

STAT 425

Simple Linear Regression. Part 1

An example

The `cats` data set from the MASS library

```
library(MASS)
help(cats)
summary(cats)

##   Sex          Bwt          Hwt
##   F:47   Min.   :2.000   Min.   : 6.30
##   M:97   1st Qu.:2.300   1st Qu.: 8.95
##             Median :2.700   Median :10.10
##             Mean   :2.724   Mean   :10.63
##             3rd Qu.:3.025   3rd Qu.:12.12
##             Max.   :3.900   Max.   :20.50
```

Heart Weight in g

18
14
10
6

2.0 2.5 3.0 3.5

Body Weight in kg

- The goal is to describe the relationship between Hwt (heart weight) and Bwt (body weight). As a starting point, we assume the relationship is linear.
- Data of the form: $(y_i, x_i), i = 1, \dots, n$ where $y_i, x_i \in \mathbb{R}$.
- Apparently the data won't be able to fit on a straight line.
Assume $y_i = \beta_0 + \beta_1 x_i + e_i$.
 (β_0, β_1) : unknown regression coefficients
 e_i 's: random errors often assumed to have zero mean and variance σ^2

Simple Linear Regression Overview (I)

- How to use Least Squares (LS) to estimate (β_0, β_1) ? We can obtain an explicit expression $(\hat{\beta}_0, \hat{\beta}_1)$. There is a nice connection between the LS estimate of the slope, β_1 , and sample correlation/variance of X and Y, which will help you to remember the expression.
- Throughout the class we'll use some jargon: fitted value, residual, Residual Sum of Squares (RSS), R-squared (used to assess the overall model fit).
- How would the LS fitting/inference be affected if the data, X and/or Y, are shifted and/or scaled (i.e., linear transformed)?
- SLR without the intercept: fit a regression line passing through the origin.
- How to use **R** to carry out all the analysis and produce relevant graphs.

Parameter estimation by Least squares

We would like to choose a line which is close to the data points. We measure the closeness by squared errors ¹.

Least Squares Estimation: find (β_0, β_1) that minimize the residual sum of squares (RSS)

$$RSS = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2$$

To find the solution, we can take the derivatives w.r.t. β_0 and β_1 and equate to zero.

$$\frac{\partial RSS}{\partial \beta_0} = -2 \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\frac{\partial RSS}{\partial \beta_1} = -2 \sum_{i=1}^n x_i(y_i - \beta_0 - \beta_1 x_i) = 0$$

¹Why squared error? Why not absolute error?

Re-arrange the equations,

$$\beta_0 n + \beta_1 \sum x_i = \sum y_i, \quad (1)$$

$$\beta_0 \sum x_i + \beta_1 \sum x_i^2 = \sum x_i y_i. \quad (2)$$

From (1), we have

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}.$$

Plug it back to (2),

$$(\bar{y} - \hat{\beta}_1 \bar{x}) \sum x_i + \beta_1 \sum x_i^2 = \sum x_i y_i$$

$$\beta_1 (\sum x_i^2 - \sum x_i \bar{x}) = \sum x_i y_i - \sum x_i \bar{y}$$

$$\hat{\beta}_1 = \frac{\sum x_i y_i - \sum x_i \bar{y}}{\sum x_i^2 - \sum x_i \bar{x}} = \frac{\sum x_i (y_i - \bar{y})}{\sum x_i (x_i - \bar{x})}.$$

Some equalities (basically centering one side is the same as centering both sides for cross-products):

$$\sum_i (x_i - \bar{x})(y_i - \bar{y}) = \sum_i x_i(y_i - \bar{y}) = \sum_i (x_i - \bar{x})y_i.$$

So the LS estimates of (β_0, β_1) can be expressed as

$$\begin{aligned}\hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x}, \\ \hat{\beta}_1 &= \frac{S_{xy}}{S_{xx}} = r_{XY} \left(\frac{S_{yy}}{S_{xx}} \right)^{1/2},\end{aligned}$$

where

$$S_{xy} = \sum (x_i - \bar{x})(y_i - \bar{y}),$$

$$S_{xx} = \sum (x_i - \bar{x})^2, \quad S_{yy} = \sum (y_i - \bar{y})^2,$$

$$r_{XY} = \frac{S_{xy}}{\sqrt{(S_{xx})(S_{yy})}} \quad (\text{the sample correlation}).$$

- Recall that SLR assumes the dependence between X and Y is linear.
- It is not surprising that the LS estimates are related to the sample correlation between X and Y.
- Correlation is exactly the measure used to quantify the linear dependence between two variables ².

²We can build an example in where variables X and Y have a non-linear relationship and their correlation is zero

Suppose that the mean and variance of X and Y, and the correlation between X and Y r_{xy} are known. Given a value of x , what is the best guess of y ?

It seems reasonable to use the *unit-free location/scale invariant* value of x multiplied by r_{xy} to get a *unit-free location/scale invariant* value of y as follows:

$$\frac{y - \mu_y}{\sigma_y} \approx r_{xy} \frac{x - \mu_x}{\sigma_x}$$

3

By using the sample estimates of the means, variances and correlation coefficient we get the corresponding sample expression:

$$\frac{y - \bar{y}}{\sqrt{S_{yy}}} \approx r_{xy} \frac{x - \bar{x}}{\sqrt{S_{xx}}} \rightarrow y - \bar{y} \approx r_{xy} \sqrt{\frac{S_{yy}}{S_{xx}}} (x - \bar{x})$$

³If you want to get x as a function of y , you need to multiply by r_{xy} on the y side of the equation.

A final equation of y as a function of x is given by:

$$y \approx \left(\bar{y} - r_{xy} \sqrt{\frac{S_{yy}}{S_{xx}}} \bar{x} \right) + \left(r_{xy} \sqrt{\frac{S_{yy}}{S_{xx}}} \right) x$$

Some jargon:

- Fitted value or prediction at x_i : $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$
- Residual at x_i : $r_i = y_i - \hat{y}_i$. If you plug-in the equations from page 7 for $\hat{\beta}_i$, you can show that:

$$\sum_i r_i = 0, \quad \sum_i x_i r_i = 0$$

4

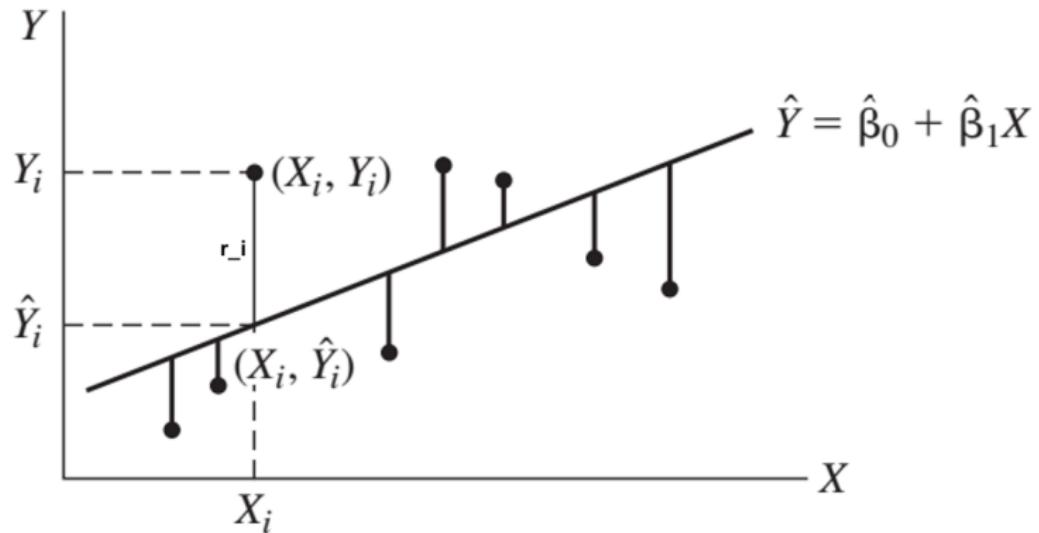
- Residual Sum of Squares (RSS): $\sum_i r_i^2$
- The error variance is estimated as:

$$\hat{\sigma}^2 = \frac{1}{n-2} RSS = \frac{1}{n-2} \sum_i r_i^2$$

- residual degrees of freedom (df): $n - 2$. Normally
 $df = \text{sample size} - \text{number of parameters}$

⁴ $\sum_i r_i = 0$ implies that the mean of $\hat{y}_i = \bar{y}$

LS fitted linear regression



Goodness of fit: R-square

The total variation of y (Total Sum of Squares (TSS)) can be decomposed into the sum of the total variation of the fitted values \hat{y} (FSS) and the Residual Sum of Squares (RSS):

$$\begin{aligned} TSS &= \sum_i (y_i - \bar{y})^2 = \sum_i (y_i - \hat{y}_i + \hat{y}_i - \bar{y})^2 = \sum_i (r_i + \hat{y}_i - \bar{y})^2 \\ &= \sum_i r_i^2 + \sum_i (\hat{y}_i - \bar{y})^2 \\ &= RSS + FSS \end{aligned} \tag{3}$$

5

Note: The average of the \hat{y}_i ($\bar{\hat{y}}$) is the same as the average of the y_i . This is true because the intercept is included in the model.

⁵The cross product $\sum_i r_i(\hat{y}_i - \bar{y}) = \hat{\beta}_0 \sum_i r_i + \hat{\beta}_1 \sum_i r_i x_i - \bar{y} \sum_i r_i = 0$

A common measure on how well the model fits the data is the so-called coefficient of determination or simply R-square:
For a given data set where TSS is fixed, the smaller the RSS, the larger the R-squared.

$$R^2 = \frac{\sum_i (\hat{y}_i - \bar{y})^2}{\sum_i (y_i - \bar{y})^2} = \frac{FSS}{TSS} = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

We can also show that $R^2 = r_{XY}^2$.

Note also that $R^2 = \frac{Var(\hat{y})}{Var(y)}$. This ratio measures how much variation in the original data y_i 's is explained or reduced by the LS fitting. If Y and X are strongly linear dependent, a linear function of X can help to reduce the uncertainty (i.e., variation) of Y.

Fitting a Linear Model in R

```
out = lm(Hwt~Bwt, data = cats)
summary(out)

##
## Call:
## lm(formula = Hwt ~ Bwt, data = cats)
##
## Residuals:
##       Min     1Q Median     3Q    Max 
## -3.5694 -0.9634 -0.0921  1.0426  5.1238 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -0.3567    0.6923  -0.515   0.607    
## Bwt         4.0341    0.2503  16.119  <2e-16 ***  
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
##
## Residual standard error: 1.452 on 142 degrees of freedom
## Multiple R-squared:  0.6466, Adjusted R-squared:  0.6441 
## F-statistic: 259.8 on 1 and 142 DF,  p-value: < 2.2e-16
```

Model output is stored in the `out` object. `out` is a [list](#) in R.

Extract Information and make some calculations

```
names(out)

## [1] "coefficients"   "residuals"      "effects"       "rank"
## [5] "fitted.values"  "assign"        "qr"           "df.residual"
## [9] "xlevels"         "call"          "terms"         "model"

out$coef

## (Intercept)      Bwt
## -0.3566624  4.0340627

cor(Hwt,Bwt)^2

## [1] 0.6466209

var(out$fitted.values)/var(Hwt)

## [1] 0.6466209

1 - sum(out$res^2)/sum((Hwt-mean(Hwt))^2)

## [1] 0.6466209

summary(out)$r.sq

## [1] 0.6466209
```

Different ways to calculate the R-square

How affine transformations on the data affect the Regression?

Affine transformation: $\tilde{y} = ay + b$ where a and b are known constants.

Changes of scale in X or Y are also affine transformations.

Suppose we have a SLR model of Y on X.

- If we rescale the data $\tilde{y} = ay + b$, and then regress \tilde{y} on x . How would the LS estimates and R^2 be affected?
- If we re-scale the covariates $x \tilde{x} = ax + b$, and then regress y on \tilde{x} . How would the LS estimates and R^2 be affected?
- If we regress X on Y instead, will the LS line be the same? How about R^2 ?

In R:

```
out1<-lm(Hwt ~ I(Bwt*1000), data=cats)
out2<-lm(I(Hwt+1) ~ Bwt, data=cats)
out3<-lm(Bwt ~ Hwt,data=cats)
cbind(out$coef, out1$coef, out2$coef, out3$coef)

## [,1] [,2] [,3] [,4]
## (Intercept) -0.3566624 -0.356662433 0.6433376 1.0196367
## Bwt 4.0340627 0.004034063 4.0340627 0.1602902

cbind(summary(out)$r.square,summary(out1)$r.square,
summary(out2)$r.square,summary(out3)$r.square)

## [,1] [,2] [,3] [,4]
## [1,] 0.6466209 0.6466209 0.6466209 0.6466209
```

Regression through the Origin

Sometimes we want to fit a line with no intercept (regression through the origin): $y_i \approx \beta_1 x_i$. For example, x_i denotes the intensity level of various exercises and y_i denotes the additional calories you burn with those exercises.

We can estimate β_1 using the LS principle

$$\min_{\beta_1} \sum_{i=1}^n (y_i - \beta_1 x_i)^2 \implies \hat{\beta}_1 = \frac{\sum_i x_i y_i}{\sum_i x_i^2}.$$

The ordinary definition of R-square is no longer meaningful; you could have RSS bigger than TSS, and therefore have a negative R-square, if you use formula $R^2 = 1 - \text{RSS}/\text{TSS}$.

The ordinary R-square measures the effect of X after removing the effect of the intercept by centering both y_i 's and \hat{y}_i 's. For regression models with no intercept, we shouldn't do the centering when computing R-square.

Let's look at the following decomposition (slightly different from (3))

$$\sum_i y_i^2 = \sum_i (y_i - \hat{y}_i + \hat{y}_i)^2 = \sum_i (y_i - \hat{y}_i)^2 + \sum_i \hat{y}_i^2.$$

Then define R-square for regression with no intercept as

$$\tilde{R}^2 = \frac{\sum_i \hat{y}_i^2}{\sum_i y_i^2} = 1 - \frac{\text{RSS}}{\sum_i y_i^2}.$$

Some remarks

- We will use the **hat** symbol for the estimators/estimates of the true model parameters: $(\hat{\beta}_1, \hat{\beta}_2, \hat{\sigma}^2)$ are the LS estimators of the population parameters: $(\beta_0, \beta_1, \sigma^2)$
- These estimators are a function of the **sample data**. If we have a different sample, we will have a different set of estimators. These implies the estimators are **random variables**.
- As a next step we will check the properties of these estimators and we will determine their probability distributions.

STAT 425

Simple Linear Regression. Part 2

Properties of the LS estimates

- We will study the statistical properties of $(\hat{\beta}_0, \hat{\beta}_1)$ as the LS estimates of the true parameter vector (β_0, β_1)
- We will compute the mean, variance and covariance of $(\hat{\beta}_0, \hat{\beta}_1)$ and check that they are **unbiased estimators**.
- We can also show that they achieve the smallest mean square error (MSE) among all unbiased estimators, but we will show this result as a general result when discussing Multiple Linear Regression (MLR).
- Until this point, we only need to make the following assumptions about the 1st and 2nd moments of residuals e_i 's:
 $E[e_i] = 0, \text{Var}(e_i) = \sigma^2, \text{Cov}(e_i, e_j) = 0, i \neq j$

- For hypothesis testing and constructing confidence/prediction intervals we need to derive the probability distribution of $(\hat{\beta}_0, \hat{\beta}_1)$
- We will assume that the e_i 's have a normal distribution and are independent and identically distributed (i.i.d.). We will use the t distribution for hypothesis testing and confidence intervals. With assumptions about the 1st and 2nd order moments and for a large sample size, we can use the CLT and assume a normal distribution instead.
- Notation: Uppercase letters are normally used for Random Variables, and lowercase letters for observed values of the random variables. Uppercase letter will be also reserved for matrices. In some occasions lowercase letter will also be used for random variables.

Least-squares estimates properties

Lets assume: $y_i = \beta_0 + \beta_1 x_i + e_i$ and 1st and 2nd moments for the e_i 's:

$$E[e_i] = 0, \text{ and } Cov[e_i, e_j] = \sigma^2 \delta_{ij}$$

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$. The assumptions 3 about the 1st and 2nd moments of the error terms leads to the following assumption on the 1st and 2nd order moments of y **conditioning on X :**

$$E[y_i|x_i] = \beta_0 + \beta_1 x_i, , Cov[y_i, y_j|x_i, x_j] = \sigma^2 \delta_{ij}$$

In stat425, the statistical assumption is on the conditional distribution of Y given X . So **when we evaluate expectations, only y_i 's are random and x_i 's are treated as known, non-random constants.**

The LS estimates are unbiased

$$\hat{\beta}_1 = \sum_i \frac{(x_i - \bar{x})}{S_{XX}} y_i = \sum_i c_i y_i \quad (\sum_i c_i = 0)$$

$$E[\hat{\beta}_1] = \sum_i c_i E[y_i] = \sum_i c_i (\beta_0 + \beta_1 x_i) = \beta_1 \left(\sum_i x_i c_i \right) = \beta_1$$

$$\hat{\beta}_0 = \bar{y} - \bar{x} \hat{\beta}_1$$

$$E[\hat{\beta}_0] = \frac{1}{n} \sum_i E[y_i] - E[\hat{\beta}_1] \bar{x} = \beta_0 + \bar{x} \beta_1 - \bar{x} \beta_1 = \beta_0$$

1

$$^1 \sum_i x_i c_i = 1$$

Mean Square Error (MSE) of the LS estimates

Since the LS estimates are unbiased, their MSE².

$$Var(\hat{\beta}_1) = Var\left(\sum_i c_i y_i\right) = \sigma^2 \sum_i c_i^2 = \sigma^2 \frac{1}{S_{XX}}$$

$$Var(\hat{\beta}_0) = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{XX}} \right)$$

³ Note that both variances depend reciprocally on S_{XX} . The larger the distance of x_i from the sample mean \bar{x} , the larger the value of S_{XX} and the smaller the Variance of the LS estimates.

² $MSE(\hat{\beta}) = E[\beta - \hat{\beta}]^2 = E[\hat{\beta} - E[\hat{\beta}]]^2$ is equal to the Variance of $\hat{\beta}$

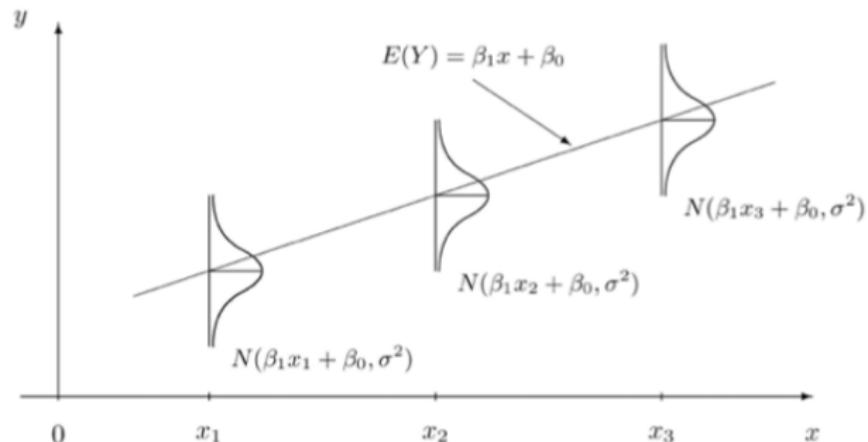
³ We can write $\hat{\beta}_0 = \bar{y} - \sum_i c_i y_i \bar{x} = \sum_i (\frac{1}{n} - c_i \bar{x}) y_i$

Normal assumptions

Assume : $y_i = \beta_0 + \beta_1 x_i + e_i$

- The residuals e_i are assumed independent and $e_i \sim N(0, \sigma^2)$.
This is equivalent to say that y_i 's are independent and
 $y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$
- The mean function $E[y_i] = \beta_0 + \beta_1 x_i$ is a linear function of x_i
- Independence of e_i 's implies Independence of y_i 's
- Normality of e_i 's implies Normality of y_i 's
- Variance homogeneity of e_i 's implies variance homogeneity of y_i 's
- Since the e_i 's are normal and independent, jointly they have a normal distribution. Therefore the y_i 's are jointly normal, and any linear combination of the y_i 's is also normal. This fact has important implications for coming inference results.

Simple Linear Regression Assumptions



Distributions of the LS estimates

- $(\hat{\beta}_0, \hat{\beta}_1)$ have a joint normal distribution with mean, variances and covariance given by:

$$E[\hat{\beta}_0] = \beta_0 \qquad \qquad \qquad Var[\hat{\beta}_0] = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{XX}} \right)$$

$$E[\hat{\beta}_1] = \beta_1 \qquad \qquad \qquad Var[\hat{\beta}_1] = \sigma^2 \frac{1}{S_{XX}}$$

$$Cov(\hat{\beta}_0, \hat{\beta}_1) = -\sigma^2 \frac{\bar{x}}{S_{XX}}$$

- The residual sum of squares $RSS = \sum(y_i - \hat{y}_i)^2 \sim \sigma^2 \chi_{n-2}^2$.

$$E[\hat{\sigma}^2] = E\left[\frac{RSS}{n-2}\right] = \sigma^2(n-2)/(n-2) = \sigma^2$$

- $(\hat{\beta}_0, \hat{\beta}_1)$ and RSS are independent.

Hypothesis testing

- We want to test: $H_0 : \beta_1 = c$ vs. $H_a : \beta_1 \neq c$ (c is a known constant)
- Test statistic:

$$t = \frac{\hat{\beta}_1 - c}{se(\hat{\beta}_1)} = \frac{\hat{\beta}_1 - c}{\hat{\sigma}/\sqrt{S_{XX}}} \sim T_{n-2}$$

- p-value = $2 \times$ the area under the T_{n-2} distribution more extreme than the observed statistic t
- The p-value returned by the **R** command **lm** is for the test with $H_0 : \beta_1 = 0$

Call:

```
lm(formula = Hwt ~ Bwt, data = cats)
```

Residuals:

Min	1Q	Median	3Q	Max
-3.5694	-0.9634	-0.0921	1.0426	5.1238

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.3567	0.6923	-0.515	0.607
Bwt	4.0341	0.2503	16.119	<2e-16 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.452 on 142 degrees of freedom

Multiple R-squared: 0.6466, Adjusted R-squared: 0.644

F-statistic: 259.8 on 1 and 142 DF, p-value: < 2.2e-16

Calculating the p-values:

```
> 2*(1-pt(abs(summary(out)$coefficients[1,3]),  
+ out$df.residual))  
[1] 0.6072131  
  
> 2*(1-pt(abs(summary(out)$coefficients[2,3]),  
+ out$df.residual))  
[1] 0
```

Using $\alpha = 0.05$

- Reject the null hypothesis $H_0 : \beta_1 = 0$ (p-value << 0.05)
- Fail to reject the null hypothesis: $H_0 : \beta_0 = 0$ (p-value >> 0.05)

F-test and ANOVA

An alternative way to test $\beta_1 = 0$ is based on the *F*-test. Recall the following decomposition of the variance: $TSS = FSS + RSS$.

Sum of Squares	Expression	df
TSS	$\sum_i (y_i - \bar{y})^2$	$n - 1$
FSS	$\sum_i (\hat{y}_i - \bar{y})^2$	1
RSS	$\sum_i (y_i - \hat{y}_i)^2$	$n - 2$

If $\beta_1 \neq 0$, we would expect a large amount of variation in Y is explained by the regression model, i.e., FSS is large. But how *large* is large? For the `cats` data, if we measure `Hwt` by kg, FSS will be much smaller, but whether `Bwt` is a good predictor for `Hwt` shouldn't be affected by the scale of `Hwt`.

Source	df	SS	MS	F
Regression	1	FSS	FSS/1	MS(reg)/MS(err)
Error	$n - 2$	RSS	$\text{RSS}/(n - 2)$	
Total	$n - 1$	TSS		

Under $H_0 : \beta_1 = 0$, the F -test statistic (scale-invariant)

$$F = \frac{\text{MS}(reg)}{\text{MS}(err)} = \frac{\text{FSS}}{\text{RSS}/(n - 2)} \sim F_{1,n-2}.$$

It can be shown that the F -test statistic is equal to the square of the t -test statistic (for testing $\beta_1 = 0$) and their p -values (for testing $\beta_1 = 0$) are the same. So they are essentially the same test; in other words, you can ignore the F -test in the R output for SLR.

ANOVA table for the *cats* example

Use function `anova`:

```
> anova(out)
```

Analysis of Variance Table

Response: Hwt

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Bwt	1	548.09	548.09	259.83	< 2.2e-16	***
Residuals	142	299.53	2.11			

Signif. codes:	0	'***'	0.001	'**'	0.01	'*'
	0.05	'.'	0.1			

F-test is equivalent to the square of the t-test for SLR:

```
> summary(out)$fstatistic  
      value    numdf    dendf  
259.8348    1.0000 142.0000  
> summary(out)$coefficients[2,3]^2  
[1] 259.8348
```

Estimation/Prediction at a New Case

The LS line can be used to obtain values of the response (Y^*) for given values of the predictor ($X = x^*$). There are two variants of this problem⁴:

- ① **Estimation:** We want to estimate the mean response at x^* . This is equivalent to estimate: $\beta_0 + \beta_1 x^*$
- ② **Prediction** of an outcome of random variable Y^* at a given value x^* , where

$$Y^* \sim N(\beta_0 + \beta_1 x^*, \sigma^2)$$

The fitted value (or point estimate) for estimation and prediction are the same: $\hat{\beta}_0 + \hat{\beta}_1 x^*$. However the accuracy for estimation and the one for prediction are different.

⁴Estimation looks to get information from the data about a fixed but parameter, while prediction looks to get information about a random variable

Measure of accuracy: expected value of the squared difference between the point estimate and the target.

- For estimation the target is $\beta_0 + \beta_1 x^*$:

$$\begin{aligned} & E[(\hat{\beta}_0 + \hat{\beta}_1 x^* - \beta_0 - \beta_1 x^*)^2] \\ &= Var(\hat{\beta}_0 + \hat{\beta}_1 x^*) \\ &= Var(\hat{\beta}_0) + x^{*2} Var(\hat{\beta}_1) + 2x^* Cov(\hat{\beta}_0, \hat{\beta}_1) \\ &= \sigma^2 \left(\frac{1}{n} + \frac{(x^* - \bar{x})}{S_{XX}} \right) \end{aligned}$$

- For prediction the target is $Y^* = \beta_0 + \beta_1 x^* + e^*$, where $e^* \sim N(0, \sigma^2)$. This new error e^* is independent of the previous n data points, ie. is independent of $(\hat{\beta}_0, \hat{\beta}_1)$

$$\begin{aligned}
 & E[(\hat{\beta}_0 + \hat{\beta}_1 x^* - Y^*)^2] \\
 &= E[(\hat{\beta}_0 + \hat{\beta}_1 x^* - \beta_0 - \beta_1 x^* - e^*)^2] \\
 &= E[(\hat{\beta}_0 + \hat{\beta}_1 x^* - \beta_0 - \beta_1 x^*)^2] + E[e^{*2}] \\
 &= \sigma^2 \left(1 + \frac{1}{n} + \frac{(x^* - \bar{x})^2}{S_{XX}} \right)
 \end{aligned}$$

- Error for estimation: $\sigma^2 \left(\frac{1}{n} + \frac{(x^* - \bar{x})}{S_{XX}} \right)$
- Error for prediction: $\sigma^2 \left(1 + \frac{1}{n} + \frac{(x^* - \bar{x})^2}{S_{XX}} \right)$
- Errors are not the same for all values of x^* . It is smaller when x^* is close to \bar{x} .
- Prediction error > estimation error.
- There are two sources of uncertainty when doing prediction at x^* :
 - ① the first is from the n sample points $(x_i, y_i) i = 1, \dots, n$ which are used to estimate the LS line.
 - ② the second is from the random error $e^* \sim N(0, \sigma^2)$, which is the error we cannot avoid even if we knew (β_0, β_1) . This is why, even when the sample size n goes to infinity, we can have the estimation error go to 0 but not the prediction error.

Confidence interval and prediction interval

- A confidence interval is always reported for a parameter, for example, $E[Y^*|x^*] = \beta_0 + \beta_1 x^*$. The $(1 - \alpha)100\%$ confidence interval for $\beta_0 + \beta_1 x^*$ is:

$$\hat{\beta}_0 + \hat{\beta}_1 x^* \pm t_{n-2}^{(1-\alpha)/2} \hat{\sigma} \sqrt{\frac{1}{n} + \frac{(x^* - \bar{x})^2}{S_{XX}}}$$

- A prediction interval is reported for the value of a random variable, for example, y^* . The $(1 - \alpha)100\%$ prediction interval for y^* is:

$$\hat{\beta}_0 + \hat{\beta}_1 x^* \pm t_{n-2}^{(1-\alpha)/2} \hat{\sigma} \sqrt{1 + \frac{1}{n} + \frac{(x^* - \bar{x})^2}{S_{XX}}}$$

Confidence and prediction interval of Hwt (in grs) for Bwt=2.15 kg

```
> predict(out, newdata=data.frame(Bwt = 2.15), se=TRUE,  
+ interval="confidence")$fit
```

	fit	lwr	upr
1	8.316572	7.945395	8.68775

```
> predict(out, newdata=data.frame(Bwt = 2.15), se=TRUE,  
+ interval="prediction")$fit
```

	fit	lwr	upr
1	8.316572	5.421611	11.21153

Association/Correlation vs. Causation

- The statement “X causes Y ” means that changing the value of X will change the distribution of Y . When X causes Y , X and Y will be associated but the reverse is not, in general, true. Association does not necessarily imply causation.
- If the data are from a **randomized study**, then the causal interpretation is correct
- If the data are from a **observational study**, then the association interpretation is correct.

STAT 425

Multiple Linear Regression. Part 1

Multiple Linear Regression Model

- In most applications we will want to use several predictors, instead of a single predictor as in simple linear regression (SLR).
- Now we have data of the form: $(y_i, \mathbf{x}_i)_{i=1}^n$, where $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^t$ with $x_{i1} = 1$
- Assume the model:

$$y_i = x_{i1}\beta_1 + x_{i2}\beta_2 + \dots + x_{ip}\beta_p + e_i$$

- $(\beta_1, \dots, \beta_p, \sigma^2)$: the unknown but true parameters.
- e_i 's: random errors

Main model assumptions:

- ① The mean function $E[y_i] = x_{i1}\beta_1 + x_{i2}\beta_2 \dots + x_{ip}\beta_p$ is linear in the p predictors.
- ② The errors e_i 's are uncorrelated with mean 0 and constant variance σ^2 . This is equivalent to: $E[e_i] = 0$ and $Cov(e_i, e_j) = \sigma^2\delta_{ij}$, with $\delta_{ij} = 0$ if $i \neq j$ and $\delta_{ij} = 1$ if $i = j$.
- ③ For hypothesis testing we further assume that e_i are i.i.d and $e_i \sim N(0, \sigma^2)$

Matrix representation of the MLR

MLR is valid for all observations for $i = 1, \dots, n$. We can write:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} x_{11}\beta_1 + x_{12}\beta_2 + \dots + x_{1p}\beta_p + e_1 \\ x_{21}\beta_1 + x_{22}\beta_2 + \dots + x_{2p}\beta_p + e_2 \\ \vdots \\ x_{n1}\beta_1 + x_{n2}\beta_2 + \dots + x_{np}\beta_p + e_n \end{pmatrix}$$
$$= \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}$$

$$\mathbf{y}_{n \times 1} = \mathbf{X}_{n \times p} \boldsymbol{\beta}_{p \times 1} + \mathbf{e}_{n \times 1}$$

Matrix \mathbf{X} is normally called the **design matrix**

Least Square Estimation

- Using matrix representation, we can express the MLR model as¹

$$\mathbf{y}_{n \times 1} = \mathbf{X}_{n \times p} \boldsymbol{\beta}_{p \times 1} + \mathbf{e}_{n \times 1}, \quad \mathbf{e} \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I}_n)$$

- The Least-Squares estimate of $\boldsymbol{\beta}$ is the vector that minimizes the Residual Sum of Squares (RSS):

$$RSS = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^t(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

¹By default the intercept is included in the model, then the 1st column of the design matrix \mathbf{X} is a vector of 1's. We further assume that the rank of \mathbf{X} is p , i.e., no columns of \mathbf{X} is a linear combination of the other columns and \mathbf{X} is a *tall and skinny matrix* ($n > p$).

