MAST90104: Introduction to Statistical Learning

Solutions to Week 8 Lab and Workshop

1. The dataset `wbca` comes from a study of breast cancer in Wisconsin. There are 681 cases of potentially cancerous tumors of which 238 are actually malignant. Determining whether a tumor is really malignant is traditionally determined by an invasive surgical procedure. The purpose of this study was to determine whether a new procedure called fine needle aspiration, which draws only a small sample of tissue, could be effective in determining tumor status.

   (a) Load the data and read descriptions of the variables using

   ```
   library(faraway)
   data(wbca)
   ?wbca
   ```

   (b) Fit a binary regression model (logistic regression in this case) using `glm`. Include all the variables in your model (shorthand for this in an R model is $\sim$.).
   Solution:

   ```
   model <- glm(cbind(Class, 1-Class)~., family=binomial, data=wbca)
   summary(model)

   ##
   ## Call:
   ## glm(formula = cbind(Class, 1 - Class) ~ ., family = binomial,
   ##     data = wbca)
   ##
   ## Deviance Residuals:
   ##      Min        1Q    Median        3Q       Max
   ## -2.48282  -0.01179   0.04739   0.09678   3.06425
   ##
   ## Coefficients:
   ##             Estimate Std. Error z value Pr(>|z|)
   ## (Intercept) 11.16678    1.41491   7.892 2.97e-15 ***
   ## Adhes       -0.39681    0.13384  -2.965  0.00303 **
   ## BNucl       -0.41478    0.10230  -4.055 5.02e-05 ***
   ## Chrom       -0.56456    0.18728  -3.014  0.00257 **
   ## Epith       -0.06440    0.16595  -0.388  0.69795
   ## Mitos       -0.65713    0.36764  -1.787  0.07387 .
   ## NNucl       -0.28659    0.12620  -2.271  0.02315 *
   ## Thick       -0.62675    0.15890  -3.944 8.01e-05 ***
   ## UShap       -0.28011    0.25235  -1.110  0.26699
   ## USize        0.05718    0.23271   0.246  0.80589
   ## ---
   ## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
   ##
   ## (Dispersion parameter for binomial family taken to be 1)
   ##
   ##     Null deviance: 881.388  on 680  degrees of freedom
   ## Residual deviance:  89.464  on 671  degrees of freedom
   ## AIC: 109.46
   ##
   ## Number of Fisher Scoring iterations: 8
   ```

   (c) Use the `step` function to search for a model with minimal AIC. Include all variables in the `scope` (type `?step` to see how to use `step`).

You should end up with the model `cbind(Class, 1 - Class)` ∼ `Adhes + BNucl + Chrom + Mitos + NNucl + Thick + UShap`.

Solution:

```
model2 <- step(model, scope=~.)
```

```
## Start:  AIC=109.46
## cbind(Class, 1 - Class) ~ Adhes + BNucl + Chrom + Epith + Mitos +
##     NNucl + Thick + UShap + USize
##
##          Df Deviance    AIC
## - USize   1   89.523 107.52
## - Epith   1   89.613 107.61
## - UShap   1   90.627 108.63
## <none>        89.464 109.46
## - Mitos   1   93.551 111.55
## - NNucl   1   95.204 113.20
## - Adhes   1   98.844 116.84
## - Chrom   1   99.841 117.84
## - BNucl   1  109.000 127.00
## - Thick   1  110.239 128.24
##
## Step:  AIC=107.52
## cbind(Class, 1 - Class) ~ Adhes + BNucl + Chrom + Epith + Mitos +
##     NNucl + Thick + UShap
##
##          Df Deviance    AIC
## - Epith   1   89.662 105.66
## - UShap   1   91.355 107.36
## <none>        89.523 107.52
## + USize   1   89.464 109.46
## - Mitos   1   93.552 109.55
## - NNucl   1   95.231 111.23
## - Adhes   1   99.042 115.04
## - Chrom   1  100.153 116.15
## - BNucl   1  109.064 125.06
## - Thick   1  110.465 126.47
##
## Step:  AIC=105.66
## cbind(Class, 1 - Class) ~ Adhes + BNucl + Chrom + Mitos + NNucl +
##     Thick + UShap
##
##          Df Deviance    AIC
## <none>        89.662 105.66
## - UShap   1   91.884 105.88
## + Epith   1   89.523 107.52
## + USize   1   89.613 107.61
## - Mitos   1   93.714 107.71
## - NNucl   1   95.853 109.85
## - Adhes   1  100.126 114.13
## - Chrom   1  100.844 114.84
## - BNucl   1  109.762 123.76
## - Thick   1  110.632 124.63
```

```
summary(model2)
```

```
##
## Call:
## glm(formula = cbind(Class, 1 - Class) ~ Adhes + BNucl + Chrom +
##     Mitos + NNucl + Thick + UShap, family = binomial, data = wbca)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.44161  -0.01119   0.04962   0.09741   3.08205
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  11.0333     1.3632   8.094 5.79e-16 ***
## Adhes        -0.3984     0.1294  -3.080  0.00207 **
## BNucl        -0.4192     0.1020  -4.111 3.93e-05 ***
## Chrom        -0.5679     0.1840  -3.085  0.00203 **
## Mitos        -0.6456     0.3634  -1.777  0.07561 .
## NNucl        -0.2915     0.1236  -2.358  0.01837 *
## Thick        -0.6216     0.1579  -3.937 8.27e-05 ***
## UShap        -0.2541     0.1785  -1.423  0.15461
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 881.388  on 680  degrees of freedom
## Residual deviance:  89.662  on 673  degrees of freedom
## AIC: 105.66
##
## Number of Fisher Scoring iterations: 8
```

