        y
54.07696 54.07696

> vif(simu2[,-3])
       x        y
1.902763 1.902763

> vif(cbind(vul[,c("ln_events","ln_fert","ln_pop","hdi")],vul$hdi^2))

ln_events   ln_fert    ln_pop       hdi vul$hdi^2
 2.433686  4.076010  2.460708 65.156728 71.792326
```

Orthogonal polynomials

The typical approach to fitting with polynomials is not to toss in higher-order monomials, but instead to orthogonalize the variables as they are introduced into the regression

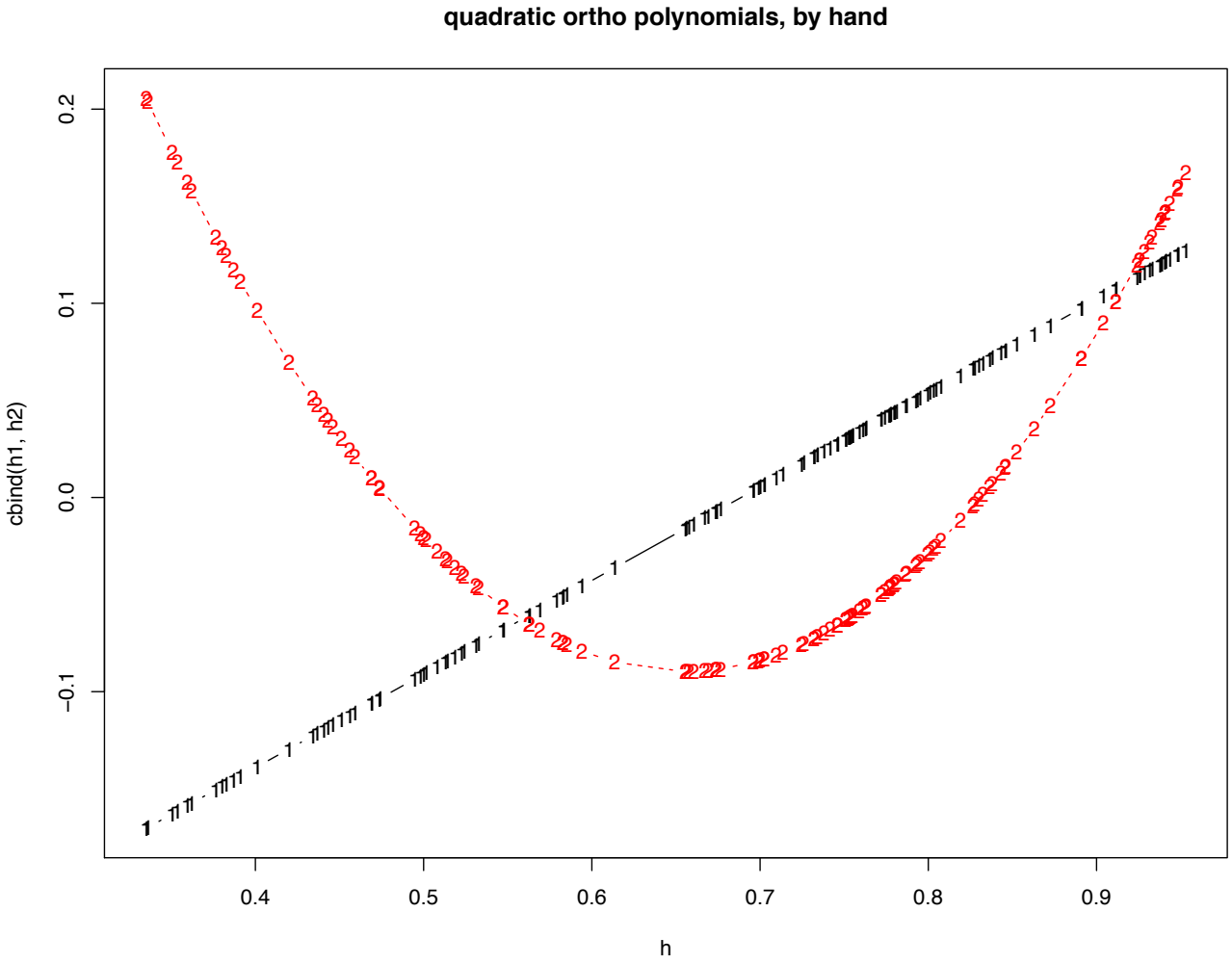
We know that the space of quadratic polynomials in HDI are spanned by three columns, say, $1, \mathrm{HDI}, \mathrm{HDI}^2$

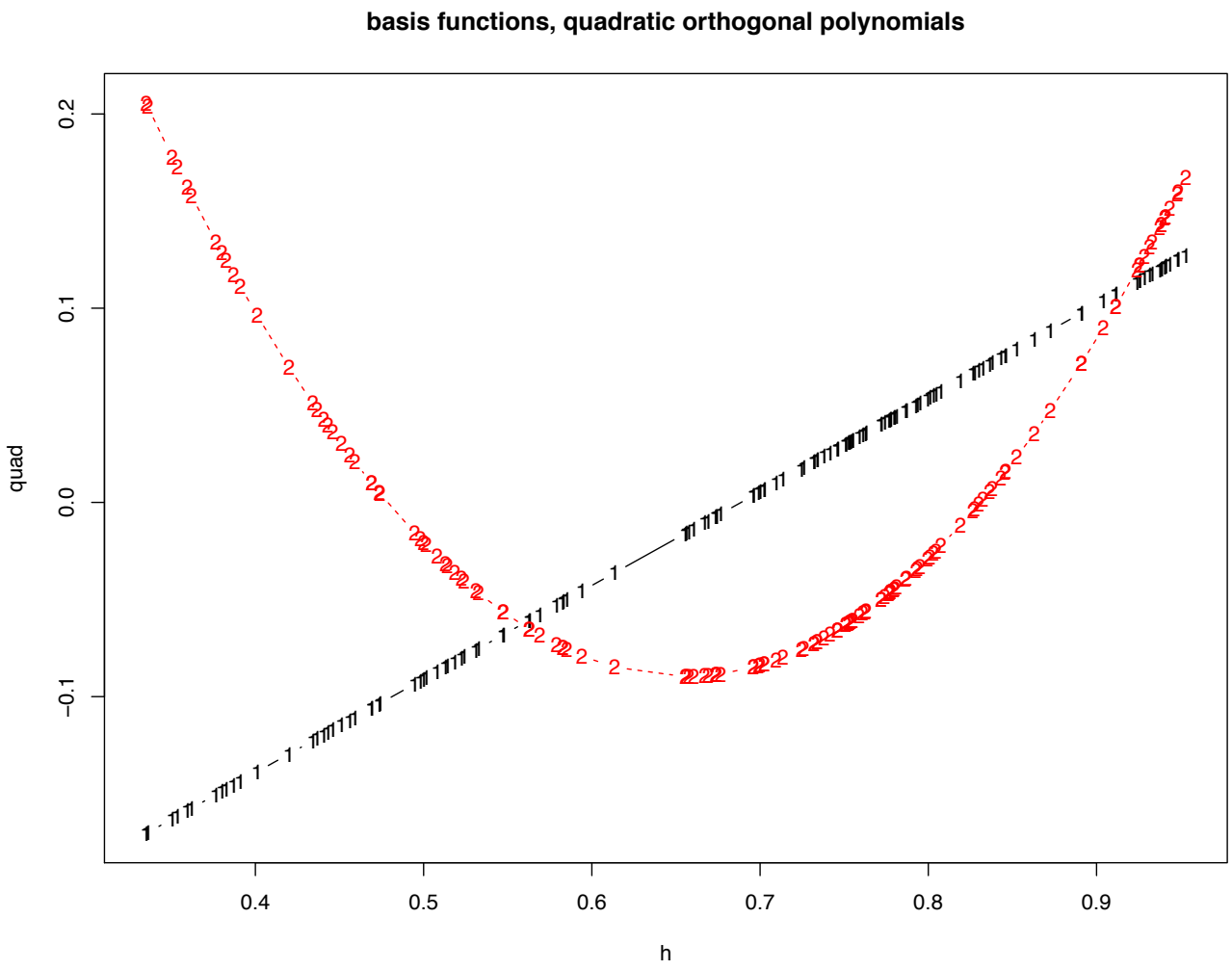
Suppose we add them into the model sequentially, each time adding not the whole variable, but instead the residuals after having regressed out the terms that came before

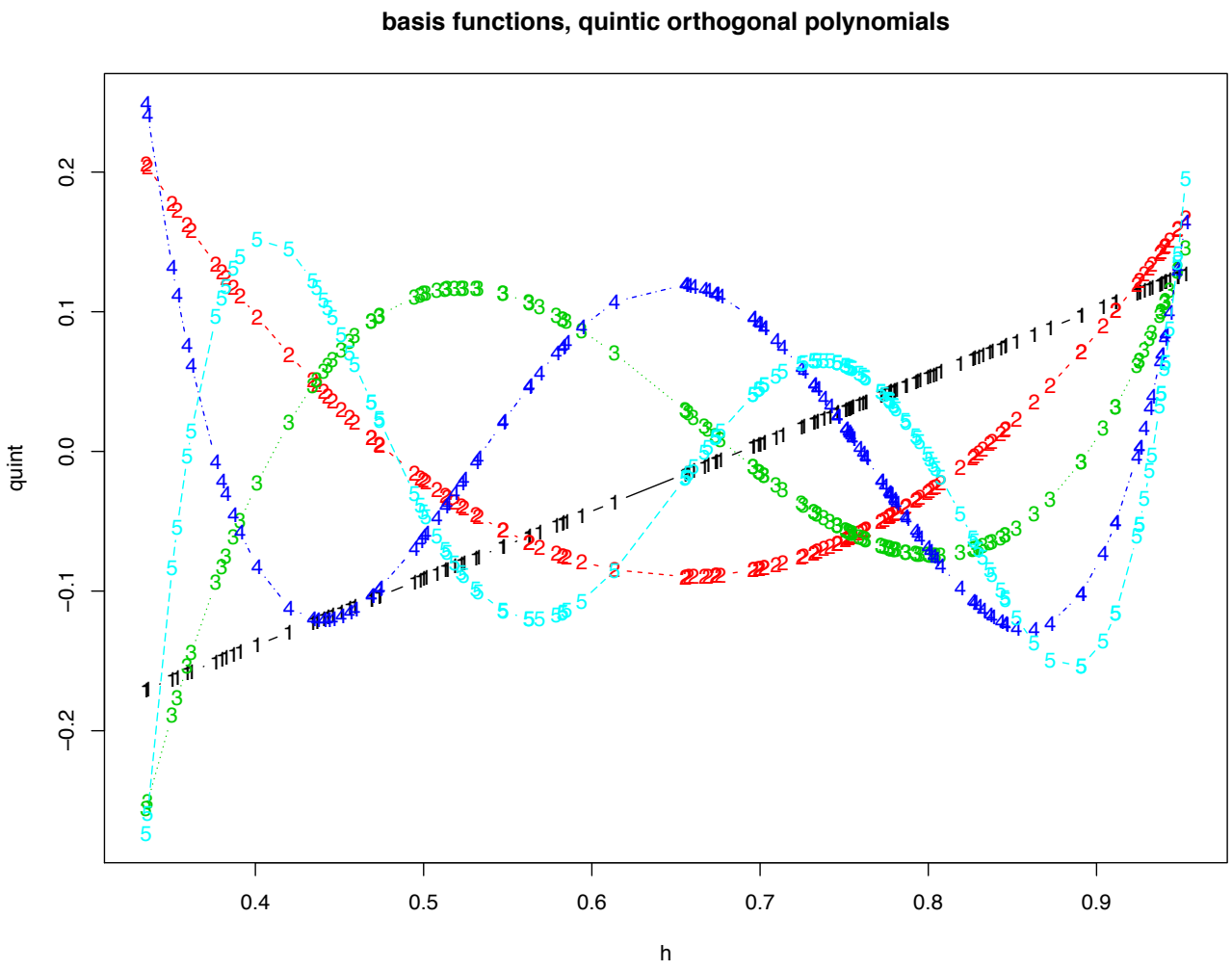
Orthogonal polynomials

So the first term to enter is simply the constant vector -- To regress the variable HDI onto this “variable” means...