Differentiating RSS with respect to β and setting to zero, we have

$$\begin{aligned}\frac{\partial \text{RSS}}{\partial \beta} &= -2\mathbf{X}_{p \times n}^t(\mathbf{y} - \mathbf{X}\beta)_{n \times 1} = \mathbf{0}_{p \times 1} \\ \implies \mathbf{X}^t(\mathbf{y} - \mathbf{X}\beta) &= \mathbf{0} \quad \text{normal equation} \\ \implies (\mathbf{X}^t\mathbf{X})\beta &= \mathbf{X}^t\mathbf{y} \\ \implies \hat{\beta} &= (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{y} \quad (*)\end{aligned}$$

Note that the inverse of the $p \times p$ matrix $(\mathbf{X}^t\mathbf{X})$ exists since we assume the rank of \mathbf{X} is p .

Next let's check the equation (*) for SLR.

$$\mathbf{X}^t \mathbf{X} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \end{pmatrix} \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \cdots & \cdots \\ 1 & x_n \end{pmatrix} = \begin{pmatrix} n & n\bar{x} \\ n\bar{x} & \sum x_i^2 \end{pmatrix}$$

$$(\mathbf{X}^t \mathbf{X})^{-1} = \frac{1}{n \sum x_i^2 - (n\bar{x})^2} \begin{pmatrix} \sum x_i^2 & -n\bar{x} \\ -n\bar{x} & n \end{pmatrix}$$

$$\mathbf{X}^t \mathbf{y} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{pmatrix} = \begin{pmatrix} n\bar{y} \\ \sum x_i y_i \end{pmatrix}$$

$$\begin{aligned}
 \hat{\beta} &= (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X} \mathbf{y} \\
 &= \frac{1}{n \sum x_i^2 - (n \bar{x})^2} \begin{pmatrix} \sum x_i^2 & -n \bar{x} \\ -n \bar{x} & n \end{pmatrix} \begin{pmatrix} n \bar{y} \\ \sum x_i y_i \end{pmatrix}
 \end{aligned}$$

So $\hat{\beta}_1$ is given by a

$$\hat{\beta}_1 = \frac{-n^2 \bar{x} \bar{y} + n \sum x_i y_i}{n \sum x_i^2 - (n \bar{x})^2} = \frac{\sum x_i y_i - n \bar{x} \bar{y}}{\sum x_i^2 - n \bar{x}^2} = \frac{S_{xy}}{S_{xx}}$$

Similarly we can check the calculation for $\hat{\beta}_0$.

^a $\sum (x_i - \bar{x})(y_i - \bar{y}) = \sum x_i y_i - n \bar{x} \bar{y}$ and $\sum (x_i - \bar{x})(x_i - \bar{x}) = \sum x_i^2 - n \bar{x}^2$.

Fitted values and Residuals

- Fitted values

$$\hat{\mathbf{y}}_{n \times 1} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{y} = \mathbf{H}_{n \times n}\mathbf{y}_{n \times 1}$$

$\mathbf{H}_{n \times n}$: is called the **hat** matrix, since it returns *y-hat*.

- Residuals

The estimated residuals are given by:

$$\mathbf{r}_{n \times 1} = \mathbf{y} - \hat{\mathbf{y}} = (\mathbf{I} - \mathbf{H})\mathbf{y}$$

- The residuals \mathbf{r} are used to estimate the **error variance**:

$$\hat{\sigma}^2 = \frac{1}{n-p} \sum_{i=1}^n r_i^2 = \frac{RSS}{n-p}$$

Note that the LS estimate $\hat{\beta}$ satisfies the normal equations:

$$\mathbf{X}^t(\mathbf{y} - \mathbf{X}\hat{\beta}) = \mathbf{0}$$

From this equation we can say that the residuals $\mathbf{r} = \mathbf{y} - \mathbf{X}\hat{\beta}$ satisfy:

- $\mathbf{X}^t\mathbf{r} = \mathbf{0}$. This implies that when we calculate the inner product of each column of matrix \mathbf{X} with the residual vector \mathbf{r} , this product is zero.
- In particular, when we include the intercept, the first equation implies that $\mathbf{1}^t\mathbf{r} = \sum_{i=1}^n r_i = 0$.
- The inner product $\hat{\mathbf{y}}^t\mathbf{r} = \hat{\beta}^t\mathbf{X}^t\mathbf{r} = 0$. This means that the residual vector is **orthogonal** to each column of \mathbf{X} and $\hat{\mathbf{y}}^t$

The Hat Matrix

The hat matrix is defined as:

$$\mathbf{H}_{n \times n} = \mathbf{X}(\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t$$

- Consider a linear combination of the columns of \mathbf{X} of the form $\mathbf{v} = \mathbf{X}\mathbf{a}_{p \times 1}$. The $\mathbf{H}\mathbf{v} = \mathbf{v}$, since:

$$\mathbf{H}\mathbf{X} = \mathbf{X}(\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t \mathbf{X} = \mathbf{X}$$

Properties of matrix \mathbf{H}

- Symmetric: $\mathbf{H}^t = [\mathbf{X}(\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t]^t = \mathbf{X}(\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t = \mathbf{H}$
- Idempotent²: $\mathbf{HH} = \mathbf{HH}^t = \mathbf{H}$
$$\mathbf{HH} = \mathbf{X}(\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{X}(\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t = \mathbf{X}(\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t = \mathbf{H}$$
- $trace(\mathbf{H}) = p$, the number of LS coefficients to be estimated.

²This property also implies that $\mathbf{H}(\mathbf{I} - \mathbf{H}) = \mathbf{0}$

Goodness of Fit: R-square

We use the R^2 to measure how well the model fits the data. R^2 is the fraction of the total variance explained by the model:

$$R^2 = \frac{\sum_i (\hat{y}_i - \bar{y})^2}{\sum_i (y_i - \bar{y})^2}$$

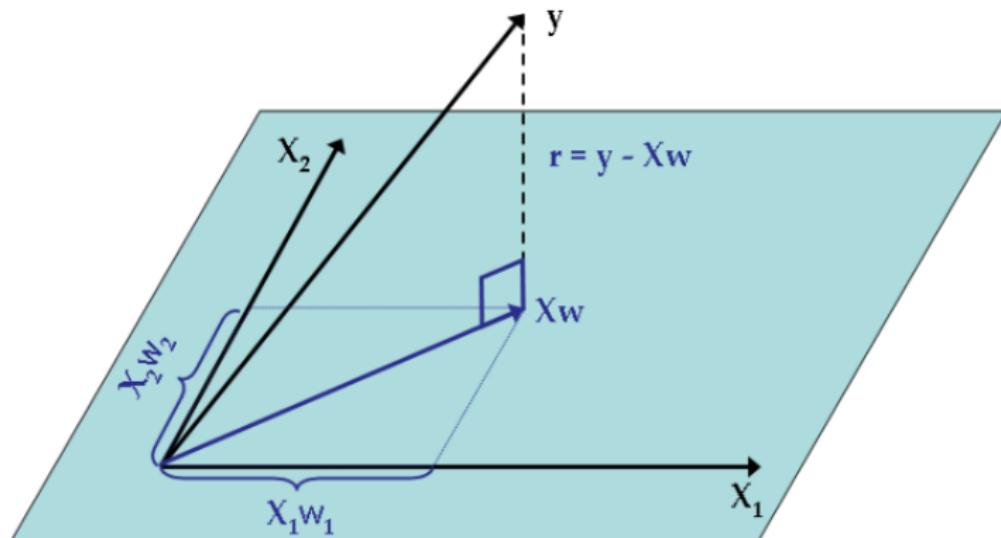
$$0 \leq R^2 \leq 1$$

This can also be written as:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} = 1 - \frac{RSS}{TSS}$$

Geometrical interpretation of the LS estimation

In \mathbb{R}^3 :



- All linear combinations $\mathbf{X}\mathbf{w}$ ($\mathbf{w} \in \mathbb{R}^p$) of the columns of matrix \mathbf{X} form a sub-space of dimension p in \mathbb{R}^n (denoted by $C(\mathbf{X})$). In the previous figure think about all the linear combinations of X_1 and X_2 .
- Finding $\hat{\boldsymbol{\beta}}$ that minimizes $||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2$ is equivalent to finding a vector $\hat{\mathbf{y}}$ from the estimation space that minimizes $||\mathbf{y} - \hat{\mathbf{y}}||^2$. From the figure it is intuitive that the **fitted value** is the **projection** of \mathbf{y} onto the estimation space.
- Matrix $\mathbf{H}_{n \times n}$ is the projection matrix:

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{y} = \mathbf{H}_{n \times n}\mathbf{y}$$

\mathbf{H} is symmetric, unique and idempotent, and the $trace(\mathbf{H}) = p$, which is the dimension of vector space $C(\mathbf{X})$.

- **Error space:** this sub-space of dimension $(n - p)$ is denoted by $C(\mathbf{X})^T$, and it is orthogonal to the estimation space. The matrix $(\mathbf{I}_n - \mathbf{H})$ is the projection matrix of the error space.
- **Residuals:** The estimated residuals can be calculated as:

$$\hat{\mathbf{e}} = \mathbf{r} = \mathbf{y} - \hat{\mathbf{y}} = (\mathbf{I}_n - \mathbf{H})\mathbf{y}$$

If the intercept is included in the model $\sum_i r_i = 0$. Due to the normal equation $\mathbf{X}^t(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \mathbf{0}$:

$$\sum_{i=1}^n r_i X_{ij} = 0 \quad \text{for } j = 1, \dots, p$$

Geometric Interpretation: \mathbf{r} is the projection of \mathbf{y} onto the error space orthogonal to $C(\mathbf{X})$. So \mathbf{r} is orthogonal to any vector in $C(\mathbf{X})$. Especially, \mathbf{r} is orthogonal to each column of \mathbf{X} .

An example

Savings Data set

##Savings rates in 50 countries The savings data frame has 50 rows and 5 columns. The data is averaged over the period 1960-1970. This data frame contains the following columns:

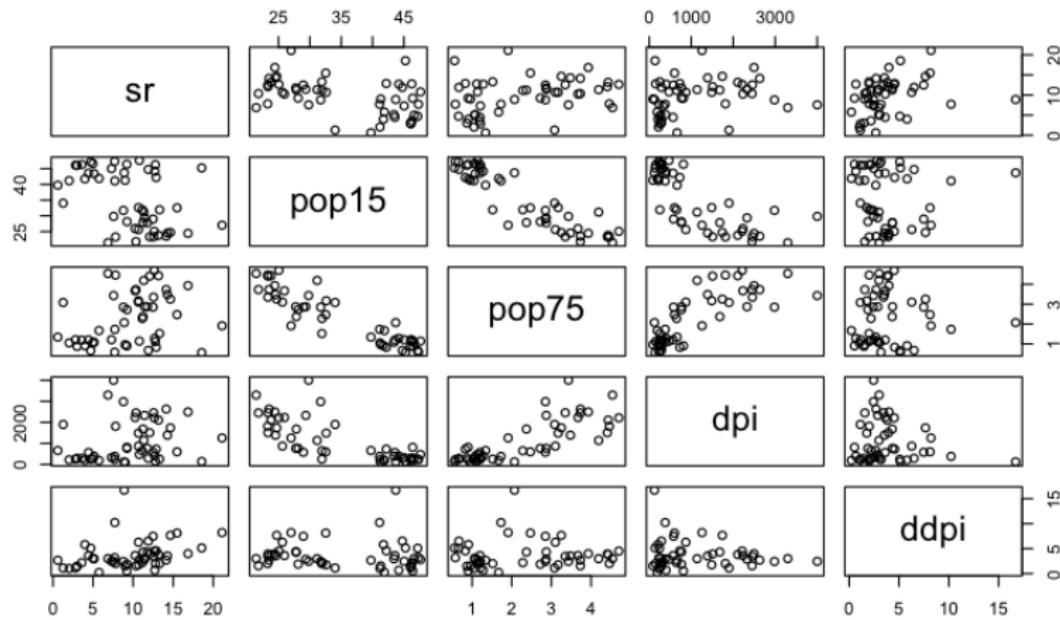
- sr – personal saving divided by disposable income
- pop15 – percent population under age of 15
- pop75 – percent population over age of 75
- dpi – per-capita disposable income in dollars
- ddpi – percent growth rate of dpi

```
library(faraway)
?savings
head(savings)
```

```
##           sr  pop15  pop75      dpi  ddpi
## Australia 11.43 29.35   2.87 2329.68 2.87
## Austria  12.07 23.32   4.41 1507.99 3.93
## Belgium  13.17 23.80   4.43 2108.47 3.82
## Bolivia   5.75 41.89   1.67  189.13 0.22
## Brazil   12.88 42.19   0.83  728.47 4.56
## Canada   8.79 31.72   2.85 2982.88 2.43
```

Plotting the data using the function **pairs(.)**

```
>pairs(saving)
```



Simple Linear Regression using function **lm**: $sr \sim pop75$

```
summary(lm(sr ~ pop75, data=savings))
```

```
##  
## Call:  
## lm(formula = sr ~ pop75, data = savings)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -9.2657 -3.2295  0.0543  2.3336 11.8498  
##  
## Coefficients:  
##                 Estimate Std. Error t value Pr(>|t|)  
## (Intercept)  7.1517     1.2475   5.733  6.4e-07 ***  
## pop75        1.0987     0.4753   2.312   0.0251 *  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 4.294 on 48 degrees of freedom  
## Multiple R-squared:  0.1002, Adjusted R-squared:  0.08144  
## F-statistic: 5.344 on 1 and 48 DF,  p-value: 0.02513
```

Multiple Linear Regression: $sr \sim pop15 + pop75 + dpi + ddpi$

```
fullmodel=lm(sr~pop15+pop75+dpi+ddpi, data=savings)
summary(fullmodel)

##
## Call:
## lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = savings)
##
## Residuals:
##      Min      1Q  Median      3Q     Max 
## -8.2422 -2.6857 -0.2488  2.4280  9.7509 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 28.5660865  7.3545161   3.884 0.000334 ***
## pop15       -0.4611931  0.1446422  -3.189 0.002603 **  
## pop75       -1.6914977  1.0835989  -1.561 0.125530    
## dpi        -0.0003369  0.0009311  -0.362 0.719173    
## ddpi        0.4096949  0.1961971   2.088 0.042471 *   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.803 on 45 degrees of freedom
## Multiple R-squared:  0.3385, Adjusted R-squared:  0.2797 
## F-statistic: 5.756 on 4 and 45 DF,  p-value: 0.0007904
```

Contradictory results for the estimated $\hat{\beta}_{pop75}$? Some predictors might be highly correlated:

```
# Lets look at the correlation matrix
cor(savings[,-1])

##          pop15      pop75       dpi      ddpi
## pop15  1.00000000 -0.90847871 -0.7561881 -0.04782569
## pop75 -0.90847871  1.00000000  0.7869995  0.02532138
## dpi   -0.75618810  0.78699951  1.0000000 -0.12948552
## ddpi  -0.04782569  0.02532138 -0.1294855  1.00000000
```

This correlation might cause contradictory results, with some regression coefficients having an unexpected sign.

Rank deficiency

- The design matrix \mathbf{X} is an $n \times p$ matrix³. If this matrix is not of full rank (i.e., its columns are not linearly independent), the matrix $\mathbf{X}^t\mathbf{X}$ can not be inverted (singular matrix).
- If the matrix $\mathbf{X}^t\mathbf{X}$ is singular the LS solutions is not unique (identifiability problem)
- **R** can cope well with this problem. To solve the LS equations **R** uses the [QR decomposition](#). You can read more on this in the supplemental material.

³You can use function `model.matrix()` in **R** to extract the model matrix of a fitted model

STAT 425

Multiple Linear Regression. Part 2

Properties of the Least-Square estimates

- In MLR the LS estimate $\hat{\beta}$ is a random vector, since it is a function of y
- For hypothesis testing we want to find the probability distribution of $\hat{\beta}$
- We will review the definitions of the mean and variance of a random vector first, and how to calculate mean and variances of affine transformations of this random vector.

Review: Mean and Variances of Random Vectors

Mean of a Random Vector

Let \mathbf{Z} a random vector of size $m \times 1$, with components

Z_1, Z_2, \dots, Z_m . The mean of \mathbf{Z} is equal to vector $\boldsymbol{\mu}$ defined as:

$$\boldsymbol{\mu} = E[\mathbf{Z}] = \begin{pmatrix} E[Z_1] \\ E[Z_2] \\ \dots \\ E[Z_m] \end{pmatrix}$$

Variance of a Random Vector

The Variance of a random vector \mathbf{Z} is a matrix (the Variance-Covariance matrix). This matrix is symmetric of size $m \times m$ with component (i, j) equal to the $Cov(Z_i, Z_j)$

$$\begin{aligned}\Sigma_{m \times m} = Cov(\mathbf{Z}) &= E[(\mathbf{Z} - \boldsymbol{\mu})(\mathbf{Z} - \boldsymbol{\mu})^t] \\ &= \begin{pmatrix} Var(Z_1) & \dots & Cov(Z_1, Z_m) \\ \dots & \dots & \dots \\ Cov(Z_m, Z_1) & \dots & Var(Z_m) \end{pmatrix}\end{aligned}$$

Mean and Covariance matrix of an affine transformation

Assume an affine transformation of the form:

$$\mathbf{W} = \mathbf{a}_{n \times 1} + \mathbf{B}_{n \times m} \mathbf{Z}_{m \times 1}$$

$$E[\mathbf{W}] = \mathbf{a} + \mathbf{B}\boldsymbol{\mu}, \quad Cov(\mathbf{W}) = \mathbf{B}\boldsymbol{\Sigma}\mathbf{B}^t$$

In particular consider a transformation of the form:

$$W = \mathbf{v}^t \mathbf{Z} = v_1 Z_1 + v_2 Z_2 + \dots + v_m Z_m$$

$$E[W] = \mathbf{v}^t \boldsymbol{\mu} = \sum_{i=1}^m v_i \mu_i$$

$$Var(W) = \mathbf{v}^t \boldsymbol{\Sigma} \mathbf{v} = \sum_{i=1}^m v_i^2 Var(Z_i) + 2 \sum_{i < j} v_i v_j Cov(Z_i, Z_j)$$

Mean and Covariance of the LS estimates

We use the assumption: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}$ with

$$E[\mathbf{e}] = \mathbf{0}, \quad Cov(\mathbf{e}) = \sigma^2 \mathbf{I}_n$$

Using these assumptions we get:

$$E[\mathbf{y}] = \mathbf{X}\boldsymbol{\beta}, \quad Cov(\mathbf{y}) = \sigma^2 \mathbf{I}_n$$

Under these assumptions we also get:

$$\begin{aligned} E[\hat{\boldsymbol{\beta}}] &= E[(\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t \mathbf{y}] \\ &= (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t E[\mathbf{y}] \\ &= (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t \mathbf{X} \boldsymbol{\beta} = \boldsymbol{\beta} \end{aligned}$$

We have shown that the LS estimate is **unbiased**.

Mean and Covariance of the LS estimates (Cont.)

$$\begin{aligned}\text{Cov}(\hat{\beta}) &= (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t \text{Cov}(\mathbf{y}) [(\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t]^t \\ &= (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t \sigma^2 \mathbf{X} (\mathbf{X}^t \mathbf{X})^{-1} \\ &= \sigma^2 (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t \mathbf{X} (\mathbf{X}^t \mathbf{X})^{-1} = \sigma^2 (\mathbf{X}^t \mathbf{X})^{-1};\end{aligned}$$

Using the previous results we can also show:

$$E[\hat{\mathbf{y}}] = \mathbf{X}\boldsymbol{\beta}, \quad Cov(\hat{\mathbf{y}}) = \sigma^2 \mathbf{H}$$

$$E[\mathbf{r}] = \mathbf{0}, \quad Cov(\mathbf{r}) = \sigma^2 (\mathbf{I}_n - \mathbf{H})$$

$$E[\hat{\sigma}^2] = \frac{1}{n-p} E[\mathbf{r}^t \mathbf{r}] = \frac{1}{n-p} \sigma^2 (n-p) = \sigma^2$$

1

¹It can be shown that $\frac{\mathbf{r}^t \mathbf{r}}{\sigma^2} = \frac{RSS}{\sigma^2} \sim \chi_{n-p}^2$

- $\hat{\beta}$ and $\hat{\sigma}^2$ are unbiased estimators of β and σ^2 respectively
- We can plug-in the variance estimator $\hat{\sigma}^2$ to get the covariance of $\hat{\beta}$
- The **standard errors** of the $\hat{\beta}_i$ are the square roots of the elements of the diagonal of the covariance matrix
 $Cov(\hat{\beta}) = \hat{\sigma}^2(\mathbf{X}^t\mathbf{X})^{-1}$. For example:

$$se(\hat{\beta}_1) = \hat{\sigma} \sqrt{[(\mathbf{X}^t\mathbf{X})^{-1}]_{11}}$$

The Gauss-Markov Theorem

The main reason why we use LS estimation is because of the [Gauss-Markov theorem](#). If the errors are uncorrelated, have equal variance and mean equal to zero, the LS estimators have a very nice property: they have the lowest variance within the class of linear estimators.

- Suppose we are interested in estimating a linear combination of β of the form:

$$\theta = \mathbf{c}^t \beta = \sum_{j=1}^p c_j \beta_j$$

For example, estimating any element of β and estimating the mean response at a new value x^* are all special cases of this setup.

The Gauss-Markov Theorem (Cont.)

- Naturally, we can form an estimate of θ by plugging in the LS estimate $\hat{\beta}$ in the equation for θ :

$$\hat{\theta}_{LS} = \mathbf{c}^t \hat{\beta} = \mathbf{c}^t (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t \mathbf{y}$$

This is a **linear**² and **unbiased estimator** of θ . Its mean square error can be calculated as:

$$MSE(\hat{\theta}_{LS}) = E[\hat{\theta}_{LS} - \theta]^2 = Var(\hat{\theta}_{LS})$$

²It is a linear combination of the n data points y_1, y_2, \dots, y_n

Gauss-Markov theorem (Cont.)

- Suppose there is another estimate of θ , which is also linear and unbiased. The following Theorem states that $\hat{\theta}_{LS}$ is always better in the sense that its MSE is always smaller (or at least, not bigger)
- **Gauss-Markov Theorem:** The estimator $\hat{\theta}_{LS} = \mathbf{c}^t \hat{\boldsymbol{\beta}}$ is the **BLUE** (best linear unbiased estimator) of the parameter $\mathbf{c}^t \boldsymbol{\beta}$ for any vector $\mathbf{c} \in \mathbb{R}^p$.

Proof: Please see Supplemental Material.

Maximum Likelihood Estimation

Recall the normal assumptions for the regression model:

$$y_i = \mathbf{x}_i^t \boldsymbol{\beta} + e_i \quad (i = 1, \dots, n)$$

with $e_i \sim N(0, \sigma^2)$. This implies:

$$\mathbf{y} \sim \mathbf{N}_{\mathbf{n}}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I})$$

We can show that the likelihood function can be written as:

$$L(\boldsymbol{\beta}, \sigma^2 | \mathbf{y}) \propto \frac{RSS^{-\frac{n}{2}}}{n}$$

The value of $\boldsymbol{\beta}$ that maximizes the Likelihood function is the Maximum Likelihood Estimator (MLE) of $\boldsymbol{\beta}$. This estimator is equal to the LS estimate of $\boldsymbol{\beta}$.

Distributions of the Least-Squares estimates

Recall the assumption for the linear regression model:

$$\mathbf{y} \sim \mathbf{N}_{\mathbf{n}}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I})$$

Any affine transformation of \mathbf{y} will also have a Normal distribution³. We can use the identities to calculate the mean and variance of an affine transformation of a random vector to get the following results:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t \mathbf{y} \sim \mathbf{N}_{\mathbf{p}}(\boldsymbol{\beta}, \sigma^2 (\mathbf{X}^t \mathbf{X})^{-1})$$

$$\hat{\mathbf{y}} = \mathbf{H}\mathbf{y} \sim \mathbf{N}_{\mathbf{n}}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{H})$$

$$\mathbf{r} = (\mathbf{I}_n - \mathbf{H})\mathbf{y} \sim \mathbf{N}_{\mathbf{n}}(\mathbf{0}, \sigma^2 (\mathbf{I}_n - \mathbf{H}))$$

³They will also have a **joint** Normal distribution

Note that for the fitted values $\hat{\mathbf{y}}$ and the estimated residuals $\hat{\mathbf{e}} = \mathbf{r}$ we can calculate the mean and covariance matrices as follows:

$$E[\hat{\mathbf{y}}] = \mathbf{H}E[\mathbf{y}] = \mathbf{H}\mathbf{X}\boldsymbol{\beta} = \mathbf{X}\boldsymbol{\beta}$$

$$Cov(\hat{\mathbf{y}}) = \mathbf{H}\sigma^2\mathbf{H}^t = \sigma^2\mathbf{H}$$

$$E[\mathbf{r}] = (\mathbf{I}_n - \mathbf{H})\mathbf{X}\boldsymbol{\beta} = \mathbf{0}$$

$$Cov(\mathbf{r}) = (\mathbf{I}_n - \mathbf{H})\sigma^2(\mathbf{I}_n - \mathbf{H})^t = \sigma^2(\mathbf{I}_n - \mathbf{H})$$

- Although \mathbf{r} is a vector of dimension n , it always lies in a subspace of dimension $(n - p)$.
- \mathbf{r} behaves like a random vector with a distribution $\mathbf{N}_{n-p}(\mathbf{0}, \sigma^2 \mathbf{I}_{n-p})$, so we have:

$$\hat{\sigma}^2 = \frac{||\mathbf{r}||^2}{n - p} \sim \sigma^2 \frac{\chi_{n-p}^2}{n - p}$$

- It can be shown that $\hat{\mathbf{y}}$ and \mathbf{r} are uncorrelated since they are in orthogonal spaces. Since they have a joint normal distribution, they are independent.⁴

⁴Note that if two random variables are uncorrelated, they are not necessarily independent, unless they have a joint Normal distribution

STAT 425

Multiple Linear Regression. Part 3

Hypothesis testing in MLR

Testing a single predictor: Suppose you have a certain number of predictors in your regression model and you want to test the hypothesis¹:

$$H_0 : \beta_j = c \text{ vs. } H_a : \beta_j \neq c$$

- We use the t-test statistic:

$$t = \frac{\hat{\beta}_j - c}{se(\hat{\beta}_j)} = \frac{\hat{\beta}_j - c}{\hat{\sigma} \sqrt{[(\mathbf{X}^t \mathbf{X})^{-1}]_{jj}}} \sim T_{n-p}$$

under the null hypothesis H_0

- p-value = $2 \times$ the area under the curve of a T_{n-p} distribution **more extreme** than the observed statistic.
- The p-value returned by the `lm` function command is for $c = 0$.

¹The test result might vary depending on which other predictors are included in the model

Different t-tests

We've learned various t -tests in class and each seems to have a different degree of freedom. How can I find out the correct df for a t -test?

All t -tests we've encountered so far involve an estimate of the error variance σ^2 . The df of a t -test is determined by the denominator of $\hat{\sigma}^2$.

- $Z_1, \dots, Z_n \sim N(\theta, \sigma^2)$. To test $\theta = a$, we have

$$\frac{\hat{\theta} - a}{\text{se}(\hat{\theta})} = \frac{\bar{Z} - a}{\sqrt{\hat{\sigma}^2/n}} \sim T_{n-1}, \quad \hat{\sigma}^2 = \frac{\sum_i (Z_i - \bar{Z})^2}{n-1}.$$

- For SLR, to test $\beta_1 = c$, we have

$$\frac{\hat{\beta}_1 - c}{\text{se}(\hat{\beta}_1)} = \frac{\hat{\beta}_1 - c}{\hat{\sigma}/\sqrt{S_{xx}}} \sim T_{n-2}, \quad \hat{\sigma}^2 = \frac{\text{RSS}}{n-2}.$$

- For MLR with p predictors (including the intercept), to test $\beta_j = c$,

$$\frac{\hat{\beta}_j - c}{\text{se}(\hat{\beta}_j)} = \frac{\hat{\beta}_j - c}{\hat{\sigma}[(\mathbf{X}^t \mathbf{X})^{-1}]_{jj}} \sim T_{n-p}, \quad \hat{\sigma}^2 = \frac{\text{RSS}}{n-p}.$$

F-test and ANOVA table

Testing all predictors: Suppose we want to test the hypothesis:

$$H_0 : \beta_2 = \beta_3 = \dots = \beta_p = 0 \quad vs. \quad H_a : \beta_j \neq 0$$

for some j , $j = 2, \dots, p$.

Under the Null hypothesis, the test statistic:

$$F = \frac{MS(Reg)}{MS(Error)} \sim F_{p-1, n-p}$$

All *F-test* components can be organized in the ANOVA table, where
 $TSS = FSS + RSS$

ANOVA Table for the overall F -test

Source	df	SS	MS	F
Regression	$p - 1$	FSS	$FSS/(p - 1)$	$MS(\text{reg})/MS(\text{err})$
Error	$n - p$	RSS	$RSS/(n - p)$	
Total	$n - 1$	TSS		

Savings example

- Suppose we start our analysis with the **full model**:

$$y_i = \beta_1 + \beta_2 pop15_i + \beta_3 pop75_i + \beta_4 dpi_i + \beta_5 ddpi_i + e_i$$

- We want to test the hypothesis that *saving* is independent of age
- We fit a **reduced model**. This implies to remove the columns corresponding to variables *pop15* and *pop75* from the design matrix:

$$y_i = \beta_1 + \beta_4 dpi_i + \beta_5 ddpi_i + e_i$$

- How can we compare the results from the two fitted models?

Savings example (Cont.)

We want to test the hypothesis:

H_0 : The reduced model is adequate (age is not needed)

H_a : The full model is required

Under H_0 we assume that the following model is correct:

$$y_j = \beta_1 + \beta_4 dpi + \beta_5 ddpi + e_i$$

We consider the following partition of the design matrix into two sub-matrices \mathbf{X}_1 and \mathbf{X}_2 :

$$\mathbf{X}_{n \times p} = (\mathbf{X}_{1n \times (p-q)}, \mathbf{X}_{2n \times q})$$

The corresponding partition of the regression parameter is:

$$\boldsymbol{\beta}^t = (\boldsymbol{\beta}_1^t, \boldsymbol{\beta}_2^t)$$

where $\boldsymbol{\beta}_1$ is $(p - q) \times 1$ and $\boldsymbol{\beta}_2$ is $q \times 1$. This partition is used to test the hypothesis:

$$H_0 : \boldsymbol{\beta}_2 = \mathbf{0}, \text{ i.e., } \mathbf{y} = \mathbf{X}_1 \boldsymbol{\beta}_1 + \mathbf{error}$$

$$H_a : \boldsymbol{\beta}_2 \neq \mathbf{0}, \text{ i.e., } \mathbf{y} = \mathbf{X}_1 \boldsymbol{\beta}_1 + \mathbf{X}_2 \boldsymbol{\beta}_2 + \mathbf{error}$$

Partial F test

We use the test statistic:

$$F = \frac{(RSS_0 - RSS_a)/q}{RSS_a/(n-p)} \sim F_{q,n-p}$$

where RSS_0 = Residual sum of squares for the model under H_0 ;
 RSS_a = Residual sum of squares for the model under H_a .

