R Markdown demo

DP

September 4, 2016

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

"my cutoff"

Everything above the line ("my cutoff") was created automatically when I created a new file by selecting File > New File > R Markdown in RStudio. I was prompted to provide a title and author.

The file originally contained more about R Markdown, below my cutoff line. I replaced that material by some Linear Models stuff then clicked on "Knit PDF" in the toolbar.

```
set.seed(10)   # for reproducibility
mydata <- data.frame(y=rnorm(10),
    x1=1:10,x2= 11:20, x3= 0.5*(1:10)-3*(11:20))
out <- lm(y ~ ., data=mydata)
summary(out)
```

```
##
## Call:
## lm(formula = y ~ ., data = mydata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0211 -0.5231  0.1832  0.4320  0.9085
##
## Coefficients: (2 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.18175    0.49193  -0.369    0.721
## x1          -0.05616    0.07928  -0.708    0.499
## x2                NA         NA      NA       NA
## x3                NA         NA      NA       NA
##
## Residual standard error: 0.7201 on 8 degrees of freedom
## Multiple R-squared:  0.05903,    Adjusted R-squared:  -0.05859
## F-statistic: 0.5019 on 1 and 8 DF,  p-value: 0.4988
```

Now let's try to figure out what **R** has done. First determine which matrix **R** fed to the $qr()$ function:

```
M <- model.matrix(out) # should have out$qr equal to qr(M)
round(cbind(M,mydata)[1:4,],3) # for comparison
```

```
##   (Intercept) x1 x2    x3      y x1 x2    x3
## 1           1  1 11 -32.5  0.019  1 11 -32.5
## 2           1  2 12 -35.0 -0.184  2 12 -35.0
## 3           1  3 13 -37.5 -1.371  3 13 -37.5
## 4           1  4 14 -40.0 -0.599  4 14 -40.0
```

As expected, **R** prepended a column of 1's to the predictors in *mydata*. You might want to compare *out*$*qr* with $qr(M)$.

Now extract the matrices for the QR decomposition of the model matrix:

```
Q <- qr.Q(out$qr)
R <- qr.R(out$qr)
# What would you expect
# round(cbind( Q %*% R,M),3)
#to show?
round(Q[1:4,],2) # Why are there four columns?
```

```
##       [,1]  [,2]  [,3]  [,4]
## [1,] -0.32 -0.50 -0.34 -0.32
## [2,] -0.32 -0.39 -0.24 -0.12
## [3,] -0.32 -0.28  0.90 -0.08
## [4,] -0.32 -0.17 -0.09  0.92
```

```
round(R,3) # Why is it 4 by 4 ?
```

```
##   (Intercept)      x1      x2      x3
## 1      -3.162 -17.393 -49.015 138.350
## 2       0.000   9.083   9.083 -22.707
## 3       0.000   0.000   0.000   0.000
## 4       0.000   0.000   0.000   0.000
```

Notice that R[3:4,] is all zeros. That means that only the first two columns of Q are being used to span the model space; the model matrix has rank 2. Let me split both matrices in the way described in the QR.pdf handout:

```
Q1 <- Q[,1:2]
R1 <- R[1:2,1:2]
R2 <- R[1:2,3:4]
# as a check look at
# round(cbind( M, Q1 %*% R1, Q1 %*% R2),3)
```

According to QR.pdf, the fitted vector $\widehat{y}$ should equal $Q_1 Q_1^T y$:

```
# round(out$fitted.values- Q1 %*% t(Q1) %*% mydata$y,4) # all zero?
```

The matrix $Q_1 Q_1^T$ projects ten-dimensional Euclidean space orthogonally onto the two-dimensional subspace spanned by a column of 1's and x_1, x_2, x_3.

If you read through QR.pdf you should see how to calculate other parts the summary using only~y and~*out*$*qr*. Homework~1 essentially asks you to add some more calculations to this handout.