Lecture 5: Shrinky dink

From last time...

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 Z 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

And now today’s material...

Prelude: The singular value decomposition

Let A be an n-by-p matrix -- Then there exists an n-by-p matrix U and a p-by-p matrix V, each with orthonormal columns, such that

$$A = U\,D\,V^t$$

where D is a p-by-p diagonal matrix with elements $s_1 \geq s_2 \geq \cdots \geq s_p \geq 0$ referred to as the singular values of A

Prelude: The singular value decomposition

Those of you in 202b will see an extended treatment of the SVD, a proof of its existence and some of its basic properties

From the perspective of our class, there are some facts about A that you can read directly from the SVD -- For example, the column space of U spans the column space of A and the number of non-zero singular values equals the number of linearly independent columns of A (also known as the rank of A)

We can also use it to simplify our solution of least squares problems (an alternative to, say, the QR decomposition) -- In particular, if for an n-by-p model matrix M with full rank we write $\mathsf{M} = \mathsf{UDV}^t$ then

$$
\begin{aligned}
\widehat{\mu} &= \mathsf{M}(\mathsf{M}^t\mathsf{M})^{-1}\mathsf{M}^t\mathsf{y} \\
&= \mathsf{UDV}^t(\mathsf{VDU}^t\mathsf{UDV}^t)^{-1}\mathsf{VDU}^t\mathsf{y} \\
& \quad \text{(grinding noise of lots of things cancelling)} \\
&= \mathsf{UU}^t\mathsf{y}
\end{aligned}
$$

Again, since U has orthonormal columns, this is equivalent to an ordinary regression using the columns of U as our predictor variables (more on U later)

Prelude: The singular value decomposition

Today we will make use of this decomposition as an interpretive device for a new kind of regression procedure...

Notation, again

As usual, we are going to assume we have n observations (inputs and responses) denoted $(x_{i1}, \ldots, x_{ip}),\ y_i,$ for $i = 1, \ldots, n$, but today we will not include the intercept among the p inputs

We then consider the usual ordinary least squares criterion

$$\sum_{i=1}^{n} (\, y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip}\,)^2$$

Rewriting things a little we find

$$\begin{aligned} &\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} \\ &\quad = \underbrace{(\beta_0 + \beta_1 \bar{x}_1 + \cdots \beta_p \bar{x}_p)}_{\text{constant}} + \underbrace{\beta_1 (x_{i1} - \bar{x}_1) + \cdots + \beta_p (x_{ip} - \bar{x}_p)}_{\text{centered predictors}} \end{aligned}$$

where $\bar{x}_j = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} x_{ij}, j = 1, \ldots, p$

Notation, again

For today's lecture, we are going to work with centered predictors (the residuals after regressing out the constant function from each)

$$\widetilde{x}_{ij} = x_{ij} - \overline{x}_j \quad \text{where} \quad \overline{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$$

It's not hard to show that the least squares solution to

$$\sum_{i=1}^{n} (\, y_i - \gamma_0 - \beta_1 \widetilde{x}_{i1} - \cdots - \beta_p \widetilde{x}_{ip} \,)^2$$

has $\widehat{\gamma}_0 = \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$

Notation again

Therefore, today we are going to work with a slightly altered model where **both the inputs and the responses are centered**

$$\widetilde{x}_{ij} = x_{ij} - \overline{x}_j \quad \text{and} \quad \widetilde{y}_i = y_i - \overline{y}$$

and we use ordinary least squares to find $\widehat{\beta} = (\widehat{\beta}_1, \ldots, \widehat{\beta}_p)^t$ that minimizes

$$\sum_{i=1}^{n} (\, \widetilde{y}_i - \beta_1 \widetilde{x}_{i1} - \cdots - \beta_p \widetilde{x}_{ip}\,)^2$$

In the end, we have $\widehat{\mu}_i = \overline{y} + \widehat{\beta}_1 \widetilde{x}_{i1} + \cdots + \widehat{\beta}_{ip} \widetilde{x}_{ip}$, where the estimated coefficients $\widehat{\beta}_1, \ldots, \widehat{\beta}_p$ and conditional means **are identical to those we would have obtained by solving the original least squares problem** using an intercept and the uncentered inputs/response

Notation again

In sympathy with our last lecture, we will again let $\widetilde{\mathsf{M}}$ denote the model matrix associated with our centered predictors

$$\widetilde{\mathsf{M}} = \left[\, \mathsf{x}_1 - \overline{\mathsf{x}}_1 \,|\, \mathsf{x}_2 - \overline{\mathsf{x}}_2 \,|\, \cdots \,|\, \mathsf{x}_\mathsf{p} - \overline{\mathsf{x}}_\mathsf{p} \,\right]$$

where $\mathsf{x}_\mathsf{j} = (\mathsf{x}_{1\mathsf{j}}, \cdots, \mathsf{x}_{\mathsf{nj}})^\mathsf{t}$ is an n-dimensional column vector representing the jth input variable, j=1,...,p

Our least squares estimates are then $\widehat{\beta} = (\widetilde{\mathsf{M}}^\mathsf{t}\widetilde{\mathsf{M}})^{-1}\widetilde{\mathsf{M}}^\mathsf{t}\widetilde{\mathsf{y}}$ where under the normal linear model we know that $\widehat{\beta}$ has a multivariate normal distribution with mean β and variance-covariance matrix $\sigma^2(\widetilde{\mathsf{M}}^\mathsf{t}\widetilde{\mathsf{M}})^{-1}$

Motivation: Examining expected errors

Using the eigen-decomposition introduced last time, we can write $\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}} = \mathsf{O}\Lambda\mathsf{O}^{\mathsf{t}}$ where the columns of O are orthonormal and Λ is a diagonal matrix with positive elements $\mathsf{d}_{\mathsf{max}} = \mathsf{d}_1 \geq \mathsf{d}_2 \cdots \geq \mathsf{d}_{\mathsf{p}} = \mathsf{d}_{\mathsf{min}}$

Now, consider the length of the error vector $\widehat{\beta} - \beta$

$$\|\widehat{\beta} - \beta\|^2 = (\widehat{\beta} - \beta)^{\mathsf{t}}(\widehat{\beta} - \beta)$$

Using our distributional result above, we know that

$$\mathsf{E}\left[(\widehat{\beta} - \beta)^{\mathsf{t}}(\widehat{\beta} - \beta)\right] = \sigma^2 \ \mathsf{trace}\ (\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}})^{-1}$$

Motivation: Examining expected errors

Finally, we can use the fact that the trace of square matrix is equal to the sum of its eigenvalues, we find that

$$\mathsf{E}\|\widehat{\beta} - \beta\|^2 = \sigma^2 \sum_{\mathsf{i}=1}^{\mathsf{p}} \frac{1}{\mathsf{d_i}}$$

since $(\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}})^{-1}$ has eigen-decomposition $\mathsf{O}\Lambda^{-1}\mathsf{O}^{\mathsf{t}}$

Motivation: Examining expected errors

From here we conclude the following

1. If our our model matrix has orthonormal columns, then $\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}}$ is the p-by-p identity matrix with a trivial eigen-decomposition and unit eigenvalues -- Therefore the error vector $\widehat{\beta} - \beta$ has expected length

$$\sigma^2 \text{ trace } (\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}})^{-1} = \mathsf{p}\sigma^2$$

2. If one or more columns of our model matrix are highly correlated (the set of columns of M are nearly linearly dependent), then we will have one or more small eigenvalues which will result in a large expected error

$$\mathsf{E}\|\widehat{\beta} - \beta\|^2 = \sigma^2 \text{ trace } (\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}})^{-1} \geq \sigma^2/\mathsf{d}_{\mathsf{p}}$$

recalling our assumption that the eigenvalues are sorted and $\mathsf{d}_{\mathsf{p}} = \mathsf{d}_{\mathsf{min}}$ is the smallest