- **Numerator:** variation in the data not explained by the reduced model, but explained by the full model.
- **Denominator:** variation in the data not explained by the full model (i.e., not explained by either model), which is used to estimate the error variance.
- Reject H_0 , if F -stat is large, that is, the variation missed by the reduced model, when being compared with the error variance, is significantly large.

Partial F test calculation using the **anova(.)** function

```
anova(reducedmodel, fullmodel)
```

```
## Analysis of Variance Table
##
## Model 1: sr ~ dpi + ddpi
## Model 2: sr ~ pop15 + pop75 + dpi + ddpi
##   Res.Df   RSS Df Sum of Sq    F    Pr(>F)
## 1     47 824.72
## 2     45 650.71  2      174 6.0167 0.004835 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We reject the null hypothesis that the reduced model is correct.

Partial F test calculation using summary outputs from the two

```
#  
rss.full=sum(fullmodel$res^2)  
# rss.full=deviance(fullmodel) # you can use "deviance" to extract RSS  
rss.reduced=sum(reducedmodel$res^2)  
#rss.reduced = deviance(reducedmodel)  
Fstat=(rss.reduced-rss.full)/2/(rss.full/45)  
Fstat
```

```
## [1] 6.016652
```

```
1-pf(Fstat, 2, 45)
```

```
## [1] 0.004834923
```

models

Examples of F-tests

- Example 1: Testing all predictors (The default F-test returned by the function `lm(.)`):

$$H_0 : \mathbf{y} = \mathbf{1}_n \boldsymbol{\alpha} + \text{error}$$

$$H_a : \mathbf{y} = \mathbf{X}_{n \times p} \boldsymbol{\beta} + \text{error}$$

- Example 2: Testing one-predictor (the F-test that is equivalent to the t-test ($H_0 : \beta_j = 0$)):

$$H_0 : \mathbf{y} = \mathbf{X}[, -\mathbf{j}]_{n \times (p-1)} \boldsymbol{\alpha} + \text{error}$$

$$H_a : \mathbf{y} = \mathbf{X}_{n \times p} \boldsymbol{\beta} + \text{error}$$

where $\mathbf{X}[, -\mathbf{j}] = \mathbf{X}$ without the j -th column, and $\boldsymbol{\alpha}$ is $(p-1) \times 1$

Examples of F-tests (Cont.)

- Example 3: Testing a subset of predictors:

$$H_0 : \mathbf{y} = \mathbf{X}_1 \boldsymbol{\beta}_1 + \text{error}$$

$$H_a : \mathbf{y} = \mathbf{X}_1 \boldsymbol{\beta}_1 + \mathbf{X}_2 \boldsymbol{\beta}_2 + \text{error}$$

where $(\mathbf{X}_1, \mathbf{X}_2)$ is a partition of matrix \mathbf{X}

- Example 4: Testing a sub-space (For example $H_0 : \beta_2 = \beta_3$)

$$H_0 : \mathbf{y} = \mathbf{X}_1 \boldsymbol{\alpha} + \text{error}$$

$$H_a : \mathbf{y} = \mathbf{X} \boldsymbol{\beta} + \text{error}$$

where \mathbf{X}_1 is a $n \times (p - 1)$ matrix that is almost the same as \mathbf{X} , but replaces the 2nd and 3rd columns of \mathbf{X} by their sum, and $\boldsymbol{\alpha}$ is $(p - 1) \times 1$.

STAT 425

Multiple Linear Regression. Part 4

Confidence Intervals for the β_j 's

- A $(1 - \alpha)$ CI for β_j is given by

$$\left(\hat{\beta}_j \pm t_{n-p}^{(\alpha/2)} \text{se}(\hat{\beta}_j) \right) = \left(\hat{\beta}_j \pm t_{n-p}^{(\alpha/2)} \hat{\sigma} \sqrt{[(\mathbf{X}^t \mathbf{X})^{-1}]_{jj}} \right)$$

where $t_{n-p}^{(\alpha/2)}$ is the $(1 - \alpha/2)$ percentile of the student T-dist with $(n - p)$ degree-of-freedom.

In R we can use the function `confint(.)`

Use the command `confint` to obtain confidence intervals for regression coefficients.

```
confint(fullmodel)
```

```
##                      2.5 %      97.5 %
## (Intercept) 13.753330728 43.378842354
## pop15        -0.752517542 -0.169868752
## pop75        -3.873977955  0.490982602
## dpi          -0.002212248  0.001538444
## ddpi         0.014533628  0.804856227
```

```
confint(fullmodel, 'pop15', level=0.99)
```

```
##                      0.5 %      99.5 %
## pop15 -0.8502207 -0.07216559
```

Confidence Region

Just as we can use estimated standard errors and t-stats to form confidence intervals for a **single parameter**, we can also obtain a $(1 - \alpha) \times 100\%$ confidence region for the entire vector β . In particular:

$$\beta - \hat{\beta} \sim N(\mathbf{0}, \sigma^2 (\mathbf{X}^t \mathbf{X})^{-1})$$

Thus, the quadratic form:

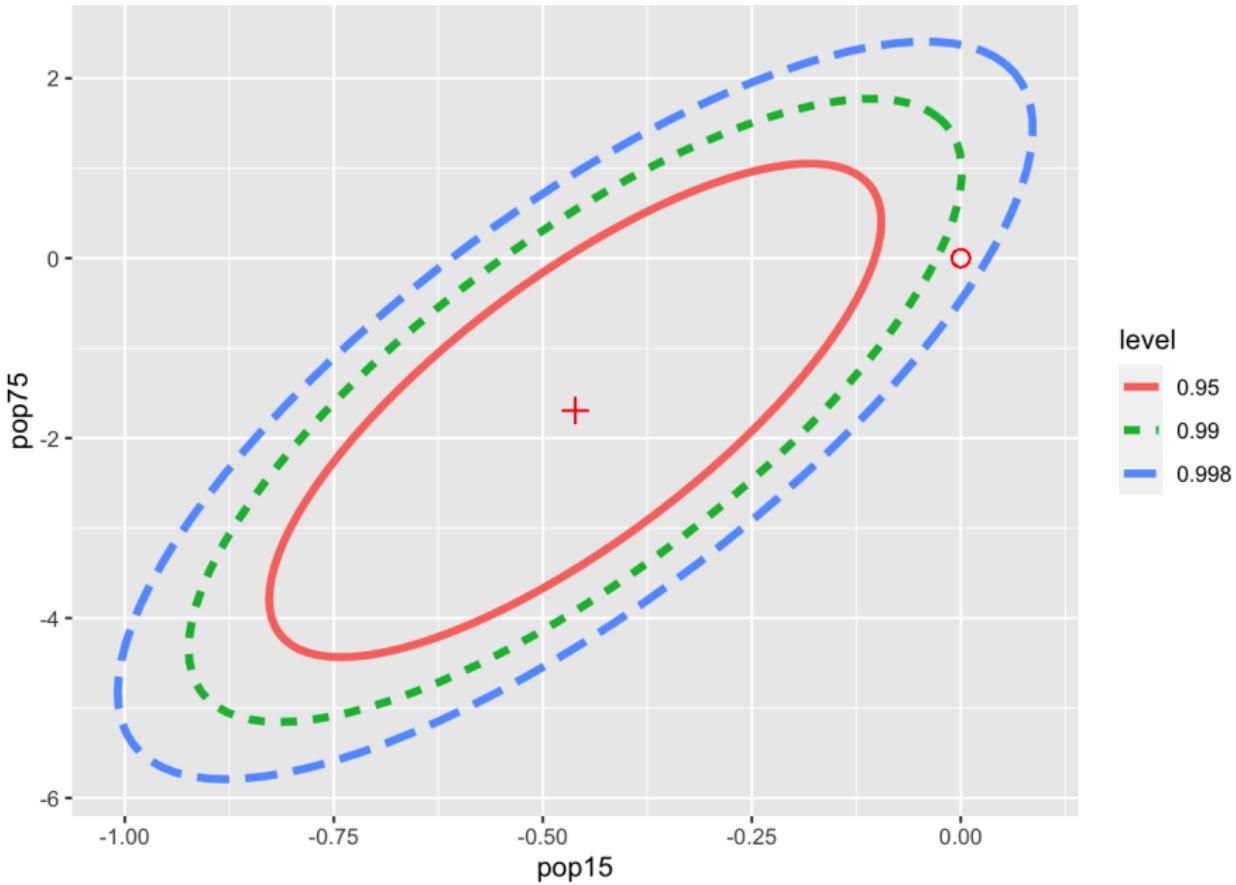
$$\frac{(\beta - \hat{\beta})^t \mathbf{X}^t \mathbf{X} (\beta - \hat{\beta})}{p \hat{\sigma}^2} \sim F_{p, n-p}$$

Then we can construct a $(1 - \alpha) \times 100\%$ confidence region for β to be all the points in the following ellipsoid

$$\frac{(\beta - \hat{\beta})^t \mathbf{X}^t \mathbf{X} (\beta - \hat{\beta})}{p \hat{\sigma}^2} < F(\alpha; p, n - p)$$

where $F(\alpha; p, n - p)$ is defined to be the point such that:

$$Pr[F_{p,n-p} > F(\alpha; p, n - p)] = \alpha$$



Confidence Interval and Prediction Interval at a future observation \mathbf{x}^*

- We are interested in obtaining an estimate $E[Y|\mathbf{x}^*] = \mu^* = (\mathbf{x}^*)^t \boldsymbol{\beta}$ and a prediction for a future observation Y^* at \mathbf{x}^* . We also want to have a CI for μ^* and a PI for y^* .
- The Gauss-Markov theorem tells us that the BLUE (Best Linear Unbiased Estimate) of μ^* is:

$$\hat{\mu}^* = (\mathbf{x}^*)^t \hat{\boldsymbol{\beta}} = (\mathbf{x}^*)^t (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t \mathbf{y}$$

This is just a linear transformation of \mathbf{y} , so we can easily derive its variance, and find its standard error.

It can be shown that:

$$se(\hat{\mu}^*) = \hat{\sigma} \sqrt{(\mathbf{x}^*)^t (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{x}^*}$$

- A Confidence Interval for μ^* is given by:

$$(\hat{\mu}^* - t_{n-p}^{(\alpha/2)} se(\hat{\mu}^*), \hat{\mu}^* + t_{n-p}^{(\alpha/2)} se(\hat{\mu}^*))$$

- The best estimate for y^* at a future observation \mathbf{x}^* is also

$$\hat{y}^* = (\mathbf{x}^*)^t \hat{\boldsymbol{\beta}}$$

- In order to find a prediction interval (PI), we need to consider the variance due to $\hat{\boldsymbol{\beta}}$ in addition to the variance associated with a new observation, which is σ^2 .
- The standard error of a prediction estimate \hat{y}^* is:¹

$$se(\hat{y}^*) = \hat{\sigma} \sqrt{1 + (\mathbf{x}^*)^t (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{x}^*}$$

- A $(1 - \alpha)100\%$ PI for a new observation Y^* at \mathbf{x}^* is given by:

$$(\hat{y}^* - t_{n-p}^{(\alpha/2)} se(\hat{y}^*), \hat{y}^* + t_{n-p}^{(\alpha/2)} se(\hat{y}^*))$$

¹Note that no matter how large the sample size becomes, the width of a PI, unlike a CI, will never approach 0.

```
# create a data frame on which you'd like to predict  
meanvalue=apply(savings[,2:5],2,mean)  
meanvalue
```

```
##      pop15      pop75      dpi      ddpi  
## 35.0896  2.2930 1106.7584   3.7576
```

```
x=data.frame(t(meanvalue))  
predict.lm(fullmodel,x,interval="confidence")
```

```
##      fit      lwr      upr  
## 1 9.671 8.587858 10.75414
```

```
predict.lm(fullmodel,x,interval="prediction")
```

```
##      fit      lwr      upr  
## 1 9.671 1.935822 17.40618
```

Simultaneous CIs and PIs

- Consider a simple linear regression $y_i = \beta_0 + \beta_1 x_i + e_i$
- Given the values of x^* , the $(1 - \alpha)100\%$ Confidence Interval for $\mu^* = E[y|x^*] = \beta_0 + \beta_1 x_i$ is:

$$I(x^*) = (\hat{\mu}^* \pm t_{n-2}^{(\alpha/2)} se(\hat{\mu}^*)) \quad (1)$$

where

$$\hat{\mu}^* = \hat{\beta}_0 + \hat{\beta}_1 x^* \text{ and } se(\hat{\mu}^*) = \hat{\sigma} \sqrt{\frac{1}{n} + \frac{(x^* - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}$$

- If we want CIs at multiple points $(x_1^*, x_2^*, \dots, x_m^*)$, we can use formula (1) to have CIs at the m points:

$$I(x_1^*), I(x_2^*), \dots, I_m(x^*)$$

Bonferroni Correction

- We know that:

$$Pr[\mu_i^* \in I(x_i^*)] = (1 - \alpha)$$

This is the **point-wise** coverage probability for μ_i^* and formula (1) gives the **point-wise** CI.

- What about the **simultaneous** coverage probability? i.e.:

$$Pr[\mu_i^* \in I(x_i^*), \text{ for } i = 1, \dots, m] = ?$$

- To make sure that (for example):

$$Pr[\mu_i^* \in I(x_i^*), \text{ for } i = 1, \dots, m] = .95$$

we need to set $\alpha = 5\% / m$, which is known as the **Bonferroni correction**

Bonferroni Correction

Let A_k denotes the event that the k th confidence interval covers μ_k^* with:

$$Pr(A_k) = (1 - \alpha)$$

Then we can show:

$$\begin{aligned} & Pr(\text{All CIs cover the corresponding } \mu_k^* \text{ values}) \\ &= Pr(A_1 \cap A_2 \dots \cap A_m) \\ &= 1 - Pr(A_1^c \cup A_2^c \dots \cup A_m^c) \\ &\geq 1 - Pr(A_1^c) - \dots - Pr(A_m^c) \\ &= 1 - m\alpha \end{aligned}$$

If we choose α/m instead of α , the simultaneous coverage probability will be $(1 - \alpha)$.

STAT 425

Model Diagnostics. Part 1

Regression model diagnostics

Regression Model assumptions:

- $\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_n)$
- Error: assumed to be iid, $e_i \sim N(0, \sigma^2)$
- Model: assumed to be linear in the parameters, i.e., $E[\mathbf{y}] = \mathbf{X}\boldsymbol{\beta}$
- We might have unusual observations.
- We will use both, graphical and numerical tools for diagnosis.

Finding unusual observations

Why discuss unusual observations first?

- Least squares regression is very sensitive to individual data points. (Later this will motivate discussion of robust regression procedures.)
- It is possible the inference, p -values, parameter estimation, CI's are all driven by a single data point.
- Sometimes, the estimated parameters and other related statistics (such as R^2) depend heavily on one observation, in the sense that if that observation was removed, the result of the analysis would change.

Types of unusual observations

- **High leverage points:** We will define some measure called “leverage” which quantifies how far a data point is from the center of the whole sample (remember the Mahalanobis distance?). Points with a large value of leverage are flagged as the high leverage points. High leverage points could be “good” or “bad”.
- **Outliers:** data point that does not fit the model as the other data points. We will introduce a formal testing procedure to identify outliers.
- **Highly influential points:** How does each individual observation affect the estimation of the model? We will define some measure, “Cook’s distance”, to quantify the aforementioned change for each data point, and data points with a large value of Cook’s distance are called high influential points.

Leverage

- The diagonal elements of \mathbf{H} ,

$$h_i = H_{ii}$$

are called **leverages** and are very useful diagnostics.

- h_i gives a measure (invariant under any affine transformation of \mathbf{X}) of how far the i -th observation is from the center of the data (in the X -space). This measure also arises in our discussion on the width of CI and standard error of prediction/estimation at \mathbf{x}_i .

- For simple regression:

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_i (x_i - \bar{x})^2}$$

- In general:

$$h_i = \mathbf{x}_i^t (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{x}_i \quad (1)$$

$$= \frac{1}{n} + \frac{1}{n-1} (\mathbf{z}_i - \bar{\mathbf{z}})^t \hat{\Sigma}^{-1} (\mathbf{z}_i - \bar{\mathbf{z}}) \quad (2)$$

where $\hat{\Sigma}_{(p-1) \times (p-1)} = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{z}_i - \bar{\mathbf{z}})(\mathbf{z}_i - \bar{\mathbf{z}})^t$ is the sample covariance of the $(p-1)$ predictor variables. The second term in the right hand side of (2) is the so-called Mahalanobis distance from \mathbf{z}_i to the data center $\bar{\mathbf{z}}$

Properties of the leverage

The two following properties of \mathbf{H} :

$$\text{tr}(\mathbf{H}) = p, \quad \mathbf{H} = \mathbf{H}\mathbf{H}^\top$$

imply that $\sum_i h_i = p$ and $\sum_j H_{ij}^2 = h_i$. For a given i we can decompose the last sum as follows:

$$\sum_j H_{ij}^2 = H_{ii}^2 + \sum_{i \neq j} H_{ij}^2 = h_i$$

$$\Rightarrow \sum_{i \neq j} H_{ij}^2 = h_i(1 - h_i) \Rightarrow h_i(1 - h_i) > 0$$

From this we can conclude the following properties of h_i :

$$0 < h_i < 1, \quad \sum_i h_i = p$$

Recall the equation $\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$.

In matrix form:

$$\begin{pmatrix} \hat{y}_1 \\ \dots \\ \hat{y}_i \\ \dots \\ \hat{y}_n \end{pmatrix} = \begin{pmatrix} H_{11} & \dots & H_{1n} \\ \dots & \dots & \dots \\ H_{i1} & \dots & H_{in} \\ \dots & \dots & \dots \\ H_{n1} & \dots & H_{nn} \end{pmatrix} \begin{pmatrix} y_1 \\ \dots \\ y_i \\ \dots \\ y_n \end{pmatrix}$$

$$\begin{aligned} \hat{y}_i &= H_{i1}y_1 + \dots + H_{ii}y_i + \dots + H_{in}y_n \\ &= H_{i1}y_1 + \dots + \textcolor{red}{h}_i y_i + \dots + H_{in}y_n \end{aligned}$$

- Note that the LS fit, \hat{y}_i , is a linear combination of the n data points:

$$\hat{y}_i = h_i y_i + \sum_{i \neq j} H_{ij} y_j$$

This means that $h_i = \frac{d\hat{y}_i}{dy_i}$

- When h_i is large (close to 1), \hat{y}_i relies heavily on y_i (instead of using the information from other data points), therefore \hat{y}_i will be “forced” to be close to the observed y_i . Consequently, the variance for the residual r_i will be small, and the variance for the fit \hat{y}_i will be large (since the fit from another data set would be quite different).

$$var[\hat{y}_i] = \sigma^2 h_i, \quad var[r_i] = \sigma^2(1 - h_i)$$

High-leverage points

High-leverage points: Since $\sum_i h_i = p$, a rule-of-thumb is that observations with leverages more than $2p/n$ (twice the mean leverage) should be flagged as high-leverage points and should be examined closely.

- **Good high-leverage points:** its y point follows the pattern of the rest of the data, but with an x_i value that is far away from the sample mean.
- **Bad high-leverage points:** its y value does not follow the pattern suggested by the rest of the data; the LS fitting might change a lot if we remove this point.

Example: Leverages in Savings data set

Use the function `influence` to extract the leverages, and the function `halfnorm` to plot the leverages in increasing order.

```
library(faraway)
attach(savings)
country=dimnames(savings)[[1]]
```

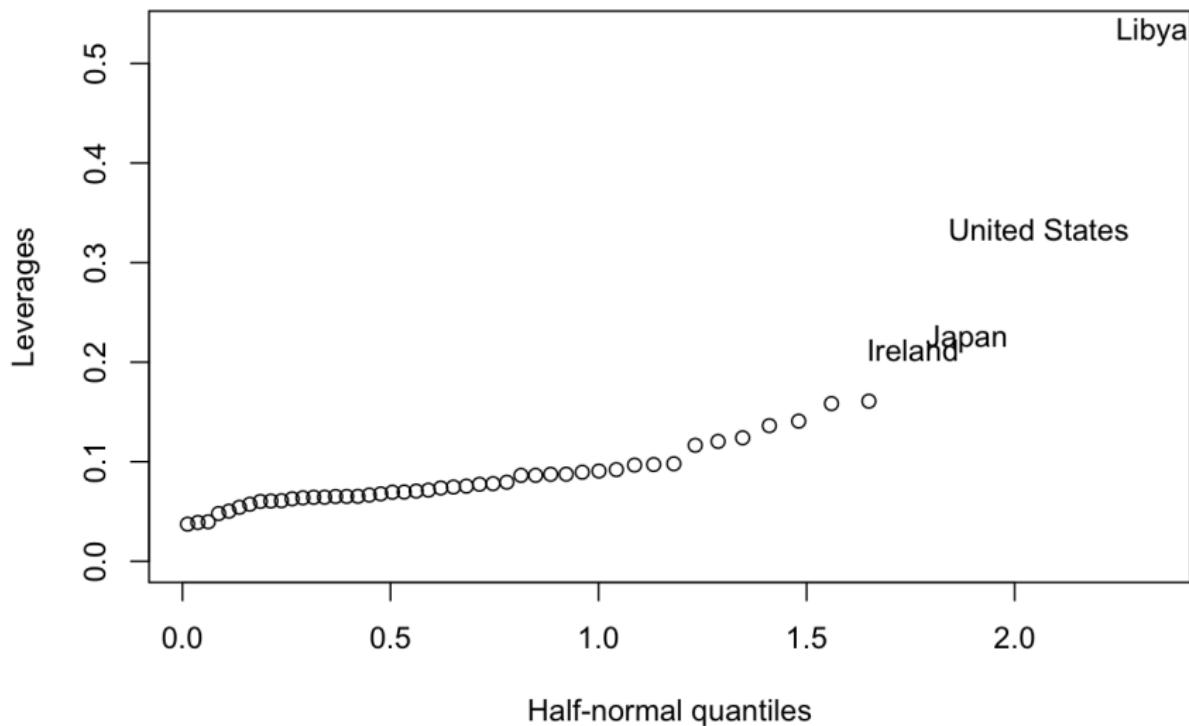
- Take a look of the data
- Check for high-leverage points

```
n=50; p=5;
g = lm(sr ~., data=savings);
lev=influence(g)$hat
lev[lev>2*p/n]
```

##	Ireland	Japan	United States	Libya
##	0.2122363	0.2233099	0.3336880	0.5314568

```
halfnorm(lev, 4, labs=row.names(savings), ylab="Leverages")
```

Example: Leverages in Savings data set



Residuals

The residuals $r_i = y_i - \hat{y}_i$ do NOT have a constant variance. So they need to be standardized. There are two versions of the residuals:

- **Standardized Residuals** r_i^* : They are internally standardized. Under the model assumptions they follow approximately a Normal distribution.
- **Studentized residuals** t_i : They are externally standardized. They follow a t distribution and will be used in our outlier test.

Residuals are very useful in regression diagnostics. Some authors recommend using the standardized version of the residuals instead of the raw residuals in all diagnostic plots.

Difference between e and r

e : true residuals

r : estimated residuals

- Both residuals are normally distributed, but:

$$\mathbf{e} \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I}_n), \quad \mathbf{r} \sim N_n(\mathbf{0}, \sigma^2 (\mathbf{I}_n - \mathbf{H}))$$

where \mathbf{H} is the projection/hat matrix.

- The errors e_i 's have equal variance and are independent, while the residuals r_i 's have unequal variance and are correlated.
- $E[\mathbf{e}] = E[\mathbf{r}] = \mathbf{0}$. But

$$\sum_i e_i \neq 0, \quad \sum_i r_i = 0$$

(by default we assume an intercept is included in the model)

Standardized Residuals

Since $r_i \sim N(0, \sigma^2(1 - h_i))$, it is reasonable to consider a standardization of the residuals in this form:

$$r_i^* = \frac{r_i}{\hat{\sigma}\sqrt{1 - h_i}}, \quad i = 1, \dots, n$$

- $\sum_i r_i^*$ is no longer zero
- Since the r_i is not independent of $\hat{\sigma}$, each r_i^* is not distributed as a t distribution.
- As an approximation, we can view the r_i^* 's as *iid* $N(0, 1)$ random variables, although they are not Normally distributed and they are slightly correlated.

Studentized Residuals

- The studentized residuals are based on the idea of leave-one-out (also known as jackknife residuals).
- Here is the leave-one-out idea: run a regression model on the $(n - 1)$ samples with the i -th sample (x_i, y_i) removed. Denote the leave-one-out estimates of the regression coefficient and error variance by $\hat{\beta}_{(i)}$ and $\hat{\sigma}_{(i)}$, where the notation (i) means "excluding the i -th observation."
- Then check the discrepancy between observations y_i and the fitted value $\hat{y}_{(i)} = \mathbf{x}^t \hat{\beta}_{(i)}$

Studentized Residuals (Cont.)

- Define the Studentized Residuals as:

$$t_i = \frac{y_i - \hat{y}_{(i)}}{\hat{\sigma}_{(i)}[1 + x_i^t (\mathbf{X}_{(\mathbf{i})}^t \mathbf{X}_{(\mathbf{i})})^{-1} x_i]^{1/2}} = \frac{y_i - \hat{y}_{(i)}}{\hat{\sigma}_{(i)} \sqrt{1 - h_i}}$$

which follows a t_{n-p-1} distribution if $y_i \sim N(\mathbf{x}_i^t \boldsymbol{\beta}, \sigma^2)$

The last equality above is not trivial (you can read the proof in the Appendix). One can also show that r_i^* and t_i are a monotone transformation of each other.

We do not need to run the model n times to get the estimates $\hat{\beta}_{(i)}$ and $\hat{\sigma}_{(i)}$ since it can be shown that:

$$t_i = r_i^* \left(\frac{n - p - 1}{n - p - r_i^{*2}} \right)^{1/2}$$

An Outlier test

- Outliers are observations that do not fit the model, but Outliers are not necessarily observations with large residuals.
- An outlier test is a useful tool to distinguish observations that have large residuals from outliers. We need to used the studentized residuals for the outlier test.
- Under the Null hypothesis H_0 , $t_i \sim t_{n-p-1}$ distribution. So we can use a **t-test** to test whether the i -th observation is an outlier or not.

- Generally, we would want to perform this outlier test for all n observations, doing the tests one at a time.
- If we perform the test on the largest observed residuals this would be an example of **data snooping**, unless somehow these cases were identified before data collection.
- In order to be certain that the overall **type I error** rate is no greater than α , the **Bonferroni correction** may be used. When doing so, each case would be tested at level α/n .

Bonferroni Correction

Suppose we are testing m hypothesis simultaneously. For each test, we use a significant level α . That is, the chance to make a Type I error is α . If we want to control the overall type I error rate (for all m tests) to be 95%, then we should set the individual significance levels to be $\alpha = 5\%/m$. Why? First note that

$$\begin{aligned} & Pr(\text{Type I error in any of } m \text{ tests}) \\ &= 1 - Pr(\text{No type I errors among } m \text{ tests}). \end{aligned}$$

Furthermore,

$$\begin{aligned} & Pr(\text{No type I error among } m \text{ tests}) \\ &= 1 - Pr(\text{type I error for test 1 OR for test 2 ... OR for test } m) \\ &\geq 1 - m\alpha \end{aligned}$$

Therefore, if we choose α/m for each of the m tests, then we will have an overall type I test of size $\leq \alpha$

What we should do with outliers?

- Delete them.
- But points should not be routinely deleted simply because they do not fit the model. No data snooping.
- Outliers, as well as other unusual observations discussed here, often flag potential problems of the current model. Instead of dropping them, maybe, try a new alternative model. (Outliers maybe thought are normal points that haven't found their distribution yet.)

Example: Outliers in Savings data set

Use the function `rstudent` to get the studentized residuals, and the function `sort` to sort the residuals in decreasing order.

```
jack=rstudent(g);
qt(.05/(2*n), 44) # Bonferroni correction

## [1] -3.525801

qt(.05/2, 44) # Without Bonferroni correction

## [1] -2.015368

sort(abs(jack), decreasing=TRUE)[1:5] # No outliers.