(d) Using the reduced model, use `predict` to estimate the outcome for a new patient with predictors 1, 1, 3, 1, 1, 4, 1. You will need to put `newdata = list(Adhes=1, BNucl=1, Chrom=3, Mitos=1, NNucl=1, Thick=4, UShap=1)` and `type="response"`.

To get a 95% CI for your estimate, use `predict` with `type="link"` and `se.fit=TRUE`, to obtain the estimate and its standard error *on the linear scale*. Use these to get a symmetric CI on the linear scale, which you can then transform back to the response scale.

Solution:

```
predict(model2, newdata = list(Adhes=1, BNucl=1, Chrom=3, Mitos=1, NNucl=1,
Thick=4, UShap=1), type="response")

##         1
## 0.9921115

(x <- predict(model2, newdata = list(Adhes=1, BNucl=1, Chrom=3, Mitos=1, NNucl=1,
Thick=4, UShap=1), type="link", se.fit=TRUE))

## $fit
##        1
## 4.834428
##
## $se.fit
## [1] 0.5815185
##
## $residual.scale
## [1] 1

ilogit(c(x$fit-2*x$se.fit, x$fit, x$fit+2*x$se.fit))

##         1         1         1
## 0.9751901 0.9921115 0.9975211
```

(e) Suppose that a cancer is classified as benign if $p > 0.5$ and malignant if $p < 0.5$. Compute the number of errors of both types that will be made if this method is applied to the current data with the reduced model.

Solution:

```
pfit <- predict(model2, type="response")
(false_neg <- sum(pfit >= 0.5 & !wbca$Class)/sum(!wbca$Class))

## [1] 0.04621849

(false_pos <- sum(pfit < 0.5 & wbca$Class)/sum(wbca$Class))

## [1] 0.02031603
```

(f) Suppose we change the cutoff to 0.9 so that $p < 0.9$ is classified as malignant and $p > 0.9$ as benign. Compute the number of errors in this case.

Consider how you might determine the cutoff in practice.

Solution:

```
pfit <- predict(model2, type="response")
(false_neg <- sum(pfit >= 0.9 & !wbca$Class)/sum(!wbca$Class))

## [1] 0.004201681

(false_pos <- sum(pfit < 0.9 & wbca$Class)/sum(wbca$Class))

## [1] 0.03611738
```

Clearly there is a trade-off between false positives and false negatives. Where you choose the cut-off depends on the relative costs (individial and societal) in each case. For medical tests we usually prefer to reduce the false negative rate at the expense of increasing the false positive rate, especially for a screening test, where there is the opportunity for further testing following a positive result.

2. The National Institute of Diabetes and Digestive and Kidney Diseases conducted a study on 768 adult female Pima Indians living near Phoenix. The purpose of the study was to investigate factors related to diabetes. The data may be found in the the dataset `pima`.

(a) Read the help file (`?pima`) to get a description of the predictor and response variables, then use `pairs` and `summary` to perform simple graphical and numerical summaries of the data.

There are some obvious irregularities in the data. Take appropriate steps to correct the problems.

Solution:

It is clear that there are missing observations for many variables, which have been recorded as zeros. The easiest (not necessarily the only or best) way to deal with these is to remove the relevant observations from the data set. On the other hand, 0 is a plausible value for insulin, diabetes and test so these 0's are not excluded.

```
missing <- with(pima, missing <- glucose==0 | diastolic==0 | triceps==0 | bmi == 0)
pima <- pima[!missing,]
```

(b) Fit a model with `test` as the response and all the other variables as predictors.

Can you tell whether this model fits the data?