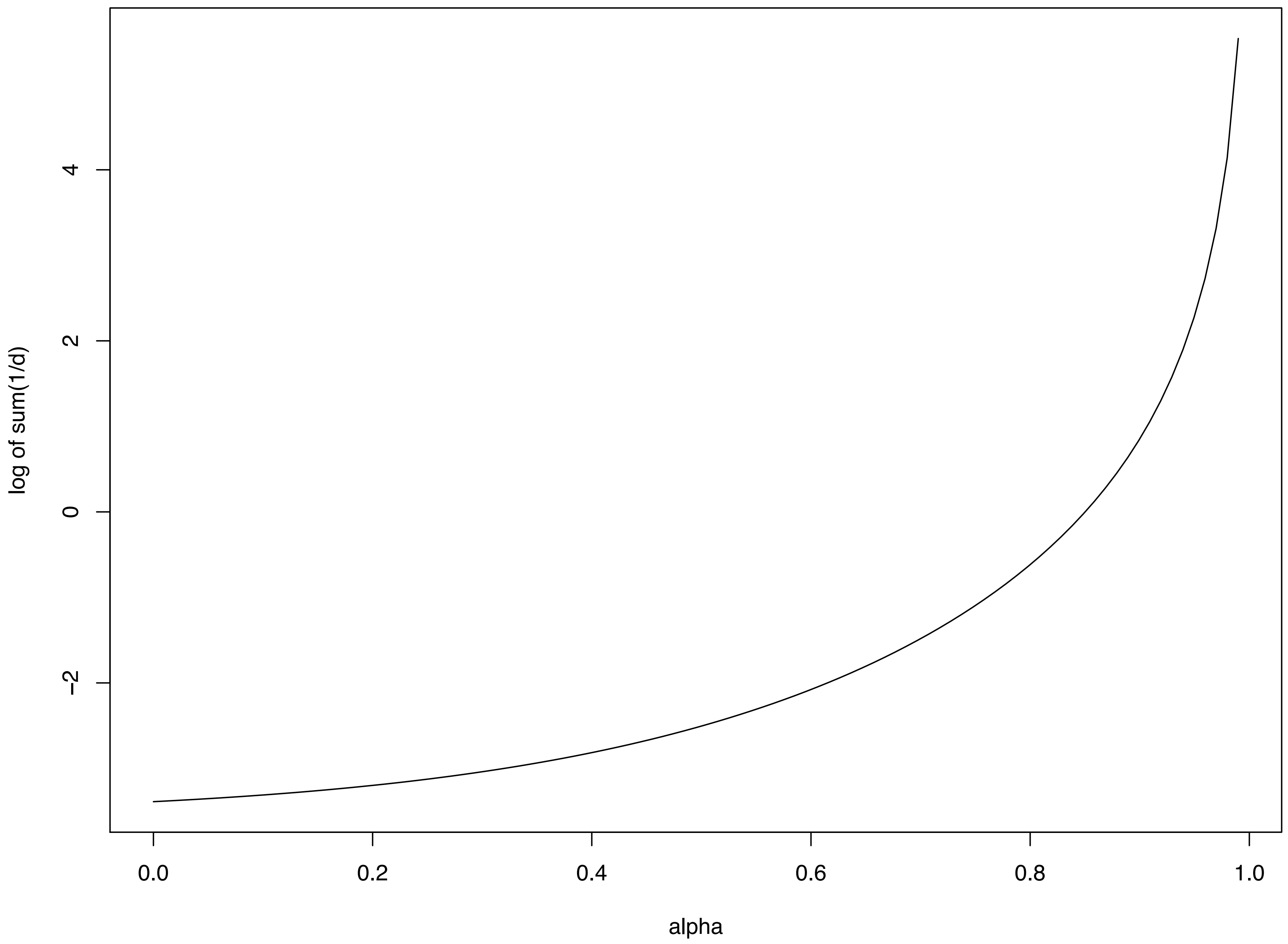
Aside: The singular value decomposition, again

If we let $\widetilde{M} = UDV^t$ then $\widetilde{M}^t\widetilde{M}$ is just

$$\widetilde{M}^t\widetilde{M} = VDU^t\ UDV^t = VD^2V^t$$

meaning that V is the matrix of eigenvectors associated with $\widetilde{M}^t\widetilde{M}$ and the squared singular values of $\widetilde{M}$ are the eigenvalues of $\widetilde{M}^t\widetilde{M}$

```
alpha <- seq(0,1,len=100)

# drop 1
alpha <- alpha[-100]

# simulate! create a series of model matrices that have columns
#       x1, x2, alpha*x2 + (1-alpha)*x3
# so that when alpha is small, we have essentially 3 (linearly)
# independent columns and when alpha is large we have just two

x1 <- rnorm(100)
x2 <- rnorm(100)
x3 <- rnorm(100)

# in the loop below we literally form MtM -- next lecture we will
# show how to do this using the svd and just M

out <- rep(NA,length(alpha))

for(i in 1:length(alpha)){

  M <- cbind(x1,x2,alpha[i]*x2+(1-alpha[i])*x3)
  out[i] <- sum(1/svd(M)$d^2)

}

plot(alpha,log(out),type="l",xlab="alpha",ylab="log of sum(1/d)")
```


log of sum(1/d)
4
2
0
−2
0.0
0.2
0.4
0.6
0.8
1.0
alpha

Motivation: Examining expected errors

With this last expression, we find that it’s possible, even when we are fitting with a correct model, to incur large errors in expectation if our model matrix has one or more small eigenvalues

But what does it mean for $\widetilde{M}^t\widetilde{M}$ this to happen? To understand this we can appeal to the framework we developed last lecture...

Motivation: Principal components

The eigenvectors of $\widetilde{M}^t\widetilde{M}/(n-1)$ are used to define a new set of variables from $\widetilde{x}_1,\dots,\widetilde{x}_p$ that have the same span, the new variables $z_1,\dots,z_p$ being referred to as principal components -- They are an orthogonal set and are usually defined in terms of a variance criterion

Let $a=(a_1,\dots,a_p)^t$ be any p-dimensional vector and consider the n scalar values $z=a_1\widetilde{x}_{i1}+\cdots+a_p\widetilde{x}_{ip},\ i=1,\dots,n$ -- Or, put another way $z=\widetilde{M}\,a$

It's not hard to show that the elements of z have mean 0 and sample variance

$$\text{var } z = a^t\frac{\widetilde{M}^t\widetilde{M}}{n-1}a$$

Motivation: Principal components

The usual definition of principal components is given more like an algorithm than a matrix decomposition

1. For any p-dimensional vector a_1 with $\|a_1\| = 1$, the sample variance of the elements of the n-vector z_1 where $z_{i1} = a_{11}\widetilde{x}_{i1} + \cdots + a_{p1}\widetilde{x}_{ip}$ takes a maximum d_1 when $a_1 = o_1$

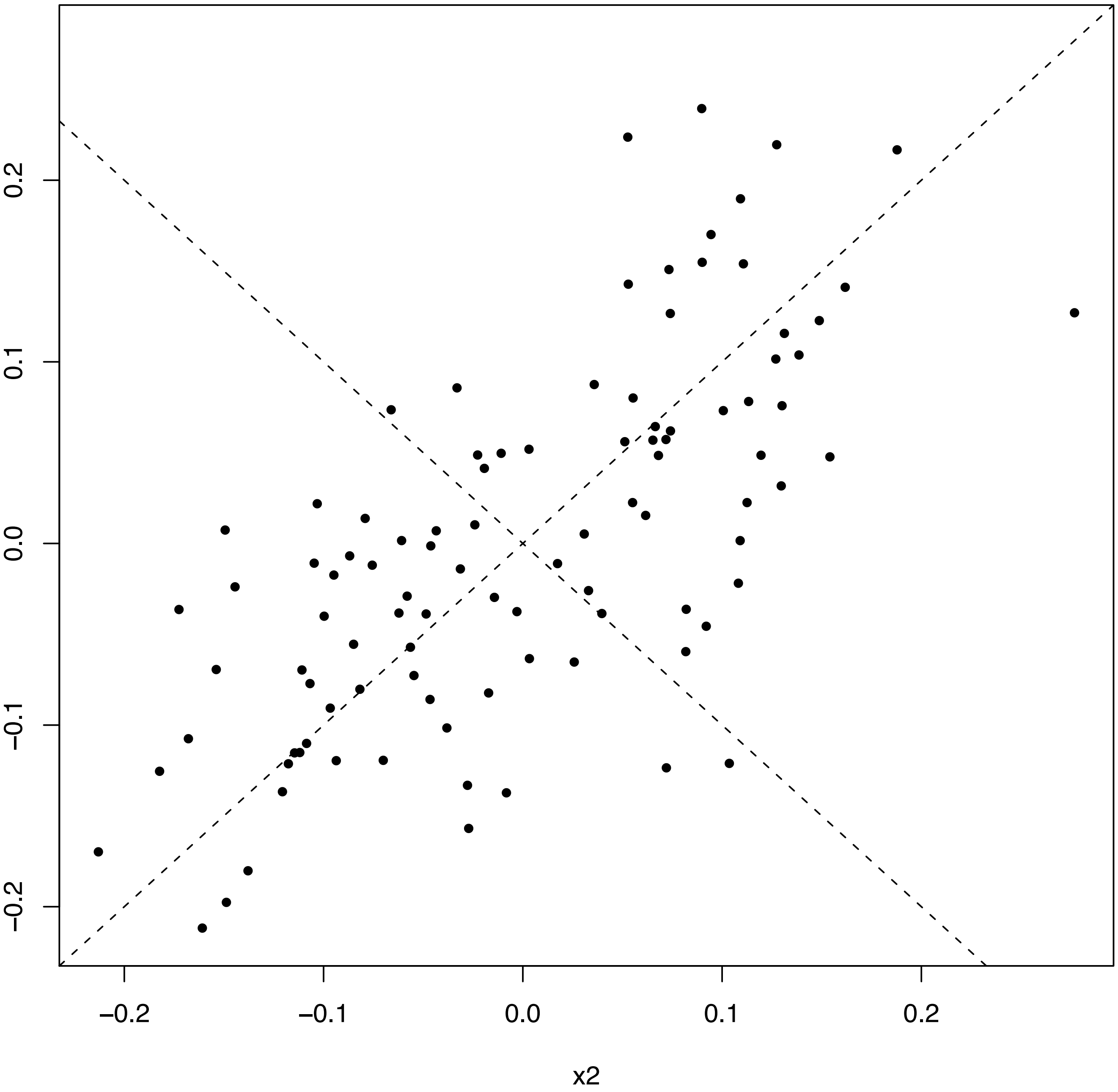
2. For any vector a_j such that $\|a_j\| = 1$ and $a_j^t o_k = 0, k = 1, \ldots, j-1$, the sample variance of the elements in the n-vector z_j where $z_{ij} = a_{1j}\widetilde{x}_{i1} + \cdots + a_{pj}\widetilde{x}_{ip}$ takes its maximum d_j when $a_j = o_j$

