The QR decomposition

How is the LS estimate $\hat{\boldsymbol{\beta}}$ solved in R? Denote the QR decomposition (also called the QR factorization) of $\mathbf{X}$ as

$$\mathbf{X}_{n\times p} = \mathbf{Q}_{n\times p}\mathbf{R}_{p\times p}$$

where $\mathbf{Q}$ is an orthogonal matrix (i.e., $\mathbf{Q}^t\mathbf{Q} = \mathbf{I}_p$) and $\mathbf{R}$ is an upper triangular matrix, i.e., all the entries in $\mathbf{R}$ below the diagonal are equal to 0. Then

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

The last equation, $\mathbf{R}\hat{\boldsymbol{\beta}} = \mathbf{Q}\mathbf{y}$, can be solved pretty easily via *backsolving* since $\mathbf{R}$ is an upper triangular matrix.

One methods for computing the QR decomposition is the *Gram-Schmidt* algorithm. Let's work with a matrix

$$\mathbf{A}_{n\times p} = \big[\mathbf{a}_1 \mid \mathbf{a}_2 \mid \cdots \mid \mathbf{a}_p\big],$$

where $\mathbf{a}_j$ denotes the jth column of $\mathbf{A}$. Then

- $\mathbf{e}_1 = \mathbf{a}_1, \quad \mathbf{q}_1 = \frac{\mathbf{e}_1}{\|\mathbf{e}_1\|}$
- $\mathbf{e}_2 = \mathbf{a}_2 - (\mathbf{a}_2^t\mathbf{q}_1)\mathbf{q}_1, \quad \mathbf{q}_2 = \frac{\mathbf{e}_2}{\|\mathbf{e}_2\|}$
- $\cdots$
- $\mathbf{e}_{k+1} = \mathbf{a}_{k+1} - \sum_{j=1}^{k}(\mathbf{a}_j^t\mathbf{q}_j)\mathbf{q}_j, \quad \mathbf{q}_{k+1} = \frac{\mathbf{e}_{k+1}}{\|\mathbf{e}_{k+1}\|}$

The resulting QR decomposition is

$$A = \big[\mathbf{a}_1 \mid \mathbf{a}_2 \mid \cdots \mid \mathbf{a}_p\big] = \big[\mathbf{q}_1 \mid \cdots \mid \mathbf{q}_p\big]\mathbf{R} = \mathbf{Q}\mathbf{R}.$$

It is each to check that $\mathbf{R}$ is an upper triangular matrix.

Partial regression coefficients

Consider a multiple linear regression model with 4 predictors and an intercept

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \text{err}.$$

The LS estimate $\hat{\beta}_k$ describes the partial correlation between Y and X_k adjusted for the other predictors. Mathematically, the LS estimate $\hat{\beta}_k$ is what we could get if we

- first regress Y onto all other predictors except X_k, denote the the corresponding residuals as a new variable Y^*;

- regress X_k onto all other predictors except X_k, denote the corresponding residuals as a new variable X_k^*;
- then fit a simple linear regression model with Y^* as the response and X_k^* as the predictor.

```
> fullmodel=lm(sr~pop15+pop75+dpi+ddpi, data=savings)
> round(summary(fullmodel)$coef, dig=3)

            Estimate Std. Error t value Pr(>|t|)
(Intercept)   28.566      7.355   3.884    0.000
pop15         -0.461      0.145  -3.189    0.003
pop75         -1.691      1.084  -1.561    0.126
dpi            0.000      0.001  -0.362    0.719
ddpi           0.410      0.196   2.088    0.042

> new.y=lm(sr~pop15+pop75+dpi, data=savings)$res
> new.ddpi = lm(ddpi~pop15+pop75+dpi, data=savings)$res
> parmodel=lm(new.y ~ new.ddpi)
> round(summary(parmodel)$coef, dig=3)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.00      0.521   0.000    1.000
new.ddpi        0.41      0.190   2.157    0.036
```

Note that although the estimated coefficients agree, the standard error and p-value for the corresponding t-test are different. This is because the sample size is miscounted in the 2nd regression: the sample size shouldn't be n, but $n-4$ since both the regressor Y^* and the predictor X_k^* are in an $n-4$ subspace which is orthogonal to the intercept and the other 3 predictors.

```
> sd=summary(parmodel)$coef[2,2]*sqrt(48/45)  # correct std. error
> sd
[1] 0.1961971
> 2*(1-pt(abs(summary(parmodel)$coef[2,1]/sd), 45)) # correct p-value
[1] 0.04247114
```

Since the t-test for β_k is testing the effect of X_k adjusted for the other predictors, it is not surprising to see the equivalence of the t-test and the F-test of comparing the full model to the model including all predictors except X_k.