Solution:

```
model <- glm(cbind(test, 1-test)~., family=binomial, data=pima)
model2 <- step(model, scope=~.)

## Start:  AIC=483.23
## cbind(test, 1 - test) ~ pregnant + glucose + diastolic + triceps +
##     insulin + bmi + diabetes + age
##
##             Df Deviance    AIC
## - triceps    1   465.41 481.41
## - diastolic  1   466.02 482.02
## - insulin    1   466.32 482.32
## <none>           465.23 483.23
## - age        1   468.85 484.85
## - pregnant   1   473.05 489.05
## - bmi        1   479.04 495.04
## - diabetes   1   479.29 495.29
## - glucose    1   543.21 559.21
##
## Step:  AIC=481.41
## cbind(test, 1 - test) ~ pregnant + glucose + diastolic + insulin +
##     bmi + diabetes + age
##
##             Df Deviance    AIC
## - diastolic  1   466.20 480.20
## - insulin    1   466.53 480.53
## <none>           465.41 481.41
## - age        1   469.14 483.14
## + triceps    1   465.23 483.23
## - pregnant   1   473.39 487.39
## - diabetes   1   479.61 493.61
## - bmi        1   490.31 504.31
## - glucose    1   543.93 557.93
##
## Step:  AIC=480.2
## cbind(test, 1 - test) ~ pregnant + glucose + insulin + bmi +
##     diabetes + age
##
##             Df Deviance    AIC
## - insulin    1   467.08 479.08
## <none>           466.20 480.20
## - age        1   469.30 481.30
## + diastolic  1   465.41 481.41
## + triceps    1   466.02 482.02
## - pregnant   1   474.31 486.31
## - diabetes   1   480.67 492.67
## - bmi        1   490.99 502.99
## - glucose    1   544.07 556.07
##
## Step:  AIC=479.08
## cbind(test, 1 - test) ~ pregnant + glucose + bmi + diabetes +
##     age
##
##             Df Deviance    AIC
## <none>           467.08 479.08
## + insulin    1   466.20 480.20
## - age        1   470.30 480.30
## + diastolic  1   466.53 480.53
## + triceps    1   466.88 480.88
## - pregnant   1   475.43 485.43
## - diabetes   1   481.12 491.12
## - bmi        1   491.22 501.22
## - glucose    1   553.65 563.65
```

```
summary(model2)

##
## Call:
## glm(formula = cbind(test, 1 - test) ~ pregnant + glucose + bmi +
##     diabetes + age, family = binomial, data = pima)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9894  -0.6530  -0.3700   0.6442   2.5417
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.879992   0.918614 -10.755  < 2e-16 ***
## pregnant     0.123887   0.043514   2.847 0.004412 **
## glucose      0.035026   0.004201   8.338  < 2e-16 ***
## bmi          0.085123   0.018110   4.700  2.6e-06 ***
## diabetes     1.321554   0.362538   3.645 0.000267 ***
## age          0.023844   0.013311   1.791 0.073244 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 676.79  on 531  degrees of freedom
## Residual deviance: 467.08  on 526  degrees of freedom
## AIC: 479.08
##
## Number of Fisher Scoring iterations: 5
```

Odds are sometimes a better scale than probability to represent chance. The odds o and probability p are related by

$$o = \frac{p}{1-p} \quad p = \frac{o}{1+o}$$

In a binomial regression model with a logit link we have

$$\text{logit}(p_j) = \log\left(\frac{p_j}{1-p_j}\right) = \eta_j = \beta_0 + \beta_1 x_{1,j} + \cdots + \beta_q x_{q,j}.$$

That is $\log o_j = \eta_j$, where o_j are the odds for the j-th observation.

(c) By what proportion do the odds of testing positive for diabetes change for a woman with a BMI at the first quartile compared with a woman at the third quartile, assuming that all other factors are held constant? Give a confidence interval for this difference.

Solution:

For $i = 1, 3$, let o_i, p_i, η_i be the odds, probability and linear response for a woman with `bmi` at the first and third quartiles respectively (27.87 and 36.90). We have

$$\begin{aligned}\frac{o_1}{o_3} &= \exp(\log(o_1/o_3)) \\ &= \exp(\eta_1 - \eta_3) \\ &= \exp(\beta_{bmi}(27.87 - 36.90))\end{aligned}$$

A point estimate and 95% CI for $\beta_{bmi}(27.87 - 36.90)$ are

$$-9.03(0.085123 \pm 2 \times 0.018110) = -0.7687 \pm 0.3271.$$

Transforming this to the odds scale, we get an odds ratio of $e^{-0.7687} = 0.4636$ with 95% CI $(0.3342, 0.6430)$. So, roughly speaking, all else being equal, the odds of showing evidence of diabetes are between 33 to 64 percent less for a woman with bmi 27.87 compared to a woman with bmi 36.90.

