

Linear Models: R Examples — The less than full rank model: estimation and estimability

Another running example

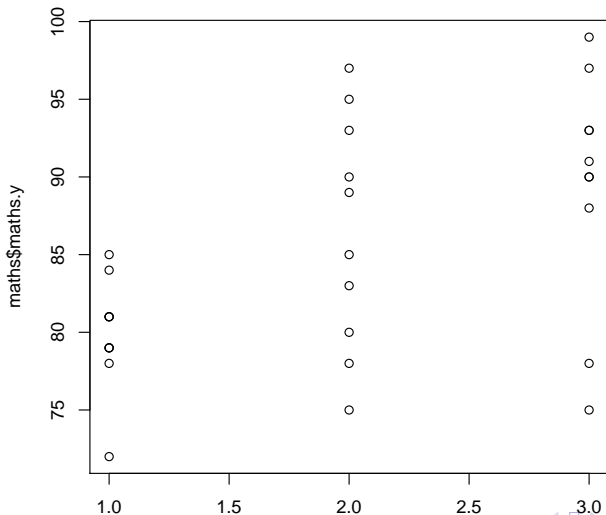
We compare the marks of students in 3 different mathematics classes. There is another factor (IQ), but we ignore this for the time being.

```
> maths <- read.csv("../data/maths.csv")
> str(maths)

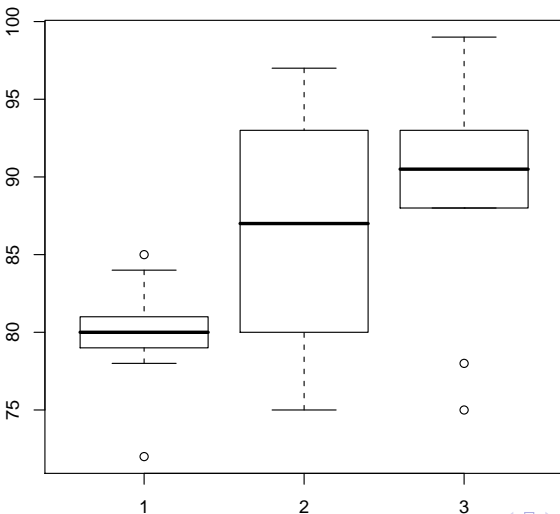
'data.frame': 30 obs. of  5 variables:
 $ X      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ maths.y: int  81 84 81 79 78 79 81 85 72 79 ...
 $ iq     : int  99 103 108 109 96 104 96 105 94 91 ...
 $ class  : int  1 1 1 1 1 1 1 1 1 1 ...
 $ class.f: int  1 1 1 1 1 1 1 1 1 1 ...

> maths$class.f <- factor(maths$class.f)
```

```
> plot(maths$class, maths$maths.y)
```



```
> plot(maths$class.f, maths$maths.y)
```



```
> y <- maths$maths.y
> y

 [1] 81 84 81 79 78 79 81 85 72 79 85 78 93 80 83 95 90 89
[26] 91 88 93 90 78

> n <- dim(maths)[1]
> k <- 3
> X <- matrix(0, n, k + 1)
> X[, 1] <- 1
> X[maths$class.f == 1, 2] <- 1
> X[maths$class.f == 2, 3] <- 1
> X[maths$class.f == 3, 4] <- 1
```

```
> X
```

```
      [,1] [,2] [,3] [,4]
[1,]    1    1    0    0
[2,]    1    1    0    0
[3,]    1    1    0    0
[4,]    1    1    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    1    0
[18,]   1    0    1    0
[19,]   1    0    1    0
[20,]   1    0    1    0
```

```
> Xre <- X[, -1]
> library(car)
> b <- inv(t(Xre) %*% Xre) %*% t(Xre) %*% y
> b
```

```
      [,1]
[1,] 79.9
[2,] 86.5
[3,] 89.4
```

```
> modelre <- lm(y ~ 0 + X[, 2] + X[, 3] + X[, 4])
> summary(modelre)
```

Call:

```
lm(formula = y ~ 0 + X[, 2] + X[, 3] + X[, 4])
```

Residuals:

Min	1Q	Median	3Q	Max
-14.40	-1.80	0.85	3.60	10.50

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
X[, 2]	79.900	2.053	38.92	<2e-16 ***
X[, 3]	86.500	2.053	42.14	<2e-16 ***
X[, 4]	89.400	2.053	43.55	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.492 on 27 degrees of freedom

Multiple R-squared: 0.9948, Adjusted R-squared: 0.9942

F-statistic: 1729 on 3 and 27 DF, p-value: < 2.2e-16

Conditional inverses

```
> library(MASS)
> A <- matrix(c(2, -6, 3, 1, 6, 4, -2, -1, 0), 3, 3)
> A
```

```
      [,1] [,2] [,3]
[1,]    2    1  -2
[2,]   -6    6  -1
[3,]    3    4    0
```

```
> Ac <- ginv(A)
```

```
> Ac
```

```
          [,1]      [,2]      [,3]
[1,] 0.04494382 -0.08988764 0.1235955
[2,] -0.03370787 0.06741573 0.1573034
[3,] -0.47191011 -0.05617978 0.2022472
```

```
> inv(A)
```

```
          [,1]      [,2]      [,3]
[1,] 0.04494382 -0.08988764 0.1235955
[2,] -0.03370787 0.06741573 0.1573034
[3,] -0.47191011 -0.05617978 0.2022472
```

```
> A <- matrix(c(2, -6, 3, 1, 6, 4, 3, 0, 7), 3, 3)
```

```
> A
```

```
      [,1] [,2] [,3]
[1,]    2    1    3
[2,]   -6    6    0
[3,]    3    4    7
```

```
> Ac <- ginv(A)
```

```
> Ac
```

```
           [,1]           [,2]           [,3]
[1,] 0.025713835 -0.084240416 0.03659883
[2,] 0.009149708  0.080454330 0.04369774
[3,] 0.034863543 -0.003786086 0.08029658
```

```
> A %*% Ac %*% A
```

```
           [,1] [,2]           [,3]
[1,]      2     1 3.000000e+00
[2,]     -6     6 4.163336e-16
[3,]      3     4 7.000000e+00
```

```
> Ac2 <- matrix(0, 3, 3)
> Ac2[1:2, 1:2] <- t(inv(A[1:2, 1:2]))
> Ac2 <- t(Ac2)
> Ac2
```

```
      [,1]      [,2] [,3]
[1,] 0.3333333 -0.05555556 0
[2,] 0.3333333 0.11111111 0
[3,] 0.0000000 0.00000000 0
```

```
> A %*% Ac2 %*% A
```

```
      [,1] [,2] [,3]
[1,] 2     1     3
[2,] -6    6     0
[3,] 3     4     7
```

```
> Ac3 <- matrix(0, 3, 3)
> Ac3[2:3, 1:2] <- t(inv(A[2:3, 1:2]))
> Ac3 <- t(Ac3)
> Ac3
```

```
      [,1]      [,2]      [,3]
[1,]    0 -0.09523810 0.1428571
[2,]    0  0.07142857 0.1428571
[3,]    0  0.00000000 0.0000000
```

```
> A %*% Ac3 %*% A
```

```
      [,1] [,2] [,3]
[1,]    2    1    3
[2,]   -6    6    0
[3,]    3    4    7
```

```
> r <- function(A) sum(svd(A)$d > 1e-15)
> r(Ac2 %*% A)
```

```
[1] 2
```

```
> A %*% ginv(t(A) %*% A) %*% t(A) %*% A
```

```
      [,1] [,2]      [,3]
[1,]    2    1 3.000000e+00
[2,]   -6    6 2.775558e-17
[3,]    3    4 7.000000e+00
```

```
> A %*% ginv(t(A) %*% A) %*% t(A)
      [,1]      [,2]      [,3]
[1,] 0.16516801 -0.09938476 0.35778514
[2,] -0.09938476 0.98816848 0.04259347
[3,] 0.35778514 0.04259347 0.84666351

> AtAc2 <- matrix(0, 3, 3)
> AtAc2[1:2, 1:2] <- inv((t(A) %*% A)[1:2, 1:2])
> A %*% AtAc2 %*% t(A)
      [,1]      [,2]      [,3]
[1,] 0.16516801 -0.09938476 0.35778514
[2,] -0.09938476 0.98816848 0.04259347
[3,] 0.35778514 0.04259347 0.84666351
```