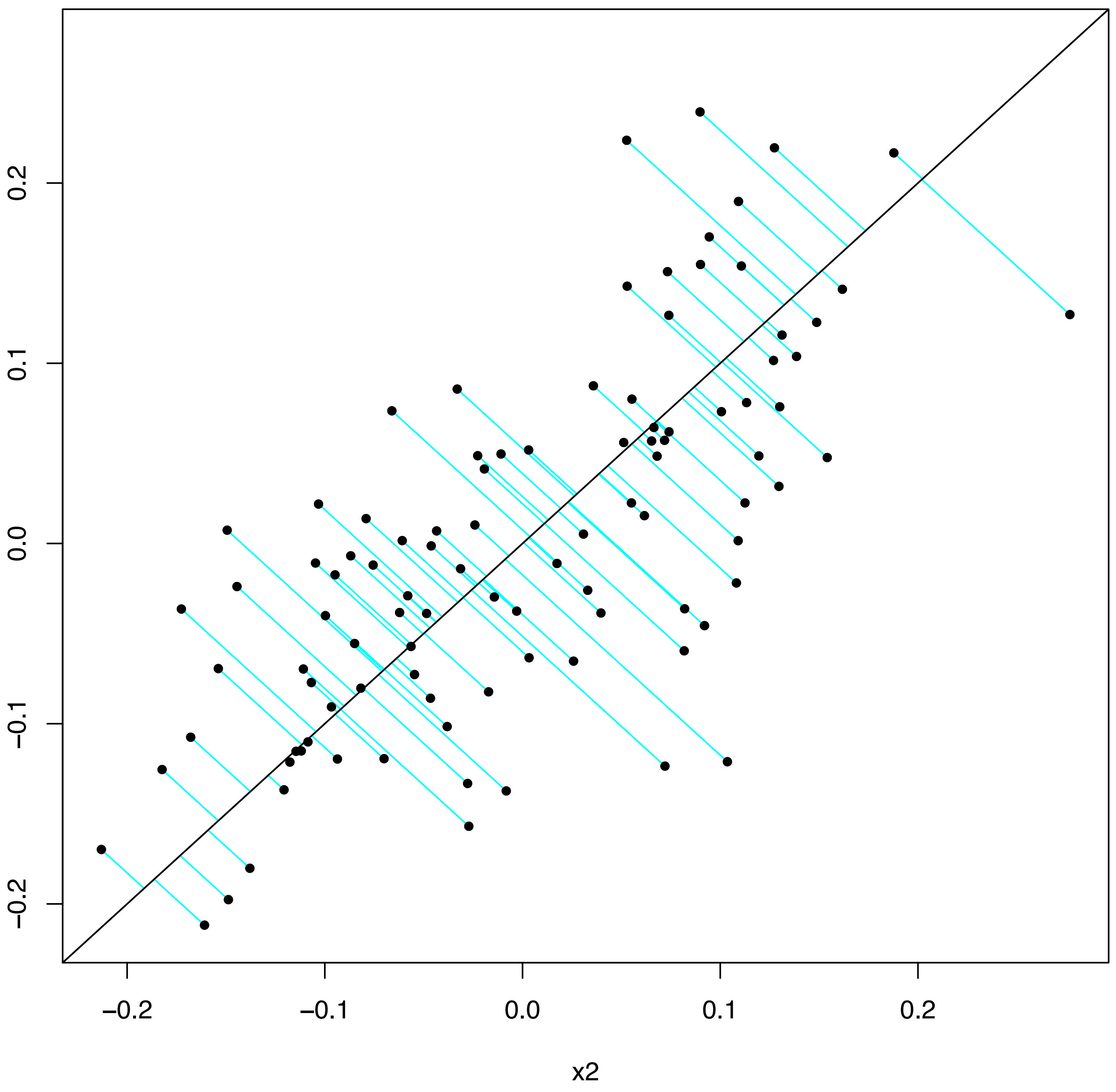
3. $\sum_{j=1}^{p} \text{var } z_j = \sum_{j=1}^{p} \text{var } x_j = \text{trace } \frac{\widetilde{M}^t \widetilde{M}}{n-1}$

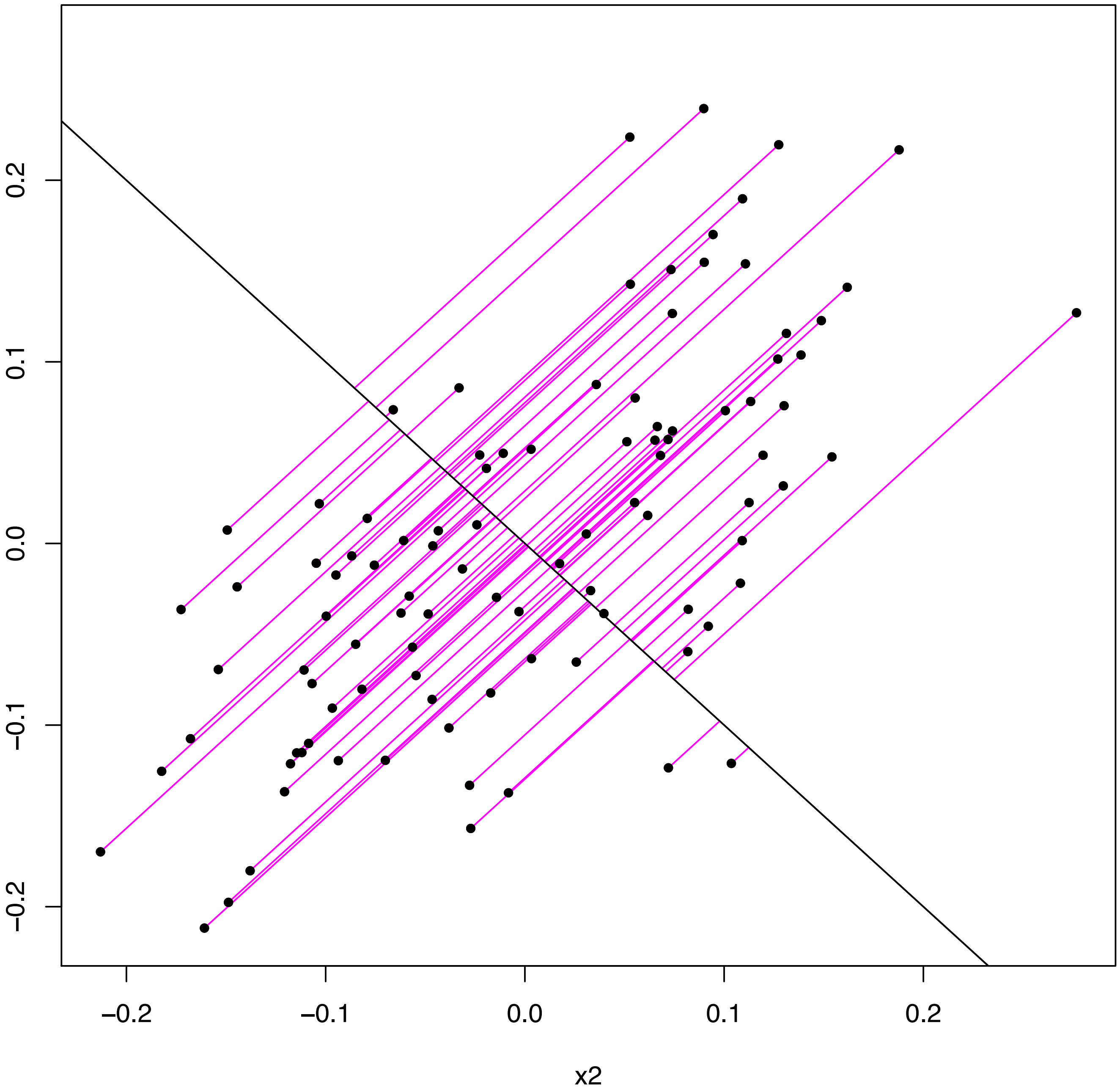
Motivation: Principal components

Intuitively, we are trying to come up with a new orthogonal coordinate system, one that is “aligned” with the data in the sense that the first direction captures most of the variability