(d) Do women who test positive have higher diastolic blood pressures? Is the diastolic blood pressure significant in the regression model? Explain the distinction between the two questions and discuss why the answers are only apparently contradictory.

Solution:

Recall that `model` used all the predictor variables.

```
cor(pima)
```

```
##               pregnant   glucose   diastolic    triceps      insulin
## pregnant   1.000000000 0.1253296 0.204663421 0.09508511 -0.006568130
## glucose    0.125329647 1.0000000 0.219177950 0.22659042  0.459904606
## diastolic  0.204663421 0.2191779 1.000000000 0.22607244  0.007051676
## triceps    0.095085114 0.2265904 0.226072440 1.00000000  0.126240293
## insulin   -0.006568130 0.4599046 0.007051676 0.12624029  1.000000000
## bmi        0.008576282 0.2470793 0.307356904 0.64742239  0.191167600
## diabetes   0.007435104 0.1658174 0.008047249 0.11863557  0.151531103
## age        0.640746866 0.2789071 0.346938723 0.16133614  0.081126066
## test       0.252585511 0.5036139 0.183431874 0.25487371  0.212204307
##                   bmi    diabetes        age      test
## pregnant  0.008576282 0.007435104 0.64074687 0.2525855
## glucose   0.247079294 0.165817411 0.27890711 0.5036139
## diastolic 0.307356904 0.008047249 0.34693872 0.1834319
## triceps   0.647422386 0.118635569 0.16133614 0.2548737
## insulin   0.191167600 0.151531103 0.08112607 0.2122043
## bmi       1.000000000 0.151107136 0.07343826 0.3009007
## diabetes  0.151107136 1.000000000 0.07165413 0.2330739
## age       0.073438257 0.071654133 1.00000000 0.3150968
## test      0.300900748 0.233073898 0.31509683 1.0000000
```

```
summary(model)
```

```
##
## Call:
## glm(formula = cbind(test, 1 - test) ~ ., family = binomial, data = pima)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8627  -0.6639  -0.3672   0.6347   2.4942
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.677562   1.005400  -9.626  < 2e-16 ***
## pregnant     0.121235   0.043926   2.760 0.005780 **
## glucose      0.037439   0.004765   7.857 3.92e-15 ***
## diastolic   -0.009316   0.010446  -0.892 0.372494
## triceps      0.006341   0.014853   0.427 0.669426
## insulin     -0.001053   0.001007  -1.046 0.295651
## bmi          0.085992   0.023661   3.634 0.000279 ***
## diabetes     1.335764   0.365771   3.652 0.000260 ***
## age          0.026430   0.013962   1.893 0.058371 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 676.79  on 531  degrees of freedom
## Residual deviance: 465.23  on 523  degrees of freedom
## AIC: 483.23
##
## Number of Fisher Scoring iterations: 5
```

`diastolic` is not significant in the presence of the other variables.

There is positive correlation between `diastolic` and `test`, yet in the model `diastolic` has a negative coefficient. This is possible because `diastolic` is correlated with other (more significant) variables: the test *is* more likely to be positive when `diastolic` is large, but this is because `glucose`, `triceps`, `bmi` and `age` are all more likely to be large, and these all have the effect of increasing the chance of a positive test.

(e) Predict the outcome for a woman with predictor values 1, 99, 64, 22, 76, 27, 0.25, 25 (same order as in the dataset). Give a confidence interval for your prediction.

Solution:

```
x <- predict(model2, newdata = list(pregnant=1, glucose=99, bmi=27, diabetes=.25, age=25),
type="link", se.fit=TRUE)
ilogit(c(x$fit-2*x$se.fit, x$fit, x$fit+2*x$se.fit))
```

```
##          1          1          1
## 0.02658525 0.04463032 0.07399266
```

3. Consider the binomial regression model with logit link fitted to the Challenger data in class. Using the log likelihood ratio, plot a 95% confidence region for (α, β).

One way of doing this is to use the function `contour`:

(a) Let $(\hat{\alpha}^*, \hat{\beta}^*)$ be the MLE, then for a grid of α and β values calculate $2l(\hat{\alpha}^*, \hat{\beta}^*) - 2l(\alpha, \beta)$.