```
> h <- sort(vul$hdi)

> h0 <- rep(1,length(h))
> h0 <- h0/sqrt(sum(h0^2))

> h1 <- h
> h1 <- h1-mean(h1)
> h1 <- h1/sqrt(sum(h1^2))

> h2 <- h^2
> h2 <- residuals(lm(h2~h0+h1-1))
> h2 <- h2/sqrt(sum(h2^2))

> matplot(h,cbind(h1,h2),type="b",main="quadratic ortho polynomials, by hand")

> t(cbind(h0,h1,h2))%*%cbind(h0,h1,h2)
              h0            h1            h2
h0  1.000000e+00 -3.625572e-16 -8.153200e-17
h1 -3.625572e-16  1.000000e+00  3.122502e-16
h2 -8.153200e-17  3.122502e-16  1.000000e+00

> vif(cbind(vul[,c("ln_events","ln_fert","ln_pop")],vul$hdi,vul$hdi^2))
ln_events   ln_fert    ln_pop   vul$hdi vul$hdi^2
 2.433686  4.076010  2.460708 65.156728 71.792326

> vif(cbind(vul[,c("ln_events","ln_fert","ln_pop")],poly(vul$hdi,2)))
ln_events   ln_fert    ln_pop         1         2
 2.433686  4.076010  2.460708  4.092598  1.145813
```


quadratic ortho polynomials, by hand
cbind(h1, h2)
0.2
0.1
0.0
−0.1
0.4
0.5
0.6
0.7
0.8
0.9
h


basis functions, quadratic orthogonal polynomials
quad
h
0.2
0.1
0.0
−0.1
0.4
0.5
0.6
0.7
0.8
0.9

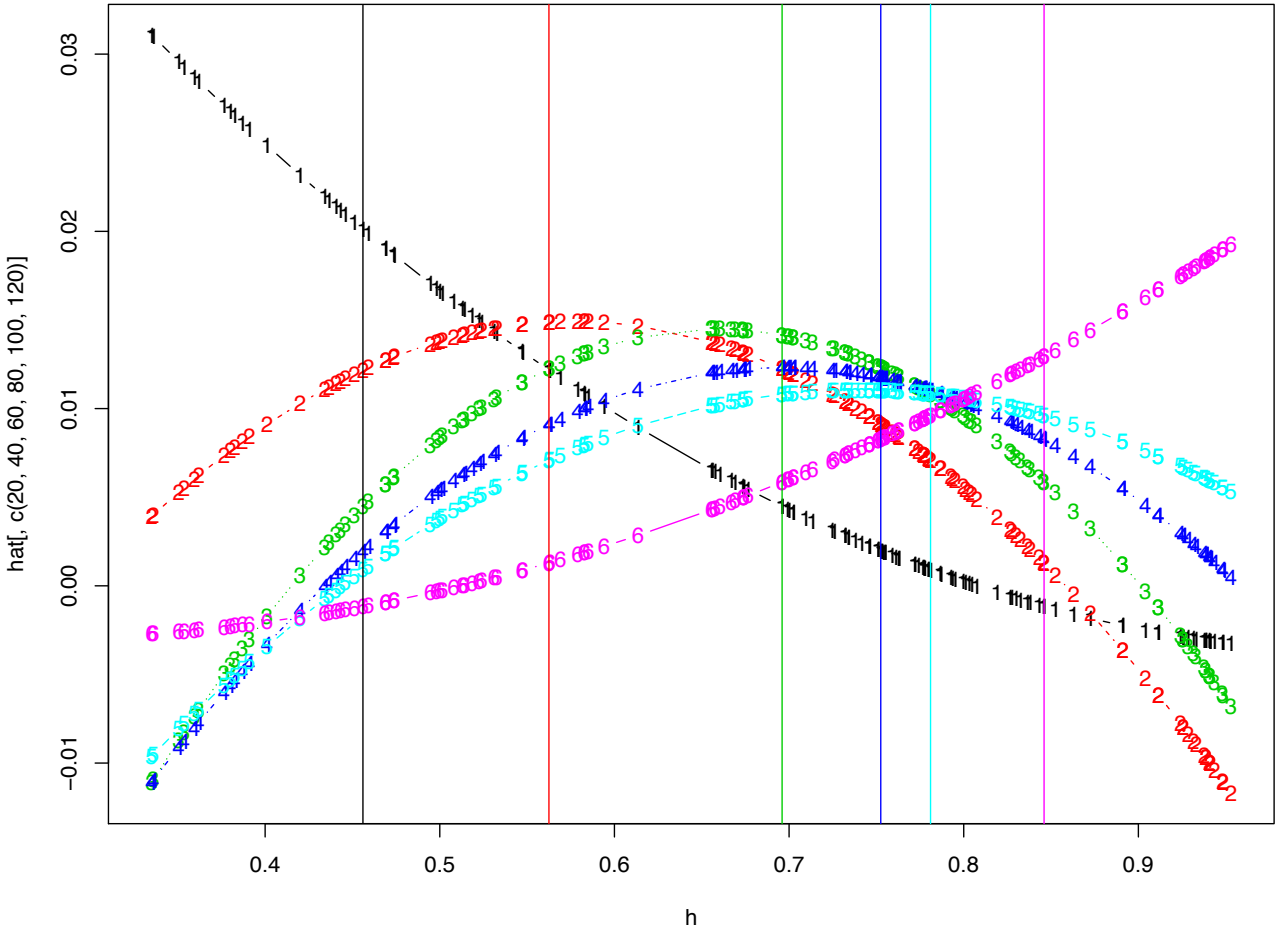

basis functions, quintic orthogonal polynomials
quint
h
0.2
0.1
0.0
−0.1
−0.2
0.4
0.5
0.6
0.7
0.8
0.9

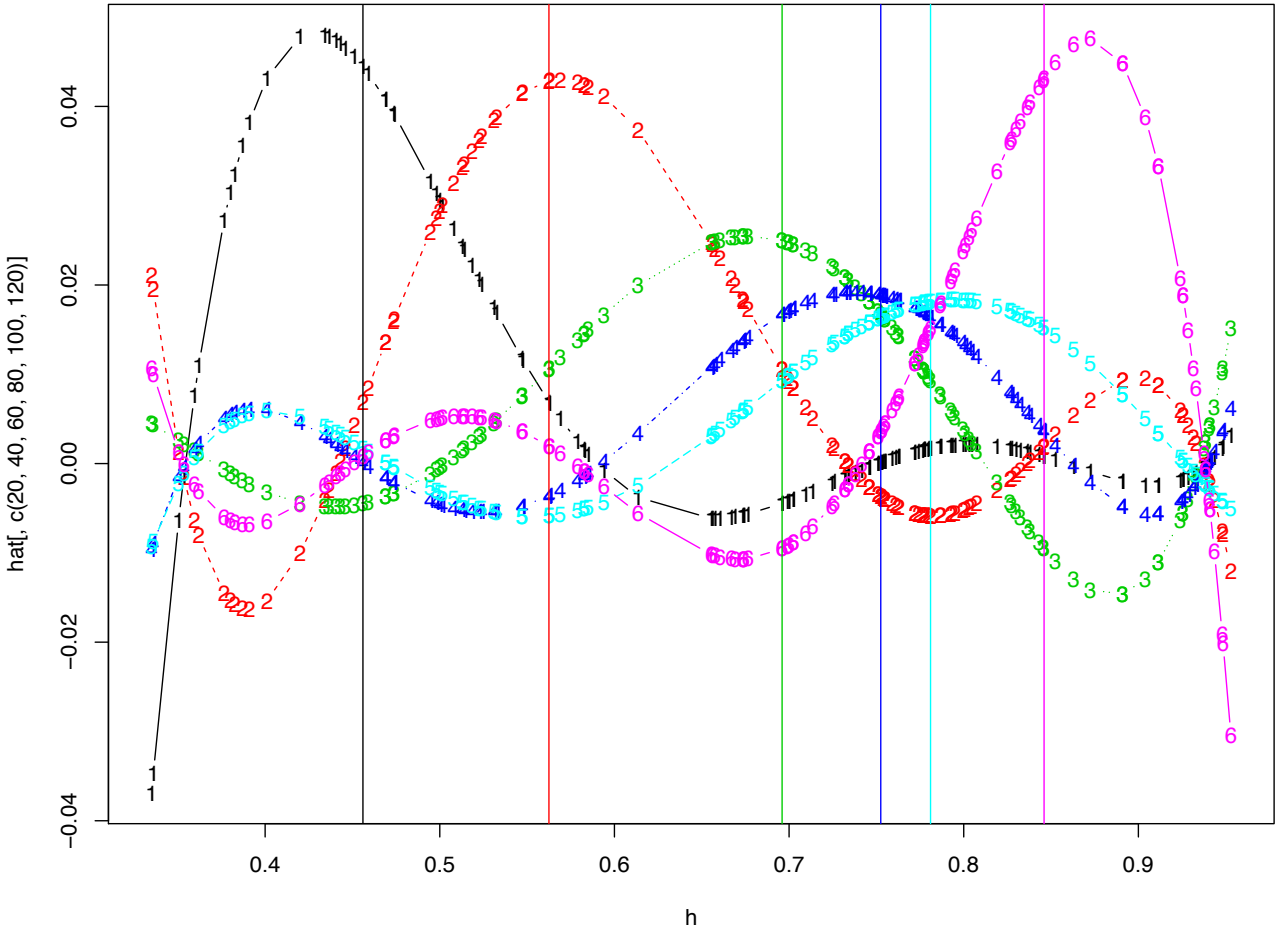
Equivalent kernels

Suppose for the moment that we model solely with HDI as our predictor and we consider (orthogonal) polynomials of higher order -- We can use what we know about the hat matrix to get a glimpse into what this kind of fitting is doing

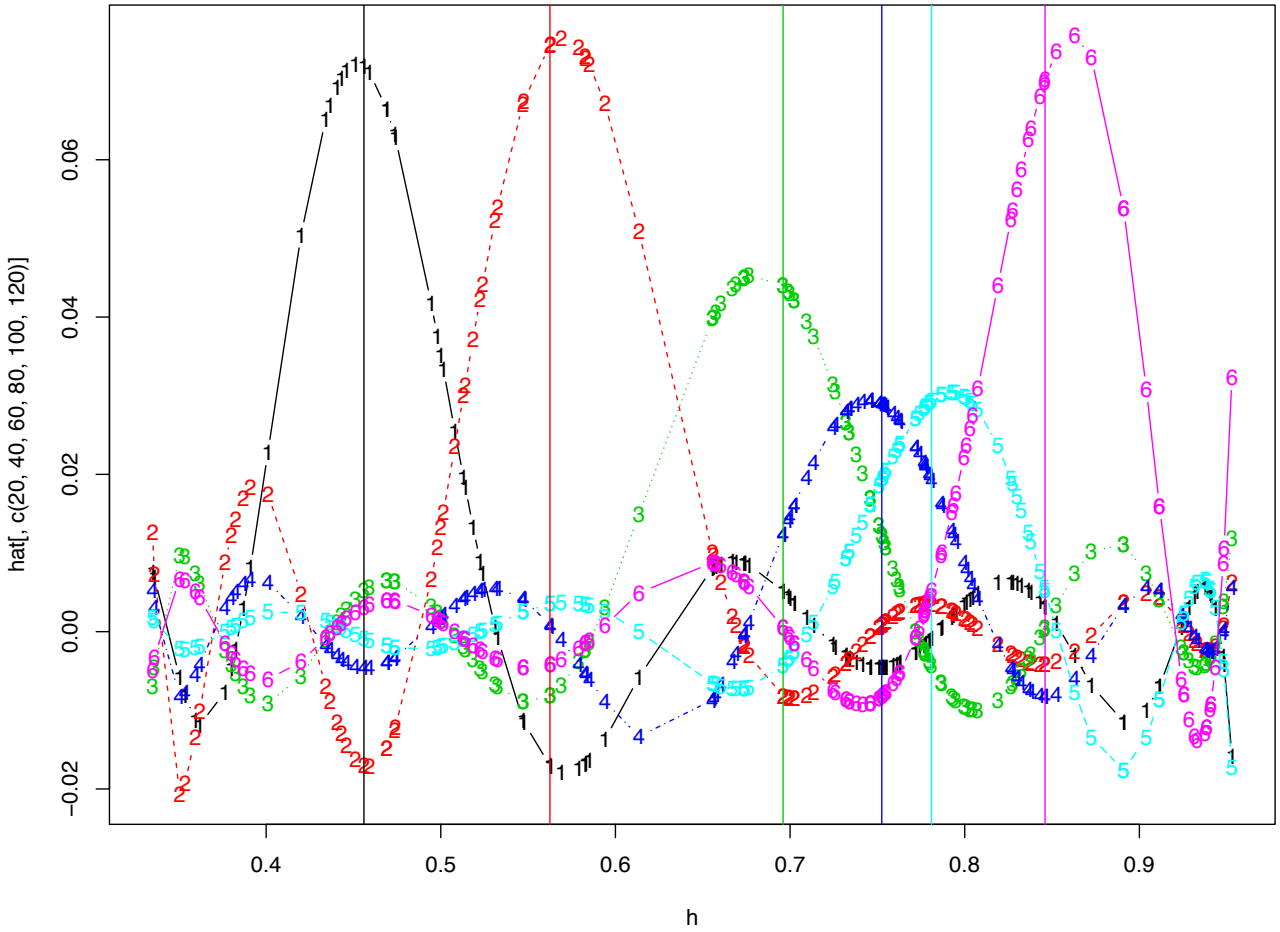
On the next slides, we present rows of the hat matrix (the 20th, 40th, 60th, 80th, 100th and 120th) as functions of HDI