##      Zambia      Chile Philippines       Peru      Iceland
## 2.853558 2.313429 1.863826 1.823914 1.731200
```

There are no outliers in this data set since all studentized residual (in abs. values) are lower than 3.5258.

Influential observations

- Observations whose removal greatly affects the regression analysis are called **influential observations**.
- An influential observation may be (or may not) an outlier or a high-leverage observation; or may be both: an outlier and a high-leverage observation.
- We will use the Cook's distance to detect influential observations.

$$\begin{aligned} D_i &= \frac{\|\mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{X}\hat{\boldsymbol{\beta}}_{(i)}\|^2}{p\hat{\sigma}^2} = \frac{\|\hat{\mathbf{y}} - \hat{\mathbf{y}}_{(i)}\|^2}{p\hat{\sigma}^2} \\ &= \frac{r_i^{*2}}{p} \left(\frac{h_i}{1 - h_i} \right) \end{aligned}$$

which indicates that highly influential points are either outliers (large $|r_i^*|$) or high-leverage points (large h_i) or both.

- A **rule-of-thumb**: observations with $D_i \geq 1$ are highly influential.

Example: Influential observations in Savings data set

Use the function `cooks.distance` to calculate the Cook's distance for each observation and the function `halfnorm` to plot the CD's in increasing order.

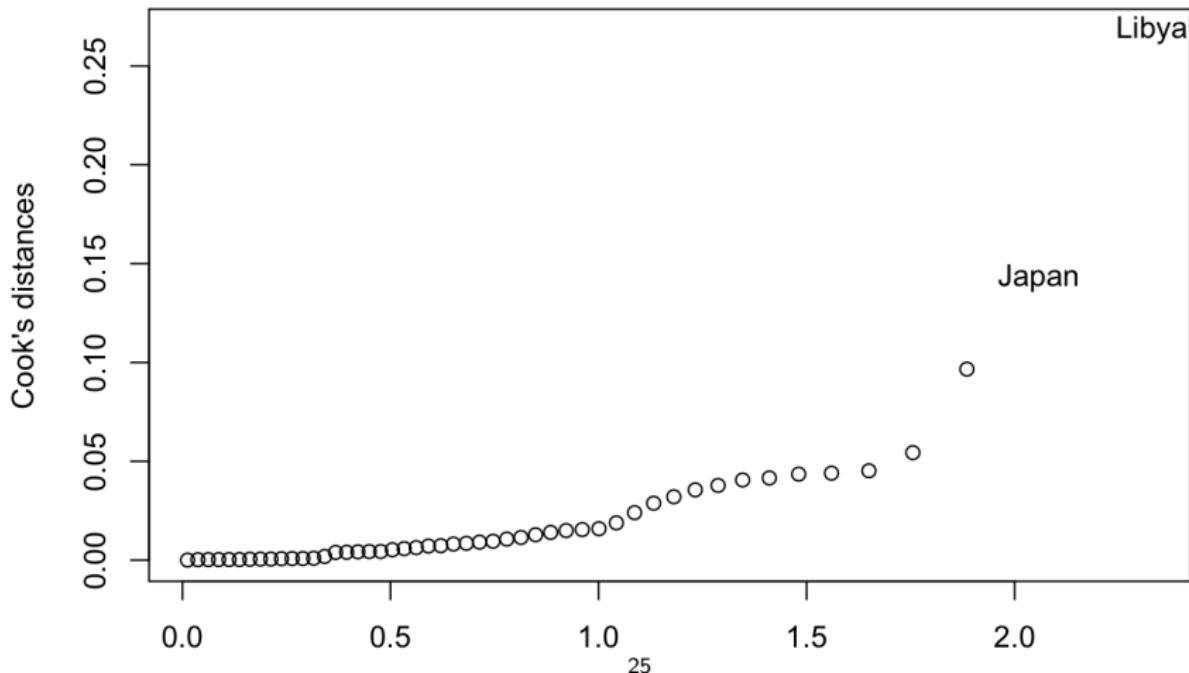
```
cook = cooks.distance(g)
max(cook)
```

```
## [1] 0.2680704
```

```
halfnorm(cook, labs=row.names(savings), ylab="Cook's distances")
```

According to the rule-of-thumb ($CD \geq 1$), there are not influential observations. However, there is one observation that is too far from the rest.

Example: Influential observations in Savings data set



Summary about Unusual Observations

- **High-leverage points:** $h_i = H_{ii} > 2p/n$. High-leverage points are far away from the center of the data (in terms of the Mahalanobis distance). Keep in mind that:

$$\text{var}[\hat{y}_i] = \sigma^2 h_i, \quad \text{var}[r_i] = \sigma^2(1 - h_i)$$

- **Outliers:** Perform a t-test on the studentized residuals using the Bonferroni correction. This is equivalent to removing the i -th point, run LS on the remaining $(n - 1)$ data points, and then form a PI at \mathbf{x}_i ; if PI covers y_i , then the i -th point is NOT an outlier.
- **Highly influential points;** Use Cook's distance and check whether $D_i \geq 1$:

$$D_i = \frac{r_i^{*2}}{p} \left(\frac{h_i}{1 - h_i} \right)$$

which indicates that high influential points are either outliers or high-leverage points or both.

STAT 425

Model Diagnostics. Part 2

Model Diagnostics: Checking Error Assumptions

Which assumptions?

- Constant Variance
- Normality
- Uncorrelated errors

How can we check these assumptions?

- Graphical tools: Residual plots, QQ-plots

Which remedies?

- Transformations, Generalized Least-Squares, nonlinear regression

Residual Plots

- Plot residuals r_i or studentized t_i against fitted values \hat{y}_i .
- Plot residuals r_i or studentized t_i against each predictor x_i .
- Plot residuals r_i or studentized t_i against an index variable such as time or case number.
- Look for systemic patterns (non-constant variance, non-linearity) and large absolute values of residuals.

Example: Cleaning data (Sheather)

```
# Read cleaning data
cleaning<-read.table("cleaning.txt",header=TRUE)

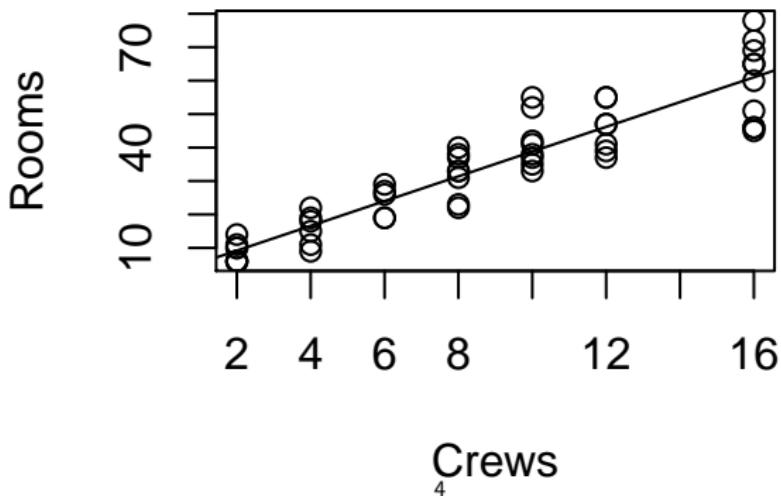
# Display basic statistical measures
summary(cleaning)

##          Case          Crews          Rooms
##  Min.   : 1   Min.   : 2.000   Min.   : 6.00
##  1st Qu.:14  1st Qu.: 4.000  1st Qu.:19.00
##  Median :27  Median : 8.000  Median :35.00
##  Mean   :27  Mean   : 8.679  Mean   :33.91
##  3rd Qu.:40  3rd Qu.:12.000 3rd Qu.:46.00
##  Max.   :53  Max.   :16.000  Max.   :78.00

modclean<-lm(Rooms ~ Crews,data=cleaning)
```

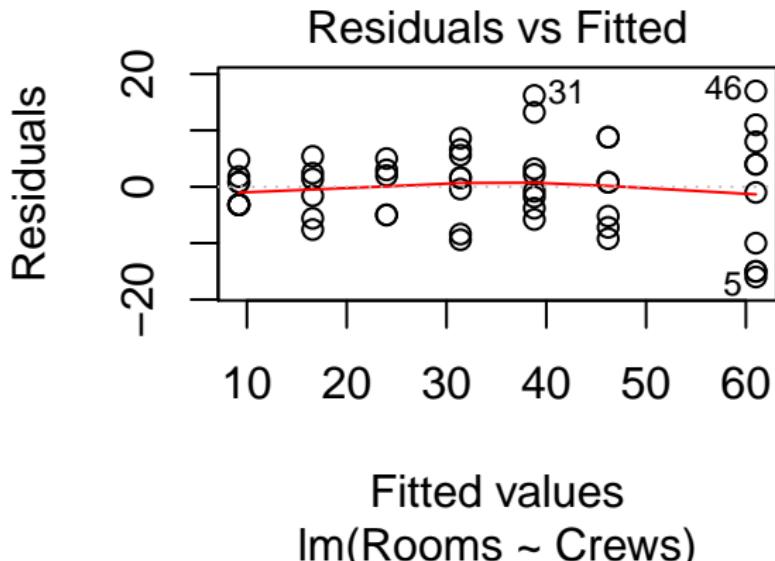
Example: Cleaning data (Sheather)

```
plot(Rooms ~ Crews, data=cleaning)  
abline(modclean)
```



Example: Cleaning data (Sheather)

```
plot(modclean, which=1)
```



Non-Constant Variance

- Check residual plots and look for a “fan” type shape or trends
- A less rigorous but quick way:
 $\text{lm}(\text{abs}(\text{mod\$residuals}) \sim \text{mod\$fitted.values})$
- A formal test: Breusch-Pagan Test (`bptest` in package `lmtest`)
- Remedy: transformation.

Breusch-Pagan test example: saving data set

Suppose $g : sr \sim pop15 + pop75 + dpi + ddpi$

We use function **bptest** from library **lmtest**

```
bptest(g)
```

```
##  
## studentized Breusch-Pagan test  
##  
## data: g  
## BP = 4.9852, df = 4, p-value = 0.2888
```

```
tmp.fit = lm(g$res^2 ~ pop15 + pop75 + dpi + ddpi)  
#summary(tmp.fit)  
summary(tmp.fit)$r.sq*50 #Compare this value with the BP statistic.
```

```
## [1] 4.985161
```

```
# We fail to reject the null hypothesis of homocedasticity
```

Variance stabilizing transformations

The goal is to find a transformation of the response $h(Y)$ to achieve constant variance. The method for finding these transformations is based on the following. Suppose h is a smooth function. Then by the Taylor's theorem, the following expansion of $h(Y)$ around the $E[Y]$ holds:

$$h(Y) = h(E[Y]) + h'(E[Y])(Y - E[Y]) + \dots$$

where the dots \dots represent the reminder of this approximation. This reminder is assumed small with high probability and we can ignore it. Then we have:

$$\text{var}[h(Y)] \approx (h'(E[Y]))^2 \text{var}[Y]$$

We want to choose a transformation h such that:

$$\text{var}[h(Y)] \approx (h'(E[Y]))^2 \text{var}[Y]$$

is approximately **constant**.

For example, suppose that the variance of Y is proportional to the mean of Y , i.e., $\text{Var}(Y) \propto E[Y]$, then if we select h such that:

$$h'(z) = \frac{1}{\sqrt{z}}$$

$$\Rightarrow h(z) \propto \sqrt{z}$$

When plugging-in the value of $h'(z)$ evaluated in $E[Y]$ in the variance of $h(Y)$ equation, the variance of $h(Y)$ will be approximately constant.

Another example:

Suppose $\text{var}(Y) \propto E[Y]^2$, then

$$h'(z) = \frac{1}{z} \Rightarrow h(z) = \log(z)$$

Residual plots can give an idea of the relationship between $\text{var}(Y) = \text{var}(e)$ (Residual variance) and the estimated $E[Y]$ (fitted values).

A summary a variance stabilizing transformations:

- When $\text{var}(e) \propto E[Y]$, $h(Y) = \sqrt{Y}$. Suitable for counts from the Poisson distribution.
- When $\text{var}(e) \propto E[Y]^2$, $h(Y) = \log(Y)$ or $\log(Y + 1)$. Suitable for data whose range of Y is very broad, e.g., from 1 to several thousands; suitable for estimating percentage effect ($Y \propto CX^\alpha$.)
- When $\text{var}(e) \propto E[Y]^4$, $h(Y) = 1/Y$ or $1/(Y + 1)$. Suitable for data where Y measures the waiting time or survival time. Taking reciprocals changes the scale from time (time per response) to rate (response per unit time).

Assessing Normality

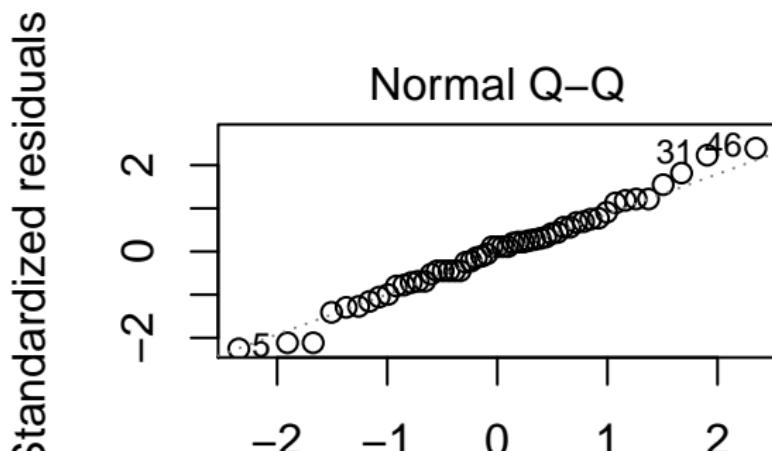
Suppose that we have a sample z_1, z_2, \dots, z_n , and we wish to examine the hypothesis that the z 's are a sample from a normal distribution with mean μ and variance σ^2 . A standard graphical method for inspecting the normal assumption is the **QQ-plot**. It is calculated as follows:

- ① Order the z 's: $z_{(1)}, z_{(2)}, \dots, z_{(n)}$
- ② Compute $u_i = \Phi^{-1}\left(\frac{i}{n+1}\right)$, where Φ is the cdf of the $N(0, 1)$ and i is the order if the data ($i = 1, 2, \dots, n$)
- ③ Plot $z_{(i)}$ against u_i . If the z 's are normal, the plot should be approximately a straight line

A more formal way to test normality: [Shapiro-Wilks test](#).

Example: Cleaning data (Sheather)

```
plot(modclean, which=2)
```



Theoretical Quantiles
lm(Rooms ~ Crews)

Shapiro-Wilks test

H_0 : The residuals follow a Normal distribution

```
# Shapiro-Wilks test
shapiro.test(residuals(modclean))

##
##  Shapiro-Wilk normality test
##
## data: residuals(modclean)
## W = 0.98681, p-value = 0.822
```

Since the p-value is large, we fail to reject the Null hypothesis

Correlated Errors

- Correlation is normally present when we have data with temporal, or spatial predictors
- We can plot residuals against time or other index, such as case number and look whether data above or below the mean tend to be followed by data above or below the mean
- To detect correlation: use formal tests like the Durbin-Watson test ([dwtest](#) in package [lmtest](#))

Checking Model Structure Assumptions (Non-linearity)

How do we check that the linearity assumption $E[y] = \mathbf{X}\beta$ is correct?

- We can apply the Lack-of-fit test when replicates are available (will be discussed later)
- Use partial regression plots
- Use partial residual plots
- Remedies to lack of linearity: Apply transformations, nonlinear regression (will be discussed later)

Partial Regression PPlot (added Variable Plot)

- We want to know the relationship between the response Y and a predictor X_k after the effect of the other predictors has been removed.
- To remove the effect of the other predictors, run the following two regression models:

$$Y \sim X_1 + \dots + X_{k-1} + X_{k+1} + \dots \quad (1)$$

$$X_k \sim X_1 + \dots + X_{k-1} + X_{k+1} + \dots \quad (2)$$

Get the following residuals:

\mathbf{r}_y = residuals from (1)

\mathbf{r}_k^X = residuals from (2)

- Plot \mathbf{r}_y vs. \mathbf{r}_k^X : For a valid model, then the added-variable plot should produce points randomly scattered around a line through the origin with slope $\hat{\beta}_k$. This is also a useful plot to detect high influential data points.

Using transformations to overcome non-linearity

Examples of linearizing transformations:

- Use $\log(Y)$ vs. $\log(X)$ (apply logarithm to the response and the predictors): Suitable when $E[Y] = \alpha X_1^{\beta_1} \dots X_p^{\beta_p}$
- $\log(Y)$ vs. X (apply logarithm to the response only): Suitable when $E[Y] = \alpha \exp \sum_j X_j \beta_j$
- $1/Y$ vs. X (Take the inverse of the response): Suitable when $E[Y] = \frac{1}{\alpha + \sum_j X_j \beta_j}$

Box-Cox transformation of the Y variable

- Box and Cox (1964) suggested a family of transformations (for positive response) designed to reduce non-normality of the errors. It turns out that in doing this, it often reduces non-linearity as well.
- Suppose each $y_i > 0$, and consider the following transformation:
¹:

$$g_\lambda(y) = \begin{cases} \frac{y^\lambda - 1}{\lambda}, & \lambda \neq 0 \\ \log(y), & \lambda = 0 \end{cases}$$

¹The transformation for $\lambda = 0$ is justified because $\lim_{\lambda \rightarrow 0} \frac{y^\lambda - 1}{\lambda} = \log(y)$

The aim is to choose λ that maximizes the likelihood of the data, under the normal assumption that the transformed data $g_\lambda(\mathbf{y})$ has a normal distribution:

$$g_\lambda(\mathbf{y}) = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}, \quad \mathbf{e} \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I})$$

- The maximum log-likelihood function for $\lambda \neq 0$ is:

$$L(\lambda) = -\frac{n}{2} \log(RSS_\lambda/n) + (\lambda - 1) \sum_{i=1}^n \log(y_i)$$

where RSS_λ is the residual sum of squares when $g_\lambda(\mathbf{y})$ is the response, and for $\lambda = 0$ is:

$$L(0) = -\frac{n}{2} \log(RSS_0/n) - \sum_{i=1}^n \log(y_i)$$

The second term in these log-likelihood function corresponds to the Jacobian of the transformation.

- Note that it doesn't make sense to simply pick λ that minimizes RSS_λ since for each λ , the residual sum of squares are measured in a different scale.

- In **R**, we can graph the log-likelihood as a function of λ ($L(\lambda)$) versus $\lambda \in (-2, 2)$ ² and then pick the maximizer $\hat{\lambda}$.
- It is common to round $\hat{\lambda}$ to a nearby value like:

$-1, -0.5, 0, 0.5,$ or 1

then the transformation defined by $\hat{\lambda}$ is easier to interpret.

²The method tends to work well for λ in this range

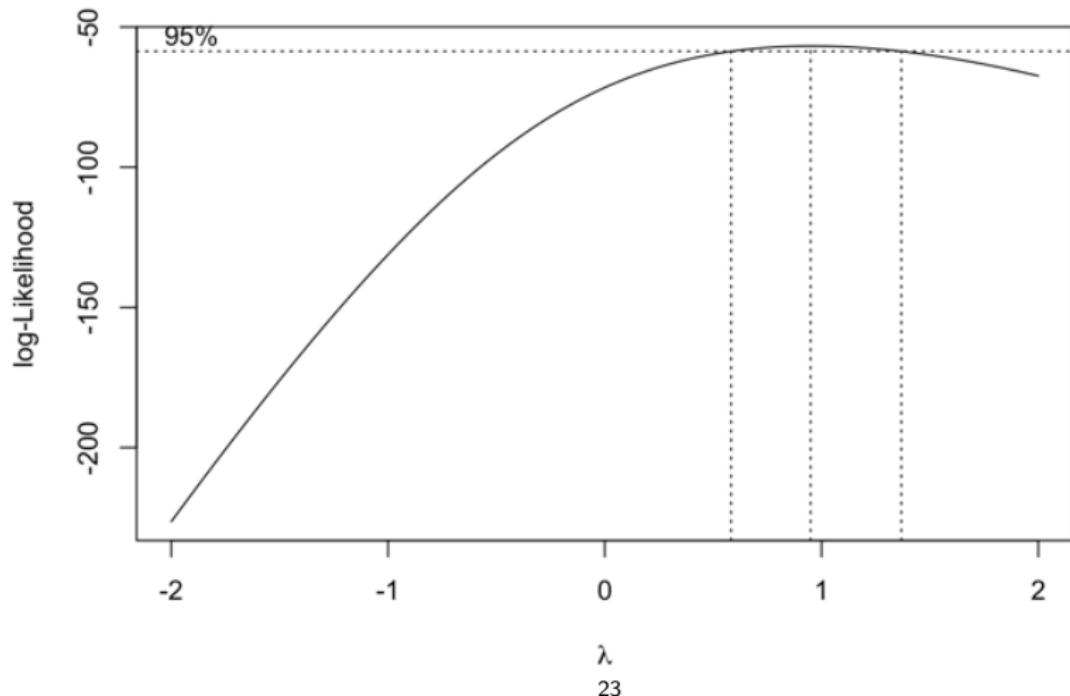
- To answer the question whether we really need the transformation g_λ , we can do hypothesis testing ($H_0 : \lambda = 1$), or equivalently construct a Confidence Interval for λ as follows³:

$$\{\lambda : L(\lambda) > L(\hat{\lambda}) - \frac{1}{2}\chi_1^2(1 - \alpha)\}$$

³This is based on the result that $2(L(\hat{\lambda}) - L(\lambda_0)) \sim \chi_1^2$ under H_0

Box-cox transformation example

```
boxcox(g,plotit=T) # plotit=T is the default setting
```



STAT 425

Collinearity

Collinearity

Consider a MLR model with a design matrix $\mathbf{X}_{n \times p}$ including the intercept. If the columns of \mathbf{X} are **orthogonal** to each other (i.e., the sample correlation of any two predictors is equal to 0), then the LS problem is greatly simplified:

$$\hat{\beta}_j = [(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}]_j = \frac{\mathbf{X}_{\cdot j}^\top \mathbf{y}}{\|\mathbf{X}_{\cdot j}\|^2}$$

where $\mathbf{X}_{\cdot j}$ denotes the j -th column of \mathbf{X} .

In other words, in this case (only) the LS regression coefficient for the j -th predictor does not depend on whether other predictors are included in the model or not.

Collinearity

- In practice, we often encounter problems in which many of the predictors are highly correlated.
- In such cases, the values and sampling variance of regression coefficients can be highly dependent on the particular predictors chosen for the model.

Exact Collinearity

- If there exists a set of constants c_1, c_2, \dots, c_p (at least one of them is non-zero), such that the corresponding linear combination of the columns of \mathbf{X} is zero, i.e.:

$$\sum_{j=1}^p c_j \mathbf{X}_{\cdot j} = \mathbf{0}$$

then the columns of \mathbf{X} are called **linearly dependent** and there is **exact collinearity**. That is, at least one column in the design matrix \mathbf{X} can be expressed as a linear combination of other columns.

What happens when the columns of \mathbf{X} are collinear?

- ① $(\mathbf{X}^\top \mathbf{X})^{-1}$ does not exist,
- ② The LS estimate $\hat{\boldsymbol{\beta}}$ is not unique, and
- ③ The coefficients of the linear model are not identifiable.

Example: Suppose the 1st column of \mathbf{X} is the intercept, and the 2nd column of \mathbf{X} is the vector $(2, 2, \dots, 2)^\top$. Then if $(\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3, \dots)^\top$ is one LS estimate of $\boldsymbol{\beta}$, the vector $(\hat{\beta}_1 - c, \hat{\beta}_2 + c/2, \hat{\beta}_3, \dots)^\top$ is also an estimate of $\boldsymbol{\beta}$, where c is any real number.

Note: In case of exact collinearity the column space of \mathbf{X} has dimension $< p$. In this case we can often fit an equivalent model by eliminating one or more redundant variables.

Approximate Collinearity

- We generally do not need to worry about exact collinearity¹, but approximate collinearity. That is, at least one column $\mathbf{X}_{\cdot j}$ can be approximated by the others:

$$\mathbf{X}_{\cdot k} \approx - \sum_{j \neq k} c_j \mathbf{X}_{\cdot j} / c_k$$

A simple diagnostic for this is to obtain the regression of $\mathbf{X}_{\cdot k}$ on the remaining predictors, and if the corresponding R^2_k is close to 1, we would diagnose approximate collinearity.

¹R can detect it and fix it automatically

Why approximate collinearity is a problem?

- In a multiple regression $Y = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p + e$, the LS estimate $\hat{\beta}_k$ is unbiased with variance:

$$var(\hat{\beta}_k) = \sigma^2 \left(\frac{1}{1 - R_k^2} \right) \left(\frac{1}{\sum_{i=1}^n (x_{ik} - \bar{x}_{.k})^2} \right)$$

where R_k^2 is the R-square from the regression of $\mathbf{X}_{.k}$ on the remaining predictors. When R_k^2 is close to 1, the variance of $\hat{\beta}_k$ is large. Consequently we will have:

- ① large Mean Square Error
- ② large (inflated) p-value to the corresponding t-test, i.e, we could **miss** a significant predictor.
- The quantity $\left(\frac{1}{1 - R_k^2} \right)$ is the **variance inflation factor** (VIF) for the k -th coefficient of the model

Example: Car position data

Data on 38 drivers:

- Age: Drivers age in years
- Weight: Drivers weight in lbs
- HtShoes: height with shoes in cm
- Ht: height without shoes in cm
- Seated: seated height in cm
- Arm: lower arm length in cm
- Thigh: thigh length in cm
- Leg: lower leg length in cm
- hipcenter: horizontal distance of the midpoint of the hips from a fixed location in the car in mm

```
library(faraway)
data(seatpos)
attach(seatpos)
g=lm(hipcenter ~ ., seatpos)
summary(g)
```

Example: Car position data

Collinearity Symptoms: None of the individual variables is significant.
Large standard errors. High correlation among variables

```
##  
## Call:  
## lm(formula = hipcenter ~ ., data = seatpos)  
##  
## Residuals:  
##      Min      1Q  Median      3Q     Max  
## -73.827 -22.833 -3.678  25.017  62.337  
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 436.43213 166.57162  2.620  0.0138 *  
## Age          0.77572   0.57033   1.360  0.1843  
## Weight        0.02631   0.33097   0.080  0.9372  
## HtShoes      -2.69241   9.75304  -0.276  0.7845  
## Ht           0.60134  10.12987   0.059  0.9531  
## Seated       0.53375   3.76189   0.142  0.8882  
## Arm          -1.32807   3.90020  -0.341  0.7359  
## Thigh         -1.14312   2.66002  -0.430  0.6706  
## Leg          -6.43905   4.71386  -1.366  0.1824  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 37.72 on 29 degrees of freedom  
## Multiple R-squared:  0.6866, Adjusted R-squared:  0.6001  
## F-statistic: 7.94 on 8 and 29 DF,  p-value: 1.306e-05
```

Calculate Variance Inflation Factor of model matrix X (after removing the first column) using function **vif(.)**.

```
# Variance Inflation Factor (VIF)
round(vif(x), dig=2)
```

```
##      Age   Weight HtShoes       Ht   Seated      Arm    Thigh     Leg
## 2.00    3.65  307.43  333.14    8.95    4.50    2.76    6.69
```

```
sqrt(307.43)
```

```
## [1] 17.53368
```

Standard error of the estimated predictor $\hat{\beta}_{HtShoes}$ is approximately 17 times larger than it would have been without collinearity.

A global measure of collinearity

- A global measure of collinearity is given by examining the eigenvalues of $\mathbf{X}^\top \mathbf{X}$. A popular measure is the **condition number** of $\mathbf{X}^\top \mathbf{X}$, denoted by:

$$\kappa = (\text{largest eigenvalue}/\text{smallest eigenvalue})^{1/2}$$

An empirical rule for declaring collinearity is $\kappa \geq 30$

- Note that κ is not scale-invariant, so we should standardize each column of \mathbf{X} (i.e. each column should have zero mean and sample variance equal to 1, before calculating the condition number).

Example: Car Seat Position data

```
# Standardize matrix
x = model.matrix(g)[,-1]
x = x - matrix(apply(x, 2, mean), 38, 8, byrow=TRUE)
x = x / matrix(apply(x, 2, sd), 38, 8, byrow=TRUE)
apply(x, 2, mean)
```

```
##           Age      Weight      HtShoes        Ht      Seated
## -2.193512e-17 2.810252e-16 9.566280e-16 1.941574e-16 -1.073010e-15
##           Arm      Thigh       Leg
## -1.070022e-16 8.909895e-17 -9.114182e-17
```

```
apply(x, 2, var)
```

```
##      Age  Weight HtShoes        Ht      Seated      Arm      Thigh      Leg
##         1       1       1       1       1       1       1       1       1
```

```
e = eigen(t(x) %*% x)
sqrt(e$val[1]/e$val)
```

```
## [1] 1.000000 2.141737 3.497636 4.852243 5.404643 6.384606 10.615424
## [8] 59.766197
```

Symptoms and Remedies of Collinearity

- Possible symptoms of collinearity:
 - ① high pair-wise (sample) correlation between predictors
 - ② high VIF
 - ③ high condition number
 - ④ R^2 is relatively large but none of the predictor is significant.
- What to do with collinearity?

Remove some predictors from highly correlated groups of predictors.

Another method we study later: regularize the model using penalized Least Squares estimation

STAT 425

Generalized Least Squares (GLS)

Generalized Least Squares

What do we do if the errors are correlated or heteroscedastic?

Suppose $\mathbf{e} \sim N_n(\mathbf{0}, \Sigma)$, where Σ is the variance-covariance matrix.
We will consider two cases:

- Σ known (this is an idealized case from which we can get some insight)
- Σ unknown (e.g. regression with time series data, spatial data, etc.)

We will discuss some examples and R code

GLS: Σ known

- Assume $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}$ and $\mathbf{e} \sim N_n(\mathbf{0}, \Sigma)$ where Σ is a known, symmetric, positive definite covariance matrix.
- Transform this problem back to Ordinary Least-Squares (OLS). Write $\Sigma = SS^\top$ where we assume S^{-1} exists. We could use, for example, the Cholesky decomposition from linear algebra to obtain S . Multiply the model equation by S^{-1} on both sides:

$$S^{-1}\mathbf{y} = S^{-1}(\mathbf{X}\boldsymbol{\beta} + \mathbf{e})$$

$$\mathbf{y}^* = \mathbf{X}^*\boldsymbol{\beta} + \mathbf{e}^*$$

$$\mathbf{e}^* \sim (S^{-1}\mathbf{0}, S^{-1}\Sigma(S^{-1})^\top) = N(\mathbf{0}, \mathbf{I})$$

- Now we can solve for β using OLS:

$$\mathbf{y}^* = \mathbf{X}^* \boldsymbol{\beta} + \mathbf{e}^*, \quad \mathbf{y}^* = S^{-1} \mathbf{y}, \quad \mathbf{X}^* = S^{-1} \mathbf{X}$$

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= [\mathbf{X}^{*\top} \mathbf{X}^*]^{-1} \mathbf{X}^{*\top} \mathbf{y}^* \\ &= (\mathbf{X}^\top (S^{-1})^\top S^{-1} \mathbf{X})^{-1} \mathbf{X}^\top (S^{-1})^\top S^{-1} \mathbf{y} \\ &= (\mathbf{X}^\top \Sigma^{-1} \mathbf{X})^{-1} \mathbf{X}^\top \Sigma^{-1} \mathbf{y}\end{aligned}$$

- Note that the solution minimizes:

$$\|\mathbf{y}^* - \mathbf{X}^* \boldsymbol{\beta}\|^2 = (\mathbf{y} - \mathbf{X} \boldsymbol{\beta})^\top \Sigma^{-1} (\mathbf{y} - \mathbf{X} \boldsymbol{\beta})$$

Weighted Least Squares (WLS)

- Suppose that Σ is a diagonal matrix of unequal error variances:

$$\Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2)$$

- The GLS estimate of β minimizes:

$$(\mathbf{y} - \mathbf{X}\beta)^\top \Sigma^{-1} (\mathbf{y} - \mathbf{X}\beta) = \sum_{i=1}^n \frac{(y_i - \mathbf{x}_i^\top \beta)^2}{\sigma_i^2}$$

This problem is known as the [Weighted Least-Squares \(WLS\)](#).

- Note that the errors are weighted by $1/\sigma_i^2$: smaller weights for samples with larger variances.

WLS Example

strongx data set from the *faraway* library.

A large number of observations taken for each *momentum* measurement, allows to have a good estimate of the standard deviation *sd* for each value of the response *crossx* at each energy level. We can use *weights* = $1/sd^2$ as a parameter in the *lm(.)* call.

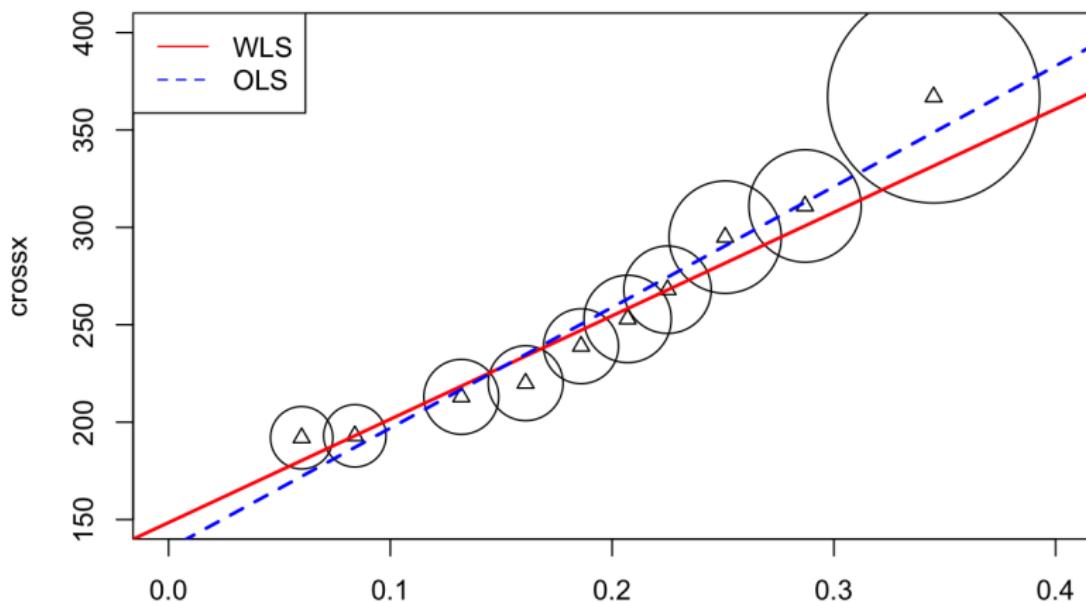
```
data("strongx")
names(strongx)
```