Normal equations

We look at the normal equations for our example dataset.

```
> t(X) %*% X
```

```
      [,1] [,2] [,3] [,4]
[1,]   30   10   10   10
[2,]   10   10    0    0
[3,]   10    0   10    0
[4,]   10    0    0   10
```

```
> t(X) %*% y
```

```
      [,1]
[1,] 2558
[2,]  799
[3,]  865
[4,]  894
```

```
> XtXc <- matrix(0, 4, 4)
> XtXc[2:4, 2:4] <- inv((t(X) %*% X)[2:4, 2:4])
> b <- XtXc %*% t(X) %*% y
> b
```

```
      [,1]
[1,]  0.0
[2,] 79.9
[3,] 86.5
[4,] 89.4
```

```
> t(X) %*% X %*% b - t(X) %*% y
```

```
      [,1]
[1,]    0
[2,]    0
[3,]    0
[4,]    0
```

```
> b2 <- ginv(t(X) %*% X) %*% t(X) %*% y
```

```
> b2
```

```
      [,1]
```

```
[1,] 63.95
```

```
[2,] 15.95
```

```
[3,] 22.55
```

```
[4,] 25.45
```

```
> t(X) %*% X %*% b2 - t(X) %*% y
```

```
      [,1]
```

```
[1,] 4.547474e-13
```

```
[2,] 3.410605e-13
```

```
[3,] 2.273737e-13
```

```
[4,] 1.136868e-13
```

```
> I4 <- diag(rep(1, 4))
> z <- as.vector(c(2, 8, -2, 1))
> b3 <- b + (I4 - XtXc %*% t(X) %*% X) %*% z
> b3
```

```
      [,1]
[1,]  2.0
[2,] 77.9
[3,] 84.5
[4,] 87.4
```

```
> t(X) %*% X %*% b3 - t(X) %*% y
```

```
      [,1]
[1,]    0
[2,]    0
[3,]    0
[4,]    0
```

```
> b + (I4 - XtXc %*% t(X) %*% X) %*% b3
```

```
      [,1]
```

```
[1,]  2.0
```

```
[2,] 77.9
```

```
[3,] 84.5
```

```
[4,] 87.4
```

```
> b3
```

```
      [,1]
```

```
[1,]  2.0
```

```
[2,] 77.9
```

```
[3,] 84.5
```

```
[4,] 87.4
```

Estimability

We return to the maths dataset.

```
> tt <- as.vector(c(0, 1, -1, 0))
```

```
> tt
```

```
[1] 0 1 -1 0
```

```
> tt %*% XtXc %*% t(X) %*% X
```

```
      [,1] [,2] [,3] [,4]  
[1,]    0    1  -1    0
```

```
> tt2 <- as.vector(c(1, 1, 1, 1))
> tt2

[1] 1 1 1 1

> tt2 %*% XtXc %*% t(X) %*% X
      [,1] [,2] [,3] [,4]
[1,]     3     1     1     1
```

```
> tt %*% b
```

```
      [,1]
```

```
[1,] -6.6
```

```
> tt %*% b2
```

```
      [,1]
```

```
[1,] -6.6
```

```
> tt %*% b3
```

```
      [,1]
```

```
[1,] -6.6
```

```
> tt2 %*% b
```

```
      [,1]
```

```
[1,] 255.8
```

```
> tt2 %*% b2
```

```
      [,1]
```

```
[1,] 127.9
```

```
> tt2 %*% b3
```

```
      [,1]
```

```
[1,] 251.8
```

Using R

For the less than full rank model, R uses contrasts for its tests. The two main contrast sets that you need to know about are `contr.treatment` and `contr.sum`.