```
# exhibit rows of hat matrix as a function of hdi, quadratic fit

M <- cbind(1/sqrt(length(h)),poly(h,2))
hat <- M%*%t(M)
matplot(h,hat[,c(20,40,60,80,100,120)],type="b")
for(i in 1:6) abline(v=h[20*i],col=i)

# and for a quintic fit...

M <- cbind(1/sqrt(length(h)),poly(h,5))
hat <- M%*%t(M)
matplot(h,hat[,c(20,40,60,80,100,120)],type="b")
for(i in 1:6) abline(v=h[20*i],col=i)
```


hat[, c(20, 40, 60, 80, 100, 120)]
0.03
0.02
0.01
0.00
−0.01
0.4
0.5
0.6
0.7
0.8
0.9
h


hat[, c(20, 40, 60, 80, 100, 120)]
0.04
0.02
0.00
−0.02
−0.04
0.4
0.5
0.6
0.7
0.8
0.9
h

And for a 10th degree poly...



Equivalent kernels

While this is skipping ahead a little, the rows of the hat matrix formed when performing a regression with polynomials, or piecewise polynomials or splines can be thought of as kernel functions

When we use linear regression and polynomials the projection or hat matrix that takes $\widehat{\mu} = \mathrm{M}(\mathrm{M}^{\mathrm{t}}\mathrm{M})^{-1}\mathrm{M}\mathrm{y} = \mathrm{H}\mathrm{y}$ -- It turns out that a large class of so-called linear smoothers are of the general form $\widehat{\mu} = \mathrm{S}\mathrm{y}$

The ith row of S then represent weights applied to the observations in y to form the estimated $\widehat{\mu}_{\mathrm{i}}$

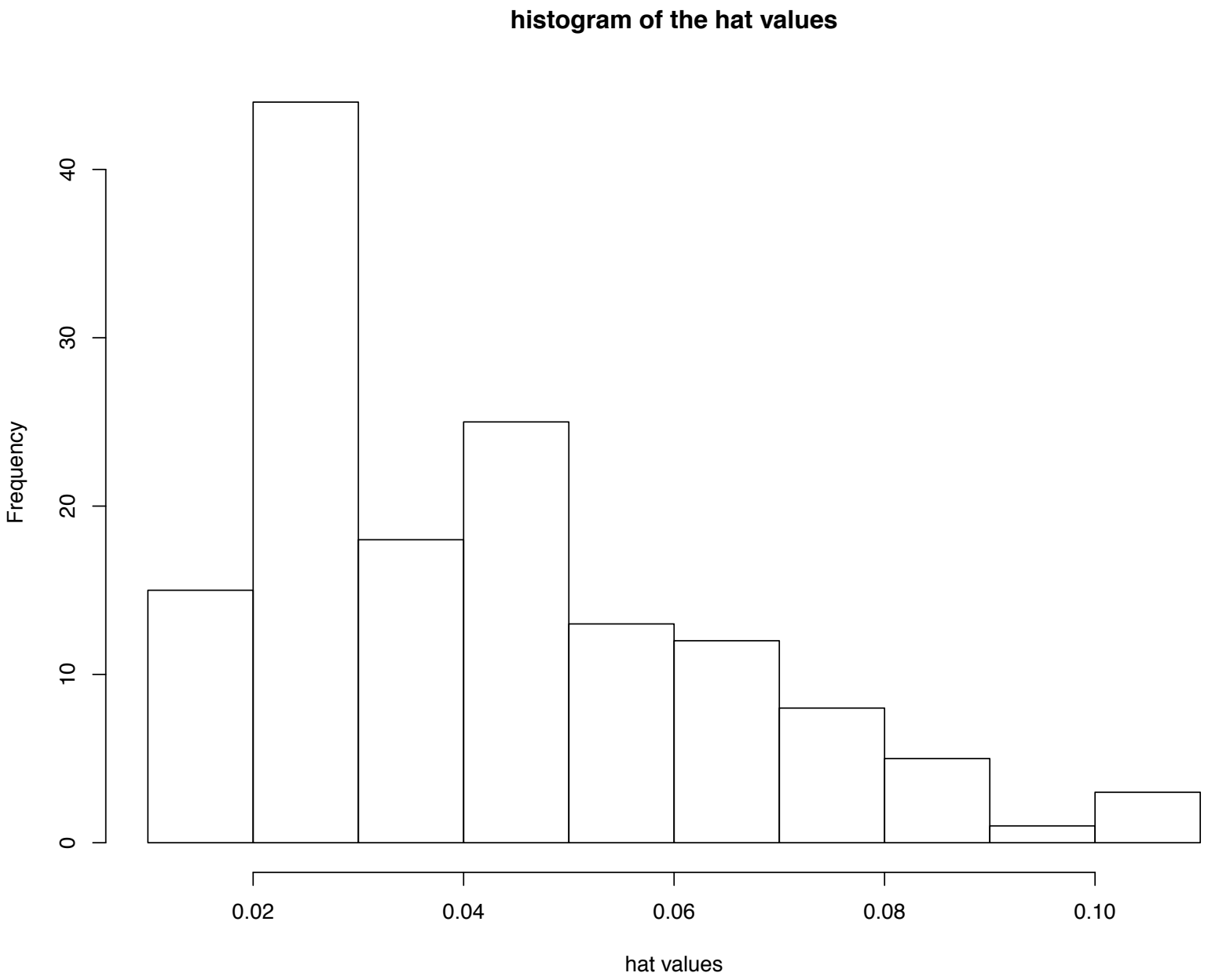
Orthogonal polynomials

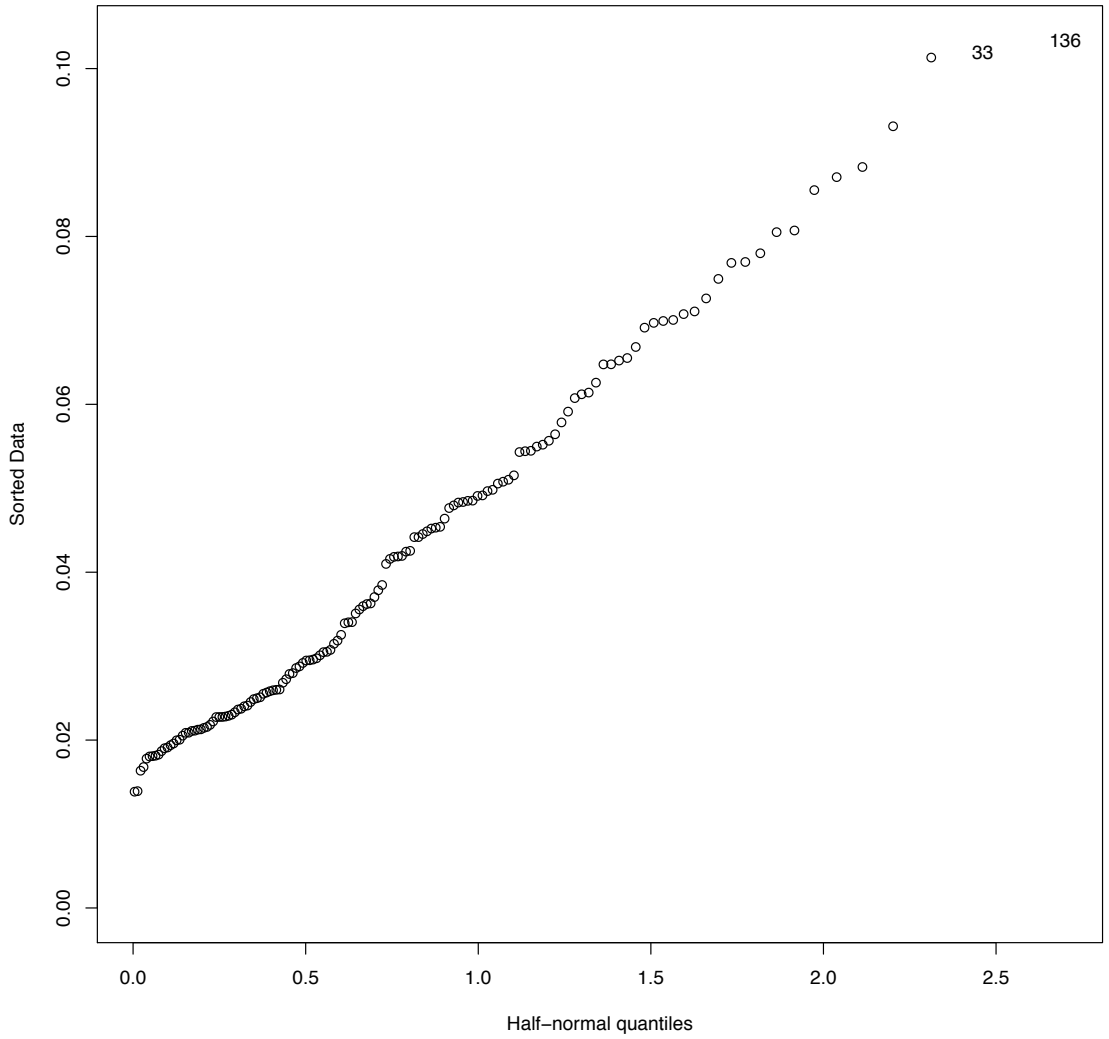
Finally, let’s have a look at the fit using these variables instead of HDI and its square -- What do you notice?

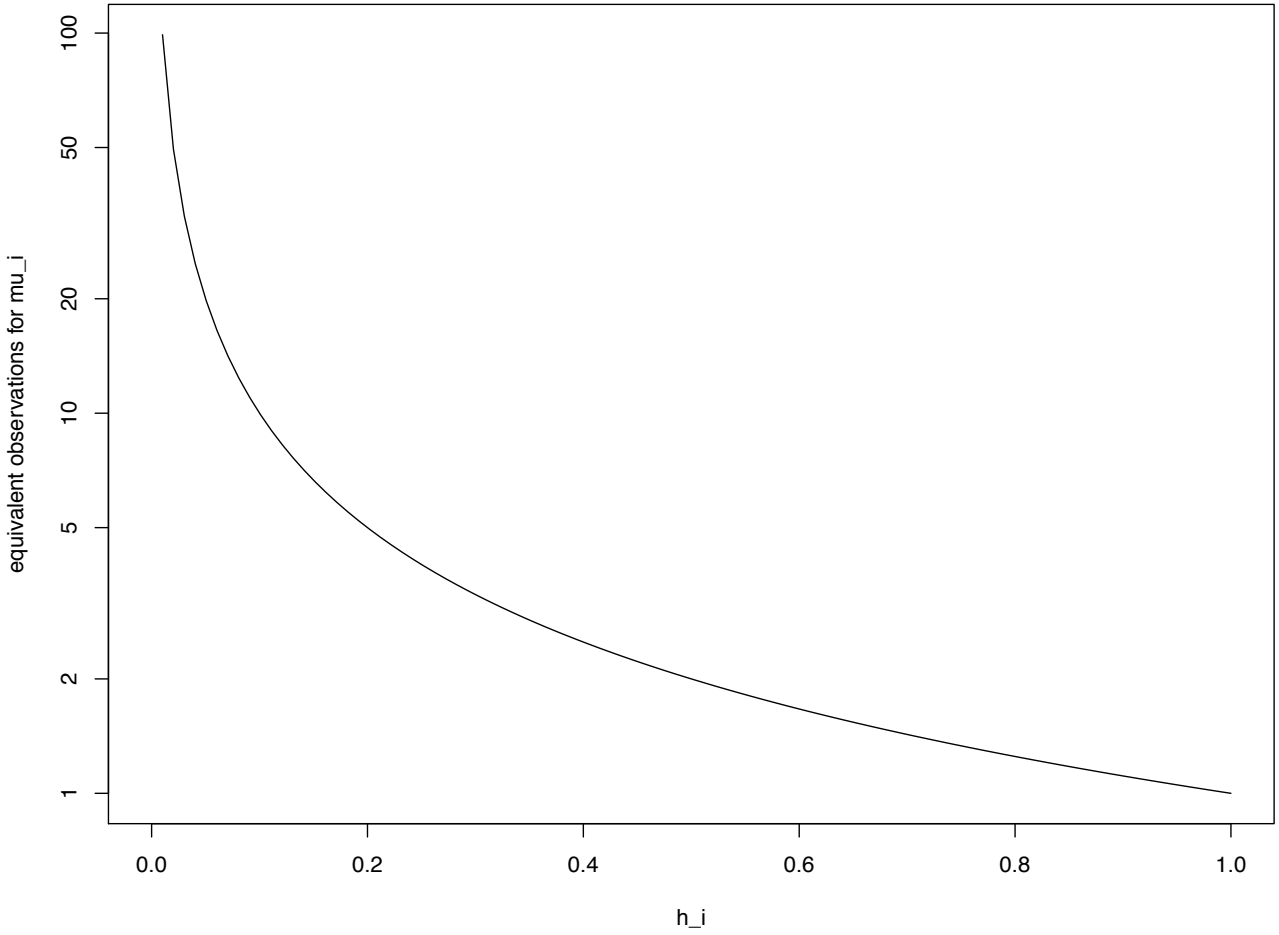
```
# fit without quadratic on hdi for the moment

fit <- lm(ln_death_risk~ln_events+ln_fert+ln_pop+hdi,data=vul)
summary(fit)