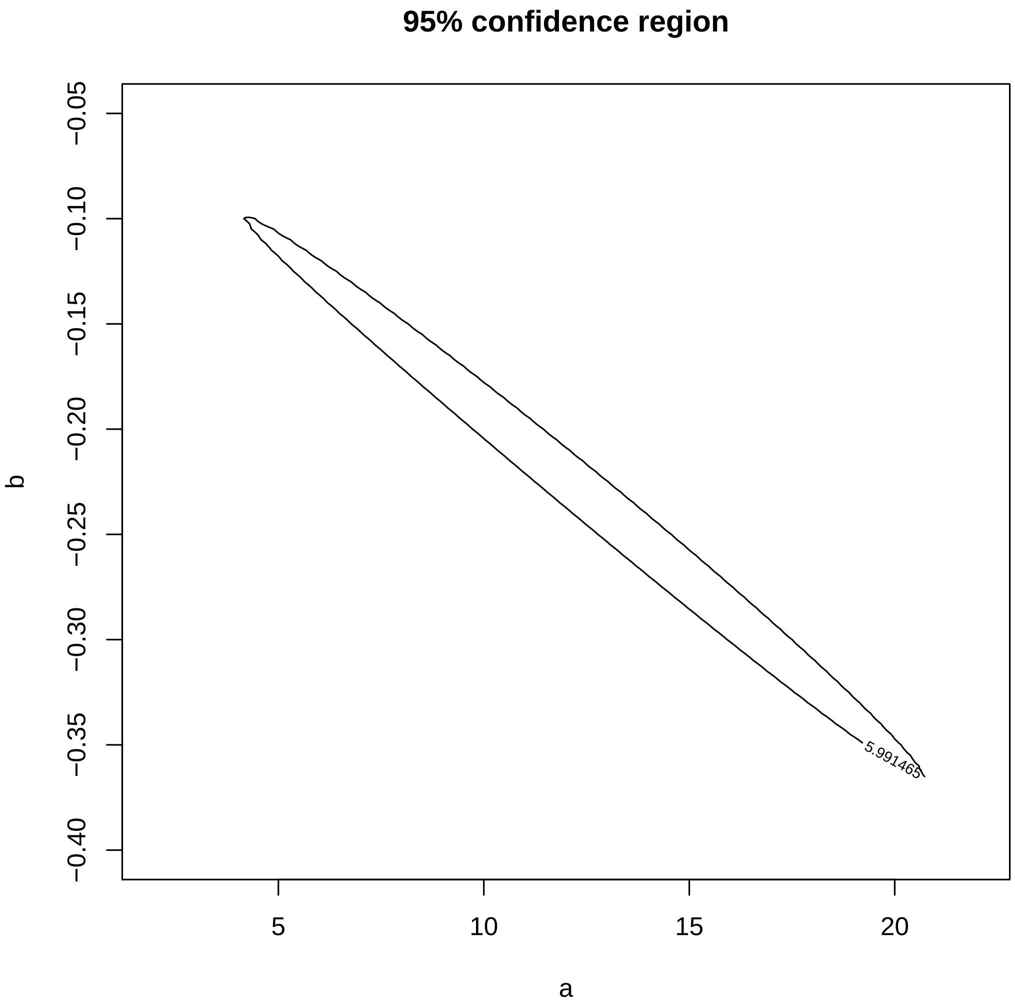
(b) The contour line with value $\chi^2_2(0.95)$ will delineate the confidence region.

Solution:

```
# load data and fit model
library(faraway)
data(orings)
logitmod <- glm(cbind(damage,6-damage) ~ temp, family=binomial, orings)

# log-likelihood function
logL <- function(beta, orings) {
eta <- cbind(1, orings$temp) %*% beta
return( sum(orings$damage*eta - 6*log(1 + exp(eta))) )
}
# log-likelihood ratio for beta = c(a, b) against beta = betafit
logLR <- function(a, b, betafit, orings) 2*logL(betafit, orings) - 2*logL(c(a, b), orings)

# interested in c(a, b) such that f(a, b, ...) <= qchisq(0.95, 2)
a_vec <- seq(2, 22, 0.1)
b_vec <- seq(-0.4, -0.05, .005)
z <- matrix(0, nrow = length(a_vec), ncol = length(b_vec))
for (i in 1:length(a_vec)) {
for (j in 1:length(b_vec)) {
z[i,j] <- logLR(a_vec[i], b_vec[j], logitmod$coefficients, orings)
}
}
# a vectorised alternative for R afficionados
# z <- outer(a_vec, b_vec, Vectorize(logLR, c("a", "b")),
#            betafit = logitmod$coefficients, orings = orings)
contour(a_vec, b_vec, z, levels = qchisq(0.95, 2),
xlab="a", ylab="b", main="95% confidence region")
```



4. Suppose the less than full rank matrix X is $n \times p$ of rank r and that C is $p \times r$. Suppose further that X has r linearly independent columns and that the corresponding rows of C are also linearly independent. The following parts combine to show that XC is full rank if, and only if, $I_r + DE$ is rank r where, if necessary by reordering the rows and columns of X and the rows of C, X & C have been partitioned as

$$X = \left[\begin{array}{c|c} X_r & X_rD \\ \hline FX_r & FX_rD \end{array}\right] \quad C = \left[\begin{array}{c} C_r \\ \hline EC_r \end{array}\right],$$

X_r, F, D, C_r, E are respectively $r \times r, n-r \times r, r \times p-r, r \times r \,\&\, p-r \times r$ and X_r, C_r are both rank r.