```
## [1] "momentum" "energy"    "crossx"     "sd"
```

```
g=lm(crossx ~ energy, strongx, weights=1/sd^2)
summary(g)
```

OLS vs. WLS

The WLS line departs from values with higher variance (smaller weights)



WLS Special case: Replicated Observations

Suppose we collected multiple observations for each \mathbf{x}_i . We use double subscripts to indicate the replicate observations:

$$(\mathbf{x}_i, y_{i1}, y_{i2}, \dots, y_{in_i})$$

Let y_i denote the average of the n_i observations sharing \mathbf{x}_i . Then the residual sum of squares for β equals

$$\sum_{i=1}^n \sum_{j=1}^{n_i} (y_{ij} - \mathbf{x}_i^\top \boldsymbol{\beta})^2 = \sum_{i=1}^n n_i (y_i - \mathbf{x}_i^\top \boldsymbol{\beta})^2 + \sum_{i=1}^n \sum_{j=1}^{n_i} (y_{ij} - y_i)^2$$

Minimizing the RSS to solve for $\boldsymbol{\beta}$ is the same as minimizing the first term on the right only (why?). Because $Var(y_i) = \sigma^2/n_i$, we use WLS on the y_i :

$$\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}} \sum_{i=1}^n n_i (y_i - \mathbf{x}_i^\top \boldsymbol{\beta})^2$$

In R: Use weights in the lm(.) function: $lm(y_i \sim \dots, \text{weights}=n_i, \dots)$

Maximum Likelihood Estimation when Σ is known

- Model: $\mathbf{y} \sim N_n(\mathbf{X}\boldsymbol{\beta}, \Sigma)$
- Log-likelihood:

$$\begin{aligned} & \log(p(\mathbf{y}|\boldsymbol{\beta}, \Sigma)) \\ &= \log \left\{ \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top \Sigma^{-1} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right] \right\} \\ &= -\frac{1}{2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top \Sigma^{-1} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + Constant. \end{aligned}$$

- Therefore the MLE is given by

$$\hat{\boldsymbol{\beta}}_{mle} = \arg \min_{\boldsymbol{\beta}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top \Sigma^{-1} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

Generalized Least-Squares: Σ unknown

How about using the following iterative approach?

- ① Start with some initial guess of Σ
- ② Use Σ to estimate β
- ③ Use residuals (since we have known β) to estimate Σ
- ④ Iterate until convergence

It looks like a good idea; however the methods will not work if we do not assume some structure about Σ (too many parameters to be estimated).

Usually, based on the application, we can assume a particular structure for Σ that does not involve too many parameters. Then we can model β and Σ simultaneously. For example , for AR(1) times series (auto-regressive model of order 1), the structure of Σ would be:

$$\Sigma = \sigma^2 \begin{pmatrix} 1 & \rho & \rho^2 & \rho^3 & \dots \\ \rho & 1 & \rho & \rho^2 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \rho^{n-1} & \rho^{n-2} & \dots & \dots & 1 \end{pmatrix}$$

Σ as a function of ρ and σ^2 .

Use the **nlme** package in **R**

Example with auto-correlated errors

Time series data

- Longley's Economic Regression Data: A data frame with 7 economical variables, observed yearly from 1947 to 1962 (n=16).
- GNP.deflator: GNP implicit price deflator (1954=100)
- GNP: Gross National Product.
- Unemployed: number of unemployed.
- Armed.Forces: number of people in the armed forces.
- Population: 'noninstitutionalized' population \geq 14 years of age.
- Year
- Employed: number of people employed.

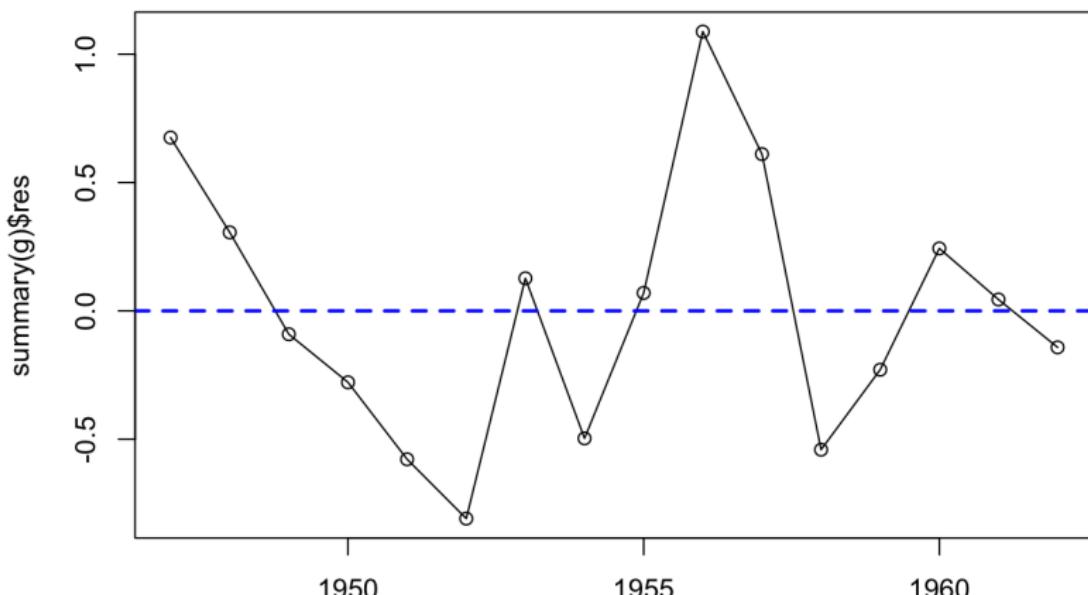
```
library(faraway)
data("longley")
head(longley)
```

	GNP.deflator	GNP	Unemployed	Armed.Forces	Population	Year	Employed
## 1947	83.0	234.289	235.6	159.0	107.608	1947	60.323
## 1948	88.5	259.426	232.5	145.6	108.632	1948	61.122
## 1949	88.2	258.054	368.2	161.6	109.773	1949	60.171
## 1950	89.5	284.599	335.1	165.0	110.929	1950	61.187
## 1951	96.2	328.975	209.9	309.9	112.075	1951	63.221
## 1952	98.1	346.999	193.2	359.4	113.270	1952	63.639

Example with auto-correlated errors

Residuals after fitting the model:

```
g = lm(Employed ~ GNP + Population, data=longley)
```



Test for autocorrelation

Use Durbin-Watson test from the lmtest library to test autocorrelation.

Null hypothesis: Errors are not auto-correlated

```
dwtest(g)
```

```
##  
##  Durbin-Watson test  
##  
## data: g  
## DW = 1.3015, p-value = 0.02245  
## alternative hypothesis: true autocorrelation is greater than 0
```

#D-W test shows the errors are significantly correlated
#Solution: Fit a Regression with autocorrelated errors.

```
library(nlme)  
g = gls(Employed ~ GNP + Population, correlation = corAR1(form= ~ Year), data=longley)  
summary(g)
```

Use function **glS** from library **nlme**

STAT 425

Lack of Fit Testing

Gaussian Model Assumptions

Recall our idealized modeling assumptions, which can be summarized concisely as:

$$\mathbf{y} \sim N_n(\mathbf{X}\beta, \sigma^2\mathbf{I})$$

Under these assumptions:

$$\hat{\beta} = (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t \mathbf{y} \sim N_p(\beta, \sigma^2 (\mathbf{X}^t \mathbf{X})^{-1}),$$

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} \sim N_n(\mathbf{X}\beta, \sigma^2 \mathbf{H}), \quad \mathbf{H} = \mathbf{X}(\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t,$$

and, independently,

$$\hat{\sigma}^2 = \frac{RSS}{n-p} = \frac{\|\mathbf{y} - \hat{\mathbf{y}}\|^2}{n-p} \sim \sigma^2 \frac{\chi_{n-p}^2}{n-p}.$$

Testing for Lack of Fit

How can we test whether the model $\mathbf{X}\beta$ fits the data?

- Intuition: If the model is correct then $\hat{\sigma}^2$ is an unbiased estimate of σ^2 . In the very special case where we knew σ^2 , we could construct a test based on the ratio $\hat{\sigma}^2/\sigma^2$, a measure of lack-of-fit.
- If σ^2 is unknown and we have some **replication** in the design (repeat rows of \mathbf{X}), then we'll see how to devise an F test for lack of fit.

Lack of Fit test when σ^2 is known

- In this case we want to test the hypothesis:

$$H_0 : \text{There is no lack of fit,} \quad \text{vs.} \quad H_a : \text{There is lack of fit}$$

- We use the test statistic:

$$\frac{\hat{\sigma}^2}{\sigma^2} = \frac{RSS/(n-p)}{\sigma^2} \sim \frac{\chi_{n-p}^2}{n-p}$$

Lack of fit means the error variance is large related to the value of σ^2 , i.e., the test statistic is large.

- Conclude that there is lack of fit (i.e. Reject H_0), if:

$$(n-p) \frac{\hat{\sigma}^2}{\sigma^2} \geq \chi_{n-p}^2(1-\alpha)$$

Example: Lack of fit test assuming σ^2 is known

In this example, all individual variances have been accounted for by using the *weights* parameter, so we take $\sigma^2 = 1$. Then our test is based on $(n - p)\hat{\sigma}^2 \sim \chi^2_{n-p}$ under H_0 .

```
g=lm(crossx ~ energy, strongx, weights=1/sd^2)
# Lack-of-fit Test
# Assume sigma^2=1 since all variances have been account for in the weights parameter
1 - pchisq(summary(g)$sig^2*8, 8)

## [1] 0.005004345
```

```
#Conclude that there is lack of fit
summary(g)
```

Since the p-value < 0.05 we reject the null hypothesis and conclude there is a lack of fit. This might be the case even with a high value of R^2 .

Lack of Fit test when σ^2 is unknown

- If σ^2 is unknown, a general approach is to compare an estimate of σ^2 based on a **much bigger/general model**.
- If we can derive the distribution (under H_0) of $\hat{\sigma}_{LinearModel}^2 / \hat{\sigma}_{BigModel}^2$, then we reduce this problem to a two model comparison test problem.
- The null hypothesis is the current model:

$$H_0 : E(y_i) = \mathbf{x}_i^\top \boldsymbol{\beta}, \quad i = 1, 2, \dots, n, \quad \text{for some vector } \boldsymbol{\beta}$$

- The more general model is assumed under the alternative hypothesis:

$$H_a : E(y_i) = f(\mathbf{x}_i), \quad i = 1, 2, \dots, n, \quad \text{for some function } f$$

Lack of Fit test when σ^2 is unknown

Can we estimate σ^2 for the big model in H_a ?

- The answer is yes if there is some replication in the data, i.e., there are multiple observations (replicates) for some (at least) of the same x_i values
- Schematically we can represent these replicates as:

$$(\mathbf{x}_i, y_{i1}, y_{i2}, \dots, y_{in_i}), \quad i = 1 : m, \quad n = \sum_i n_i$$

Lack of Fit test

Under the null hypothesis H_0 :

- $y_{ij} = \mathbf{x}_i^\top \beta + e_{ij}$, some β , $e_{ij} \sim \text{iid } N(0, \sigma^2)$
- RSS_0 with $df = n - p$

Under the alternative big-model hypothesis H_a :

- $y_{ij} = f(\mathbf{x}_i) + e_{ij}$, some function f , $e_{ij} \sim \text{iid } N(0, \sigma^2)$
- RSS_a with $df = n - m = \sum_i (n_i - 1)$, where

$$RSS_a = \sum_{i=1}^m \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_{i\cdot})^2$$

All of the degrees of freedom for RSS_a come from the replications.

Therefore, with replication we can do an F test for lack of fit:

$$F = \frac{(RSS_0 - RSS_a)/(m - p)}{RSS_a/(n - m)} \sim F_{m-p, n-m}$$

Example: Corrosion Data Set

For a given value of iron content (x_i), we have several observations of weight loss (y_{ij})

- Fe: Iron content in percent loss
- loss: Weight loss in mg per square decimeter per day

```
data("corrosion")
corrosion[order(corrosion$Fe),]
```

```
##      Fe   loss
## 1  0.01 127.6
## 6  0.01 130.1
## 11 0.01 128.0
## 2  0.48 124.0
## 7  0.48 122.0
## 3  0.71 110.8
## 9  0.71 113.1
## 4  0.95 103.9
## 5  1.19 101.5
## 8  1.44  92.3
## 12 1.44  91.4
## 10 1.96  83.7
## 13 1.96  86.2
```

Model Comparison

The model under H_0 is compared with a more general model in where each level of X is considered as a factor.

```
## Analysis of Variance Table
##
## Model 1: loss ~ Fe
## Model 2: loss ~ factor(Fe)
##   Res.Df   RSS Df Sum of Sq    F    Pr(>F)
## 1     11 102.850
## 2      6  11.782  5    91.069 9.2756 0.008623 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
1-pf(9.2756,5,6) #There is lack of fit
```

```
## [1] 0.008622884
```

Since the p-value < 0.5 we have Lack of Fit. The model under H_0 is not adequate for this data set.

STAT 425

Polynomial Regression

Transformations of the Predictors

- We discussed transformations on the response variable Y to stabilize the variance and to normalize the response. Some of these transformations are the Box-Cox transformation, log, square-root and so on. We can apply these transformations to the predictors too.
- In this section we focus on the type of transformations of X 's which in fact generates new predictors:
 - Polynomials Regression
 - Local Polynomials (Splines) Regression
- From now on, assume we have only one predictor

Polynomial Regression

Assume $x \in \mathbb{R}$ and a model of the form:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_d x^d + \text{error}$$

d is the degree of the polynomial component. How do we choose d ?

- **Forward approach:** Keep adding terms until the last added term is not significant
- **Backward approach:** start with a large d , keep eliminating the insignificant terms starting with the highest order term.

- **Question:** Suppose we have picked a value of d , then should we test whether the other terms, x^j 's with $j = 1, \dots, d - 1$, are significant or not? Usually we do not test the significance of the lower-order terms. When we decide to use a polynomial of degree d , by default, we include all the lower-order terms in our model.
- **Why is this?**. For regression analysis, we usually do not want our results to be affected by a change of location/scale of the data (For example, suppose that the temperature is recorded in F instead of C). Suppose the data $\{y_i, x_i\}_{i=1}^n$ are generated by the model:

$$y_i = x_i^2 + e_i, \quad e_i \sim N(0, \sigma^2)$$

But the data are recorded as $\{z_i, x_i\}_{i=1}^n$, where $z_i = x_i + 2$, that is,

$$y_i = (z_i - 2)^2 + e_i = 4 - 4z_i + z_i^2 + e_i$$

So the linear term could become significant if we shift the x values

- **However**, if you have a particular polynomial function in mind, e.g., the data are collected to test a particular physics formula $Y \approx X^2 + \text{constant}$, then you should test whether you can drop the linear term.
- Or if experts believe the relationship between Y and X should be $Y \approx (X - 2)^2$, then you should check the **R** output for the model $\text{lm}(Y \sim X + I((X - 2)^2))$ to test whether you can drop the linear term and the intercept.

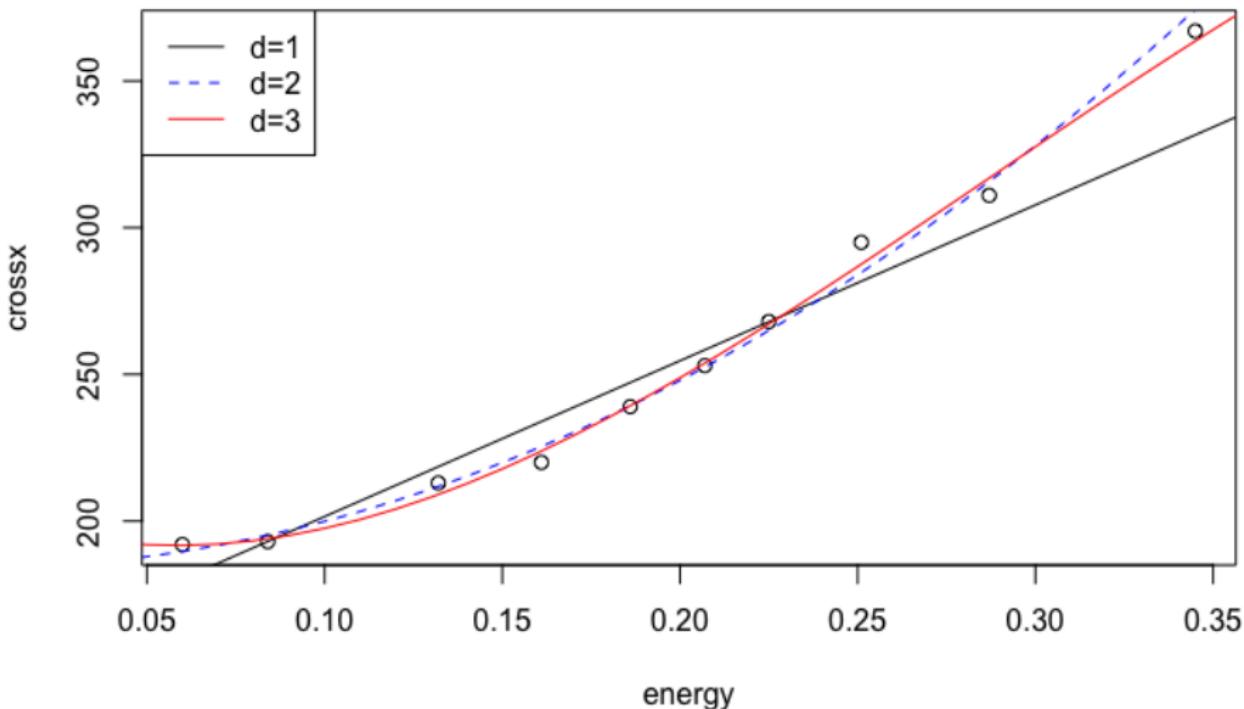
Example: strongx data set

Selecting order d for the polynomial fitted variable `crossx` as a function of `energy`

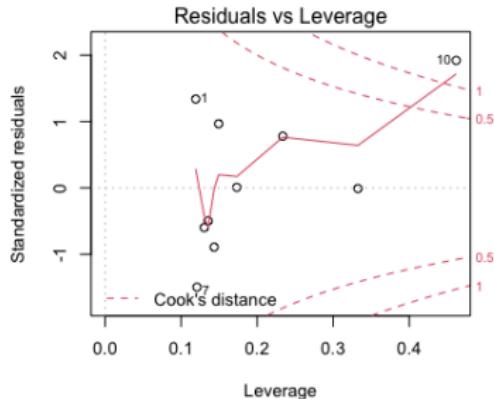
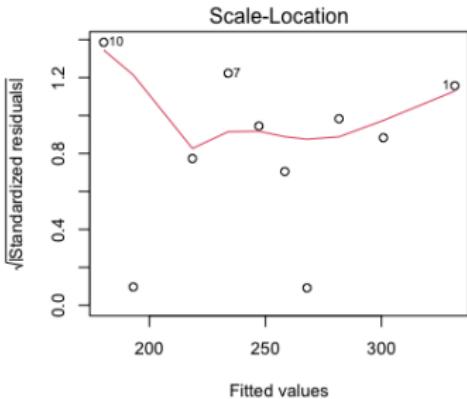
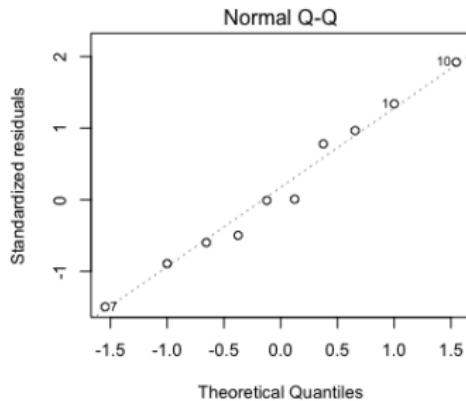
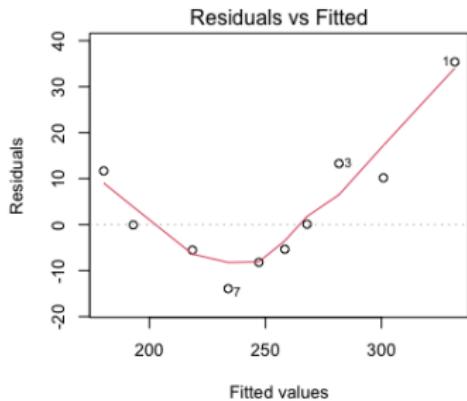
```
round(summary(lm(crossx ~ energy + I(energy^2) + I(energy^3), weights = sd^-2, strongx)  
coef, dig=3)
```

	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	204.992	12.826	15.983	0.000
## energy	-472.866	268.862	-1.759	0.129
## I(energy^2)	4504.475	1600.789	2.814	0.031
## I(energy^3)	-5220.597	2848.373	-1.833	0.117

- The cubic term is not significant. We can use a quadratic polynomial.
- Backward and forward elimination methods for d selection yield the same result ($d = 2$). This is not always the case.



Residuals for the linear model $lm(crossx \sim energy)$



Orthogonal Polynomials

- Fitting high order polynomials is generally do not recommended, since they are very unstable and difficult to interpret.
- Successive predictors x^j are highly correlated introducing multicollinearity problems.
- One way around this is to fit orthogonal polynomials of the form:

$$y_i = \beta_0 + \beta_1 z_1 + \dots + \beta_d z_d + \text{error}$$

where each $z_j = a_1 + b_2 x + \dots + \kappa_j x^j$ is a polynomial of order j . Its coefficients are chosen such that $z_i^\top z_j = 0$

- Use function **poly(.)** in **R** to fit orthogonal polynomials.

Standard polynomials vs. Orthogonal polynomials

```
# Plot polynomials with order 1 to 3
# Orthogonal polynomials
x=seq(0, 1, 0.01)
outo= poly(x, 3)
# standard polynomials
outs = cbind(x, x^2, x^3);
```

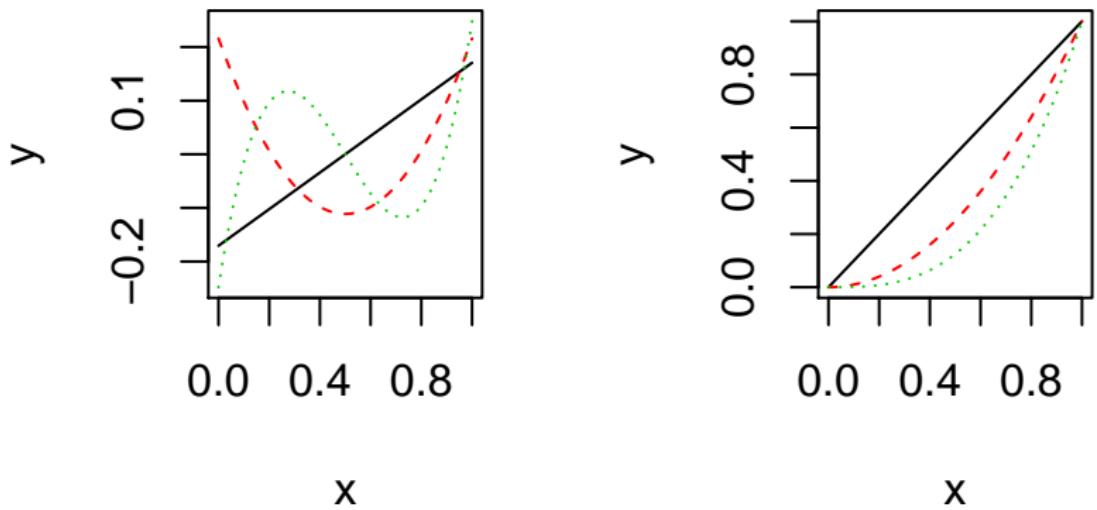


Figure: Orthogonal (left) vs. Standard (right)

Piece-wise Polynomials

- If the true mean of $E[Y|X = x] = f(x)$ is too wiggly, we might need to fit a higher order polynomial, which is not always a good idea.
- Instead we will consider **piece-wise polynomials**: we divide the range of x into several intervals, and within each interval $f(x)$ is a low-order polynomial, e.g., cubic or quadratic, but the polynomial coefficients change from interval to interval; in addition we require the overall $f(x)$ to be continuous up to certain derivatives.

Cubic Splines

We want to define a cubic spline function in the interval $[a, b]$

- Define m knots such that: $a < \xi_1 < \xi_2 < \dots < \xi_m < b$
- A function g defined on $[a, b]$ is a cubic spline with respect to (wrt) knots $\{\xi_i\}_{i=1}^m$ if:
 - ① g is a cubic polynomial in each of the $m + 1$ intervals,

$$g(x) = d_i x^3 + c_i x^2 + b_i x + a_i, \quad x \in [\xi_i, \xi_{i+1}]$$

where $i = 0 : m$, $\xi_0 = a$ and $\xi_{m+1} = b$

- ② g is continuous up to the 2nd derivative: since g is continuous up to the 2nd derivative for any point inside an interval, it suffices to check the following conditions:

$$g^{(0,1,2)}(\xi_i^+) = g^{(0,1,2)}(\xi_i^-), \quad i = 1 : m$$

This expression indicates that the function and the first and second order derivatives are continuous at the knots.

- How many free parameters we need to represent g ?

We need **4 parameters** (d_i, c_i, b_i, a_i) for each of the $(m + 1)$ intervals, but we also have **3 constraints** at each of the m knots (continuity constraints). The total number of free parameters (similar to the number of degrees of freedom) is:

$$4(m + 1) - 3m = m + 4$$

Some properties of the cubic splines

Suppose the knots $\{\xi_i\}_{i=1}^m$ are given.

- If $g_1(x)$ and $g_2(x)$ are cubic splines, the linear combination $a_1g_1(x) + a_2g_2(x)$ is also a cubic spline, where a_1 and a_2 are known constants.
That is, for a set of given knots, the corresponding cubic splines form a linear space (of functions) with $\dim(m+4)$.
- A set of basis functions for cubic splines (w.r.t knots $\{\xi_i\}_{i=1}^m$) is given by:

$$\begin{aligned} h_0(x) &= 1; h_1(x) = x; \\ h_2(x) &= x^2; h_3(x) = x^3; \\ h_{i+3}(x) &= (x - \xi_i)_+^3, \quad i = 1, 2, \dots, m \end{aligned}$$

- That is, any cubic spline $f(x)$ can be uniquely expressed as:

$$f(x) = \beta_o + \sum_{j=1}^{m+3} \beta_j h_j(x)$$

- There are many other choices of basis functions. For example, **R** uses the **B-splines basis functions**.

Natural Cubic Spline (NCS)

- A cubic spline on $[a, b]$ is a NCS if its second and third derivatives are zero at a and b .
- This condition implies that NCS is a linear function in the two extreme intervals $[a, \xi_1]$ and $[\xi_m, b]$. The linear functions in the two extreme intervals are completely determined by their neighboring intervals.
- The degree of freedom of NCS's with m knots is m :

$$4(m + 1) - 3m - 4 = m$$

(4 additional constraints)

- For a curve estimation problem with data $(x_i, y_i)_{i=1}^n$, if we put n knots at the n data points (assumed to be unique), then we obtain a smooth curve (using NCS) passing through all y 's.

STAT 425

Regression Splines

Regression Splines

Recall that for a set of given knots, the corresponding cubic splines form a linear space (of functions) with $\dim (m + 4)$.

Regression splines uses a basis expansion approach:

$$g(x) = \beta_1 h_1(x) + \beta_2 h_2(x) + \dots + \beta_p h_p(x)$$

- If cubic splines are used as basis functions $p = m + 4$
- If Natural Cubic Splines (NCS) are used as basis functions $p = m$

We can represent the model on the observed n data points using matrix notation:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}_{n \times 1} = \begin{pmatrix} h_1(x_1) & h_2(x_1) & \dots & h_p(x_1) \\ h_1(x_2) & h_2(x_2) & \dots & h_p(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(x_n) & h_2(x_n) & \dots & h_p(x_n) \end{pmatrix}_{n \times p} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix}_{p \times 1}$$

Our design matrix is the matrix \mathbf{F} of basis functions. We can find $\hat{\boldsymbol{\beta}}$ by solving the problem:

$$\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{F}\boldsymbol{\beta}\|^2$$

Regression Splines in R

- We can obtain the design matrix \mathbf{F} by commands `bs` (B-splines) or `ns` (NCS) in R, and then call the regression function `lm`.
- To select the number of knots we can use K -fold cross-validation (CV) (More on this later).

B-splines basis functions

Understand how **R** counts the degrees-of-freedom

- To generate a B-spline basis for a given set of x_i 's, you can use the command `bs`.
- You can tell **R**, the `location of knots`.
- Or you can tell **R** the `df`. Recall that a cubic spline with m knots has $m + 4$ df, so we need $m = df - 4$ knots.
- By default, **R** puts knots at the $1/(m+1), \dots, m/(m+1)$ quantiles of $x_{1:n}$.

Example: B-splines with 5 knots (9 df)

Define knots in the interval [0, 2] and get matrix F

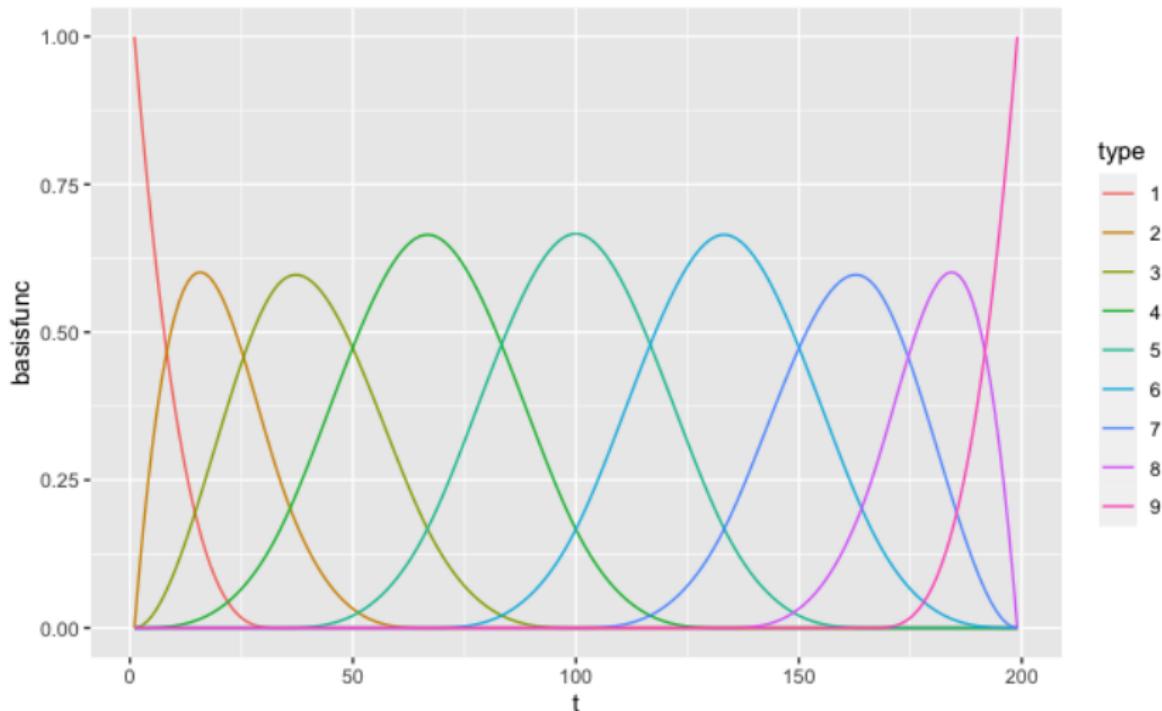
```
library(splines)
x=(1:199)/100
n = length(x)
m=5
myknots= 2*(1:m)/(m+1)
myknots

## [1] 0.3333333 0.6666667 1.0000000 1.3333333 1.6666667

#Using the Intercept option
F=bs(x,knots=myknots, intercept=TRUE)
dim(F)

## [1] 199    9
```

B-splines basis functions with df=9 (knots=5)



- How **R** counts the df might be confusing. The df in command **bs** actually means the number of columns of the design matrix returned by bs. So if the intercept is not included in the design matrix (which is the default), then the df in command bs is equal to the real df minus 1.