Label	<code>contr.treatment</code>	<code>contr.sum</code>
Intercept	μ_1	$\bar{\mu}$
factor1		$\mu_1 - \bar{\mu}$
factor2	$\mu_2 - \mu_1$	$\mu_2 - \bar{\mu}$
factor3	$\mu_3 - \mu_1$	$\mu_3 - \bar{\mu}$
\vdots	\vdots	\vdots
factor(k-1)	$\mu_{k-1} - \mu_1$	$\mu_{k-1} - \bar{\mu}$
factor(k)	$\mu_k - \mu_1$	

```
> options(contrasts = c("contr.treatment", "contr.poly"))
> model <- lm(maths.y ~ class.f, data = maths)
> summary(model)
```

Call:

```
lm(formula = maths.y ~ class.f, data = maths)
```

Residuals:

Min	1Q	Median	3Q	Max
-14.40	-1.80	0.85	3.60	10.50

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	79.900	2.053	38.922	< 2e-16 ***
class.f2	6.600	2.903	2.273	0.03117 *
class.f3	9.500	2.903	3.272	0.00292 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.492 on 27 degrees of freedom

Multiple R-squared: 0.2941, Adjusted R-squared: 0.2418

F-statistic: 5.625 on 2 and 27 DF, p-value: 0.009077

```
> options(contrasts = c("contr.sum", "contr.poly"))
> model2 <- lm(maths.y ~ class.f, data = maths)
> summary(model2)
```

Call:

```
lm(formula = maths.y ~ class.f, data = maths)
```

Residuals:

Min	1Q	Median	3Q	Max
-14.40	-1.80	0.85	3.60	10.50

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	85.267	1.185	71.943	< 2e-16 ***
class.f1	-5.367	1.676	-3.202	0.00348 **
class.f2	1.233	1.676	0.736	0.46818

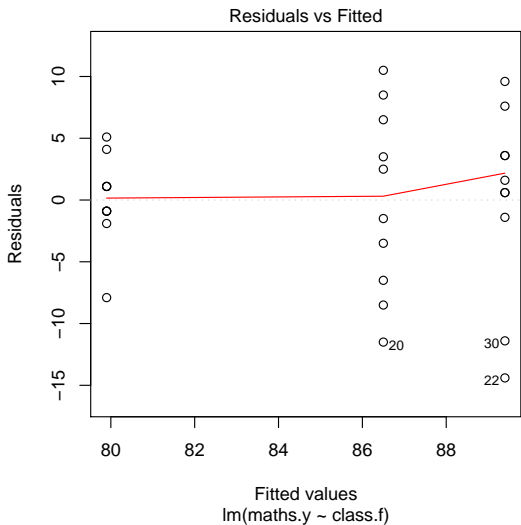
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.492 on 27 degrees of freedom

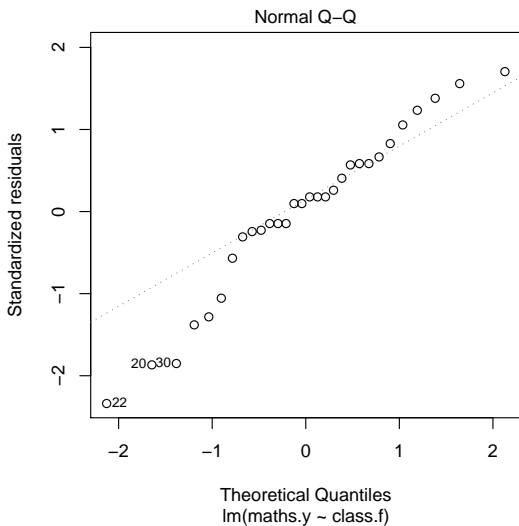
Multiple R-squared: 0.2941, Adjusted R-squared: 0.2418