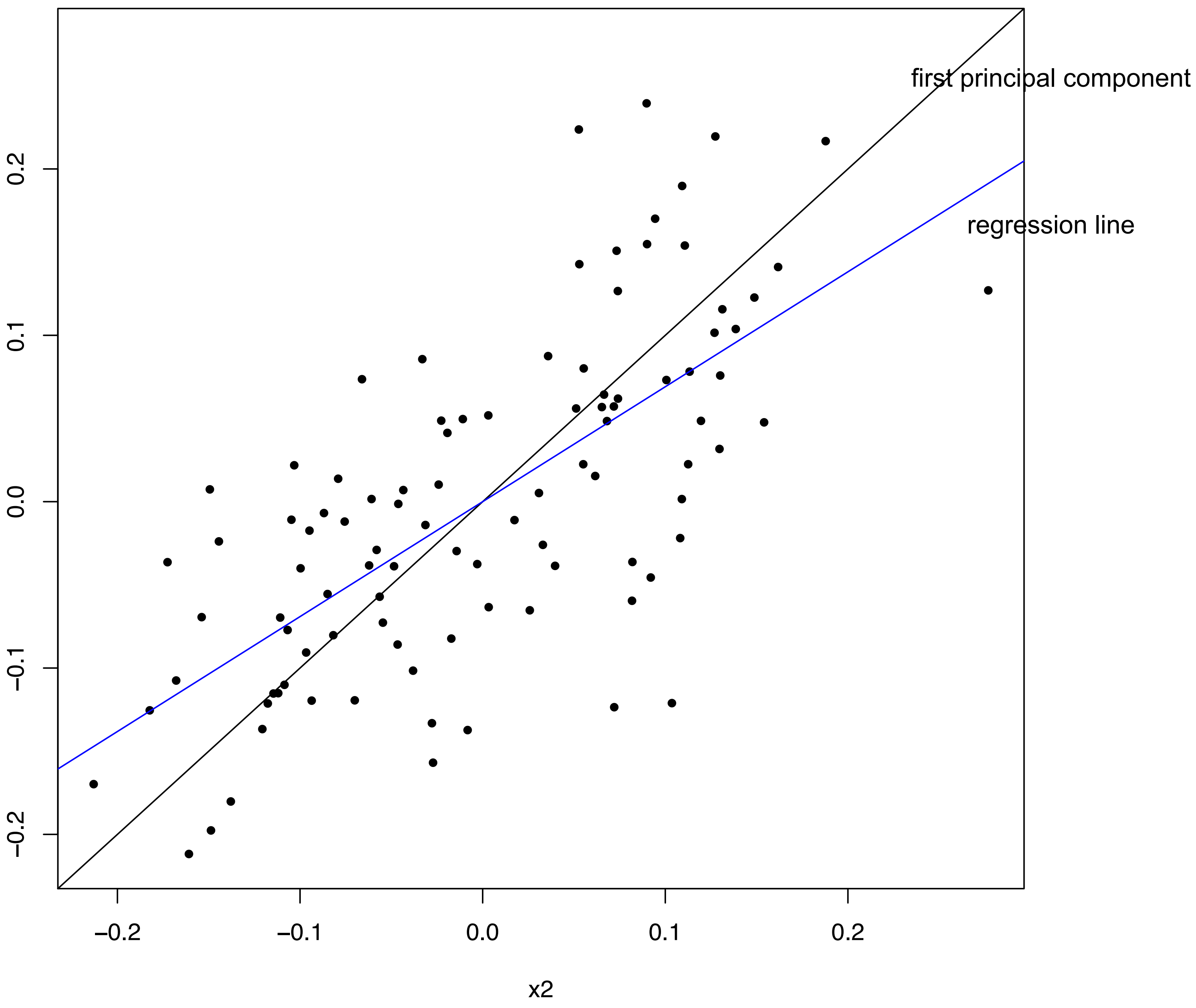
On the next few slides, we exhibit the principal components for a simple two-dimensional data set and illustrate the geometry involved -- We then compare this with a regression line fit to the same data


0.2
0.1
0.0
−0.1
−0.2
−0.2
−0.1
0.0
0.1
0.2
x2


0.2
0.1
0.0
−0.1
−0.2
−0.2
−0.1
0.0
0.1
0.2
x2



−0.2
−0.1
0.0
0.1
0.2
x2
−0.2
−0.1
0.0
0.1
0.2


first principal component
regression line
0.2
0.1
0.0
−0.1
−0.2
−0.2
−0.1
0.0
0.1
0.2
x2

Motivation: Principal components

What does it mean to have one or more small eigenvalues? It means that our data have a direction in which there is little variability -- This leads to the kind of cartoon instability we’ve seen in an earlier lecture when we motivated the Variance Inflation Factor

That seems like a long way to go to describe an unstable regression problem -- We’ll see however, that common “solutions” take us back to principal components via two paths

Motivation: Principal components

The constraint in step (2) essentially means each subsequent derived variable z_j is orthogonal to z_k for k=1,...,j-1 -- To see this just take the inner product between, say, the first two principal components

$$z_1^t z_2 = o_1^t \widetilde{M}^t \widetilde{M} o_2 = o_1^t O \Lambda O^t o_2 = 0$$

and by the same reasoning

$$z_1^t z_1 = o_1^t \widetilde{M}^t \widetilde{M} o_1 = o_1^t O \Lambda O^t o_1 = d_1$$

Motivation: Principal components

The first result is easy to see -- Because the vectors $o_1, \ldots, o_p$ are orthogonal, they form a basis for $\mathbb{R}^p$ and we can write any p-vector a as a linear combination

$$a = \alpha_1 o_1 + \cdots \alpha_p o_p = O\alpha$$

where $\alpha = (\alpha_1, \ldots, \alpha_p)^t$ -- Furthermore, if $\|a\| = 1$ then

$$\|\alpha\|^2 = \alpha^t\alpha = \alpha^t O^t O\alpha = a^t a = 1$$

Then, we have that $\text{var} \sum_{j=1}^{p} a_j x_j = a^t \widetilde{M}^t \widetilde{M} a/(n-1)$ and that

$$\begin{aligned} a^t \widetilde{M}^t \widetilde{M} a &= \alpha^t O^t \widetilde{M}^t \widetilde{M} O\alpha \\ &= \alpha^t O^t (O \Lambda O^t) O\alpha \\ &= \alpha^t D\alpha \\ &= \sum_{j=1}^{p} \alpha_j^2 d_j \end{aligned}$$

Motivation: Principal components

Condensing the chain of equalities, we can construct an upper bound

$$(n-1)\,\mathrm{var}\sum_{j=1}^{p} a_j x_j = \sum_{j=1}^{p} \alpha_j^2 d_j \leq d_1$$

By choosing $\alpha_1 = 1, \alpha_2 = 0, \ldots, \alpha_p = 0$ we can attain the maximum, which corresponds to setting $a = o_1$

The second result follows much like the first -- Try it!

Aside: The singular value decomposition, again

During this lecture, we have been working with two decompositions of two related matrices -- The SVD of $\widetilde{M} = UDV^t$ and the eigen-decomposition of $\widetilde{M}^t\widetilde{M} = O\Lambda O^t$

Recall that the squared singular values of $\widetilde{M}$ are the eigenvalues of $\widetilde{M}^t\widetilde{M}$ so that $s_j^2 = d_j, j = 1, \ldots, p$ -- Also, recall that V is the p-by-p matrix of eigenvectors of $\widetilde{M}^t\widetilde{M}$ so that O = V

Therefore, if $\widetilde{M} = UDO^t$, then $\widetilde{M}O = UD$ so that the first principal component is $\widetilde{M}o_1 = u_1d_1$ -- In general, we have that $z_j = u_jd_j$ and we often refer to the $u_j, j = 1, \ldots, p$, as “normalized” principal components because they have norm 1 (the columns of U were assumed orthonormal)

An alternative

Hoerl and Kennard (1970) describe the problem this way:

> *“The least squares estimate suffers from the deficiency of mathematical optimization techniques that give point estimates; the estimation procedure does not have built into it a method for portraying the sensitivity of the solution to the optimization criterion.”*

To get around this, they propose an alternative to OLS...

Ridge regression

In their paper, they first present the form of their solution -- Rather than consider the usual least squares estimates

$$\widehat{\beta} = (\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}})^{-1}\widetilde{\mathsf{M}}\mathsf{y}$$

They consider adding a “ridge” to $\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}}$ to yield

$$\widehat{\beta}^{*} = (\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}} + \lambda \mathbf{I}_{\mathsf{p}\times\mathsf{p}})^{-1}\widetilde{\mathsf{M}}^{\mathsf{t}}\mathsf{y}$$

where $\lambda \geq 0$

Ridge regression

There are a number of ways to arrive at this solution -- The most popular approach involves adding a constraint to the original OLS criterion

That is, find the value of β that minimizes

$$\sum_{i=1}^{n} (\widetilde{y} - \beta_1 \widetilde{x}_{i1} - \cdots - \beta_p \widetilde{x}_{ip})^2$$

subject to the constraint that $\sum_{j=1}^{p} \beta_j^2 \leq s$

This constraint is meant to have the effect of preventing the “cancellation” we saw at the start of the lecture -- A very large positive coefficient being “cancelled” by an equally large negative coefficient on another correlated variable

Ridge regression

To solve this, we could introduce a Lagrange multiplier and come up with the equivalent minimization problem

$$\sum_{i=1}^{n}(\,\widetilde{y} - \beta_1\widetilde{x}_{i1} - \cdots - \beta_p\widetilde{x}_{ip}\,)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

There is a one-to-one correspondence between s on the previous slide and the “penalty parameter” λ here -- Each act to control the size (Euclidean norm) of the coefficient vector