```
#No Intercept option
F=bs(x,knots=myknots)
dim(F)

## [1] 199    8
```

The following three design matrices (the first two are of $n \times 5$ and the last one is of $n \times 6$) correspond to the same regression model with cubic splines of $\text{df}=6$.

```
F1=bs(x, knots=quantile(x, c(1/3, 2/3)))
dim(F1)

## [1] 199    5

F2=bs(x, df=5)
dim(F2)

## [1] 199    5

F3=bs(x, df=6, intercept=TRUE)
dim(F3)

## [1] 199    6
```

NCS Basis functions

- To generate a NCS basis for a given set of x_i 's, use the command `ns`.
- Recall that the linear functions in the two extreme intervals are totally determined by the other cubic splines. So data points in the two extreme intervals (i.e., outside the two boundary knots) are wasted since they do not affect the fitting. Therefore, by default, **R** puts the two boundary knots as the min and max of the x_i 's
- You can tell **R** the location of knots, which are the interior knots. Recall that a NCS with m knots has m df. So the df is equal to the number of (interior) knots plus 2, where 2 means the two **boundary knots**.

- Or you can tell **R** the df. If $intercept = \text{TRUE}$, then we need $m = df - 2$ knots, otherwise we need $m = df - 1$ knots. Again, by default, **R** puts knots at the $1/(m+1), \dots, m/(m+1)$ quantiles of $x_{1:n}$.
- The following three design matrices (the first two are of $n \times 3$ and the last one is of $n \times 4$) correspond to the same regression model with NCS of $df = 4$.

```
F1=ns(x, knots=quantile(x, c(1/3, 2/3)))
dim(F1)

## [1] 199    3

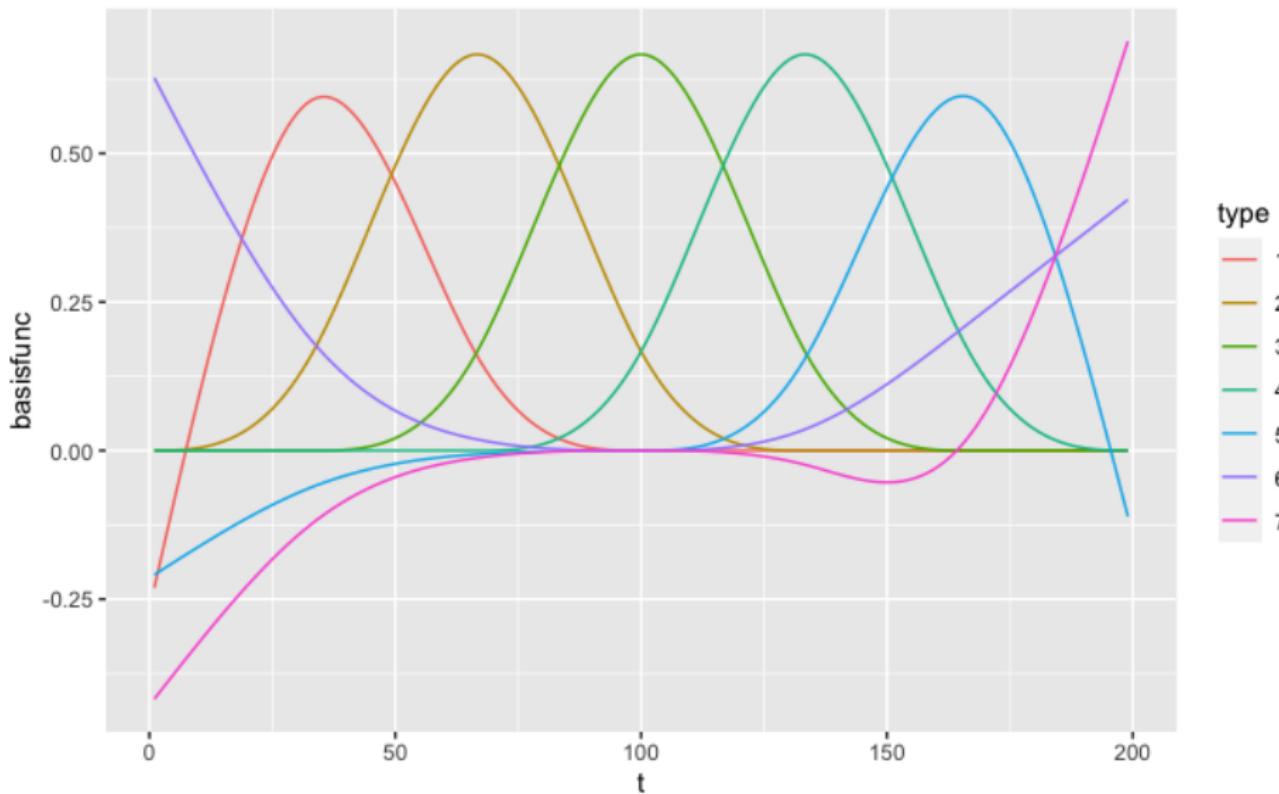
F2=ns(x, df=3)
dim(F2)

## [1] 199    3

F3=ns(x, df=4, intercept=TRUE)
dim(F3)

## [1] 199    4
```

Natural cubic splines basis functions with df=7 (knots=5)



Example: Birth rate data set

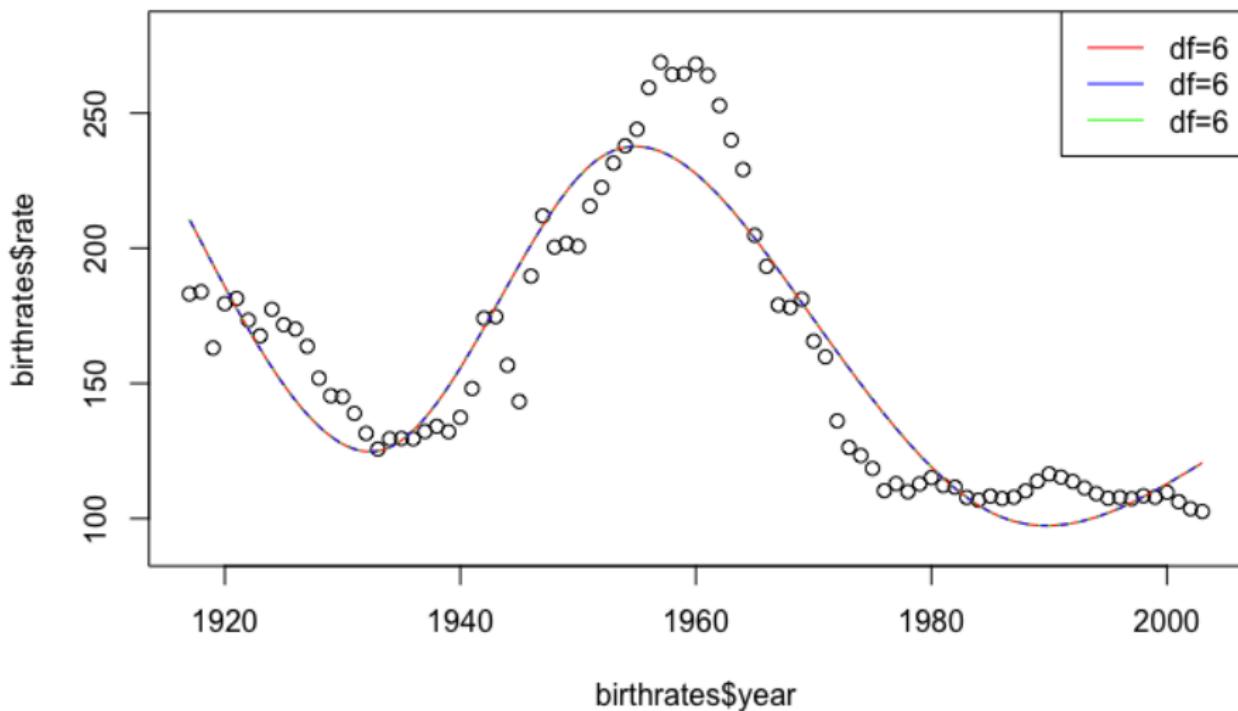
This data set lists the number of live births per 10,000 23-year-old women in the United States between 1917 and 2003.

Use NCS with number of knots=4 and df=6 (including the two boundary knots)

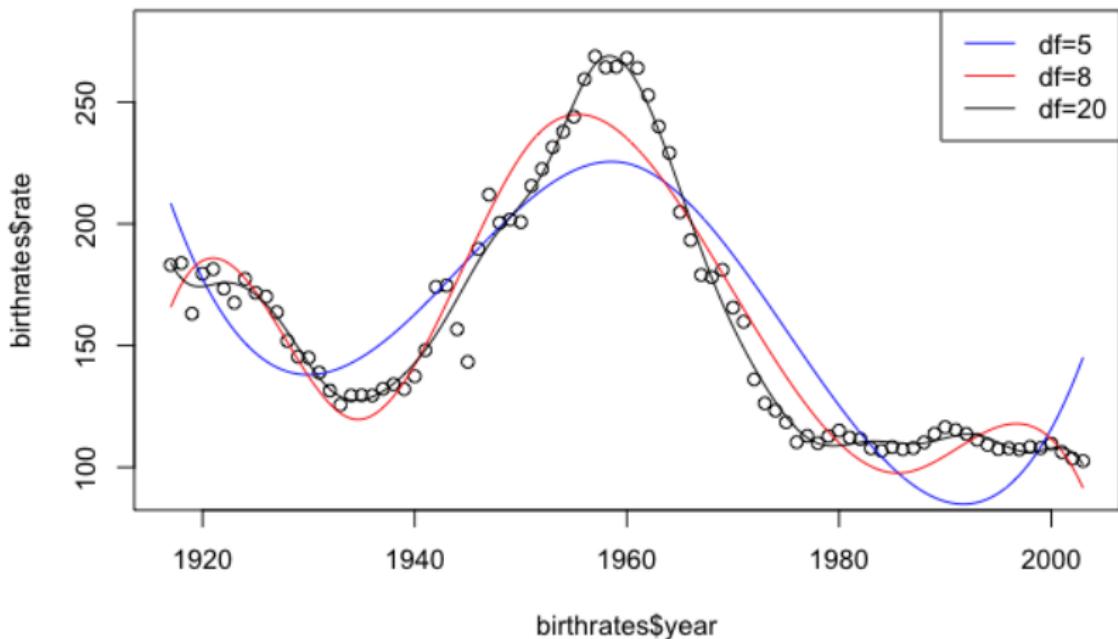
```
source("birthrates.txt");
birthrates = as.data.frame(birthrates)
names(birthrates) = c("year", "rate")
# The following fitted models provide the same results
fit1=lm(rate~ns(year, knots=quantile(year, (1:4)/5)),
data=birthrates);
fit2=lm(rate~ns(year, df=5), data=birthrates);
fit3=lm(rate~ns(year, df=6, intercept=TRUE),
data=birthrates)
```

Birth rate example

All fitted models provide the same results. However the number of knots might not be optimal.



Comparing different number of knots in the birth rate example



A good way to select the optimal number of knots (or df) is to use K-fold Cross-Validation:

- ① Set a fixed number of knots (or df)
- ② Divide the set of observations into k groups (or *folds*)
- ③ Leave the first fold as a validation set (not used to fit the model). Fit the Regression Spline with a fixed number of knots using the remaining $k - 1$ folds.
- ④ Calculate the Mean Square Error for fold 1: MSE_1
- ⑤ Repeat the previous steps k times. Each time a new validation set is used to calculate MSE_i
- ⑥ Calculate the average k -fold Cross-Validation error:
$$CV(k) = \frac{1}{k} \sum_{i=1}^k MSE_i$$
- ⑦ Repeat 2 to 6 with a new number of knots (or df)
- ⑧ Select the number of knots that minimizes the k -fold CV error or $CV(k)$

STAT 425

ANCOVA Models

ANCOVA Models

- ANCOVA stands for **AN**alysis of **COVA**riance: These are regression problems where some predictors are quantitative (i.e. numerical) and some are qualitative (i.e. categorical).
- For simplicity we will focus on examples with just two predictors: X (numerical) and D (categorical).

A two-level example

- Suppose we model the response Y by two predictors X and D , where X is a numerical variable and D is categorical with two levels (such as male, female)
- You can code D as 0 or 1, e.g., 1 for male and 0 for female.
Note: you can code the two levels using any two different values, which will not change \hat{y} , but the interpretation of the estimated coefficients.
- In general, a factor with k levels corresponds to $k - 1$ variables, when there is an additional intercept.

Consider the general model:

$$y = \beta_0 + \beta_1 x + \beta_2 d + \beta_3(x \cdot d) + e$$

The cats example revisited

We want to build a model to predict Hwt based on Bwt . For simplicity, assume we have $n = 4$ observations and the first two are female. **What are the possible regression models?**

- ① **Coincident regression lines** (the simplest model): the same regression line for both groups, i.e., the categorical variable D has no effect on Y :

$$y = \beta_0 + \beta_1 x + e$$

- 1' **Two-mean model** (another simplest model): the numerical variable X has no effect on Y :

$$y = \beta_0 + \beta_2 d + e = \begin{cases} \beta_0 + e, & d = 0 \\ (\beta_0 + \beta_2) + e, & d = 1 \end{cases}$$

- ② Parallel regression lines: the categorical variable D **only** changes the intercept, i.e., it produces an additive effect only:

$$y = \beta_0 + \beta_2 d + \beta_1 x + e = \begin{cases} \beta_0 + \beta_1 x + e, & d = 0 \\ (\beta_0 + \beta_2) + \beta_1 x + e, & d = 1 \end{cases}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & x_1 \\ 1 & 0 & x_2 \\ 1 & 1 & x_3 \\ 1 & 1 & x_4 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_2 \\ \beta_1 \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix}$$

β_2 : measures the **change** of the additive effect (i.e., difference of the intercept).

Alternative choices for the design matrix (they should give us the same \hat{y})

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & x_1 \\ 1 & 0 & x_2 \\ 1 & 1 & x_3 \\ 1 & 1 & x_4 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_2 \\ \beta_1 \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & x_1 \\ 1 & 1 & x_2 \\ 1 & 2 & x_3 \\ 1 & 2 & x_4 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_2 \\ \beta_1 \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix}$$

- ③ Regression lines with equal intercepts but different slopes: the categorical variable D only changes the effect of X on Y :

$$y = \beta_0 + \beta_1 x + \beta_3(x \cdot d) + e$$

$$= \begin{cases} \beta_0 + \beta_1 x + e, & d = 0 \\ \beta_0 + (\beta_1 + \beta_3)x + e, & d = 1 \end{cases}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} 1 & x_1 & 0 \\ 1 & x_2 & 0 \\ 1 & x_3 & x_3 \\ 1 & x_4 & x_4 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_3 \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix}$$

β_3 : measures the **change** of the slope.

- ④ Unrelated regression lines (the most general model): the categorical variable D produces an additive change in Y and also changes the effect of X on Y . Then, should we just divide the data into two sets and run lm separately on them?

$$y = \beta_0 + \beta_1 x + \beta_2 d + \beta_3(x \cdot d) + e$$

$$= \begin{cases} \beta_0 + \beta_1 x + e, & d = 0 \\ (\beta_0 + \beta_2) + (\beta_1 + \beta_3)x + e, & d = 1 \end{cases}$$

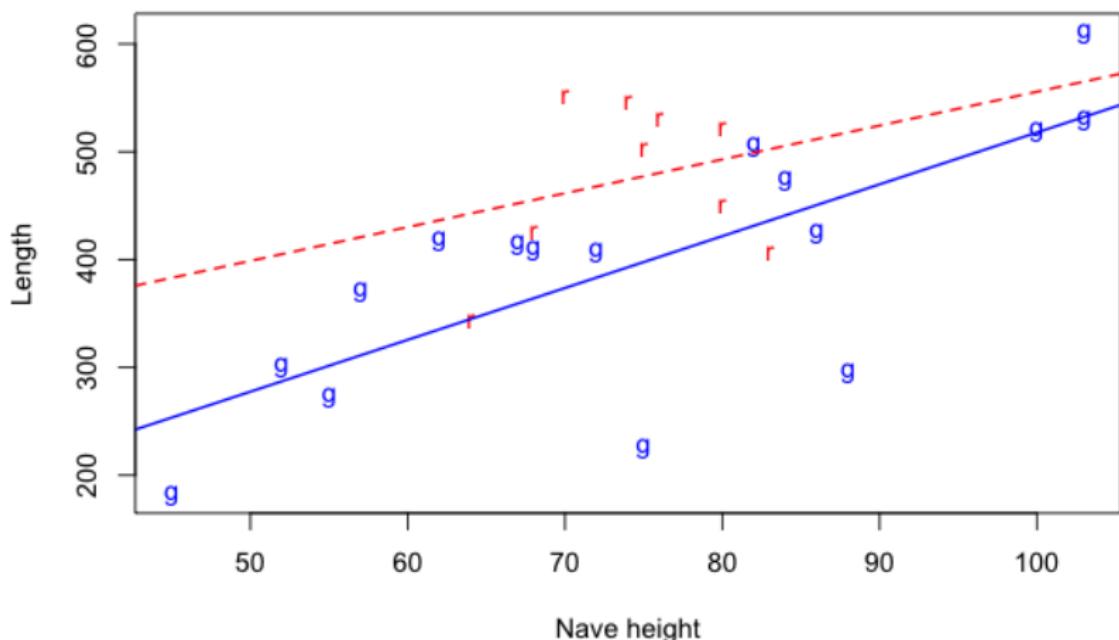
$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & x_1 & 0 \\ 1 & 0 & x_2 & 0 \\ 1 & 1 & x_3 & x_3 \\ 1 & 1 & x_4 & x_4 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_2 \\ \beta_1 \\ \beta_3 \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix}$$

How to interpret the LS coefficients from model 4?

- The usual: β_1 measures the effect of X_1 on Y when other predictors are held unchanged, does not make much sense for models with interactions. We cannot change x while holding d and $(x \cdot d)$ unchanged.
- Check the Cathedral R example.

Example: cathedral data set

$x = \text{nave height}$; $y = \text{total length in feet}$ for English medieval cathedrals. r represents Romanesque style and g represents Gothic style.



Fit the Full Model

full model: different intercepts and different slopes. How to interpret the coefficients?

```
g.full = lm(y~x+style+x:style, data=cathedral)
# same as lm(y~x*style, data=cathedral)
summary(g.full)

##
## Call:
## lm(formula = y ~ x + style + x:style, data = cathedral)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -172.68   -30.22    23.75   55.78   89.50
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  37.111     85.675   0.433  0.669317    
## x            4.808      1.112   4.322  0.000301 ***  
## styler      204.722    347.207   0.590  0.561733    
## x:styler    -1.669      4.641  -0.360  0.722657    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 79.11 on 21 degrees of freedom
## Multiple R-squared:  0.5412, Adjusted R-squared:  0.4757 
## F-statistic: 8.257 on 3 and 21 DF,  p-value: 0.0008072
```

Which model to pick?

You can use F-test to select the appropriate model.

- First test whether the interaction term is significant:

$$H_0 : \text{model 2} \quad H_a : \text{model 4}$$

If reject the null, stop and take model 4.

Otherwise, decide whether you can further reduce model 2 to model 1 or model 1'.

- What if β_3 (the interaction) is significant, but, β_1 or β_2 , is not significant? What about model 3?

- The **Hierarchical Rule** for interactions: an interaction term will be included in a model only if all its main effects have been included. Due to this rule, we would include both β_1 and β_2 , once β_3 is significant.
- In practice we could test $\beta_1 = 0$ or $\beta_2 = 0$. We just need to understand what the model looks like when β_1 or β_2 equals zero.

- When $\beta_1 = 0$ (it doesn't mean that X is not significant):

$$y = \begin{cases} \beta_0 + e, & d = 0 \\ (\beta_0 + \beta_2) + \beta_3 x + e, & d = 1 \end{cases}$$

- When $\beta_2 = 0$ (gives us model 3; it does not mean D is not significant):

$$y = \begin{cases} \beta_0 + \beta_1 x + e, & d = 0 \\ \beta_0 + (\beta_1 + \beta_3)x + e, & d = 1 \end{cases}$$

A Multi-level example

- Model the response Y by two predictors X and D , where X is a numerical variable and D is categorical with k levels.
- We need to generate $k - 1$ dummy variables: D_2, \dots, D_k where:

$$D_i = \begin{cases} 0 & \text{if not level } i \\ 1 & \text{if level } i \end{cases}$$

Level 1 is the reference level.

The main purpose of the analysis is to decide which of the following models fits the data:

- Model 0: $Y \sim 1$
- Model 1a: $Y \sim X$
- Model 1b: $Y \sim D$
- Model 2: $Y \sim D + X$
- Model 4: $Y \sim D + X + D : X$

The major tool is the F -test. Note that when D has more than two levels, the difference between model parameter number may not be one, so t -test is no longer appropriate.

- 1) Compare models:

$$H_0 : Y \sim X + D \quad vs. \quad H_a : Y \sim D + X + D : X$$

If the interaction $D : X$ is significant, stop.

- 2a) If X is significant, keep X .
- 2b) If D is significant, keep D .
- 3) If neither X nor D are significant, report the intercept model $Y \sim 1$.

2a) and 2b) are a little bit tricky:

2a) Is X significant?

Test the marginal contribution of X :

$$H_0 : Y \sim 1 \quad vs. \quad H_a : Y \sim X$$

Test the contribution of X in addition to D :

$$H_0 : Y \sim D \quad vs. \quad H_a : Y \sim X + D$$

2b) Is D significant?

$$H_0 : Y \sim 1 \quad vs. \quad H_a : Y \sim D$$

$$H_0 : Y \sim X \quad vs. \quad H_a : Y \sim D + X$$

Sequential ANOVA

We can use the **anova** function to get sequential F-tests. The sequence of F -tests given by *anova(lm(Y ~ X + D + X : D))*

H_0	H_a
$Y \sim 1$	$Y \sim X$
$Y \sim X$	$Y \sim X + D$
$Y \sim X + D$	$Y \sim X + D + X : D$

The sequence of F -tests given by $\text{anova}(lm(Y \sim D + X + X : D))$ is given by:

H_0	H_a
$Y \sim 1$	$Y \sim D$
$Y \sim D$	$Y \sim D + X$
$Y \sim D + X$	$Y \sim D + X + X : D$

Be aware that: Some of the F -stats and p -values from the sequential ANOVA table are different from the ones we calculated based on usual F -test (we learned) for comparing two nested models.

Suppose we want to compare:

$$H_0 : Y \sim X \quad vs \quad H_a : Y \sim X + D$$

- The usual F -stat is given by:

$$\frac{(RSS_0 - RSS_a)/(k - 1)}{RSS_a/(n - p_a)} = \frac{(RSS_0 - RSS_a)/(k - 1)}{\hat{\sigma}_a^2}$$

which follows $F_{k-1, n-p_a}$ under the null hypothesis. k is the total number of categories of variable D

- The F -stat from the sequential ANOVA table:

$$\frac{(RSS_0 - RSS_a)/(k - 1)}{RSS_A/(n - p_A)} = \frac{(RSS_0 - RSS_a)/(k - 1)}{\hat{\sigma}_A^2}$$

which follows $F_{k-1, n-p_A}$ under the null hypothesis, where RSS_A denotes the RSS from the biggest model $Y \sim X + D + X : D$ and $p_A = 2k$

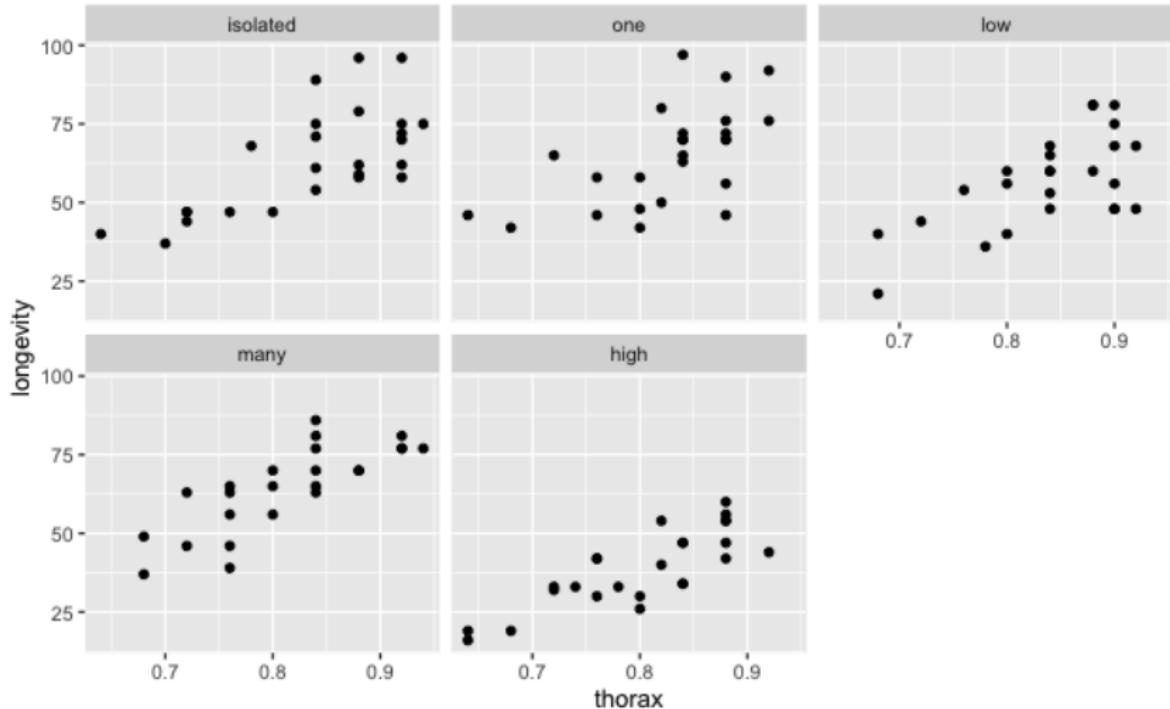
Example: fruitfly data set

The *fruitfly* data frame has 9 rows and 3 columns. 125 fruit flies were divided randomly into 5 groups of 25 each. The response is the longevity of the fruit fly in days. The following groups or categories describe the sexual activity:

- One group was kept solitary (isolated)
- One group was kept with a virgin female each day (low)
- One group was kept with 8 virgin females per day (high)
- One group was kept with one pregnant female per day (one)
- One group was kept with eight pregnant female per day (many)

Pregnant fruit flies will not mate. The thorax length of each male was measured as this was known to affect longevity. One observation in the many group has been lost. So the total sample size is 124.

Response: Longevity (days); Predictors: Thorax length (numerical) and activity (categorical)



Sequential ANOVA

```
lmod= lm(longevity ~ thorax * activity, fruitfly)
# summary(lmod)
anova(lmod)
```

```
## Analysis of Variance Table
##
## Response: longevity
##              Df  Sum Sq Mean Sq F value    Pr(>F)
## thorax          1 15003.3 15003.3 130.733 < 2.2e-16 ***
## activity         4  9634.6  2408.6  20.988 5.503e-13 ***
## thorax:activity  4     24.3      6.1    0.053    0.9947
## Residuals       114 13083.0    114.8
## ---
## Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Example: F -stat for the activity variable: $F = \frac{9634.6/4}{13083.0/114} = 20.988$.
Under the null hypothesis that $\beta_{activity}$ is non-significant, $F \sim F_{4,114}$.
From these results we conclude that the interaction term is not significant.

Additive Model

```
lmod.add = lm(longevity ~ thorax + activity, fruitfly)
summary(lmod.add)
```

```
##  
## Call:  
## lm(formula = longevity ~ thorax + activity, data = fruitfly)  
##  
## Residuals:  
##       Min     1Q   Median     3Q    Max  
## -26.108 -7.014 -1.101  6.234 30.265  
##  
## Coefficients:  
##             Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -48.749     10.850  -4.493 1.65e-05 ***  
## thorax       134.341     12.731   10.552 < 2e-16 ***  
## activityone   2.637      2.984    0.884   0.3786  
## activitylow  -7.015      2.981   -2.353   0.0203 *  
## activitymany  4.139      3.027    1.367   0.1741  
## activityhigh -20.004     3.016   -6.632 1.05e-09 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 10.54 on 118 degrees of freedom  
## Multiple R-squared:  0.6527, Adjusted R-squared:  0.638  
## F-statistic: 44.36 on 5 and 118 DF,  p-value: < 2.2e-16
```

STAT 425

Variable Selection

Variable Selection

- Consider a MLR model $Y \sim 1 + X_1 + X_2 + \cdots + X_p$ where we have p non-intercept predictors. Later we drop the intercept for notational simplicity.
- In many applications, we have a lot of potential explanatory variables, i.e., p is large and we could even have $p \gg n$, but only a small portion of the p variables are believed to be relevant to Y .
- Of interest is to find the following subset of the p predictors:

$$S = \{j : \beta_j \neq 0\}$$

- In some applications as sales prediction, the key question we need to answer is to identify this set S , e.g., which variables among the p variables are really effective for boosting the sales (Y).
- If our goal is simply to do well on prediction, then should we care about variable selection?

Recall that the LS estimate $\hat{\beta}$ is unbiased, i.e., estimates for irrelevant $\hat{\beta}_j$ (with $j \in S^c$) will eventually go to zero anyway. To understand this, let's examine the [training and the test errors](#).

Test Error vs. Training Error

- Training data: $(\mathbf{x}_i, y_i)_{i=1}^n$
- Test data: $(\mathbf{x}_i, y_i^*)_{i=1}^n$ is an independent (imaginary) data set collected at the same location \mathbf{x}_i 's (also known as, **in-sample prediction**)
- Assume the data comes from a linear model:
 $\mathbf{y}_{n \times 1}$, $\mathbf{y}_{n \times 1}^*$ are i.i.d $\sim N_n(\boldsymbol{\mu}, \sigma^2 \mathbf{I}_n)$ and $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}$
- We can also write:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}$$

$$\mathbf{y}^* = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}^*$$

with $\mathbf{e}_{n \times 1}$, $\mathbf{e}_{n \times 1}^*$ i.i.d $\sim N_n(\mathbf{0}, \sigma^2 \mathbf{I}_n)$ are independent.

Mean square testing error and Mean square training error

$$\begin{aligned} E[\text{Test Error}] &= E\|\mathbf{y}^* - \mathbf{X}\hat{\boldsymbol{\beta}}\|^2 \\ &= E\|(\mathbf{y}^* - \mathbf{X}\boldsymbol{\beta}) + (\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\hat{\boldsymbol{\beta}})\|^2 \\ &= E\|\mathbf{y}^* - \boldsymbol{\mu}\|^2 + E\|\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\hat{\boldsymbol{\beta}}\|^2 \\ &= E\|\mathbf{e}^*\|^2 + \text{Tr}(\mathbf{X}Cov(\hat{\boldsymbol{\beta}})\mathbf{X}^\top) \\ &= n \cdot \sigma^2 + \sigma^2 \text{Tr}\mathbf{H} = n \cdot \sigma^2 + p \cdot \sigma^2 \end{aligned}$$

$$\begin{aligned} E[\text{Train Error}] &= E\|\mathbf{y} - \hat{\mathbf{y}}\|^2 = E\|(\mathbf{I} - \mathbf{H})\mathbf{y}\|^2 \\ &= \text{Tr}((\mathbf{I} - \mathbf{H})Cov(\mathbf{y})(\mathbf{I} - \mathbf{H})^\top) \\ &= \sigma^2 \text{Tr}((\mathbf{I} - \mathbf{H})) = (n - p) \cdot \sigma^2 \end{aligned}$$

From the previous equations we can conclude:

- Testing error **increases** with p
- Training error **decreases** with p
- When adding more variables (p large) the testing error increases.
If our goal is pure prediction, adding more variables to matrix \mathbf{X} is not the best option. We should remove some irrelevant variables.
- The analysis on the previous slide might indicate that the best model (i.e., the one with the smallest expected test error), is the intercept-only model with $p = 0$.
- This of course is not true. The previous analysis is based on the assumption that the mean of \mathbf{y} is in the column space of \mathbf{X} , i.e., there exists some coefficient vector $\boldsymbol{\beta}$ such that $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}$. In general, we run a linear regression model using only a subset of the columns of \mathbf{X} . This means there will be an additional Bias term.