F-statistic: 5.625 on 2 and 27 DF, p-value: 0.009077

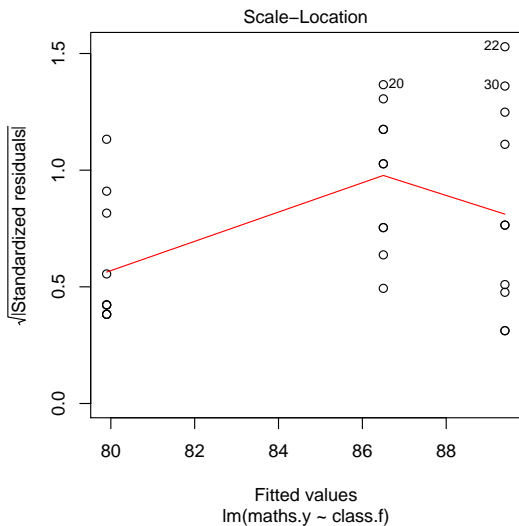
```
> plot(model, which = 1)
```



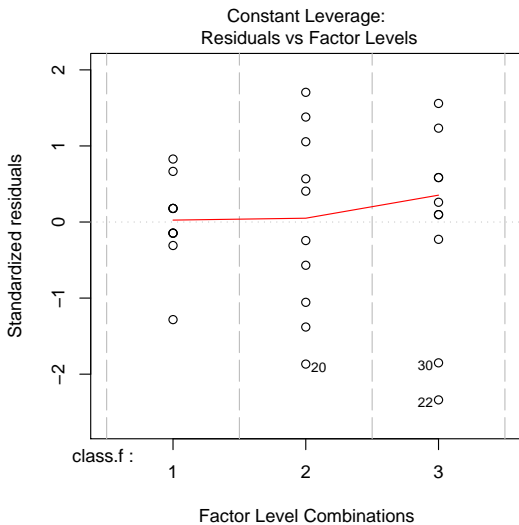
```
> plot(model, which = 2)
```



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



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



Estimating σ^2

```

> SSRes <- t(y - X %*% b) %*% (y - X %*% b)
> SSRes
      [,1]
[1,] 1137.8
> t(y - X %*% b2) %*% (y - X %*% b2)
      [,1]
[1,] 1137.8
> sum(y^2) - t(y) %*% X %*% XtXc %*% t(X) %*% y
      [,1]
[1,] 1137.8
> s2 <- SSRes/(n - r(X))
> s2
      [,1]
[1,] 42.14074

```

Estimating σ^2 (R)

```
> deviance(model)
```

```
[1] 1137.8
```

```
> deviance(model)/model$df.residual
```

```
[1] 42.14074
```

Interval estimation

We first find a confidence interval for the estimable quantity $\mu + \tau_1$, the mean of population 1.

```
> tt <- as.vector(c(1, 1, 0, 0))
> halfwidth <- qt(0.975, df = n - k) * sqrt(s2) * sqrt(t(tt) %*%
+      XtXc %*% tt)
> c(tt %*% b - halfwidth, tt %*% b + halfwidth)
```

```
[1] 75.68796 84.11204
```

```
> newdata <- data.frame(class.f = factor(1))
> predict(model, newdata, interval = "confidence", level = 0.95)
```

```
      fit      lwr      upr
1 79.9 75.68796 84.11204
```

We now find a confidence interval for the estimable quantity $\tau_1 - \tau_2$, the difference between the first two populations.

```
> tt <- as.vector(c(0, 1, -1, 0))
> halfwidth <- qt(0.975, df = n - k) * sqrt(s2) * sqrt(t(tt) %*%
+      XtXc %*% tt)
> c(tt %*% b - halfwidth, tt %*% b + halfwidth)

[1] -12.5567252  -0.6432748
```

Any estimation using R has to be done relative to the treatment contrasts used.

```
> library(gmodels)
> ci <- estimable(model, c(0, -1, 0), conf.int = 0.95)
> ci
```

	Estimate	Std. Error	t value	DF	Pr(> t)	Lower
(0 -1 0)	-6.6	2.903127	-2.273410	27	0.03117113	-12.5567252

```
> c(ci$Lower, ci$Upper)
```

```
[1] -12.5567252 -0.6432748
```

```
> confint(model, level = 0.95)
```

	2.5 %	97.5 %
(Intercept)	75.6879592	84.11204
class.f2	0.6432748	12.55673
class.f3	3.5432748	15.45673