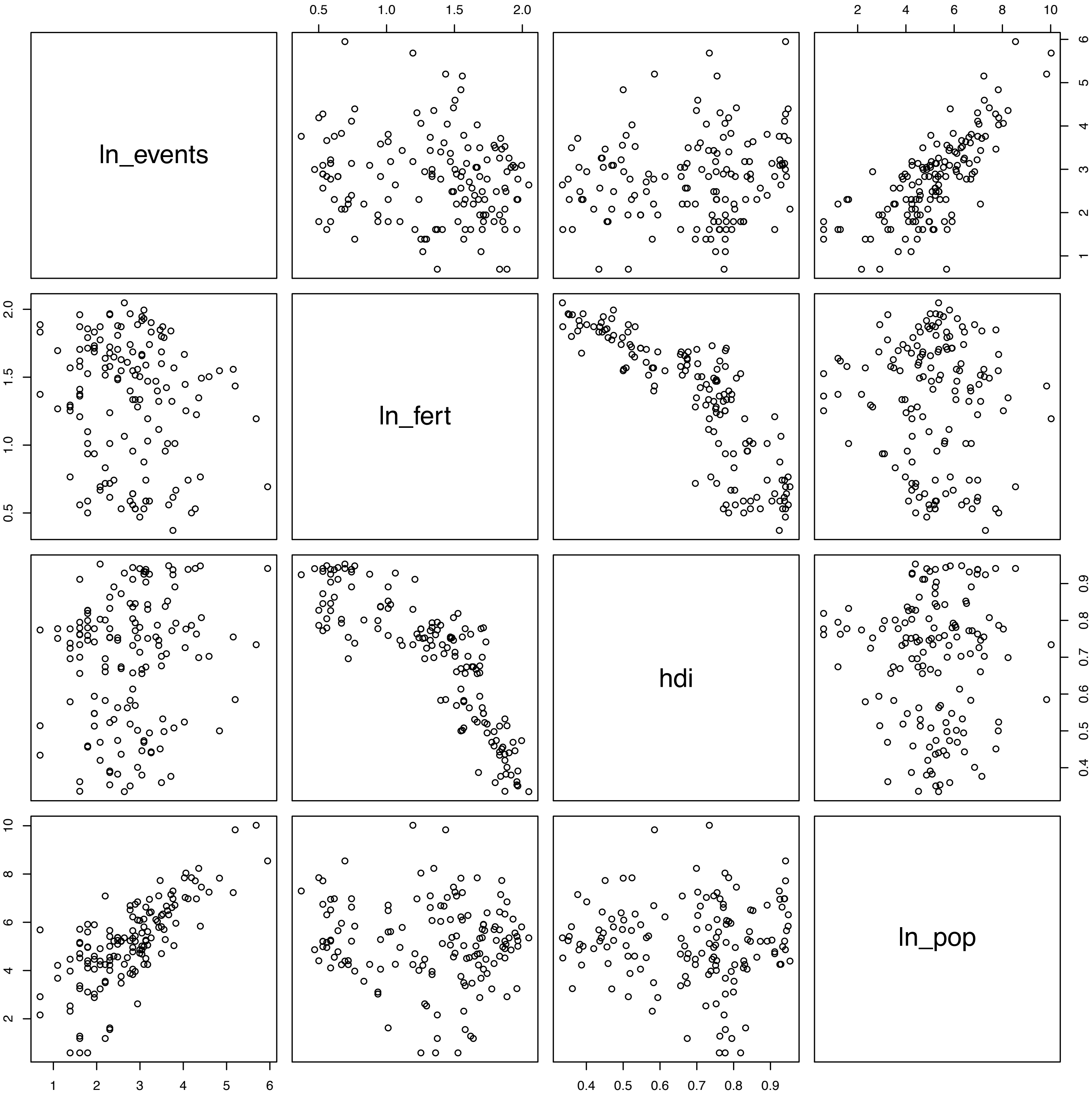
Initially, Hoerl and Kennard (1970) introduced ridge regression purely from a stability standpoint and start with the definition given two slides back -- More modern approaches to the subject start with this penalized expression, primarily because recent work examines different penalties on the coefficients

Example

Consider again our vulnerability data -- The MASS library contains a specialty function for fitting ridge regression at one or more values of the penalty parameter

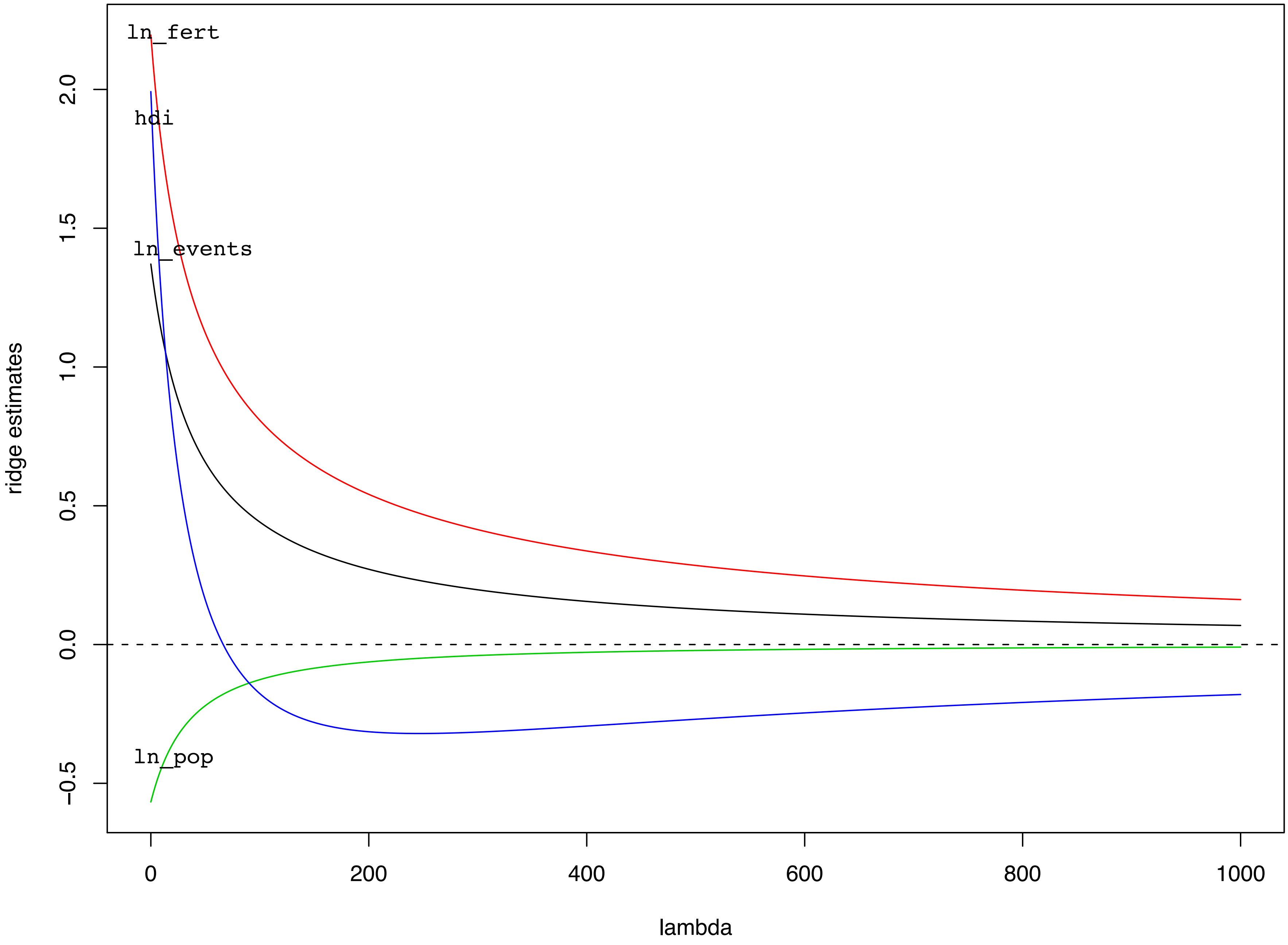
Plots of the ridge regression coefficients as a function of the penalty parameter are often referred to as the “ridge trace” and can tell us about the nature of the dependence between the variables

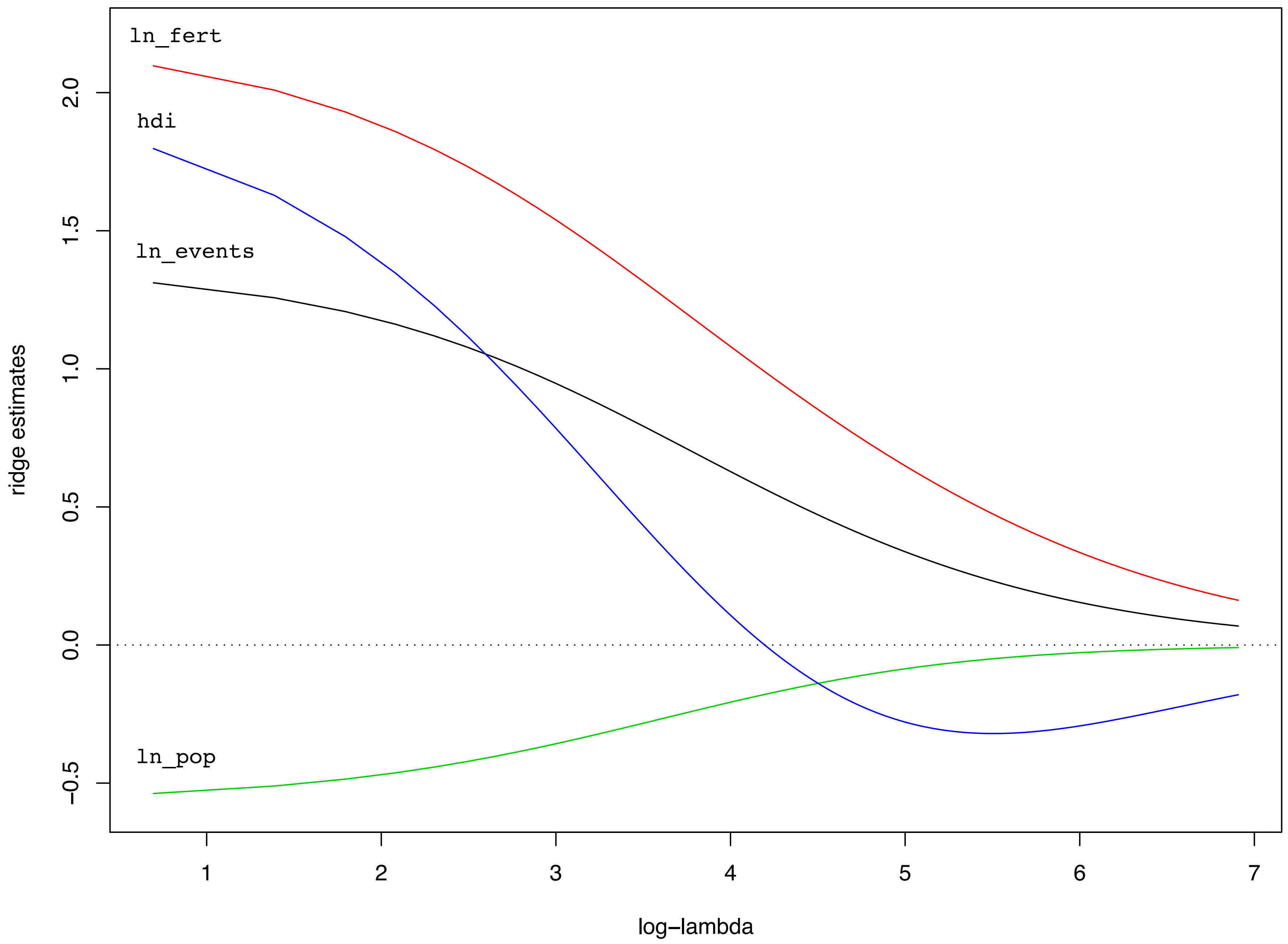
The function `lm.ridge` takes care of the centering of the variables -- On the next page, we illustrate how to use the function and illustrate the ridge traces...