# Call:
# lm(formula = ln_death_risk ~ ln_events + ln_fert + ln_pop + hdi,
#    data = vul)
#
# Residuals:
#     Min      1Q  Median      3Q     Max
# -3.4518 -0.7673 -0.1513  0.5669  6.2271
#
# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)  -5.3485     1.5175  -3.524 0.000575 ***
# ln_events     1.3708     0.1792   7.649 3.04e-12 ***
# ln_fert       2.1961     0.4614   4.760 4.81e-06 ***
# ln_pop       -0.5672     0.1026  -5.529 1.54e-07 ***
# hdi           1.9922     1.2628   1.578 0.116928
# ---
# Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 #
# Residual standard error: 1.35 on 139 degrees of freedom
# Multiple R-squared: 0.4221,!Adjusted R-squared: 0.4055
# F-statistic: 25.38 on 4 and 139 DF,  p-value: 8.522e-16
```

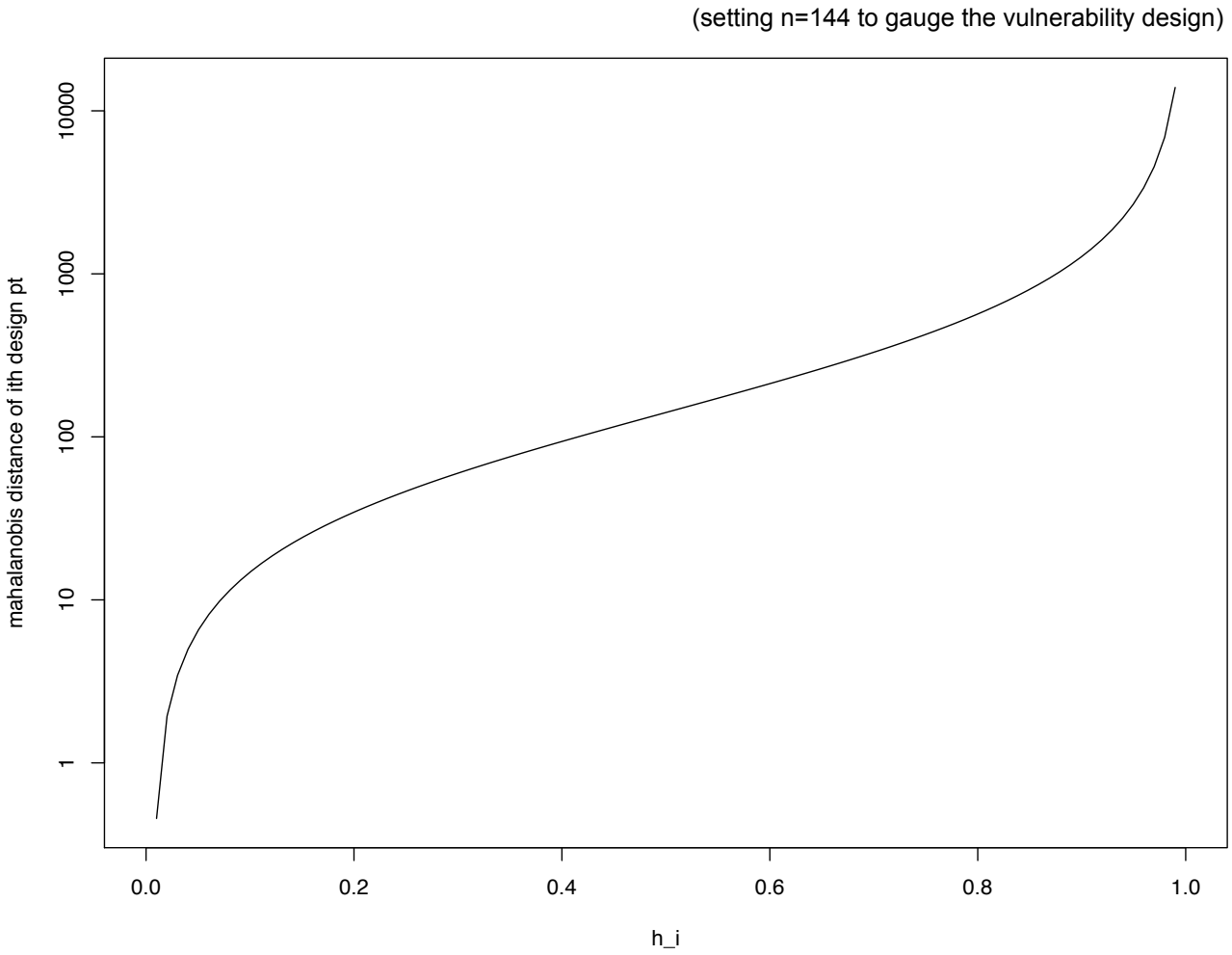
```
> fit <- lm(ln_death_risk~ln_events+ln_fert+ln_pop+poly(hdi,2),data=vul)
> summary(fit)

Call:
lm(formula = ln_death_risk ~ ln_events + ln_fert + ln_pop + poly(hdi,
    2), data = vul)

Residuals:
     Min       1Q   Median       3Q      Max
-3.81655 -0.80298 -0.04575  0.63866  5.60679

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -3.19131    0.72498  -4.402 2.13e-05 ***
ln_events      1.42774    0.16648   8.576 1.79e-14 ***
ln_fert        1.47558    0.45232   3.262  0.00139 **
ln_pop        -0.56450    0.09507  -5.938 2.22e-08 ***
poly(hdi, 2)1  0.65104    2.53050   0.257  0.79735
poly(hdi, 2)2 -6.53905    1.33895  -4.884 2.84e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.251 on 138 degrees of freedom
Multiple R-squared: 0.5073,   Adjusted R-squared: 0.4894
F-statistic: 28.41 on 5 and 138 DF,  p-value: < 2.2e-16
```

```
# fit the full model

fit <- lm(ln_death_risk~ln_events+ln_fert+ln_pop+poly(hdi,2),data=vul)

# extract the hat values

h <- hatvalues(fit)

hist(h)