Sequential analysis of variance (ANOVA)

When adding a new predictor to a regression model, we can evaluate its relevance/effect by the improved RSS. When there are multiple predictors, we can carry out this comparison in a sequential way: first compare the RSS from the model with X_1 to the RSS from the

null model (with only the intercept), then compare the RSS from the model with both X_1 and X_2 to the RSS from the model with X_1 only, and so on. That's what the `R` command `anova` would return you.

```
> rss = sum((sr-mean(sr))^2)
> model1 = lm(sr~pop15)
> rss = c(sum(model1$res^2), rss)
> model2 = lm(sr~pop15+pop75)
> rss = c(sum(model2$res^2), rss)
> model3 = lm(sr~pop15+pop75+dpi)
> rss = c(sum(model3$res^2), rss)
> model4 = lm(sr~pop15+pop75+dpi+ddpi)
> rss = c(sum(model4$res^2), rss)
> rss
[1] 650.7130 713.7670 726.1680 779.5107 983.6283
> round(diff(rss), dig=2)
[1]  63.05  12.40  53.34 204.12

> anova(fullmodel)
Analysis of Variance Table

Response: sr
          Df Sum Sq Mean Sq F value    Pr(>F)
pop15      1 204.12 204.118 14.1157 0.0004922 ***
pop75      1  53.34  53.343  3.6889 0.0611255 .
dpi        1  12.40  12.401  0.8576 0.3593551
ddpi       1  63.05  63.054  4.3605 0.0424711 *
Residuals 45 650.71  14.460

> anova(lm(sr~ddpi+pop15+pop75+dpi))
Analysis of Variance Table

Response: sr
          Df Sum Sq Mean Sq F value    Pr(>F)
ddpi       1  91.37  91.374  6.3190 0.0155920 *
pop15      1 191.70 191.702 13.2571 0.0006984 ***
pop75      1  47.95  47.946  3.3157 0.0752748 .
dpi        1   1.89   1.893  0.1309 0.7191732
Residuals 45 650.71  14.460
```

Of course, order matters: the importance of `ddpi` as the 1st variable entering the model wouldn't be expected to be the same as the last variable entering the model.

We have already seen that evaluating the effect of a single predictor is a difficult problem, since its effect depends on what else are included in the model. Later in this semester we will learn how to select an optimal subset of variables.

Errors in Predictors

Let $\mathbf{X}$ be the observed design matrix of dimension $n \times p$ where the 1st column contains only 1's and the remaining $(p-1)$ columns correspond to the $(p-1)$ non-intercept covariates (predictors). In many cases it's quite possible that there are substantial measurement errors involved. Let $\tilde{\mathbf{X}}$ denote the "true" value of the predictors, and consider the model

$$\mathbf{X} = \tilde{\mathbf{X}} + \mathbf{D},$$

where $\mathbf{D}$ is a matrix of errors of the same dimension as $\mathbf{X}$. When an intercept is in the model, the first column of $\mathbf{D}$ would contain only 0's. The remaining elements of $\mathbf{D}$ could represent rounding errors or measurment errors.

Let $\mathbf{d}_i^t$ denote the row of $\mathbf{D}$ corresponding to the i-th case (so $\mathbf{d}_i$ is a $p \times 1$ vector). We assume that $\mathbf{d}_i$ and $\mathbf{d}_j$ are statistically independent for $i \neq j$. In addition, we assume that

$$\mathbb{E}[\mathbf{d}_i] = \mathbf{0}, \quad \text{Cov}(\mathbf{d}_i) = \mathbf{S} = \text{diag}(s_i^2)_{i=1}^p,$$

where $s_1^2 = 0$ (apparently, no measurement error for the intercept).

Suppose the problem of interest is to estimate $\boldsymbol{\beta}$ in the model

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

We would like to estimate $\boldsymbol{\beta}$ with $(\tilde{\mathbf{X}}^t\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^t\mathbf{y}$. However, we observe $\mathbf{X}$ instead of $\tilde{\mathbf{X}}$. Instead, we have

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{y}. \tag{1}$$

When the predictors are measured with error, the LS estimator $\hat{\boldsymbol{\beta}}$ obtained in (1) is no longer unbiased. The bias is given by (Hodges and Moore, 1972)

$$\mathbb{E}[\hat{\boldsymbol{\beta}}] - \boldsymbol{\beta} = -(n-p-1)(\tilde{\mathbf{X}}^t\tilde{\mathbf{X}})^{-1}\mathbf{S}\boldsymbol{\beta}.$$

With a given dataset, this can be approximated by

$$\mathbb{E}[\hat{\boldsymbol{\beta}}] - \boldsymbol{\beta} \approx -(n-p-1)(\tilde{\mathbf{X}}^t\tilde{\mathbf{X}})^{-1}\mathbf{S}\hat{\boldsymbol{\beta}}.$$

In the special case of estimating the slope in simple linear regression, the approximation is

$$\mathbb{E}[\hat{\beta}_1] \approx \beta_1\Big[1 - \frac{s_1^2}{\sum_{i=1}^n (x_i - \bar{x})^2/(n-2)}\Big].$$