ln_events
ln_fert
hdi
ln_pop
0.5
1.0
1.5
2.0
2
4
6
8
10
1
2
3
4
5
6
0.4
0.5
0.6
0.7
0.8
0.9

```
# specialty function in the MASS (Modern Applied Statistics with S by Venables and Ripley)
library(MASS)

# create values for the penalty parameter
lam <- seq(0,1000,len=500)

# and fit a series of ridge regressions (it’s worth looking at the code
# to see how they are doing them all in one go)

fits <- lm.ridge(ln_death_risk~ln_events+ln_fert+ln_pop+hdi,data=vul,lambda=lam)

# exhibit the coefficient vectors

matplot(log(lam),coefficients(fits)[,-1],lty=1, type="l",
         ylab="ridge estimates", xlab="log-lambda")
abline(h=0,lty=3)

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

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

# Coefficients:
# (Intercept)    ln_events      ln_fert       ln_pop          hdi
#     -5.3485       1.3708       2.1961      -0.5672       1.9922

cor(vul[,3:6])

#            ln_events     ln_fert         hdi      ln_pop
# ln_events  1.0000000 -0.15641675  0.14891515  0.74320022
# ln_fert   -0.1564168  1.00000000 -0.84119616 -0.09589724
# hdi        0.1489151 -0.84119616  1.00000000 -0.01559138
# ln_pop     0.7432002 -0.09589724 -0.01559138  1.00000000
```


ln_fert
hdi
ln_events
ln_pop
ridge estimates
2.0
1.5
1.0
0.5
0.0
−0.5
0
200
400
600
800
1000
lambda


ln_fert
hdi
ln_events
ln_pop
ridge estimates
2.0
1.5
1.0
0.5
0.0
−0.5
1
2
3
4
5
6
7
log−lambda

Example

The ridge traces all start at the least squares values for the estimates ($\lambda = 0$) and then eventually work their way to zero as the constraint tightens

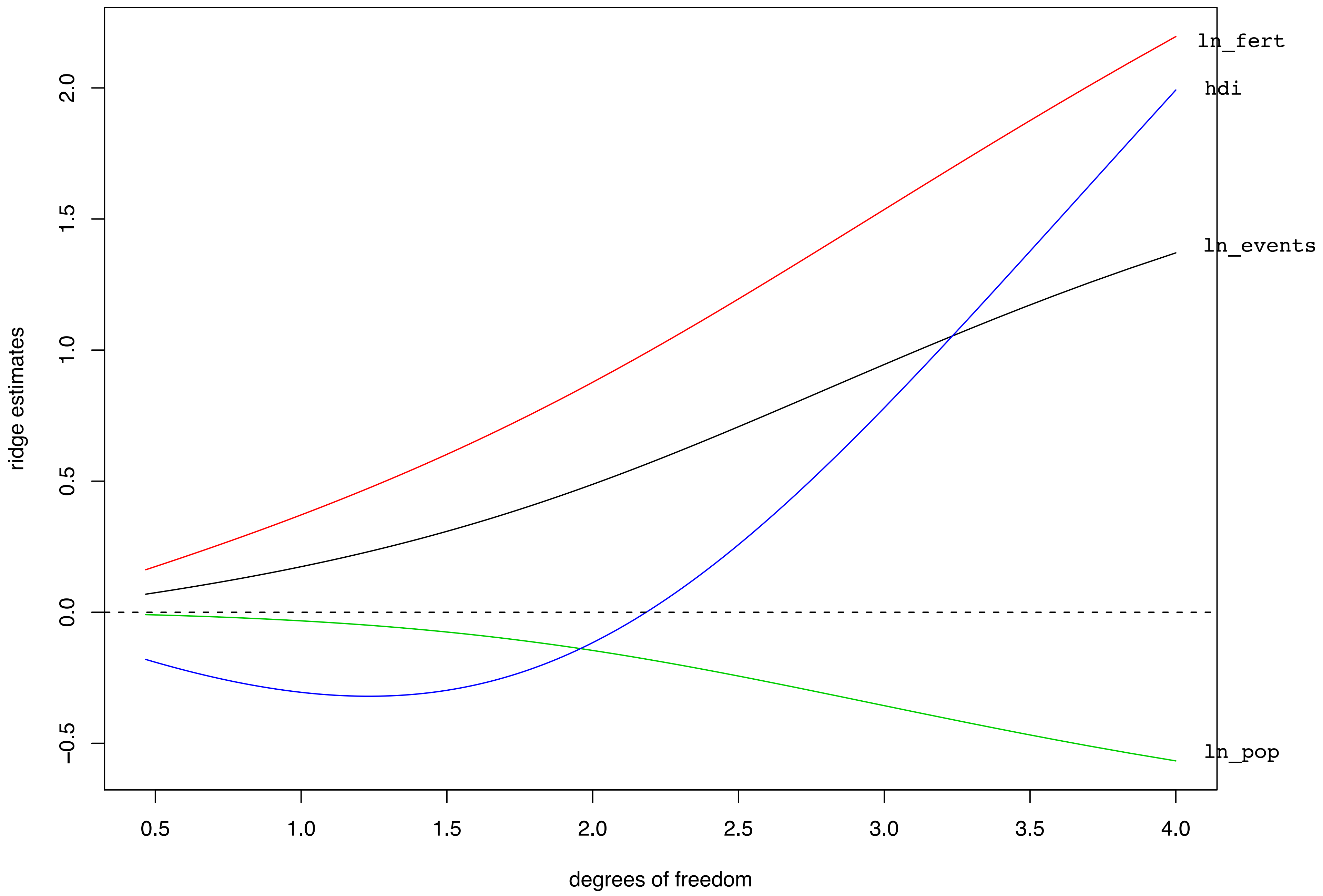
Notice that unlike the other three variables, HDI changes its sign as you increase the penalty -- This behavior is not uncommon when you have correlated predictors

In this case `ln_fert` and `HDI` are negatively correlated and so they can be thought of as representing the same factor, but with opposite signs -- As such, it doesn’t seem reasonable (although we better think about the particulars of the variables) that their contributions should have the same sign

Example

The one difficulty with these plots is that they are not well calibrated in terms of choosing the penalty parameter -- We have selected a large range of values, knowing that for 0 we have OLS and for something big, we have essentially an intercept-only model