Model Index γ

- Index each model (i.e., each subset of the p variables) by a p -dimensional binary vector γ :

$$\gamma = (\gamma_1, \gamma_2, \dots, \gamma_p), \quad \gamma_j = 0/1$$

where $\gamma_j = 1$ indicates that X_j is included in the model, and $\gamma_j = 0$ otherwise.

- So there are a total of 2^p possible subsets or sub-models. In particular

$$\gamma = (1, 1, \dots, 1)$$

refers to the full model including all p variables (largest dim), and

$$\gamma = (0, 0, \dots, 0)$$

refers to the intercept-only model (smallest dim).

Suppose that $\mu = \mathbf{X}\beta$ where μ is the mean of y . If we fit the data y with respect to model γ , i.e., we fit a linear model with a sub-design matrix X_γ where X_γ contains only columns from \mathbf{X} such that $\gamma_j = 1$. We can show that the Testing Error and the Training error for model γ are:

$$E[\text{Test Error}] = n\sigma^2 + p\sigma^2 + \text{Bias}_\gamma$$
$$E[\text{Training Error}] = n\sigma^2 - p\sigma^2 + \text{Bias}_\gamma$$

Bigger model (i.e., p large) \rightarrow small Bias, but large variance ($p\sigma^2$);
Smaller model (i.e., p small) \rightarrow large Bias, but small variance ($p\sigma^2$).
So to reduce the test error (i.e., prediction error), the key is to find the best **trade-off** between Bias and Variance.

Model selection procedures

- **Testing-based procedures:** Select best model based on statistical tests for model comparison.
- **Criterion-based procedures:** Select best model based on an information criteria (combining model fit and model complexity) for model comparison.

Testing-based procedures

Backward elimination

- ① Start with all the predictors in the model.
- ② Remove the predictor with highest $p - \text{value} > \alpha_0$ (most insignificant).
- ③ Refit the model, and repeat the above process.
- ④ Stop when all $p - \text{values} \leq \alpha_0$.
(α_0 is often set to 15% or 20% which is higher than usual)

Testing-based procedures

Forward selection

- ① Start with the intercept-only model.
- ② For all predictors not in the model, check their p -value if being added to the model. Add the one with the lowest p – value $\leq \alpha_0$ (most significant).
- ③ Refit the model, and repeat the above process.
- ④ Stop when no more predictors can be added.

Pros and Cons of Testing-based procedures

- Main advantage: Computation cost is low.
- Due to the “one-at-a-time” nature of adding/dropping variables, this type of procedures does not compare all possible models. So it’s possible to miss the “optimal” model.
- It’s not clear how to choose α_0 , the cut-off for p -values.

Criterion-based procedures

- ① Score each model according to an information criteria
- ② Use a searching algorithm to find the optimal model

Model selection criteria/scores often takes the following form:

Training error + Complexity-penalty

- In the context of linear regression models, complexity of a model increases with the number of predictor variables (i.e., p_γ).
- Why we do not use R^2 or RSS ?

Model Selection Criteria

- AIC/BIC - lower is better

$$AIC := -2 \times \loglik_{\gamma} + 2p_{\gamma}$$

$$BIC := -2 \times \loglik_{\gamma} + \log(n)p_{\gamma}$$

where p_{γ} is the number of predictors included in model γ .

- R functions AIC and BIC use $p_{\gamma} + 1$ instead of p_{γ} , due to estimating σ^2 , but the extra constant does not affect the comparison between models.
- For the Gaussian linear regression model:

$$-2 \times \loglik_{\gamma} = n \log \frac{RSS_{\gamma}}{n} + \text{constant}$$

- When n is large, adding predictors costs more in BIC than AIC. So AIC tends to pick a bigger model than BIC.

- Adjusted- R^2 for model γ

$$\begin{aligned}
 R_a^2 &= 1 - \frac{RSS/(n - p_\gamma - 1)}{TSS/(n - 1)} \\
 &= 1 - (1 - R^2) \left(\frac{n - 1}{n - p_\gamma - 1} \right) \\
 &= 1 - \frac{\hat{\sigma}_\gamma^2}{\hat{\sigma}_0^2}
 \end{aligned}$$

The higher the R_a^2 the better.

- Mallow's C_p

$$C_p = \frac{RSS_\gamma}{\hat{\sigma}^2} + 2p_\gamma - n$$

where $\hat{\sigma}^2$ is the estimate of the error variance from the full model. Mallow's C_p behaves very similar to AIC.

Searching Algorithms

- **Leaps and Bounds:** return the **global optimal solution** among all possible models, but only feasible for less than 50 variables.
 - Find the p models with the smallest RSS amongst all models of the same size¹.
 - Then evaluate the score on the p models and report the optimal one.

¹Note that step 1, we do not need to visit every model. For example, suppose we know that $RSS(X1, X2) < RSS(X3, X4, X5, X6)$; then we do not need to visit any size 2 or 3 sub-models of set $(X3, X4, X5, X6)$, which can be **leaped over**

- **Greedy algorithms**: fast, but only return a **local optimal** solution (which might be good enough in practice).
 - **Backward**: start with the full model and sequentially delete predictors until the score does not improve.
 - **Forward**: start with the null model and sequentially add predictors until the score does not improve.
 - **Stepwise**: consider both deleting and adding one predictor at each stage.

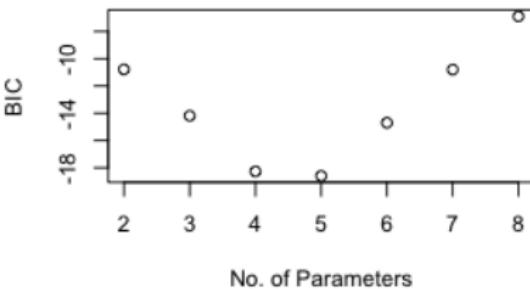
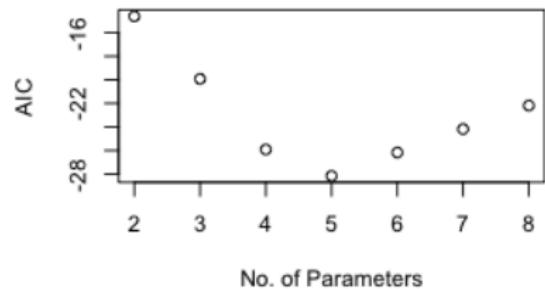
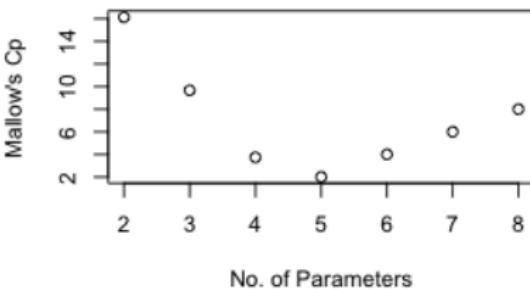
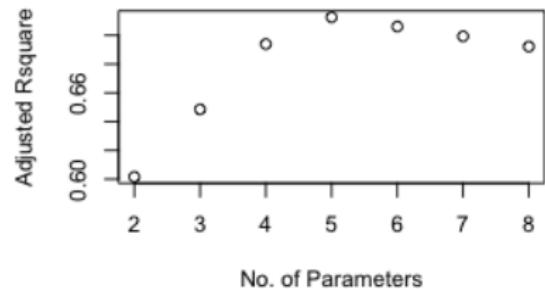
Example: Life Expectancy data set

state.77 data set from library **datasets**

```
# Read state.x77 data set
statedata=data.frame(state.x77, row.names=state.abb,
                      check.names=T)
g = lm(Life.Exp ~ ., data=statedata)
```

Leaps and Bounds method

Use function *regsubsets* from library **leaps** to evaluate different scores for sub-sets of models up to size p (including the intercept). In this example a model of size $p = 5$ is selected by all criteria. This is not always the case.



Searching methods

Use function `step` from the **stats** library to apply searching algorithms based on the AIC (default) or BIC criteria ($k = \log(n)$). The option `direction=both` uses the Stepwise searching algorithm. You can also use the options `direction=forward` and `direction=backward`.

```
```{r}
step(g, direction="both")
step(g, direction="both", k=log(n))
```
```

STAT 425

Shrinkage Methods for Regularized Regression

Shrinkage Methods - Regularized Regression

What to do if we have too many predictors?

- We have already discussed that too many predictors can create collinearity problems.
- Increasing the number of predictors might increase the **prediction error**.
- More predictors do not necessarily mean a better model, but more predictors would mean more information, and less **bias**.

We study three different methods to **shrink** the number of predictors in order to find a trade-off between model bias and prediction error.

- Principal Components Regression
- Ridge Regression
- Lasso Regression

Principal Components Regression

What to do when we have too many predictors?:

- Perform dimensionality reduction in the predictor space.
- Be aware that predictors might be highly correlated.
- Take matrix \mathbf{X} of predictors and center the columns of \mathbf{X} to have zero mean. Consider \mathbf{X} with no intercept column.
- Find directions of greater variation in the data.

Principal Component Analysis:

The steps to find directions of greater variation in matrix \mathbf{X} :

- Find \mathbf{u}_1 to maximize variance of $\mathbf{X}\mathbf{u}_1$ subject to $\mathbf{u}_1^\top \mathbf{u}_1 = 1$.
- Find \mathbf{u}_2 to maximize variance of $\mathbf{X}\mathbf{u}_2$ subject to $\mathbf{u}_1^\top \mathbf{u}_2 = 0$ and $\mathbf{u}_2^\top \mathbf{u}_2 = 1$.
- Continue looking for directions of greatest variation in the data which are orthogonal to the previous ones.
- Continue until the total number of dimensions is exhausted.

The principal components are given by the columns of matrix \mathbf{Z} , where

$$\mathbf{Z} = \mathbf{X} [\mathbf{u}_1 \mathbf{u}_2 \cdots \mathbf{u}_m] = \mathbf{XU}$$

\mathbf{z}_i and \mathbf{u}_i are the columns of \mathbf{Z} and \mathbf{U} respectively. \mathbf{U} is called the **rotation matrix**. \mathbf{Z} is a version of the data rotated in such a way that the resulting principal components are orthogonal.

- Each Principal Component is a [linear combination](#) of the original variables $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$ with weights given by each column \mathbf{u}_i of matrix \mathbf{U} :

$$\mathbf{z}_i = u_{1i}\mathbf{x}_1 + u_{2i}\mathbf{x}_2 + \dots + u_{mi}\mathbf{x}_m$$

- PCs are very sensitive to outliers.
- The [Mahalanobis distance](#) can be used to measure the distance of a point to the data mean, after adjusting for correlation in the data.
- Under the multivariate normality assumption in m dimensions, the [Mahalanobis distance](#) $d_i = \sqrt{(\mathbf{x}_i - \mu)^\top \Sigma^{-1} (\mathbf{x}_i - \mu)}$ can be estimated using the sample estimators for μ and Σ and the quantity d_i^2 follows a χ_m^2 distribution. This can be used to detect outliers in higher dimensions.

In Principal Components Regression (PCR):

- Replace model $Y \sim X$ by the model $Y \sim Z$
- Only need to use the first few columns of Z as predictors
- Interpretation of the PCAs as predictors might be challenging.
We need to use the values of \mathbf{u}_i in the rotation matrix (also called the **loadings**) for interpretation.
- Sometimes we can make better predictions with a small number of PCs in Z than with a large number of predictors in X

How many Principal Components?

- The trace of the sample variance-covariance S of \mathbf{X} (total sample variance) is equal to the sum of its eigenvalues:
$$\text{trace}(S) = s_1^2 + s_2^2 + \dots + s_m^2 = \lambda_1 + \lambda_2 + \dots + \lambda_m.$$
- Often most of the total variance of a data set is concentrated in the first principal components.
- Make a plot of the PCs standard deviations ($\sqrt{\lambda_i}$) vs. the PC index i . This is called the **scree plot**.
- Look for the PC index i where there is a big change in slope (the **elbow**) in the scree plot.
- Another way is to calculate the cumulative variance explained by the first PCs, and retain the number of PCs explaining between 70% to 90% of the total variation.
- Another option is to discard PCs such that $\lambda_i < \bar{\lambda}$.

PCR Example

- *meatspec* data set from the *faraway* library.
- The goal is to predict fat content (response) using 100-channel spectrum of absorbances (predictors) in 215 samples of finely chopped meat.
- Partition the data into *training sample* and *testing sample* to test model performance.
- Fit model with all predictors using the training data set and make a prediction on the testing set.
- Select few PCs representing the 100 predictors and repeat the process (*shrinkage effect*)
- What about if we select the number of PCs to minimize the prediction error instead?. We can use *cross-validation*.

PCR Example: *meatspec* data set

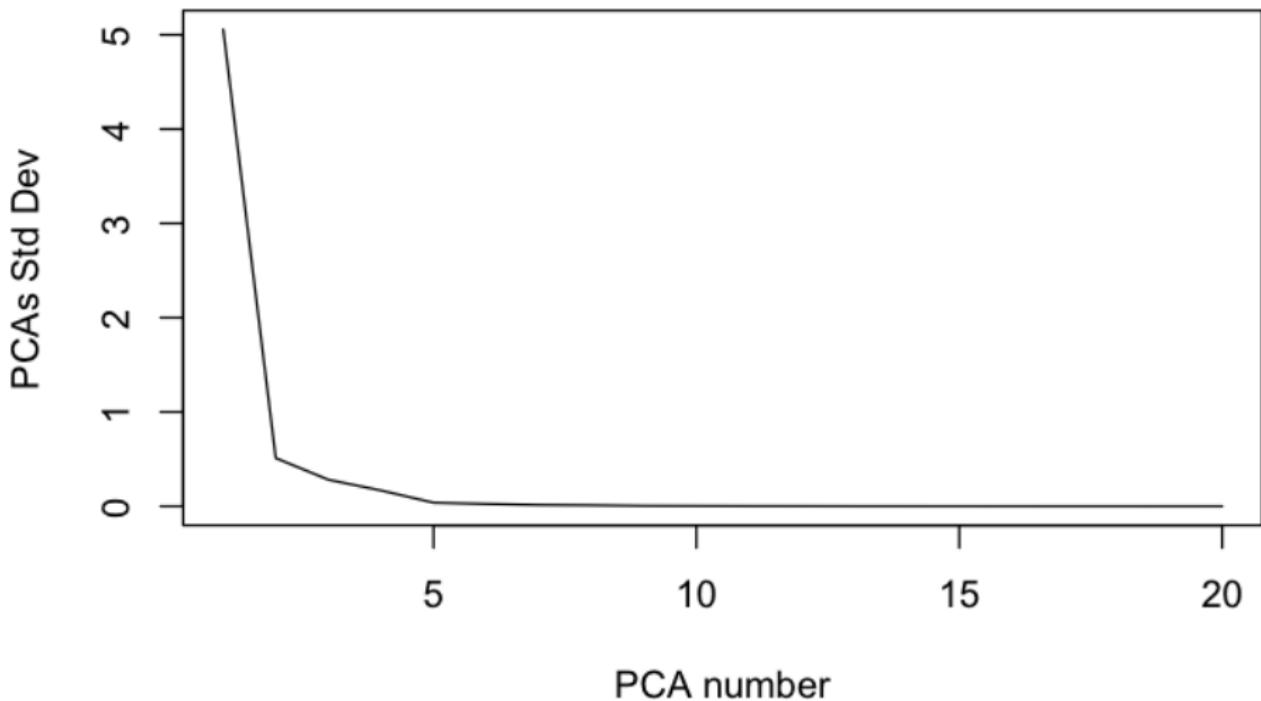
Function **prcomp** can be used to calculate the PCs and extract the λ 's squared-roots (sdev) and *eigenvectors* (rotation) of the variance-covariance matrix:

```
data(meatspec, package="faraway")
trainmeat<-meatspec[1:172,]
testmeat<-meatspec[173:215,]
mod1<-lm(fat~., trainmeat)
meatpca<-prcomp(trainmeat[, -101])
round(meatpca$sdev, 3)[1:50]
```

```
## [1] 5.055 0.511 0.282 0.168 0.038 0.025 0.014 0.011 0.005 0.003 0.002 0.002
## [13] 0.001 0.001 0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## [25] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## [37] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## [49] 0.000 0.000
```

The first two PC's explains around 90% of the total variation of meat fat content.

Scree plot



According to the scree plot, 4 PC's (elbow at 5th PC) seem adequate to represent the data.

The **pcr** function (principal component regression) from the *pls* package has useful features for prediction and cross-validation. We can easily calculate the RMSE for the training set and the testing set.

```
modpcr<-pcr(fat ~ ., data=trainmeat,ncomp=50)
#summary(modpcr)
#RMSE with 4 PCAs
rmse(predict(modpcr,ncomp=4),trainmeat$fat)
```

```
## [1] 4.064745
```

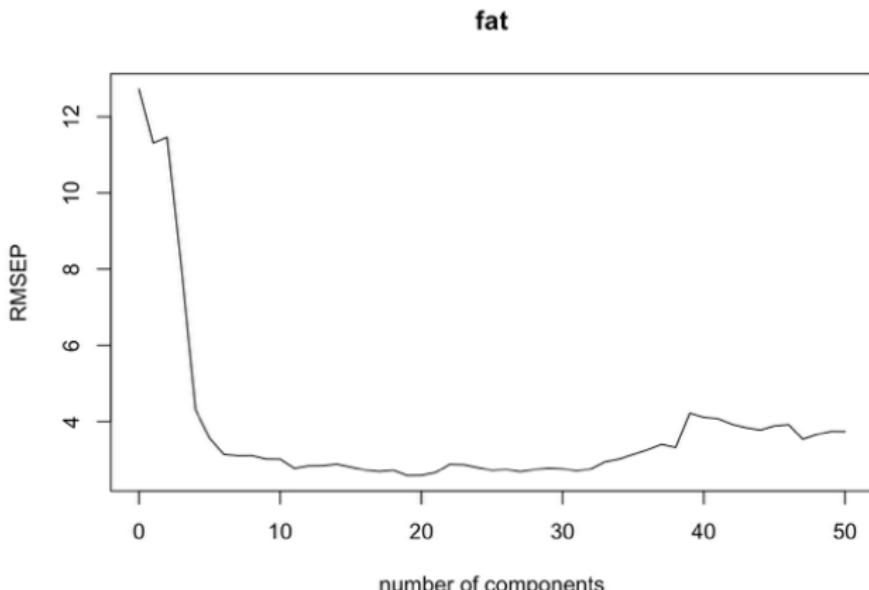
```
rmse(predict(modpcr,testmeat,ncomp=4),testmeat$fat)
```

```
## [1] 4.533982
```

As expected the testing error is larger than the training error. We can do better by selecting the number of PC's that minimize the CV error.

You can use the function **RMSEP** instead, to select the number of PC's that minimize the 10-fold Cross-Validation error. The resulting Cross-Validation error is < 2.5

```
#Use 10-fold Cross-Validation
set.seed(123)
modpcrcv<-pcr(fat~,data=trainmeat,validation="CV",ncomp=50)
pcreCV<-RMSEP(modpcrcv,estimate="CV")
plot(pcreCV)
```



Ridge Regression

- Although the aim of PCR is to reduce dimensionality in the number of predictors, you still have to measure all the predictors since each PC is a linear combination of all predictors.
- Ridge regression assumes that after normalization, some of the regression coefficients should not be very large.
- Ridge regression is very useful when you have collinearity and the LS regression coefficients are unstable.
- The method uses a **penalized regression** since the LS minimization problem has a penalty term:

$$\text{minimize} (y - X\beta)^\top (y - X\beta) + \lambda \sum_j \beta_j^2$$

for some $\lambda \geq 0$. The penalty term is $\sum_j \beta_j^2$

- Usually predictors are standardized first (centered by their means and scaled by their standard deviations) and the response y is centered.
- The ridge regression estimates are:

$$\hat{\boldsymbol{\beta}} = (X^\top X + \lambda I)^{-1} X^\top y$$

(Note the extra term λI or *ridge* in the $X^\top X$ matrix)

- The difference with standard LS is that the problem solution $\boldsymbol{\beta}$ minimizes:

$$(y - X\boldsymbol{\beta})^\top (y - X\boldsymbol{\beta}) \quad \text{subject to} \quad \sum_{j=1}^p \beta_j^2 \leq t^2$$

- The parameter λ (or t) should be chosen to have stable estimates of $\boldsymbol{\beta}$. Can use function *lm.ridge* from the *MASS* library.

- Note that when $\lambda = 0$ the ridge regression estimation problem reduces to the standard LS problem, while when $\lambda \rightarrow \infty$, $\hat{\beta} \rightarrow 0$.
- It is useful to plot the values of $\hat{\beta}_j$ as a function of λ .
- The value of λ can be also chosen using automated methods as Generalized Cross-Validation (GCV) (similar to Cross-Validation).
- Ridge regression coefficient estimates are biased.

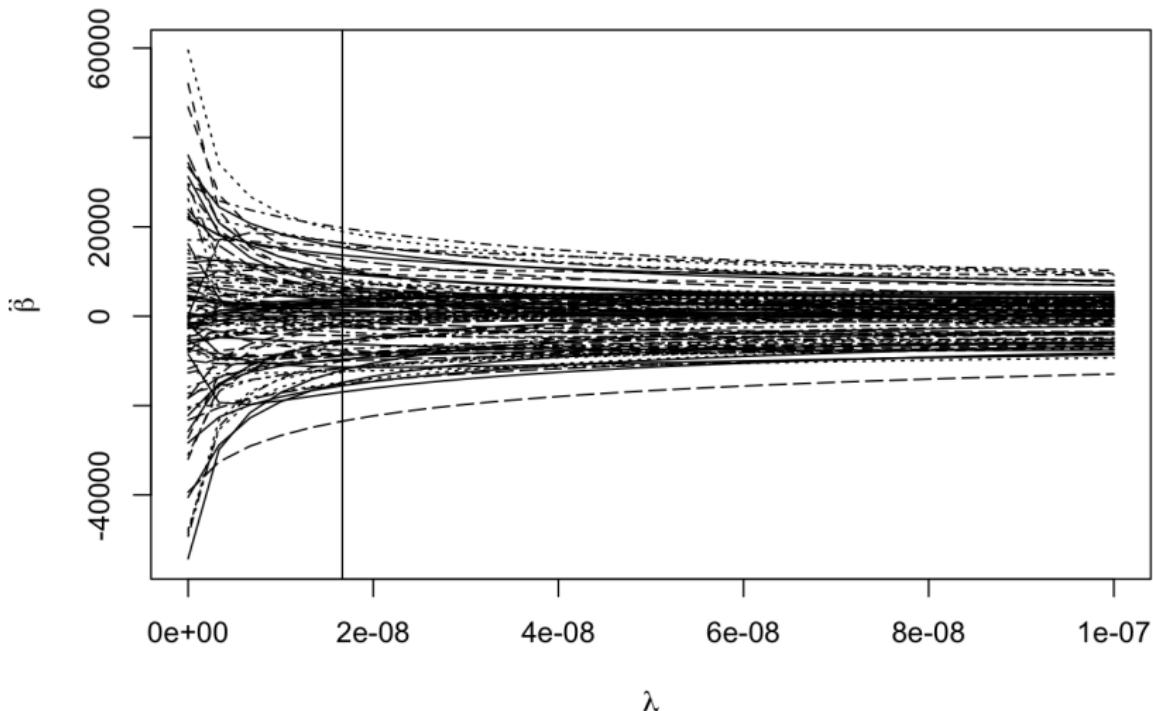
Example: meatspec data set

The value of λ is selected by minimizing the GCV. Some experimentation is required to get the initial λ range.

```
library(MASS)
library(faraway)
data(fat)
trainmeat<-meatspec[1:172,]
testmeat<-meatspec[173:215,]
ridgemod<-lm.ridge(fat~,trainmeat,lambda=seq(0,0.0000001,len=31))
# Predictors are centered and scaled.
# The response is centered
matplot(ridgemod$lambda,coef(ridgemod),type="l",xlab=expression(lambda),ylab=expression(hat(beta)),col=1)
# Look for the value of lambda that minimizes GCV
which.min(ridgemod$GCV)
```

```
## 1.666667e-08
## 6
```

```
abline(v=1.666667e-08)
```



Trace plot for the *meatspec* data. The vertical line is the value of λ that minimizes the GCV.

Lasso Regression

- In this case the estimated $\hat{\beta}$ minimizes:

$$\text{minimize} (y - X\beta)^\top (y - X\beta) + \lambda \sum_j |\beta_j|$$

for some $\lambda \geq 0$. The penalty term is $\sum_j |\beta_j|$ (L_1 constraint).

- In two-dimensions the constraint defines a square. In higher dimensions it defines a **polytope**.
- Lasso is useful when the response can be explained by few predictors with zero effect on the remaining predictors (Lasso is similar to a variable selection method).
- When $\beta_j = 0$ the corresponding predictor is eliminated. This is not the case for ridge regression.

Lasso regression

- Use Lasso when the effect of predictors is *sparse*. This means that only few predictors will have an effect on the response (e.g. gene expression data) or when number of predictors is large ($p > n$)
- Use the *lars R package* for Lasso
- Select t in the constraint $\sum_{j=1}^p |\beta|_j \leq t$ by using Cross-Validation (CV).
- As t increases, the number of predictors increases.

Example: *state* data set

We have data from 50 states:

- Population: population estimate as of July 1, 1975
- Income: per capita income (1974)
- Illiteracy: illiteracy (1970, percent of population)
- Life Exp: life expectancy in years (1969–71)
- Murder: murder and non-negligent manslaughter rate per 100,000 population (1976)
- HS Grad: percent high-school graduates (1970)
- Frost: mean number of days with minimum temperature below freezing (1931–1960) in capital or large city
- Area: land area in square miles

We want to predict *Life Expectancy*

Example: *state* data set

Use function *lars* from library *lars*

```
library(lars)

## Loaded lars 1.2

data(state)
statedata<-data.frame(state.x77, row.names = state.abb)
names(statedata)[1:4]

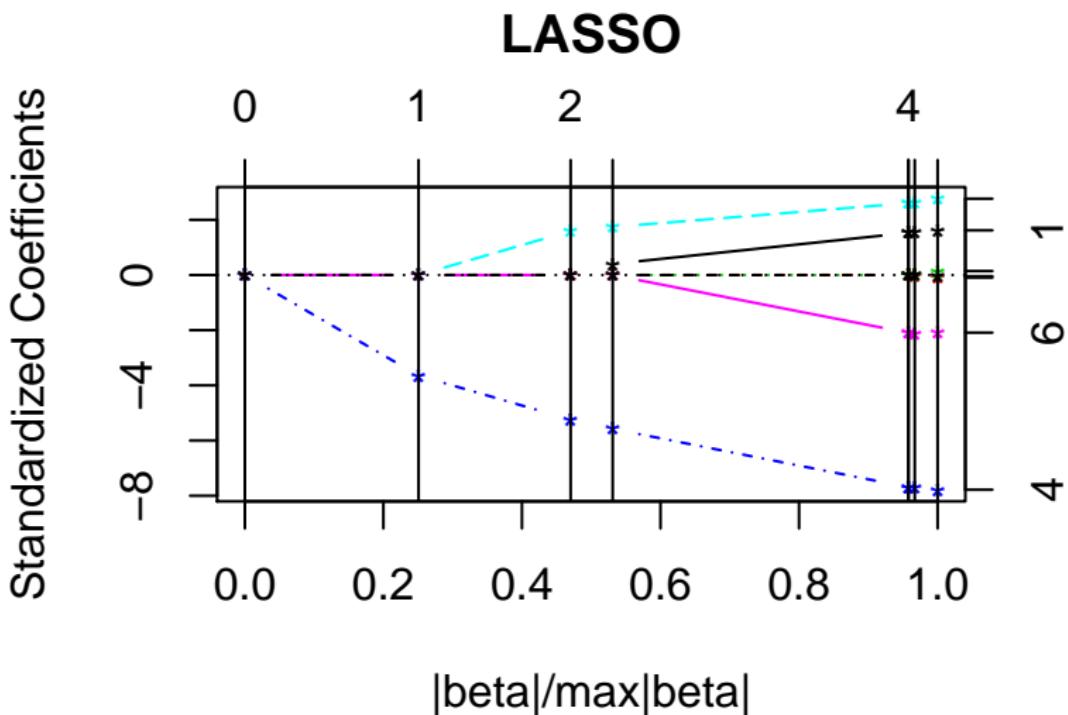
## [1] "Population" "Income"      "Illiteracy"   "Life.Exp"

names(statedata)[5:8]

## [1] "Murder"    "HS.Grad"   "Frost"     "Area"

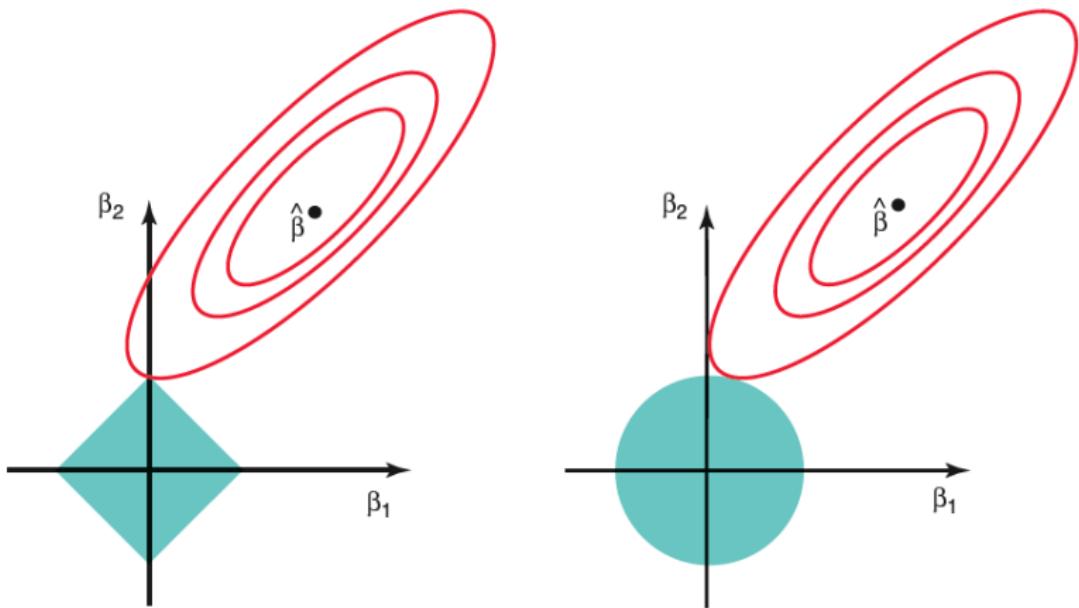
#No model equation in function lars
modlasso<-lars(as.matrix(statedata[,-4]),
                 statedata$Life.Exp)
```

```
plot(modlasso)
```



- The x-axis in the graph is the scaled value of $t = \sum_{j=1}^p |\beta_j|$
- This value has been scaled to its maximum value which is the least-square solution.
- As t increases more predictors enter into the model.
- Variable 4 (murder rate) enters first, followed by HS graduation, population and days of frost (*Frost*). The remaining variables enter when t is large (close to the LS solutions).
- t can be selected by cross-validation (lars uses 10-fold CV by default) using function *cvlars*

Comparing Ridge Regression and Lasso



Comparing Ridge Regression and Lasso

- Lasso selects a sub-set of predictors (some coefficients equal to zero).
- Ridge regression performs better when the response is a function of many predictors with coefficients around the same size.
- Lasso will perform better when a relatively small number of predictors have large coefficients and the rest are very small or equal to zero.
- Since the number of predictors is never known *a priori*, cross-validation can be used to decide which approach is better for a particular data set.