(a) Show that the rows and columns of X can be rearranged to achieve the partitions given.

Solution: Because X has rank r, it has r linearly independent rows. Arrange that these rows are on top by reordering the rows of X if necessary.

The remaining $n-r$ rows must all be linear combinations of the first r linearly independent rows - otherwise X would have rank greater than r. These rows can be written as a matrix, F , which is $n-r \times r$, premultiplying the remaining $n-r$ rows of X.

By assumption X has r linearly independent columns. Arrange that these rows are to the left by reordering the columns of X if necessary. When the columns of X are reordered, since the columns of XC are linear combinations of the columns of X with weights in the rows of C, it is necessary to rearrange the rows of C in the same way as the columns of X. By assumption, this can be done so that the rows of C at the top are linearly independent.

Because X is rank r and the first r columns are linearly independent, the remaning $p - r$ columns are linear combinations of the first r columns. Hence, those remaining columns are a matrix consisting of the first r columns postmultiplied by a matrix D which is $r \times p - r$.

Using the reordered rows and columns gives the desired partition of X.

Because C is rank r and the first r rows are linearly independent, the remaning $p - r$ rows are linear combinations of the first r rows. Hence, these remaining rows are a matrix consisting of the first r columns premultiplied by a matrix, E, which is $p - r \times r$. This gives the desired partition of C.

(b) Show that $r(XC) = r(I_r + DE)$.

Solution: Since multiplication of partitioned matrices follows the same rules as normal matrix multiplication:

$$XC = \left[\begin{array}{c} X_rC_r + X_rDEC_r \\ \hline FX_rC_r + FX_rDEC_r \end{array}\right] = \left[\begin{array}{c} X_r(I_r + DE)C_r \\ \hline FX_r(I_r + DE)C_r \end{array}\right]$$

since matrix multiplication follows the distributive laws.

Since adding rows to a matrix cannot decrease the rank, $r(XC) \geq r(X_r(I + DE)C_r) = r(I + DE)$.

Let

$$G = \left[\begin{array}{c} X_r \\ \hline FX_r \end{array}\right]$$

so that

$$XC = G(I_r + DE)C_r.$$

Then $r(XC) = r(G(I_r + DE)C_r) \leq r(I + DE)$.

Combining the two inequalities on $r(XC)$ gives the required equality.

(c) Show that XC is full rank if, and only if, $I_r + DE$ is rank r.

Solution: $I_r + DE$ is $r \times r$ and XC is full rank is the same as $r(XC) = r$ giving the result.

5. Find the matrices D and E and verify that $I_r + DE$ is rank r for C_r from the `contr.treatment` and `contr.sum` matrices in R. Use these to veryify the reparameterisation equations given in notes.

Solution: These both apply to one factor which is supposed to have r levels. It suffices to assume $n = r$. So the original design matrix is

$$X = [\mathbf{1}_r I_r]$$

where $\mathbf{x}_r$ is a column vector of r whose values are all x.

For `contr.treatment` the matrix C, with the ordering of columns and rows in X, is

$$C = [\mathbf{e}_1^{r+1}|\mathbf{e}_3^{r+1}|\cdots|\mathbf{e}_{r+1}^{r+1}].$$

where $\mathbf{e}_1^{r+1}, \cdots, \mathbf{e}_{r+1}^{r+1}$ are the standard unit $r + 1-$vectors.

In this case, the second row of C is all 0 so is not linearly independent of the other rows. Hence, the rows of C, and the corresponding columns of X, need to be reordered in order to ensure that the first r rows are linearly independent.

The re-ordering is

$$X = [\mathbf{1}_r|\mathbf{e}_2^r|\cdots|\mathbf{e}_r^r|\mathbf{e}_1^r], C = \left[\begin{array}{c} I_r \\ \mathbf{0}^T \end{array}\right].$$

The last column of X is now the first minus the sum of the remaining $r - 1$, so that

$$D = \left[\begin{array}{c} 1 \\ -\mathbf{1}_r \end{array}\right], C_r = I_r, E = \mathbf{0}^T.$$

Hence

$$I_r + DE = I_r$$

which is rank r as required.