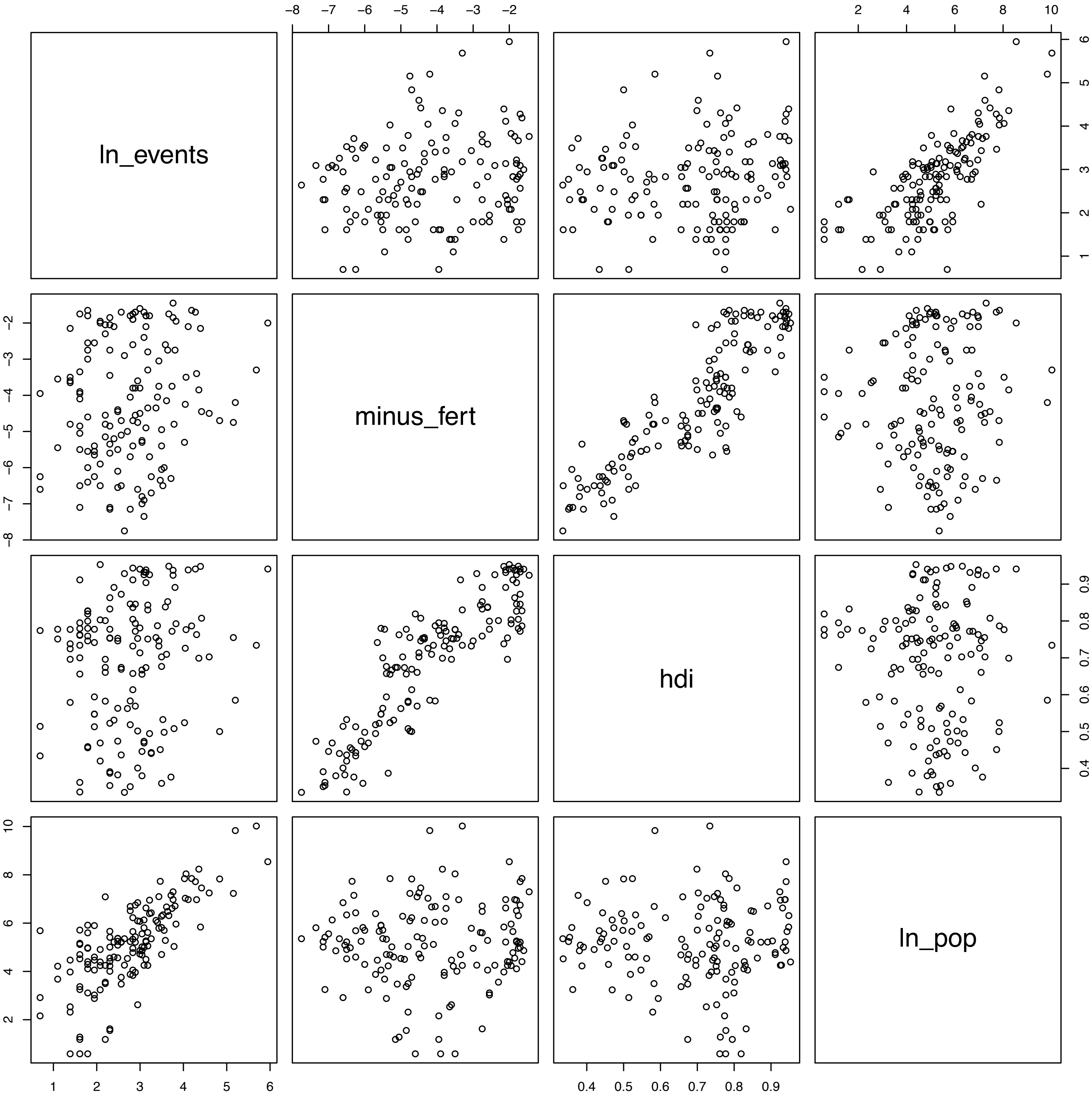
What scale might make it easier to interpret the action of our constraint?

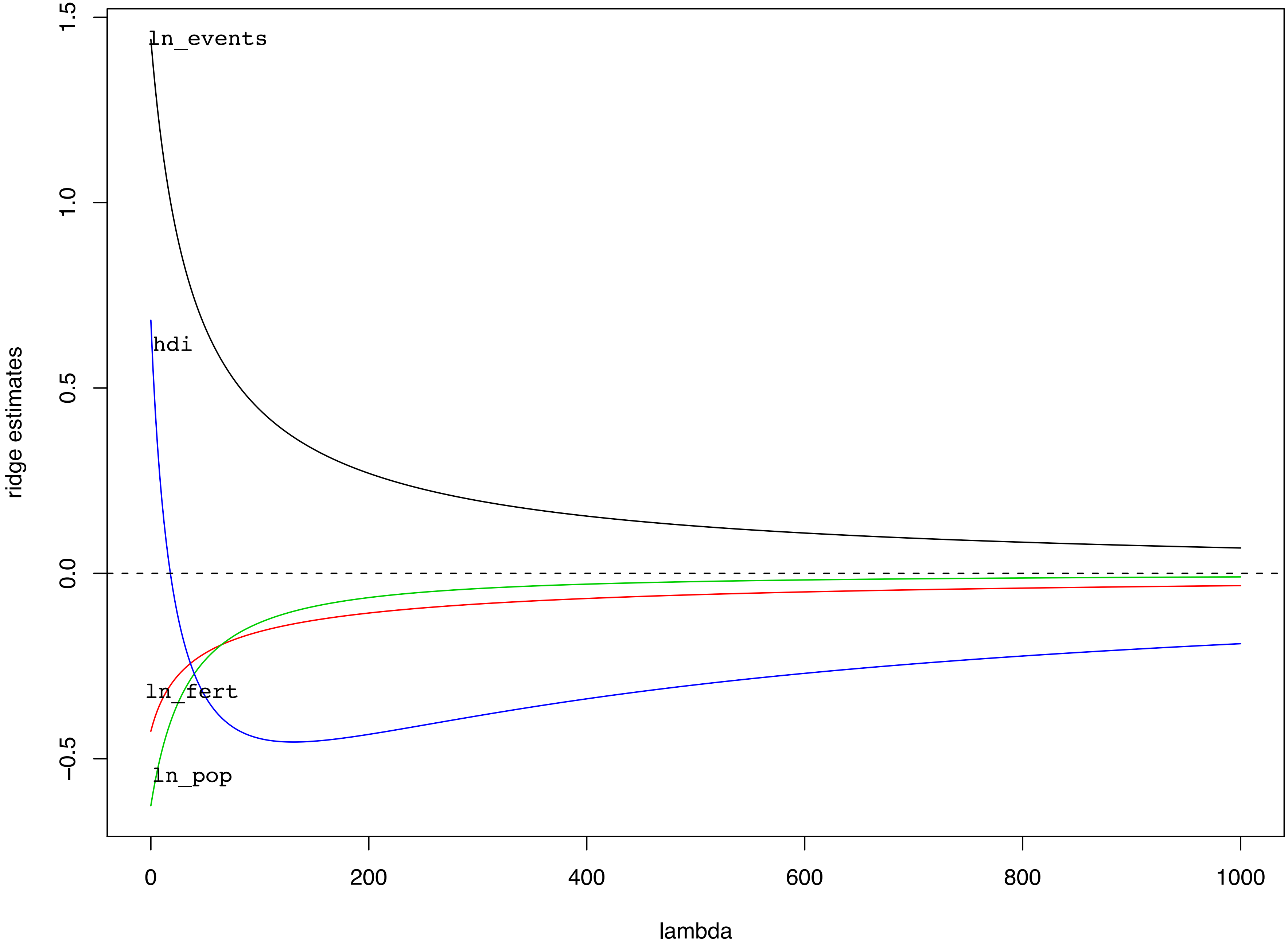

ridge estimates
2.0
1.5
1.0
0.5
0.0
−0.5
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
degrees of freedom
ln_fert
hdi
ln_events
ln_pop