STAT 425

One Way ANOVA

Comparative Experiments

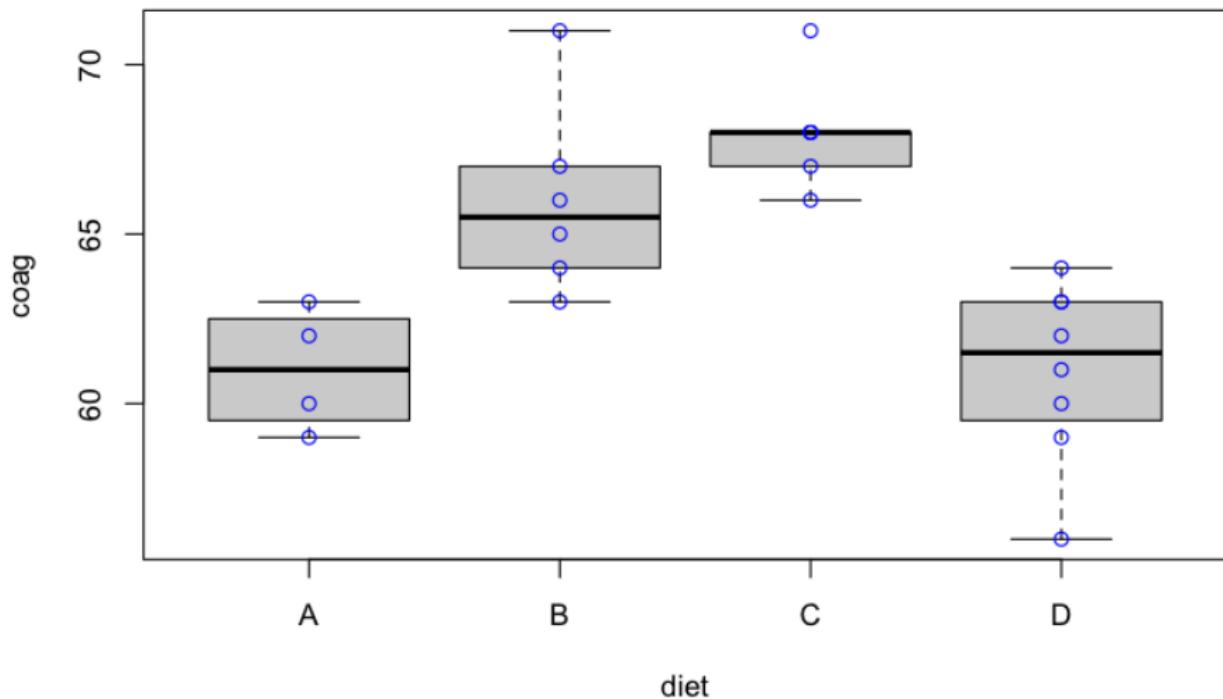
- A **comparative experiment** is intended to answer research questions regarding the differences between the effects of imposing two or more different conditions.
- The imposed conditions are the **treatments**, and they are imposed on the **experimental units**. The effects are measured using the **responses** (usually values of a single response variable).
- The way treatments are assigned to experimental units is called the **design** of the experiment. Some form of **randomization** is usually used. In that case, it is a **randomized experiment** (or sometimes **randomized study**).

Blood Coagulation Example

- 24 animals were randomly assigned to 4 different diets with goal to study blood coagulation times.
- The samples were taken in a random order.
- This data set can be found in the *faraway* library.

| ## | coag | diet |
|-------|------|------|
| ## 1 | 62 | A |
| ## 2 | 60 | A |
| ## 3 | 63 | A |
| ## 4 | 59 | A |
| ## 5 | 63 | B |
| ## 6 | 67 | B |
| ## 7 | 71 | B |
| ## 8 | 64 | B |
| ## 9 | 65 | B |
| ## 10 | 66 | B |
| ## 11 | 68 | C |
| ## 12 | 66 | C |
| ## 13 | 71 | C |
| ## 14 | 67 | C |
| ## 15 | 68 | C |
| ## 16 | 68 | C |
| ## 17 | 56 | D |
| ## 18 | 62 | D |
| ## 19 | 60 | D |
| ## 20 | 61 | D |
| ## 21 | 63 | D |
| ## 22 | 64 | D |
| ## 23 | 63 | D |
| ## 24 | 59 | D |

Blood Coagulation Example



Terminology

- **Factor:** an Independent variable. They can be experimental or observational. In our example: *Diet*
- **Level:** A particular form of the factor. In our example: *Levels of the Diet: A, B, C, D*
- **Treatments:** Factor levels or factor level combinations (if the study contains more than one factors). They provide insights into mechanisms causing the variation being studied.
Control treatments?
- **Complete Randomized Design:** Experimental units are randomly split into r groups, and r treatments are assigned, one per group.

One-Way ANOVA Model

- Data:

| | | | | |
|-----------|-----------|----------|----------|------------|
| group 1 | $y_{11},$ | y_{12} | \dots | y_{1n_1} |
| group 2 | $y_{21},$ | y_{22} | \dots | y_{2n_2} |
| \vdots | \vdots | \vdots | \vdots | \vdots |
| group r | $y_{r1},$ | y_{r2} | \dots | y_{rn_r} |

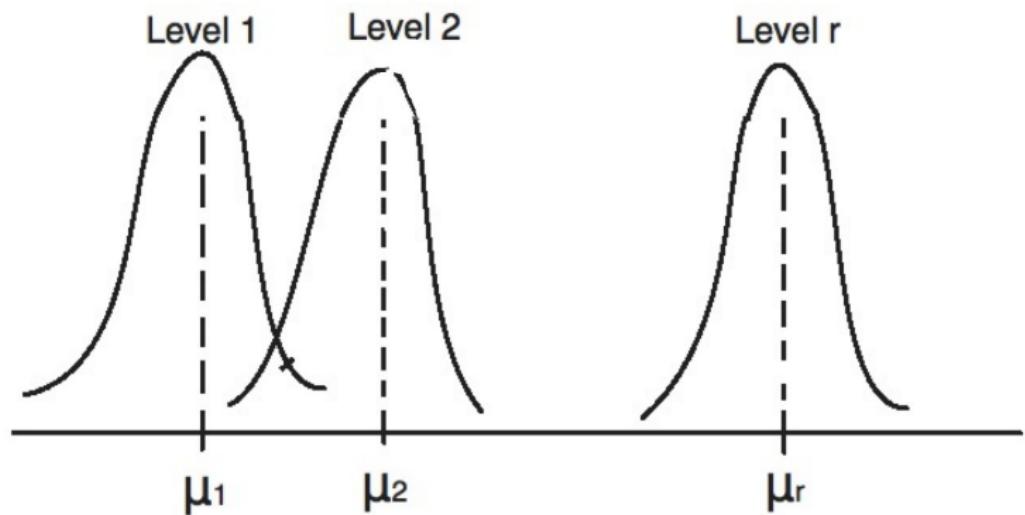
- r is the number of groups
- n_i denotes the number of obs in the i th group
- $n = \sum_{i=1}^r n_i$ is the total sample size
- $y_{ij} =$ observation j for the i th factor.

ANOVA Means Model

$$y_{ij} = \mu_i + e_{ij}, \quad i = 1, \dots, r; \quad j = 1, \dots, n_i$$

- y_{ij} : the value of the response in the j th trial for the i th factor.
- μ_i : the population mean for the i th factor level (treatment).
- $e_{ij} \sim^{iid} N(0, \sigma^2)$

ANOVA Model Representation



ANOVA Factor Effects Model

Define the effect of factor level i on the response, i.e. the treatment effect as

$$\alpha_i = \mu_i - \mu$$

where μ is the overall mean.

Factor Effects Model

$$y_{ij} = \mu + \alpha_i + e_{ij}, \quad i = 1, \dots, r; \quad j = 1, \dots, n_i$$

$$e_{ij} \sim^{iid} N(0, \sigma^2)$$

Model Parametrization

$$y_{ij} = \mu + \alpha_i + e_{ij}$$

- The factor effects model has $r + 1$ model parameters, i.e.

$$(\mu, \alpha_1, \dots, \alpha_r)$$

- In order for the α 's to be (uniquely) estimated, we need to impose restrictions.
- The restrictions on the α 's depend on how μ is defined.

| Model | μ Definition | α 's Restriction |
|----------------------|--------------------------------------|---------------------------|
| Reference Cell | $\mu = \mu_1$ | $\alpha_1 = 0$ |
| Sum-to-Zero | $\mu = \frac{1}{r} \sum_i \mu_i$ | $\sum_i \alpha_i = 0$ |
| Weighted Sum-to-Zero | $\mu = \frac{1}{n} \sum_i n_i \mu_i$ | $\sum_i n_i \alpha_i = 0$ |

- The default in R is the Reference Cell model.

Coagulation Example: Reference Cell (default)

```
contrasts(diet)=contr.treatment(4)
g=lm(coag~diet)
summary(g)
```

```
##
## Call:
## lm(formula = coag ~ diet)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.00  -1.25   0.00   1.25   5.00
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.100e+01  1.183e+00  51.554 < 2e-16 ***
## diet2       5.000e+00  1.528e+00   3.273 0.003803 **
## diet3       7.000e+00  1.528e+00   4.583 0.000181 ***
## diet4       2.991e-15  1.449e+00   0.000 1.000000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.366 on 20 degrees of freedom
## Multiple R-squared:  0.6706, Adjusted R-squared:  0.6212
## F-statistic: 13.57 on 3 and 20 DF,  p-value: 4.658e-05
```

```
model.matrix(g)
```

```
##      (Intercept) dietB dietC dietD
## 1            1     0     0     0
## 2            1     0     0     0
## 3            1     0     0     0
## 4            1     0     0     0
## 5            1     1     0     0
## 6            1     1     0     0
## 7            1     1     0     0
## 8            1     1     0     0
## 9            1     1     0     0
## 10           1     1     0     0
## 11           1     0     1     0
## 12           1     0     1     0
## 13           1     0     1     0
## 14           1     0     1     0
## 15           1     0     1     0
## 16           1     0     1     0
## 17           1     0     0     1
## 18           1     0     0     1
## 19           1     0     0     1
## 20           1     0     0     1
## 21           1     0     0     1
## 22           1     0     0     1
## 23           1     0     0     1
## 24           1     0     0     1
```

Coagulation Example: A coding that fits the Mean Model

```
g1=lm(coag~diet-1)
summary(g1)
```

```
##
## Call:
## lm(formula = coag ~ diet - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -5.00   -1.25    0.00    1.25    5.00 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## dietA     61.0000   1.1832   51.55 <2e-16 ***
## dietB     66.0000   0.9661   68.32 <2e-16 ***
## dietC     68.0000   0.9661   70.39 <2e-16 ***
## dietD     61.0000   0.8367   72.91 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.366 on 20 degrees of freedom
## Multiple R-squared:  0.9989, Adjusted R-squared:  0.9986 
## F-statistic:  4399 on 4 and 20 DF,  p-value: < 2.2e-16
```

```
model.matrix(g1)
```

```
##      dietA dietB dietC dietD
## 1      1     0     0     0
## 2      1     0     0     0
## 3      1     0     0     0
## 4      1     0     0     0
## 5      0     1     0     0
## 6      0     1     0     0
## 7      0     1     0     0
## 8      0     1     0     0
## 9      0     1     0     0
## 10     0     1     0     0
## 11     0     0     1     0
## 12     0     0     1     0
## 13     0     0     1     0
## 14     0     0     1     0
## 15     0     0     1     0
## 16     0     0     1     0
## 17     0     0     0     1
## 18     0     0     0     1
## 19     0     0     0     1
## 20     0     0     0     1
## 21     0     0     0     1
## 22     0     0     0     1
## 23     0     0     0     1
## 24     0     0     0     1
```

Coagulation Example: $\sum_i \alpha_i = 0$

```
contrasts(diet) = contr.sum(4)
g2 = lm(coag~diet)
summary(g2)
```

```
##
## Call:
## lm(formula = coag ~ diet)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.00  -1.25   0.00   1.25   5.00
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 64.0000    0.4979 128.537 < 2e-16 ***
## diet1       -3.0000    0.9736  -3.081 0.005889 **
## diet2        2.0000    0.8453   2.366 0.028195 *
## diet3        4.0000    0.8453   4.732 0.000128 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.366 on 20 degrees of freedom
## Multiple R-squared:  0.6706, Adjusted R-squared:  0.6212
## F-statistic: 13.57 on 3 and 20 DF,  p-value: 4.658e-05
```

```
model.matrix(g2)
```

```
##      (Intercept) diet1 diet2 diet3
## 1             1     1     0     0
## 2             1     1     0     0
## 3             1     1     0     0
## 4             1     1     0     0
## 5             1     0     1     0
## 6             1     0     1     0
## 7             1     0     1     0
## 8             1     0     1     0
## 9             1     0     1     0
## 10            1     0     1     0
## 11            1     0     0     1
## 12            1     0     0     1
## 13            1     0     0     1
## 14            1     0     0     1
## 15            1     0     0     1
## 16            1     0     0     1
## 17            1    -1    -1    -1
## 18            1    -1    -1    -1
## 19            1    -1    -1    -1
## 20            1    -1    -1    -1
## 21            1    -1    -1    -1
## 22            1    -1    -1    -1
## 23            1    -1    -1    -1
## 24            1    -1    -1    -1
```

Model Properties

① $E(y_{ij}) = \mu_i$

② $Var(y_{ij}) = Var(e_{ij}) = \sigma^2$

Thus, all observations have the same variance, regardless of factor level.

③ $e_{ij} \sim N(0, \sigma^2)$ and independent.

④ $y_{ij} \sim \mathcal{N}(\mu_i, \sigma^2)$ and independent.

We can re-state the model as

y_{ij} are independent $\mathcal{N}(\mu_i, \sigma^2)$

Fitting of ANOVA Model

Minimize the sum of squared deviations of the observations around their expected values with respect to the parameters:

$$Q = \sum_{i=1}^r \sum_{j=1}^{n_i} (y_{ij} - \mathbb{E}(y_{ij}))^2$$

If we re-write Q we have

$$Q = \sum_j (y_{1j} - \mu_1)^2 + \sum_j (y_{2j} - \mu_2)^2 + \dots + \sum_j (y_{rj} - \mu_r)^2$$

So the **least squares estimator** of μ_i , denoted by $\hat{\mu}_i$ is

$$\hat{\mu}_i = \bar{y}_{i\cdot} = \frac{1}{n_i} \sum_{j=1}^{n_i} y_{ij}$$

Using the appropriate constraints, we can easily extract the estimators for μ and α_i .

Using the model ‘g2’ with constraint $\sum_{i=1}^4 \alpha_i = 0$ we have:

```
g2$coef
```

```
## (Intercept)      diet1      diet2      diet3
##           64          -3           2           4
```

This implies that $\hat{\mu} = 64$ and

$$\begin{aligned}\hat{\alpha}_1 &= -3 & \hat{\mu}_1 &= 64 - 3 = 61 \\ \hat{\alpha}_2 &= 2 & \hat{\mu}_2 &= 64 + 2 = 66 \\ \hat{\alpha}_3 &= 4 & \hat{\mu}_3 &= 64 + 4 = 68\end{aligned}$$

The estimators for α_4 and the corresponding mean μ_4 , are obtained them using the constraints:

$$\hat{\alpha}_4 = -\hat{\alpha}_1 - \hat{\alpha}_2 - \hat{\alpha}_3 = 3 - 2 - 4 = -3 \text{ and } \hat{\mu}_4 = 64 - 3 = 61$$

Fitted Values & Residuals

- The LS fit for y_{ij} is the corresponding group mean

$$\hat{y}_{ij} = \bar{y}_i.$$

- Residuals

$$r_{ij} = y_{ij} - \hat{y}_{ij} = y_{ij} - \bar{y}_i.$$

- RSS

$$\sum_{i=1}^r \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2,$$

i.e. the within-group variation.

ANOVA Table

| Source of Variation | SS | df | MS |
|-----------------------|---|---------|-------------------|
| Between Groups | $FSS = \sum n_i(\bar{y}_{i\cdot} - \bar{y}_{..})^2$ | $r - 1$ | $\frac{FSS}{r-1}$ |
| Error (within groups) | $RSS = \sum \sum (y_{ij} - \bar{y}_{i\cdot})^2$ | $n - r$ | $\frac{RSS}{n-r}$ |
| Total | $TSS = \sum \sum (y_{ij} - \bar{y}_{..})^2$ | $n - 1$ | |

F-test

- We want to test whether the means of the groups are really different. We can express this as

$$\begin{cases} H_0 : \mu_1 = \mu_2 = \dots = \mu_r \\ H_a : \text{not all } \mu_i, i = 1, \dots, r \text{ are equal} \end{cases}$$

- or in terms of models

$$\begin{cases} H_0 : y_{ij} = \mu + e_{ij} \\ H_a : y_{ij} = \mu + \alpha_i + e_{ij} \end{cases}$$

F-test

- They are two nested models, so we can use the *F*-test

$$\frac{(RSS_0 - RSS_a)/(r - 1)}{RSS_a/(n - r)} \sim F_{r-1, n-r},$$

under H_0 .

- The test statistic can also be written as

$$\frac{FSS/(r - 1)}{RSS/(n - r)} = \frac{\text{Between-group Variation}/(r - 1)}{\text{Within-group Variation}/(n - r)},$$

where FSS, RSS are defined in the ANOVA table.

```
null = lm(coag ~ 1)
anova(null, g2)
```

```
## Analysis of Variance Table
##
## Model 1: coag ~ 1
## Model 2: coag ~ diet
##   Res.Df RSS Df Sum of Sq      F    Pr(>F)
## 1     23  340
## 2     20 112  3      228 13.571 4.658e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- It does not matter which coding is used for the mean/effects. The results would be the same.

Equivalently, we can get the ANOVA table that contains the same F test and p -value:

```
anova(g2)
```

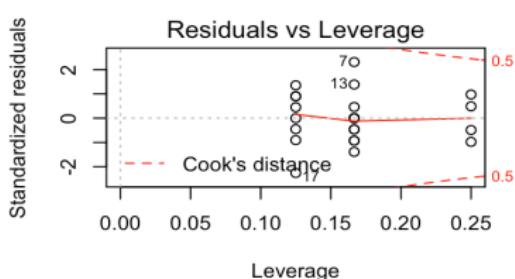
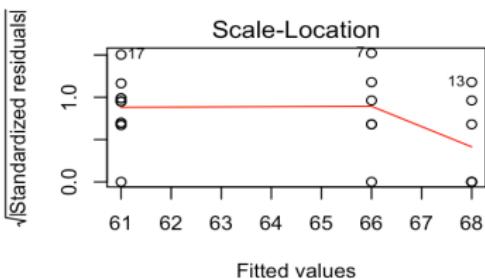
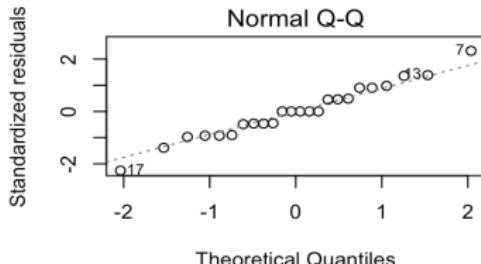
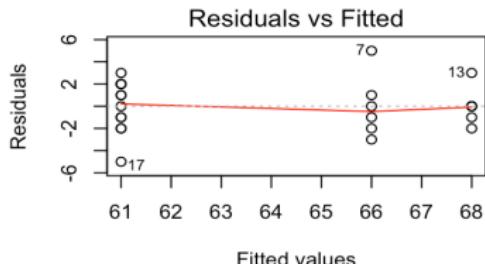
```
## Analysis of Variance Table
##
## Response: coag
##           Df Sum Sq Mean Sq F value    Pr(>F)
## diet      3    228    76.0  13.571 4.658e-05 ***
## Residuals 20   112     5.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The p -value is much less than $\alpha = 5\%$, so we reject the null and conclude that there are differences among the different types of diet.

Diagnostics for ANOVA Models

- Check for outliers/ unusual observations.
- Check the residuals vs. fitted values plot for departures from the constant variance assumption.
- Check the Q-Q plot for departures from the normality assumption.

```
par(mfrow=c(2,2))  
plot(g2)
```



Levene's Test for Equality of Variances:

- Run Regression $\text{abs}(\text{residuals}) \sim X$, i.e. use $\text{abs}(\text{residuals})$ as the response in a new one-way ANOVA.
- If the p -value for the F -test is **greater** than 1% level, then we conclude that there is no evidence of a non-constant variance.

H_0 : All group variances are equal.

```
g2=lm(coag~diet)
summary(lm(abs(g2$res) ~ diet))

##
## Call:
## lm(formula = abs(g2$res) ~ diet)
##
## Residuals:
##    Min     1Q Median     3Q    Max 
## -2.000 -1.000  0.000  0.625  3.000 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1.6250    0.3013   5.394  2.8e-05 ***
## diet1      -0.1250    0.5891  -0.212    0.834    
## diet2       0.3750    0.5115   0.733    0.472    
## diet3      -0.6250    0.5115  -1.222    0.236    
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
##
## Residual standard error: 1.432 on 20 degrees of freedom
## Multiple R-squared:  0.09559,   Adjusted R-squared:  -0.04007 
## F-statistic: 0.7046 on 3 and 20 DF,  p-value: 0.5604
```

- Since the p -value is greater than 0.01, there is no evidence of unequal variances.

STAT 425

Analysis of Factor Level Means

Question:

What happens when the F -test leads to the conclusion that the factor level means μ_i differ?

- Analysis of the factor level means of interest using *estimation* techniques.
- Statistical *tests* concerning the factor level means of interest.

Inference for Factor Level Means

- ① A single factor level mean.
- ② A difference between two factor level means.
- ③ A contrast among factor level means.
- ④ A linear combination of factor level means.

Single Factor Level Mean

- Estimation of μ_i : $\hat{\mu}_i = \bar{y}_i$.
- Distribution of $\hat{\mu}_i$: $E(\hat{\mu}_i) = \mu_i$, $Var(\hat{\mu}_i) = \frac{\sigma^2}{n_i}$.

The estimated variance of $\bar{y}_{i\cdot}$ is $s_{\bar{y}_{i\cdot}}^2 = \frac{1}{n_i} \cdot \frac{RSS}{n-r}$.

- Under the H_0

$\frac{\bar{y}_{i\cdot} - \mu_i}{s_{\bar{y}_{i\cdot}}}$ is distributed as T_{n-r} .

- Confidence Interval for μ_i :

$$\mu_i \in \bar{y}_{i\cdot} \pm T_{n-r}(\alpha/2) s_{\bar{y}_{i\cdot}}$$

- In order to obtain confidence intervals for the factor level means in R, we fit the *means model* and we use the `confint` command:

```
# CI for single Factor Level Mean  
g1=lm(coag~diet-1)  
confint(g1)
```

```
##           2.5 %   97.5 %  
## dietA 58.53185 63.46815  
## dietB 63.98477 68.01523  
## dietC 65.98477 70.01523  
## dietD 59.25476 62.74524
```

Difference between Two Factor Level Means

The difference between two factor level means ([pairwise comparison](#)) is defined as

$$D = \mu_i - \mu_{i'}$$

- Estimation of D : $\hat{D} = \bar{y}_{i\cdot} - \bar{y}_{i'\cdot}$.
- Distribution of \hat{D} : $E(\hat{D}) = \mu_i - \mu_{i'}, \text{Var}(\hat{D}) = \sigma^2 \left(\frac{1}{n_i} + \frac{1}{n_{i'}} \right)$.

The estimated variance of \hat{D} is

$$s_{\hat{D}}^2 = \frac{RSS}{n-r} \cdot \left(\frac{1}{n_i} + \frac{1}{n_{i'}} \right).$$

- Under the H_0

$\frac{\hat{D} - D}{s_{\hat{D}}}$ is distributed as T_{n-r}

- Confidence Interval for D : $D \in \hat{D} + T_{n-r}(\alpha/2) s_{\hat{D}}$
- Hypothesis Test for D :

$$\left\{ \begin{array}{l} H_0 : \mu_i = \mu_{i'} \\ H_\alpha : \mu_i \neq \mu_{i'} \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} H_0 : \mu_i - \mu_{i'} = D = 0 \\ H_\alpha : \mu_i - \mu_{i'} \neq 0 \end{array} \right\}$$

The test statistic is $t = \frac{\hat{D}}{s_{\hat{D}}} \sim T_{n-r}$.

Contrast of Factor Level Means

A contrast is a comparison involving two or more level means:

$$L = \sum_{i=1}^r c_i \mu_i, \quad \text{where } \sum_{i=1}^r c_i = 0.$$

- Estimation of L : $\hat{L} = \sum_{i=1}^r c_i \bar{y}_i$.
- Distribution of \hat{L} : $E(\hat{L}) = \sum_{i=1}^r c_i \mu_i$, $Var(\hat{L}) = \sigma^2 \sum_{i=1}^r \frac{c_i^2}{n_i}$

The estimated variance of \hat{L} is

$$s_{\hat{L}}^2 = \frac{RSS}{n-r} \cdot \sum_{i=1}^r \frac{c_i^2}{n_i}$$

- Under the H_0

$$\frac{\hat{L} - L}{s_{\hat{L}}} \text{ is distributed as } T_{n-r}$$

- Confidence Interval for L : $L \in \hat{L} + T_{n-r}(\alpha/2) s_{\hat{L}}$
- Hypothesis Testing for L :

$$\begin{cases} H_0 : L = 0 \\ H_\alpha : L \neq 0 \end{cases}$$

The test statistic is $t = \frac{\hat{L}}{s_{\hat{L}}} \sim T_{n-r}$.

Linear Combination of Factor Level Means

$$L = \sum_{i=1}^r c_i \mu_i, \quad \text{no restrictions on } c'_i s$$

- Point estimator and estimated variance same as before.
- Single Degree of Freedom Tests

$$\begin{cases} H_0 : L = c \\ H_\alpha : L \neq c \end{cases}$$

Under the H_0 the test statistic here is

$$F = t^2 = \left(\frac{\hat{L} - c}{s_{\hat{L}}} \right)^2 \sim F_{1,n-r}$$

Limitations of Inference Procedures

- The confidence coefficient $1 - \alpha$ for the estimation procedures described is a statement confidence coefficient and applies only to a particular estimate, not to a series of estimates.
- Similarly the specified Type I error rate α applies only to a particular test and not to a series of tests.

Bonferroni Correction

When? The family of interest is a **particular set of pairwise comparisons, contrasts, or linear combinations** that is specified by the user.

- Suppose m is the number of statements in the family.
- In order to control the family wise error rate to be α , we need to reduce the error rate for each individual comparison to be α/m .
- That is we need to increase the significance level from $(1 - \alpha)$ to $(1 - \alpha/m)$.
- Not applicable when m is large, since the CIs would be too wide due to the increase of the significant level.

- In R, we can obtain the p -values for the Bonferroni corrections for pairwise differences using the `pairwise.t.test` command.

```
# Bonferroni correction: this test outputs the p-values for the corresponding differences.  
pairwise.t.test(coag, diet, p.adjust.method = "bonferroni")
```

```
##  
##  Pairwise comparisons using t tests with pooled SD  
##  
## data: coag and diet  
##  
##      A          B          C  
## B 0.02282 -         -  
## C 0.00108 0.95266 -  
## D 1.00000 0.00518 0.00014  
##  
## P value adjustment method: bonferroni
```

Tukey's Paired Comparison Procedures

When? the family of interest is a set of **all pairwise comparisons** of factor level means, i.e. it consists of estimates of all pairs

$$D = \mu_i - \mu_{i'}.$$

A confidence interval is given by

$$D \in \hat{D} + \frac{q(\alpha/2; r, n-r)}{\sqrt{2}} s(\hat{D}),$$

where $q(\alpha/2; r, n-r)$ refers to the $\alpha/2$ upper quantile of the **studentized range** for r means and $n-r$ degrees of freedom.

The coverage probability is exact when the sample sizes in each group are identical and is approximate otherwise.

Remark: The **studentized range** refers to the distribution of

$$\max_{i \neq j} \sqrt{n}(\bar{y}_i - \bar{y}_j) / \hat{\sigma}$$

where \bar{y}_i and \bar{y}_j are sample means from independent samples of size n from normal distributions with common means and variance σ^2 .

- To obtain Tukey family CIs for all pairwise comparisons in R, we use the TukeyHSD command.

```
# Tukey Simultaneous 95% CI for all mean differences
TukeyHSD(aov(coag~diet), data=coagulation)
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = coag ~ diet)
##
## $diet
##      diff      lwr      upr      p adj
## B-A     5  0.5932529  9.406747 0.0228300
## C-A     7  2.5932529 11.406747 0.0013858
## D-A     0 -4.2789880  4.278988 1.0000000
## C-B     2 -1.9415144  5.941514 0.4988550
## D-B    -5 -8.7981383 -1.201862 0.0075558
## D-C    -7 -10.7981383 -3.201862 0.0002854
```

Scheffé's Method for Contrasts

When? The family of interest is the set of **contrasts** among the factor level means:

$$L = \sum c_i \mu_i, \text{ where } \sum c_i = 0$$

A confidence interval is given by

$$L \in \hat{L} + (r - 1) F_{r-1, n-r}(\alpha) s_{\hat{L}}$$

- To obtain Scheffé family CIs for all pairwise comparisons in R, we use the `ScheffeTest` in the *DescTools* library.

```
g2aov=aov(coag~diet)

# If you want all the pairwise comparisons with Scheffe's method:
ScheffeTest(g2aov)
```

```
##  
## Posthoc multiple comparisons of means: Scheffe Test  
##      95% family-wise confidence level  
  
##  
## $diet  
##      diff      lwr.ci     upr.ci    pval  
## B-A     5   0.342883  9.657117 0.03233 *  
## C-A     7   2.342883 11.657117 0.00210 **  
## D-A     0  -4.418129  4.418129 1.00000  
## C-B     2  -2.165452  6.165452 0.55494  
## D-B    -5  -8.896424 -1.103576 0.00876 **  
## D-C    -7 -10.896424 -3.103576 0.00031 ***  
##  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- If we want to obtain intervals for specific contrasts, such as

$$L_1 = \mu_A - \frac{1}{2}\mu_B - \frac{1}{2}\mu_C \text{ and } L_2 = \mu_B - \frac{1}{2}\mu_C - \frac{1}{2}\mu_D$$

then we can specify this in the contrasts argument as follows:

```
ScheffeTest(g2ao, contrasts=matrix(c(1,-0.5,-0.5,0,  
0,1,-0.5,-0.5), ncol=2))
```

```
##  
## Posthoc multiple comparisons of means: Scheffe Test  
## 95% family-wise confidence level  
##  
## $diet  
##      diff    lwr.ci   upr.ci   pval  
## A-B,C -6.0 -10.165452 -1.834548 0.0032 **  
## B-C,D  1.5 -2.031434  5.031434 0.6482  
##  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```