# load the library of convenience functions from julian faraway

library(faraway)
halfnorm(h)
```


histogram of the hat values
Frequency
0
10
20
30
40
0.02
0.04
0.06
0.08
0.10
hat values


Sorted Data
0.00
0.02
0.04
0.06
0.08
0.10
0.0
0.5
1.0
1.5
2.0
2.5
Half−normal quantiles
33
136


equivalent observations for mu_i
100
50
20
10
5
2
1
0.0
0.2
0.4
0.6
0.8
1.0
h_i


(setting n=144 to gauge the vulnerability design)
mahalanobis distance of ith design pt
10000
1000
100
10
1
0.0
0.2
0.4
0.6
0.8
1.0
h_i

Regression by successive orthogonalization

The stepwise process we followed to successively orthogonalize $1, \text{HDI}, \text{HDI}^2$ can be applied generally to any set of predictors -- The idea would be to sidestep the stability issues with correlated input variables by orthogonalizing them before they are added to the model

This process is known in linear algebra texts as Gram-Schmidt orthogonalization and is used to build an orthogonal basis for a given subspace (say the column space of a matrix)

Orthogonal predictors

Suppose we were given orthogonal predictors $q_1, \ldots, q_p$, we can solve the least squares problem easily -- If we let $Q = [q_1, \ldots, q_p]$ be our new model matrix (Q because it sorta looks like an O which would stand for “orthogonal” -- Seriously) and want to find a vector α to minimize

$$\|y - Q\alpha\|^2$$

Then our solution is (symbolically)

$$\widehat{\alpha} = (Q^t Q)^{-1} Q^t y$$

But because $q_1, \ldots, q_p$ are orthogonal, $(Q^t Q)$ is diagonal with entries $q_j^t q_j = \|q_j\|^2$ which means the components of our solution vector are simply

$$\widehat{\alpha}_j = q_j^t y / q_j^t q_j$$

Gram-Schmidt

Therefore, in a regression context, we can think of the Gram-Schmidt process in the following way

1. Set $z_1 = (1, \ldots, 1)$

2. For j=2,...,p

   Regress m_j on $z_1, \ldots, z_{j-1}$ to produce the coefficients

$$\widehat{\gamma}_{lj} = \frac{z_l^t m_j}{z_l^t z_l} = \frac{\sum_i z_{il} m_{il}}{\sum_i z_{il}^2} \quad \text{for} \quad l = 1, \ldots, j-1$$

   and form the new basis vector

$$z_j = m_j - (\widehat{\gamma}_{1j} z_1 + \cdots + \widehat{\gamma}_{j-1,j} z_{j-1})$$

Unrolling

Rearranging the residuals in step (2) we find that

$$
\begin{aligned}
m_1 &= z_1 \\
m_2 &= z_2 - \widehat{\gamma}_{12} z_1 \\
m_3 &= z_3 - \widehat{\gamma}_{23} z_2 - \widehat{\gamma}_{13} z_1 \\
&\vdots \\
m_p &= z_p - \widehat{\gamma}_{p-1,p} z_{p-1} - \cdots - \widehat{\gamma}_{1p} z_1
\end{aligned}
$$

And so each m_j is a linear combination of the z_l for $l \leq j$; since all the z_l are orthogonal, they form a basis for the column space of M

Aside: Invertible transformations

Again, this process allows us to construct a new basis for the column space of M, one that is orthogonal -- We can then easily solve for the least squares projection $\widehat{\mu}$ of y

Note, however, that we are no longer modeling with the same set of basis functions (predictor variables) we started with, and any interpretation based on the fitted coefficients in our model will change

In general, if we use an invertible p-by-p matrix A to create a new set of predictor variables from our old ones $\mathsf{M}' = \mathsf{MA}$, then the value of α (a vector) that minimizes

$$\|\mathsf{M}'\alpha - \mathsf{y}\|^2 = \|\mathsf{MA}\alpha - \mathsf{y}\|^2$$

is simply $\widehat{\alpha} = \mathsf{A}^{-1}\widehat{\beta}$, a transformation of our previous OLS estimates

Because the column spaces of M and MA are the same, the projection of y into this space, $\widehat{\mu}$, is unchanged by the transformation -- In some sense, then, you might argue that this property makes the conditional means μ more sensible parameters to think about than the $\beta'\mathsf{s}$

An observation

Let $\widehat{\beta}$ be the least squares estimate of β , so that it minimizes $\|y - M\beta\|^2$, and write out our expression for the estimated conditional means

$$\widehat{\mu} = \widehat{\beta}_1 m_1 + \cdots + \widehat{\beta}_p m_p$$

We can then used the unrolled expressions on the previous slide to rewrite $\widehat{\mu}$ in terms of the orthogonal basis functions $q_1, \ldots, q_p$

$$\widehat{\mu} = \widehat{\beta}_1 z_1 + \widehat{\beta}_2 (z_2 - \widehat{\gamma}_{12} z_1) + \cdots + \widehat{\beta}_p (z_p - \widehat{\gamma}_{p-1,p} z_{p-1} - \cdots - \widehat{\gamma}_{1p} z_1)$$

But notice that z_p only appears in $m_p$and with a coefficient 1 -- That means that $\widehat{\beta}_p$ is the same as the regression coefficient for z_p when we regress y on the orthogonal basis $z_1, \ldots, z_p$

Therefore, we've shown that

$$\widehat{\beta}_p = z_p^t y / z_p^t z_p = z_p^t y / \|z_p\|^2$$

An observation

This means that the regression coefficient $\widehat{\beta}_p$ is the result of a univariate regression of the response y on z_p where z_p is the residual after regressing m_p on the previous p-1 predictors $m_1, \ldots, m_{p-1}$

Of course there's nothing special about the ordering of our predictor variables and we could have arranged any of the p predictors to be last...

In general

The jth regression coefficient $\widehat{\beta}_j$ is the univariate regression of y on a vector z where z is the residual after regressing m_j on $m_1, \ldots, m_{j-1}, m_{j+1}, \ldots, m_p$

It's simple to show that since $\widehat{\beta}_j = z^t y / z^t z = z^t y / \|z\|^2$

$$\text{var } \widehat{\beta}_j = \frac{\sigma^2}{\|z\|^2}$$

This means that if a predictor variable is "well explained" by other variables in the model, we will not be able to estimate it stably

Aside: VIF