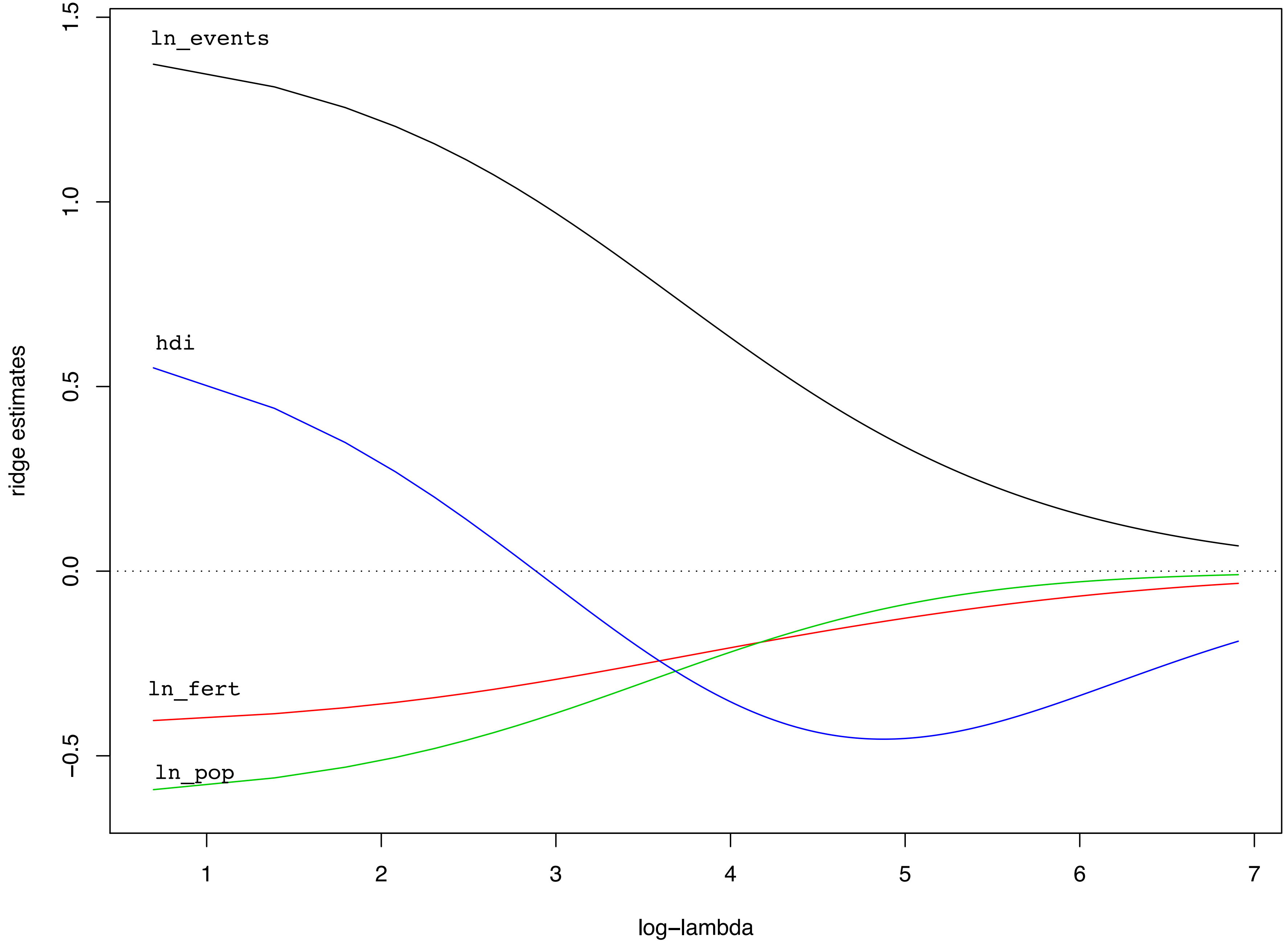
Altered example

Here we repeat the steps from before but use a transformed version of the fertility variable, one that is not correlated positively with HDI...

```
# fit a series of ridge regressions using the minus_fert variable

vul$minus_fert <- -exp(vul$ln_fert)

fits <- lm.ridge(ln_death_risk~ln_events+minus_fert+ln_pop+hdi,data=vul,lambda=lam)

# exhibit the coefficient vectors

matplot(log(lam),coefficients(fits)[,-1],lty=1, type="l",
         ylab="ridge estimates", xlab="log-lambda")
abline(h=0,lty=3)

lm(ln_death_risk~ln_events+minus_fert+ln_pop+hdi,data=vul)

# Call:
# lm(formula = ln_death_risk ~ ln_events + minus_fert + ln_pop +     hdi, data = vul)
#
# Coefficients:
# (Intercept)    ln_events   minus_fert       ln_pop          hdi
#     -3.1910       1.4404      -0.4259      -0.6268       0.6829

cor(vul[,c("ln_events","minus_fert","hdi","ln_pop")])

#            ln_events minus_fert         hdi      ln_pop
# ln_events  1.0000000 0.16203773  0.14891515  0.74320022
# minus_fert 0.1620377 1.00000000  0.88628808  0.06805284
# hdi        0.1489151 0.88628808  1.00000000 -0.01559138
# ln_pop     0.7432002 0.06805284 -0.01559138  1.00000000
```


ln_events
minus_fert
hdi
ln_pop
−8
−7
−6
−5
−4
−3
−2
2
4
6
8
10
1
2
3
4
5
6
0.4
0.5
0.6
0.7
0.8
0.9


ln_events
hdi
ln_fert
ln_pop
ridge estimates
lambda
1.5
1.0
0.5
0.0
−0.5
0
200
400
600
800
1000


ln_events
hdi
ln_fert
ln_pop
ridge estimates
1.5
1.0
0.5
0.0
−0.5
1
2
3
4
5
6
7
log−lambda

Ridge regression

The “penalized regression” on the previous slide can be solved using our good old standby calculus techniques -- Taking derivatives and forming an analog of the normal equations

$$\widehat{\beta}^* = (\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}} + \lambda\mathsf{I})^{-1}\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{y}}$$

How does this compare to our old friends the OLS estimates?

Character of the solution: Orthonormal predictors

Assume initially that $\widetilde{\mathsf{M}}$ has **orthonormal columns** so that $\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}} = \mathsf{I}$, and recall that the OLS estimates are just $\widehat{\beta} = \widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{y}}$

Then, following our nose(s) with the expression for the ridge regression, we find that the ridge solutions are given by

$$\begin{aligned} \widehat{\beta}^* &= (\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}} + \lambda \mathsf{I})^{-1}\widetilde{\mathsf{M}}\widetilde{\mathsf{y}} \\ &= (\mathsf{I} + \lambda \mathsf{I})^{-1}\widehat{\beta} \\ &= \frac{1}{1+\lambda}\widehat{\beta} \end{aligned}$$

What do you notice?

Character of the solution: Orthonormal predictors

Since $\lambda \geq 0$, we obtain our ridge regression estimates by “shrinking” their OLS cousins -- Ridge regression can be thought of as a shrinkage procedure, a class of techniques that gained popularity in the 1960s with the discovery of James-Stein estimation

We will consider the estimation properties of ridge regression in a moment -- For now we want to continue to think about the solution, and, in particular, what it means

Character of the solution: General case

When we don't have a special orthonormal model matrix, we can appeal to the singular value decomposition, writing $\widetilde{\mathsf{M}} = \mathsf{UDV}^{\mathsf{t}}$, to help us interpret the solution

$$\begin{aligned}
\widehat{\mu}^* &= \widetilde{\mathsf{M}}\widehat{\beta}^* \\
&= \widetilde{\mathsf{M}}(\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}} + \lambda \mathsf{I})^{-1}\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{y}} \\
&= \mathsf{UDV}^{\mathsf{t}}(\mathsf{VDU}^{\mathsf{t}}\,\mathsf{UDV}^{\mathsf{t}} + \lambda \mathsf{I})^{-1}\mathsf{VDU}^{\mathsf{t}}\widetilde{\mathsf{y}} \\
&\quad \text{(more grinding noises)} \\
&= \mathsf{UD}(\mathsf{D}^2 + \lambda \mathsf{I})^{-1}\mathsf{DU}^{\mathsf{t}}\widetilde{\mathsf{y}}
\end{aligned}$$

Expanding this last expression we find that

$$\widehat{\mu}^* = \sum_{\mathsf{j}=1}^{\mathsf{p}} \mathsf{u}_{\mathsf{j}} \frac{\mathsf{s}_{\mathsf{j}}^2}{\mathsf{s}_{\mathsf{j}}^2 + \lambda}(\mathsf{u}_{\mathsf{j}}^{\mathsf{t}}\widetilde{\mathsf{y}})$$

Character of the solution: General case

Recall that we could also write our least squares solution in terms of the elements of SVD

$$\widehat{\mu} = \mathsf{UU^ty} = \sum_{j=1}^{p} \mathsf{u_j(u_j^t\widetilde{y})}$$

which we can now directly compare to our ridge solutions

$$\widehat{\mu}^* = \sum_{j=1}^{p} \mathsf{u_j}\frac{\mathsf{s_j^2}}{\mathsf{s_j^2} + \lambda}(\mathsf{u_j^t\widetilde{y}})$$

The difference is the shrinkage factors $0 < \mathsf{s_j^2}/(\mathsf{s_j^2} + \lambda) < 1$

Character of the solution: The general case

Recall that the columns of U are normalized principal components -- Each column u_j can be thought of as $u_j = z_j/\|z_j\|$ where $z_1, \ldots, z_p$ are our p principal components

Therefore, our OLS fit can be thought of in terms of this set of orthonormal basis vectors $u_1, \ldots, u_p$ -- The ridge fit also uses these directions, but shrinks the OLS estimates according to factors that depend on the eigenvalues of $\widetilde{M}^t\widetilde{M}$

$$\frac{d_j}{d_j + \lambda}$$

Therefore, a greater amount of shrinkage is applied to directions associated with smaller values of d_j , the “thinner” direction of our ellipse in the previous data example

Character of the solution: The general case

In the end, we see that ridge regression protects against the instability of gradients estimated in these thin directions -- Implicitly, we’re hoping that the response will tend to vary most in the directions of high variance for the inputs

This is often a reasonable assumption, but does not need to hold in general -- Recall that the principal components are entirely a function of the input space and does not involve the actual response at all

Alternatives

If the issue is multi-collinearity, then shrinkage is just one solution -- A common alternative is so-called principal components regression

Here we use the principal components as a new orthogonal basis and add terms sequentially from 1 through p -- This is a kind of subset selection in which the basis elements have a “natural” ordering and we consider a sequence of models of the form

$$\widehat{\mu}_j = \bar{y} + \sum_{k=1}^{j} u_k(u_k^t \widetilde{y})$$

We’ll come back to the idea of how to choose j, but as an intellectual exercise, compare this “keep or kill” approach to the shrinkage from ridge regression

Alternatives

In the next lecture, we'll consider the so-called bias-variance tradeoff and see how it might help us guide some decisions (how to pick λ or how to pick j)

We will also examine another kind of penalty -- Rather than putting the Euclidean norm on β we could write

$$\sum_{i=1}^{n}(\,\widetilde{y}_i - \beta_1\widetilde{x}_{i1} - \cdots - \beta_p\widetilde{x}_{ip}\,)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

What effect does this have? We'll spend a lot of time with a simple orthogonal model in the hopes that the directness of the estimates will help us build intuition for the overall methodology

Aside: Degrees of freedom

The degrees of freedom we plotted a few slides back is essentially a trace

$$\mathsf{df}(\lambda) = \mathsf{trace}\ [\widetilde{\mathsf{M}}(\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}} + \lambda \mathsf{I})^{-1}\widetilde{\mathsf{M}}^{\mathsf{t}}] = \sum_{\mathsf{j}=1}^{\mathsf{p}} \frac{\mathsf{d}_{\mathsf{j}}}{\mathsf{d}_{\mathsf{j}} + \lambda}$$

Notice that if we are not doing any shrinkage ($\lambda = 0$), then the degrees of freedom is just p as you would hope -- With increasing λ the degrees of freedom drop to zero

Keep in mind that all of these calculations have been done omitting the intercept from the calculations -- This has been allowed to pass “as is” without any shrinkage