Using equation 1 in Module 6,

$$\boldsymbol{\beta}_r = \begin{bmatrix} \mu \\ \tau_2 \\ \vdots \\ \tau_r \end{bmatrix}, \boldsymbol{\beta}_{p-r} = \tau_1, (C_r(I_r + DE))^{-1} = I_r$$

so

$$\boldsymbol{\gamma} = \boldsymbol{\beta}_r + \tau_1 D = \begin{bmatrix} \mu + \tau_1 \\ \tau_2 - \tau_1 \\ \vdots \\ \tau_r - \tau_1 \end{bmatrix},$$

as required.

In completing the question for `contr.sum`, the following R commands show analysis of the case $r = 4$.

```
# The C matrix is obtained from adding a column of
# zeros to contr.sum and then adding a row which
# is the transpose of the first standard unit vector
#
(C <- rbind(t(c(1,0,0,0)),cbind(c(0,0,0,0),contr.sum(4))))

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

#
# Cr is the C matrix minus the last row
# Here it is the identity matrix
#
(Cr <- C[-5,])

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

#
# The original design matrix is used
# The column vector d is the same as for contr.treatment
#
(X <- cbind(c(1,1,1,1), diag(4)))
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    1    0    0    0
## [2,]    1    0    1    0    0
## [3,]    1    0    0    1    0
## [4,]    1    0    0    0    1
```

```
D <- c(1,-1,-1,-1)
E <-  t(c(0,-1,-1,-1))
#
# Use package plyr to get the pipe command, %>%
# Use package fractional to get the output as fractions
# Calculate I + DE
# Invert it to check rank
# Calculate the multiplying matrix, mult, to reparameterise
#
library(dplyr)
```

```
##
## Attaching package:  'dplyr'
## The following objects are masked from 'package:stats':
##
##    filter, lag
## The following objects are masked from 'package:base':
##
##    intersect, setdiff, setequal, union
```

```
library(fractional)
diag(4) + D%*%E
```

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

```
mult1 <- solve((diag(4) + D%*%E)%*%Cr) %>% fractional %>% print
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1  1/4  1/4  1/4
## [2,]    .  3/4 -1/4 -1/4
## [3,]    . -1/4  3/4 -1/4
## [4,]    . -1/4 -1/4  3/4
```

```
mult <- mult1%*%cbind(diag(4),D) %>% fractional %>% print
```

```
##                          D
## [1,]    1  1/4  1/4  1/4  1/4
## [2,]    .  3/4 -1/4 -1/4 -1/4
## [3,]    . -1/4  3/4 -1/4 -1/4
## [4,]    . -1/4 -1/4  3/4 -1/4
```

For general r, the original design matrix is used and

$$C = \left[\begin{array}{c|c} \multicolumn{2}{c}{I_r} \\ \hline 0 & -\mathbf{1}_{r-1} \end{array}\right], D = \left[\begin{array}{c} 1 \\ -\mathbf{1}_r \end{array}\right], C_r = I_r, E = \left[\begin{array}{c|c} 0 & -\mathbf{1}_{r-1}^T \end{array}\right].$$

and

$$I_r + DE = \left[\begin{array}{c|c} 1 & -\mathbf{1}^T_{r-1} \\ \hline \mathbf{0}_{r-1} & I_{r-1} + \mathbf{1}_{r-1\times r-1} \end{array}\right]$$

where $\mathbf{x}_{m\times n}$ is an $m \times n$ matrix filled with the values x.

From the case of $r = 4$ in the R output, the most obvious conjecture, remembering that $C_r = I_r$, is that

$$((I_r + DE)C_r)^{-1} = \frac{1}{r}\left[\begin{array}{c|c} r & \mathbf{1}^T_{r-1} \\ \hline \mathbf{0}_{r-1} & rI_{r-1} - \mathbf{1}_{r-1\times r-1} \end{array}\right].$$

Postmultiplying the matrix on the right side by $I + DE$ gives I_r as required, showing that the rank of $I + DE$ is r, as required. To see this, use the formula for multiplying partitioned matrices:

$$\begin{aligned}
&\frac{1}{r}\left[\begin{array}{c|c} r & \mathbf{1}^T_{r-1} \\ \hline \mathbf{0}_{r-1} & rI_{r-1} - \mathbf{1}_{r-1\times r-1} \end{array}\right] \times \left[\begin{array}{c|c} 1 & -\mathbf{1}^T_{r-1} \\ \hline \mathbf{0}_{r-1} & I_{r-1} + \mathbf{1}_{r-1\times r-1} \end{array}\right] \\
&\quad = \frac{1}{r}\left[\begin{array}{c|c} r + \mathbf{1}^T_{r-1}\mathbf{0}_{r-1} & -r\mathbf{1}^T_{r-1} + \mathbf{1}^T_{r-1}(I_{r-1} + \mathbf{1}_{r-1\times r-1}) \\ \hline \mathbf{0}_{r-1} \times 1 + (rI_{r-1} - \mathbf{1}_{r-1\times r-1})\mathbf{0}_{r-1} & \mathbf{0}_{r-1} \times -\mathbf{1}_{r-1} + (rI_{r-1} - \mathbf{1}_{r-1\times r-1})(I_{r-1} + \mathbf{1}_{r-1\times r-1}) \end{array}\right] \\
&\quad = \frac{1}{r}\left[\begin{array}{c|c} r & \mathbf{0}^T_{r-1} \\ \hline \mathbf{0}_{r-1} & rI_{r-1} + r\mathbf{1}_{r-1\times r-1} - \mathbf{1}_{r-1\times r-1} - \mathbf{1}_{r-1\times r-1} \times \mathbf{1}_{r-1\times r-1} \end{array}\right] \\
&\quad = I_r.
\end{aligned}$$

This now also gives:

$$\begin{aligned}
(I_r + DE)^{-1}[I_r|D] &= \frac{1}{r}\left[\begin{array}{c|c} r & \mathbf{1}^T_{r-1} \\ \hline \mathbf{0}_{r-1} & rI_{r-1} - \mathbf{1}_{r-1\times r-1} \end{array}\right] \times \left[\begin{array}{c|c|c} 1 & \mathbf{0}_{r-1} & 1 \\ \hline \mathbf{0}_{r-1} & I_{r-1} & -\mathbf{1}_{r-1} \end{array}\right] \\
&= \frac{1}{r}\left[\begin{array}{c|c} r & \mathbf{1}^T_{r} \\ \hline \mathbf{0}^T_{r-1} & I_r - \mathbf{1}_{r-1\times r-1} \end{array}\right]
\end{aligned}$$

$$.$$

Multiplying this matrix on the right side by the one factor less than full rank parameter vector gives

$$\boldsymbol{\gamma} = \begin{bmatrix} \mu + \bar{\tau} \\ \tau_1 - \bar{\tau} \\ \vdots \\ \tau_{r-1} - \bar{\tau} \end{bmatrix}$$

where

$$\bar{\tau} = \frac{\sum_{i=1}^{r} \tau_i}{r},$$

as required.

6. Prove Theorem 6.2 using the following steps.

   (a) Show that under the conditions of Theorem 6.1 (question 4 above), the column space of XC is the same as the column space of X.

   Solution: Every column of XC is a linear combination of columns in X so it in the column space of X.

   Since, under the conditions of the Theorem, XC is full rank, the columns of XC are thus a basis for the column space of X. Hence every element of the column space of X can be expressed as a linear combination of the columns of XC. That is every element of the column space of X is in the column space of XC, showing that the two column spaces are the same.

(b) Show that if two full-rank linear models have the same column space, the eigenvectors of their hat matrices are the same.

Solution:

Suppose the two full rank linear models with the same column space have hat matrices H_1, H_2.

Then their eigenvalues are all either 0 or 1 since they are idempotent, symmetric matrices.

Take an eigenvector, $\mathbf{x}$, for H_1 which has eigenvalue 1.

Then $\mathbf{x}$ is also an eigenvector with eigenvalue 1 for H_2 since

$$H_2\mathbf{x} = H_2H_1\mathbf{x} = H_1\mathbf{x} = \mathbf{x},$$

the third step following since $H_1\mathbf{x}$ is in the common column space and any hat matrix leaves elements of the column space unchanged.

This shows that for every $n-$ vector $\mathbf{y}$ is a linear combination of the common unit eigenvectors of the common column space plus a linear combination of the 0 eigenvectors for H_2.

For an eigenvector, $\mathbf{z}$, for H_1 with eigenvalue 0 this must only be a linear combination of the 0 eigenvectors of H_2, since any of the other common linearly independent eigenvectors will produce a non zero value for $H_1\mathbf{z}$. The linearity of H_2 then shows that $H_2\mathbf{z} = 0$, so $\mathbf{z}$ is a 0 eigenvector for H_2.

(c) Hence show that if the column space for two linear models is the same, the fitted values are the same.

Solution: The fitted values for $\mathbf{y}$ are $H_1\mathbf{y}$ and $H_2\mathbf{y}$. But $\mathbf{y}$ can be written as a linear combination of the common eigenvectors of H_1, H_2 and the multipliers are uniquely defined. When either hat matrix is applied to the linear combination, the result is the linear combination of the eigenvectors which have eigenvalue 1.

(d) Complete the proof of Theorem 6.2.

Solution:

All of the other quantities in the theorem are functions of the data and the fitted values.

7. Deferred to next week.