This also gives us a proof of an expression involving the VIF we saw earlier --
Let z be the residual after regressing m_j on the remaining $m_1, \ldots, m_{j-1}, m_{j+1}, \ldots, m_p$
and let R^2 denote the associated (unadjusted) correlation coefficient

Then

$$1 - R^2 = 1 - \left(1 - \frac{\sum_i (m_{ij} - \overline{m}_j)^2}{\|z\|^2}\right) = \frac{\sum_i (m_{ij} - \overline{m}_j)^2}{\|z\|^2}$$

So that the VIF for the jth variable is just

$$VIF_j = \frac{\|z\|^2}{\sum_i (m_{ij} - \overline{m}_j)^2}$$

Therefore, we can write the variance on the previous slide as

$$\frac{\sigma^2}{\|z\|^2} = \frac{\sigma^2}{\|z\|^2} \frac{\sum_i (m_{ij} - \overline{m}_j)^2}{\sum_i (m_{ij} - \overline{m}_j)^2} = \frac{\sigma^2 \; VIF_j}{\sum_i (m_{ij} - \overline{m}_j)^2}$$

A second observation

While we have presented the Gram-Schmidt algorithm as a process, we have already seen hints of other representations -- In particular, using the unrolled expression from several slides back we can write

$$M = Z\Gamma$$

where Z has the columns $z_1, \ldots, z_p$ and Γ is an upper-triangular matrix with entries $\widehat{\gamma}_{ij}$ -- Introducing a diagonal matrix D where the jth element is $\|z_j\|^2 = z_j^t z_j$ we can rewrite this expression as

$$M = (ZD^{-1})D\Gamma = QR$$

The QR decomposition

The QR decomposition breaks a matrix M into an n-by-p orthogonal matrix Q and a p-by-p upper triangular matrix R -- This decomposition is unique and it represents one convenient orthogonal basis for the column space of M

With this representation, we can show that the least squares estimates are given by the somewhat simpler expressions

$$\widehat{\beta} = \mathsf{R}^{-1}\mathsf{Q}^{\mathsf{t}}\mathsf{y} \quad \text{and} \quad \widehat{\mu} = \mathsf{Q}\mathsf{Q}^{\mathsf{t}}\mathsf{y}$$

Unlike inverting $\mathsf{M}^{\mathsf{t}}\mathsf{M}$ this form of the solution is easy because R is an upper-triangular matrix -- This is how R (now the programming environment) solves its least squares problems

Modified Gram-Schmidt

We told you that solving the normal equations, $\mathrm{M}^{\mathrm{t}}\mathrm{M}\beta = \mathrm{M}^{\mathrm{t}}\mathrm{y}$, is generally viewed as an unwise approach because of numerical difficulties -- Roundoff errors can, for example, accumulate so that your computation could be somewhat off from the “true” or exact arithmetic solution

Using Gram-Schmidt to create the QR decomposition is also viewed as being somewhat unstable -- Instead people often appeal to the so-called modified Gram-Schmidt procedure

Modified Gram-Schmidt

On the previous slides, we constructed an orthogonal basis stepwise, at step j forming an orthogonal basis $z_1, \ldots, z_{j-1}$ and projecting m_j into the orthogonal complement of their span

That is, we projected into the column space of $z_1, \ldots, z_{j-1}$ and looked at the residual from this fit

Alternately, we could project m_j first into the orthogonal complement of the space spanned by z_1 (look at the residual $m_j - (z_1^t m_j / z_1^t z_1) z_1$) and then project this into the orthogonal complement of the space spanned by z_2 and so on

This alteration is at the core of the modified Gram-Schmidt procedure...

Classical and Modified Gram-Schmidt

The change is best illustrated below

Set $z_1 = (1, \ldots, 1)$

For j=2,...,p

Set $z_j = m_j$

For l = 1,...,j-1

$$\begin{cases} \alpha = z_l^t m_j / z_l^t z_l \ (\text{CGS}) \\ \alpha = z_l^t z_j / z_l^t z_l \ (\text{MGS}) \end{cases}$$

$$z_j = z_j - \alpha z_l$$

A diagnostic

This view of the jth multiple regression coefficient $\widehat{\beta}_j$ as a simple regression onto a new predictor variable, a residual removing the effect of the other predictors in the model, provides us with another diagnostic plot that is commonly discussed in the literature

The so-called added variable plot or partial regression plot is designed to examine the contribution of m_j to the regression, given that the remaining predictors have already been included -- It is said to be an improvement over the residual plots we made last time because it isolates the effects of a single variable

As a graphic, the added variable plot relates two sets of residuals -- Let

$$M_{[j]} = [m_1, \ldots, m_{j-1}, m_{j+1}, \ldots, m_p]$$

be the model matrix removing the jth variable and let (symbolically)

$$\widehat{\epsilon}_{[j]} = y - M_{[j]}(M_{[j]}^t M_{[j]})^{-1} M_{[j]}^t y \quad \text{and} \quad \widehat{z}_j = m_j - M_{[j]}(M_{[j]}^t M_{[j]})^{-1} M_{[j]}^t m_j$$

Added variable plots

In a very real sense, the plot exhibits the regression problem “felt” by the coefficient $\widehat{\beta}_j$ -- What properties do you expect this plot to have?

Added variable plots

First, the slope of the regression line of $\widehat{\epsilon}_{[j]}$ on $\widehat{z}_j$ is exactly $\widehat{\beta}_j$ -- We derived this in the last few slides

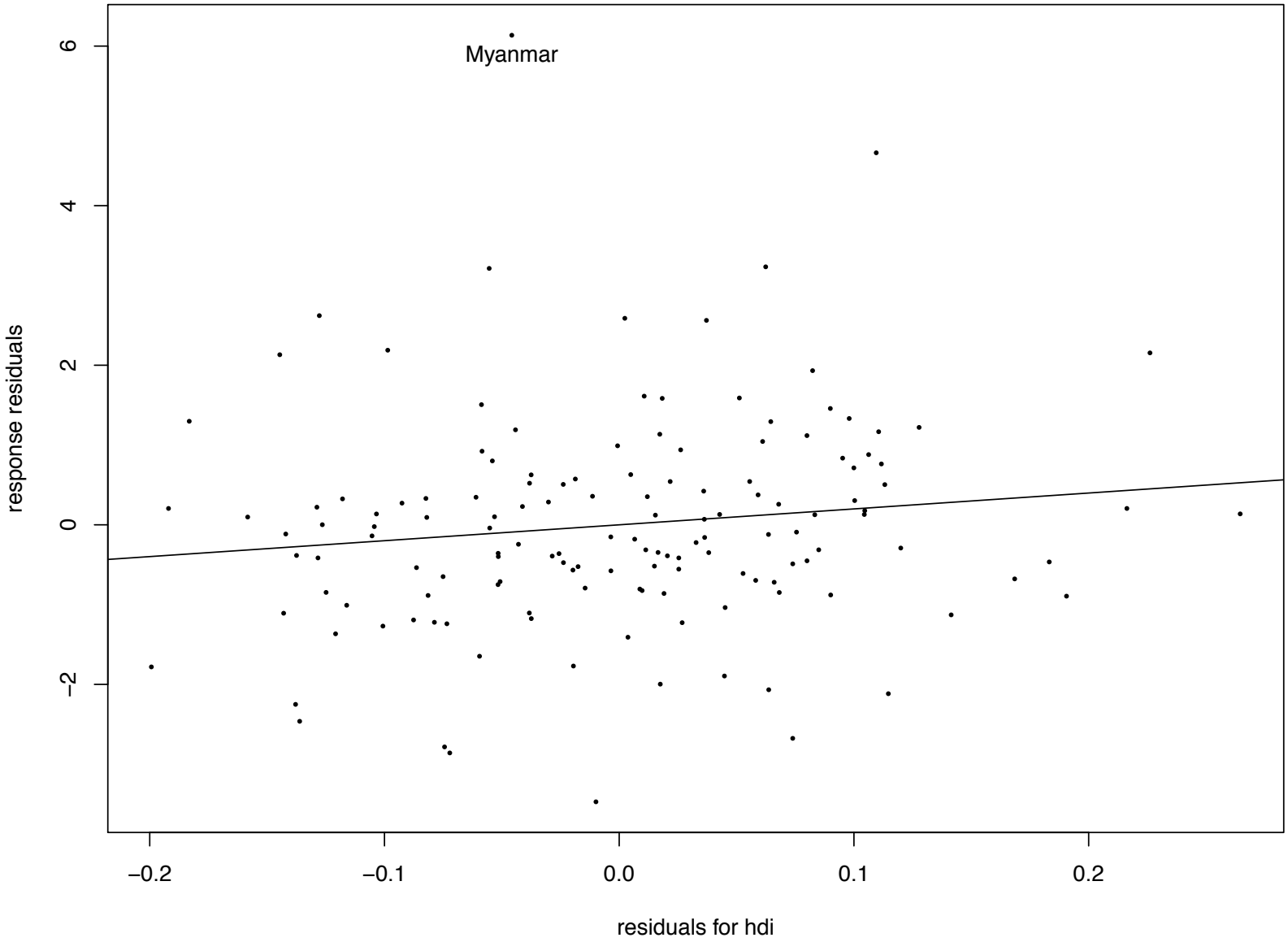
Next, the residuals around the line are exactly the residuals from the full fit -- And in this sense we "see" the actual regression that produces the coefficient estimate $\widehat{\beta}_j$

As a result, people recommend this plot to examine the significance of $\widehat{\beta}_j$ in the full regression, examine the data for extreme our outlying points and (to a lesser extent) examine whether a nonlinear function of m_j might improve the fit

An example

Consider the vulnerability data and the effect of HDI -- Let's start with just a simple linear model in all the predictors and examine the added variable plots for a few of them

```
fit <- lm(ln_death_risk~ln_events+ln_fert+ln_pop+hdi,data=vul)
fit1 <- lm(ln_death_risk~ln_events+ln_fert+ln_pop,data=vul)
fit2 <- lm(hdi~ln_events+ln_fert+ln_pop,data=vul)
fit3 <- lm(residuals(fit1)~residuals(fit2)-1)

plot(residuals(fit2),residuals(fit1),pch=20,cex=0.5)
abline(fit3)

# coefficients from the full model...

coefficients(fit)

# (Intercept)   ln_events     ln_fert      ln_pop         hdi
#  -5.3484985   1.3708219   2.1960509  -0.5672164   1.9921835

# and from the (through the origin) fit on the residuals
# from the two regressions

coefficients(fit3)

# residuals(fit2)
#        1.992184

# identify points that seem large on the plot...

text(residuals(fit2)[residuals(fit1)>5],
     residuals(fit1)[residuals(fit1)>5],
     vul$country[residuals(fit1)>5],pos=1)
```


Myanmar
response residuals
6
4
2
0
−2
−0.2
−0.1
0.0
0.1
0.2
residuals for hdi

Added variable plots

Suppose we have fit a model and want to include a new variable -- Part of the motivation for the added variable plot construction is the fact that simple residual plots can portray an incorrect image of the dependence of a new variable not yet in the model

If the response is related linearly, say, to a new variable, then plotting the residuals from the current fit against this new variable need not be linear -- It is only guaranteed to be so if the new variable is orthogonal to those already in the model

How could you demonstrate that?

Added variable plots

Added variable plots, on the other hand, correct this situation and if the response depends linearly on the new variable, we see a linear relationship in the plot

However, you can also show that just because this plot is linear, it’s not necessarily the case that the underlying relationship is linear -- Correlations with the column space of the variables already in the model are the problem

Another diagnostic

The partial residual plot is an alternative to the standard residual plot we used the other day and is often used to audition new variables not yet in a model -- Let $\widehat{\beta}$ be the least squares fit to our response y using all the variables (both old and the new candidate)

Let $\widehat{\epsilon}$ denote the residuals from this full fit and plot

$$\widehat{\epsilon} + m_j \widehat{\beta}_j \quad \text{against} \quad m_j$$

Where does this come from?

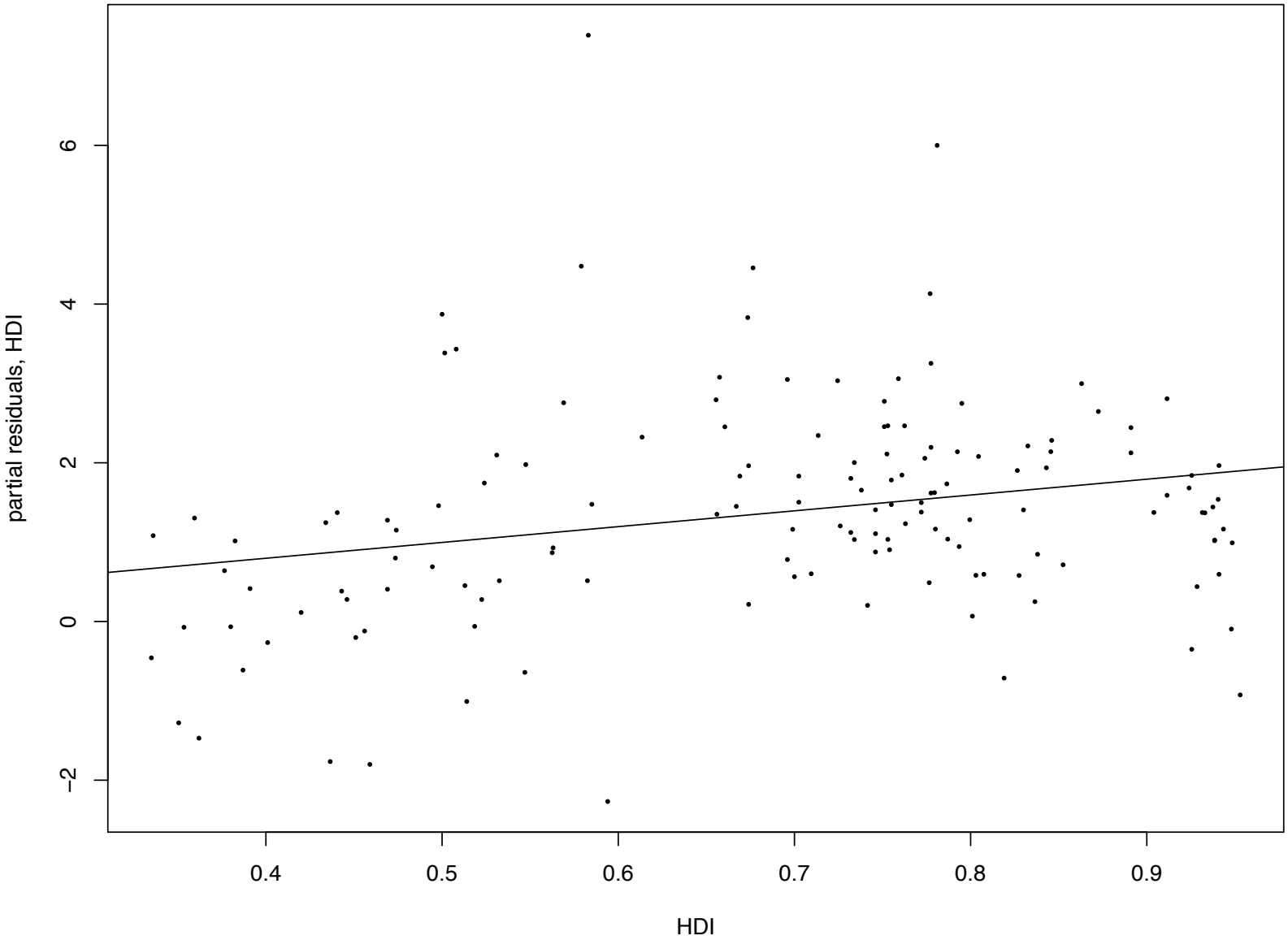
Partial residual plots

To isolate the effect of a single variable m_j, we might consider looking at the response with the effects of the other inputs removed

$$y - \sum_{l \neq j} m_l \widehat{\beta}_l = \widehat{\mu} + \widehat{\epsilon} - \sum_{l \neq j} m_l \widehat{\beta}_l = m_j \widehat{\beta}_j + \widehat{\epsilon}$$

What properties does this plot have?

```
# fit the full model

fit <- lm(ln_death_risk~ln_events+ln_fert+ln_pop+hdi,data=vul)

# extract our model matrix (MMMMMMM)

M <- model.matrix(fit)

# it's really a matrix and its column names are just the variable names in the model

colnames(M)

# [1] "(Intercept)" "ln_events"   "ln_fert"     "ln_pop"      "hdi"

# extract just the HDI column

m <- M[,"hdi"]

# and make the partial residual plot...

plot(m, coefficients(fit)["hdi"]* m + residuals(fit),
            xlab = "HDI",ylab="partial residuals, HDI")

abline(0, coefficients(fit)["hdi"])
```


partial residuals, HDI
6
4
2
0
−2
0.4
0.5
0.6
0.7
0.8
0.9
HDI

Comparisons

In general, ordinary residual plots, added variable plots and partial residual plots are all tools we might consult to examine deficiencies in a model or to audition model elaborations

It is believed that (and our single example supports it) that the partial residual plot is better at spotting nonlinearities (like the need for a quadratic term, say), while the added variable plot is better at understanding outliers and extreme points that may influence the fit

Back to SMW!

We will now use some of what we've developed (the fun with the Sherman-Morrison-Woodbury Formula and the eigen-decomposition of a variance-covariance matrix) to examine more diagnostics

We'll look at so-called standardized as well as Studentized residuals and then finish with leave-one-out statistics...

Standardized residuals

First, let's go back to adding points to a regression from the start of the lecture -- A similar kind of analysis (reasoning from an updated hat matrix) will let us examine the residuals in a new light as well

First recall that a common diagnostic strategy involves plotting the ordinary residuals against the estimated conditional means (the so-called fitted values) and the different variables in the model

Assume we are given n observations $(x_{i1}, \ldots, x_{ip})^t$, y_i for i=1,...,n so that the conditional means are given by $\widehat{\mu} = H\,y$, the residuals by $\widehat{\epsilon} = (I - H)\,y$ and the variance-covariance matrix of the residuals is simply $\sigma^2(I - H)$

Therefore, rather than plot the simple residuals, we might instead work with the so-called standardized residuals (Anscombe and Tukey, 1963)

$$\frac{\widehat{\epsilon}_i}{s\sqrt{1 - h_i}}$$

where s is the usual estimate of the residual standard deviation

$$s^2 = \frac{1}{n - p}\sum_i \widehat{\epsilon}_i^2 = \frac{RSS}{n - p}$$

Standardized v. Studentized residuals

The justification for this procedure is that the standardizing puts the residuals on “equal footing” -- Those points with high leverage (or in Huber’s interpretation, involve just a few observations) will have greater variability in their conditional mean estimates $\widehat{\mu}_i$ and less assigned to the associated residual $y_i - \widehat{\mu}_i$

For diagnostic purposes, however, these standardized residuals are not exactly what we’re after -- Instead we would like to consider the size of the residual corresponding to y_i when the ith data point has been omitted from the fit

If we let $\widehat{\mu}_{(i)}$ denote the conditional mean computed for the ith input point $(x_{i1}, \ldots, x_{ip})^t$ estimated using all but the ith observation, then we’d like to compute $y_i - \widehat{\mu}_{(i)}$ -- Let’s see how we’d do this!

Updated residuals: Adding rather than deleting points

First, recall our expression for the updated hat matrix when we add any new row m^t to an orthogonal model matrix M

$$H^+ = \left[\begin{array}{c|c} MM^t - \frac{(Mm)(Mm)^t}{1+m^tm} & \frac{Mm}{1+m^tm} \\ \hline \frac{(Mm)^t}{1+m^tm} & \frac{m^tm}{1+m^tm} \end{array}\right]$$

Suppose, then we have $m = (x_{n+1,1}, \ldots, x_{n+1,p})^t$ and y_{n+1}, a new pair of observations -- Let $\widehat{\mu}^+$ denote our estimate of the conditional means using all (n+1) data points (compared to $\widehat{\mu}$ which uses just the first n)

Examining the last row of the matrix above, we find that

$$\widehat{\mu}^+_{n+1} = (1 - h_{n+1})\,\widehat{\mu}_{n+1} + h_{n+1}\,y_{n+1}$$

where $\widehat{\mu}_{n+1}$ is the estimate at m using just the first n data points and we've let h_{n+1} be the (n+1)st diagonal element of the hat matrix H^+

In short, we see that the updated estimate (adding a point) is **a convex combination of the original estimate (just n points) and the new response** y_{n+1} with weights that are just the last diagonal entry in the updated hat matrix

Updated residuals: Adding rather than deleting points

Flipping this around, we can come up with an expression for the residuals for this updated fit (using all n+1 data points) as a function of the preliminary fit (using just n values)

$$y_{n+1} - \widehat{\mu}^{+}_{n+1} = (1 - h_{n+1})\,(y_{n+1} - \widehat{\mu}_{n+1}) \qquad (a)$$

This relation is important in that it connects the “ordinary” residual using all (n+1) points $y_{n+1} - \widehat{\mu}^{+}_{n+1}$ with the residual $y_{n+1} - \widehat{\mu}_{n+1}$ ignoring the case m, y_{n+1}

Of course relation (a) holds for arbitrary indices and not just (n+1) and it holds even for non-orthogonal model matrices M

Updated fits

From the expression on a previous slide (assuming we add a row m to an orthogonal model matrix M), we have that

$$\left[(M^+)^t M^+\right]^{-1} = I_{p\times p} - \frac{mm^t}{1+m^t m}$$

Brute force calculations allow us to relate the estimated coefficient vector $\widehat{\beta}^+$ for the (n+1) points as a function of $\widehat{\beta}$, the estimate involving just the original n

$$\widehat{\beta}^+ - \widehat{\beta} = -m\left(\frac{y_{n+1} - \widehat{\mu}^+_{n+1}}{1-h_{n+1}}\right)$$

or for general M (non-orthogonal)

$$\widehat{\beta}^+ - \widehat{\beta} = -\left[(M^+)^t M^+\right]^{-1} m\left(\frac{y_{n+1} - \widehat{\mu}^+_{n+1}}{1-h_{n+1}}\right)$$

Updated residuals: Back to deleting points

Now, to turn this result into the deletion of a point, suppose we again have just n observations $(x_{i1}, \ldots, x_{ip})^t$, y_i for i=1,...,n

Then, let $\widehat{\mu}_i$ denote the conditional mean at the ith input point estimated using all n values, and let $\widehat{\mu}_{(i)}$ denote the conditional mean at the ith input point estimated using all but the ith pair

Then, using the result from the previous slide, we have that

$$y_i - \widehat{\mu}_{(i)} = \frac{y_i - \widehat{\mu}_i}{1 - h_i} = \frac{\widehat{\epsilon}_i}{1 - h_i}$$

Again this means that we can assess the “leave one out” residuals with no further computation

Studentized residuals

In terms of notation, let m_i^t be the ith row if the n-by-p model matrix M and let $M_{(i)}$ be the (n-1)-by-p model matrix obtained by leaving this row out -- Further, $\widehat{\beta}$ will be the coefficient vector estimated from the full data $\widehat{\beta}_{(i)}$ and the estimate obtained by leaving out the ith point

Now, by direct calculation, the variance of the residual $y_i - \widehat{\mu}_{(i)} = y_i - m_i^t \widehat{\beta}_{(i)}$ is

$$\sigma^2 \left[1 + m_i^t (M_{(i)}^t M_{(i)})^{-1} m_i\right]$$

which we can estimate via

$$s_{(i)}^2 \left[1 + m_i^t (M_{(i)}^t M_{(i)})^{-1} m_i\right]$$

where $s_{(i)}$ is the residual standard deviation leaving out the ith row

Studentized residuals

With this notation in place, we can define the Studentized residual

$$\widehat{\epsilon}_i^* = \frac{y_i - \widehat{\mu}_{(i)}}{s_{(i)}\sqrt{1 + m_i^t (M_{(i)}^t M_{(i)})^{-1} m_i}}$$

Why this terminology?

Studentized residuals

Under the normal linear model, the numerator and denominator in this residual are independent so that $\widehat{\epsilon}_i^*$ has a t-distribution with n-p-1 degrees of freedom -- We can use this fact to assess “significance” of the Studentized residuals, although $\widehat{\epsilon}_i^*$ and $\widehat{\epsilon}_j^*$ will not be independent

Using our results about the updated residuals from deleting an observation from a regression fit, we can show that

$$\widehat{\epsilon}_i^* = \frac{\widehat{\epsilon}_i}{s_{(i)}\sqrt{1-h_i}}$$

and we can compute

$$(n-p-1)s^2_{(i)} = (n-p)s^2 - \frac{\widehat{\epsilon}_i^2}{1-h_i}$$

Deleting points and the coefficients

Flipping around the relation from several slides back (framed initially for adding points) we also see the effect when deleting the ith observation from a regression -- That is

$$\widehat{\beta} - \widehat{\beta}_{(i)} = (M^t M)^{-1} m_i \frac{\widehat{\epsilon}_i}{1 - h_i}$$

More diagnostics

There are a host of diagnostics that are built from these leave-one-out statistics -- One of the most famous, Cook's distance for a point i, can be motivated as a standardized (in the Mahalanobis sense) distance between $\widehat{\beta}_{(\mathrm{i})}$ and $\widehat{\beta}$

It can also be derived as sum of squared deviations $\widehat{\mu}_{(\mathrm{i})} - \widehat{\mu}_{\mathrm{i}}$, scaled to have a direct interpretation -- This is your homework!

Inference

So far we've shied away from formal probabilistic arguments (aside from, say, assessing the independence of various outputs of a linear model) -- We have, however, been blindly using the hypothesis tests behind the summary tables of R's lm objects

Given our work with variance-covariance matrices today, we're in a good place to assess the sampling distribution of the residual sum of squares -- This will be an important component in our t- and F-distribution work later

Idempotent, again

We've seen today that a square, symmetric matrix A can be written as

$$A = ODO^t$$

where O is an orthonormal matrix and D is a diagonal matrix of nonnegative values -- If A is also idempotent, we have that

$$ODO^t = A = A^2 = ODO^tODO = OD^2O^t$$

Since O is invertible, we find that $D^2 = D$ -- Therefore, the eigen-values of a symmetric and idempotent matrix must be either 0 or 1

And since any symmetric and idempotent matrix is also a projection (!) we know that the eigen-values of a projection matrix must also be 0 or 1

Idempotent, again

If a p-by-p symmetric and idempotent matrix A has k eigen-values that are 1 and p-k that are 0, we say that A has rank k

Let $O = [o_1|o_2|\cdots o_p]$ and rewrite the eigen-decomposition on the previous slide for a rank k, p-by-p as

$$A = ODO^t = \sum_{j=1}^{p} d_j o_j o_j^t = \sum_{j=1}^{k} o_j o_j^t$$

where we have arranged for the first k eigen-values to be all 1's -- In turn, we can describe the elements in the column space of A as

$$A\beta = \sum_{j=1}^{k} o_j (o_j^t \beta)$$

for p-vectors β , so that our two notions of rank agree

Idempotent, again

Finally, we recall a theorem from probability -- If A is any (nonrandom) symmetric and idempotent matrix and W has a p-dimensional multivariate standard normal distribution, Z^tAZ has a chi-square distribution with degrees of freedom equal to the rank of A

To see this, let $A = ODO^t$ and reason as we did with the Mahalanobis distance -- That is, let $W = O^tZ$ so that W again has a standard normal distribution so that

$$Z^tAZ = W^tDW = \sum_{j=1}^{p} d_j w_j^2 = \sum_{j=1}^{k} w_j^2$$

where we have assumed the rank of A is k and that the k non-zero eigen-values are arranged first in the decomposition -- The result follows

Idempotent, again

To apply this result, under the assumptions of the normal linear model, we can write our residuals as

$$\widehat{\epsilon} = (\mathsf{I} - \mathsf{H})\mathsf{y} = (\mathsf{i} - \mathsf{H})(\mathsf{M}\beta + \epsilon) = (\mathsf{I} - \mathsf{H})\epsilon$$

The residual sum of squares is then

$$(\mathsf{y} - \mathsf{M}\widehat{\beta})^{\mathsf{t}}(\mathsf{y} - \mathsf{M}\widehat{\beta}) = \widehat{\epsilon}^{\mathsf{t}}\widehat{\epsilon} = \epsilon^{\mathsf{t}}(\mathsf{I} - \mathsf{H})\epsilon$$

again because H and hence I-H are idempotent

Idempotent, again

Now, since the eigenvalues of I are all 1, the eigenvalues of I-H are 1 minus the eigenvalues of H (which are either 1 or 0) -- Therefore, if H has full rank p, I-H has rank n-p

Lastly, the errors ϵ were assumed to have mean 0 and common variance σ^2 meaning ϵ_i/σ are independent standard normals

Using the expression on the previous slide and the theorem 1 slide back, we find that $\epsilon^t(I - H)\epsilon/\sigma^2$ must have a chi-square distribution with n-p degrees of freedom

Combine this with your homework result and you have derived the t-distributional result used in your table fun in R

Next time

We will start on variable selection and penalized least squares (two topics initially, but merge into one with more “modern” or post-70s approaches)

We’ll start with ridge regression and compare it to schemes for adding and deleting variables (stepwise procedures, regression by leaps and bounds and so on)

We’ll also derive some simple model selection “rules” to assess the amount of penalization or the size of the final model for variable selection