

# Multivariate Linear Regression

---

Max Turgeon

STAT 4690—Applied Multivariate Analysis

# Multivariate Linear Regression model

- We are interested in the relationship between  $p$  outcomes  $Y_1, \dots, Y_p$  and  $q$  covariates  $X_1, \dots, X_q$ .
  - We will write  $\mathbf{Y} = (Y_1, \dots, Y_p)$  and  $\mathbf{X} = (1, X_1, \dots, X_q)$ .
- We will assume a **linear relationship**:
  - $E(\mathbf{Y} | \mathbf{X}) = B^T \mathbf{X}$ , where  $B$  is a  $(q + 1) \times p$  matrix of *regression coefficients*.
- We will also assume **homoscedasticity**:
  - $\text{Cov}(\mathbf{Y} | \mathbf{X}) = \Sigma$ , where  $\Sigma$  is positive-definite.
  - In other words, the (conditional) covariance of  $\mathbf{Y}$  does not depend on  $\mathbf{X}$ .

# Relationship with Univariate regression i

- Let  $\sigma_i^2$  be the  $i$ -th diagonal element of  $\Sigma$ .
- Let  $\beta_i$  be the  $i$ -th column of  $B$ .
- From the model above, we get  $p$  univariate regressions:
  - $E(Y_i | \mathbf{X}) = \mathbf{X}\beta_i$ ;
  - $\text{Var}(Y_i | \mathbf{X}) = \sigma_i^2$ .
- However, we will use the correlation between outcomes for hypothesis testing
- This follows from the assumption that each component  $Y_i$  is linearly associated with the *same* covariates  $\mathbf{X}$ .

## Relationship with Univariate regression ii

- If we assumed a different set of covariates  $\mathbf{X}_i$  for each outcome  $Y_i$  and still wanted to use the correlation between the outcomes, we would get the **Seemingly Unrelated Regressions** (SUR) model.
  - This model is sometimes used by econometricians.

# Least-Squares Estimation i

- Let  $\mathbf{Y}_1, \dots, \mathbf{Y}_n$  be a random sample of size  $n$ , and let  $\mathbf{X}_1, \dots, \mathbf{X}_n$  be the corresponding sample of covariates.
- We will write  $\mathbb{Y}$  and  $\mathbb{X}$  for the matrices whose  $i$ -th row is  $\mathbf{Y}_i$  and  $\mathbf{X}_i$ , respectively.
  - We can then write  $E(\mathbb{Y} \mid \mathbb{X}) = \mathbb{X}B$ .
- For Least-Squares Estimation, we will be looking for the estimator  $\hat{B}$  of  $B$  that minimises a least-squares criterion:
  - $LS(B) = \text{tr} \left[ (\mathbb{Y} - \mathbb{X}B)^T (\mathbb{Y} - \mathbb{X}B) \right]$
  - **Note:** This criterion is also known as the (squared) *Frobenius norm*; i.e.  $LS(B) = \|\mathbb{Y} - \mathbb{X}B\|_F^2$ .

## Least-Squares Estimation ii

- **Note 2:** If you expand the matrix product and look at the diagonal, you can see that the Frobenius norm is equivalent to the sum of the squared entries.
- To minimise  $LS(B)$ , we could use matrix derivatives...
- Or, we can expand the matrix product along the diagonal and compute the trace.
- Let  $\mathbf{Y}_{(j)}$  be the  $j$ -th column of  $\mathbf{Y}$ .

# Least-Squares Estimation iii

- In other words,  $\mathbf{Y}_{(j)} = (Y_{1j}, \dots, Y_{nj})$  contains the  $n$  values for the outcome  $Y_j$ . We then have

$$\begin{aligned} LS(B) &= \text{tr} \left[ (\mathbb{Y} - \mathbb{X}B)^T (\mathbb{Y} - \mathbb{X}B) \right] \\ &= \sum_{j=1}^p (\mathbf{Y}_{(j)} - \mathbb{X}\beta_j)^T (\mathbf{Y}_{(j)} - \mathbb{X}\beta_j) \\ &= \sum_{j=1}^p \sum_{i=1}^n (Y_{ij} - \beta_j^T \mathbf{x}_i)^2. \end{aligned}$$

## Least-Squares Estimation iv

- For each  $j$ , the sum  $\sum_{i=1}^n (Y_{ij} - \beta_j^T \mathbf{X}_i)^2$  is simply the least-squares criterion for the corresponding univariate linear regression.
- $\hat{\beta}_j = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \mathbf{Y}_{(j)}$
- But since  $LS(B)$  is a sum of  $p$  positive terms, each minimised at  $\hat{\beta}_j$ , the whole is sum is minimised at

$$\hat{B} = \begin{pmatrix} \hat{\beta}_1 & \cdots & \hat{\beta}_p \end{pmatrix}.$$

- Or put another way:

$$\hat{B} = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \mathbb{Y}.$$



- We still have not made any distributional assumptions on  $\mathbf{Y}$ .
  - We do not need to assume normality to derive the least-squares estimator.
- The least-squares estimator is *unbiased*:

$$\begin{aligned}E(\hat{B} \mid \mathbb{X}) &= (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X} E(\mathbf{Y} \mid \mathbb{X}) \\&= (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \mathbb{X} B \\&= B.\end{aligned}$$

## Comments ii

- We did not use the covariance matrix  $\Sigma$  anywhere in the estimation process. But note that:

$$\begin{aligned}\text{Cov}(\hat{\beta}_i, \hat{\beta}_j) &= \text{Cov}\left((\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \mathbf{Y}_{(i)}, (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \mathbf{Y}_{(j)}\right) \\ &= (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \text{Cov}\left(\mathbf{Y}_{(i)}, \mathbf{Y}_{(j)}\right) \left((\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T\right)^T \\ &= (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T (\sigma_{ij} I_n) \mathbb{X} (\mathbb{X}^T \mathbb{X})^{-1} \\ &= \sigma_{ij} (\mathbb{X}^T \mathbb{X})^{-1},\end{aligned}$$

where  $\sigma_{ij}$  is the  $(i, j)$ -th entry of  $\Sigma$ .

## Example i

```
# Let's revisit the plastic film data
```

```
library(heplots)
```

```
library(tidyverse)
```

```
Y <- Plastic %>%
```

```
  select(tear, gloss, opacity) %>%
```

```
  as.matrix
```

```
X <- model.matrix(~ rate, data = Plastic)
```

```
head(X)
```

## Example ii

```
##      (Intercept) rateHigh
## 1             1         0
## 2             1         0
## 3             1         0
## 4             1         0
## 5             1         0
## 6             1         0
```

```
(B_hat <- solve(crossprod(X)) %*% t(X) %*% Y)
```

## Example iii

```
##           tear gloss opacity
## (Intercept) 6.49  9.57    3.79
## rateHigh    0.59 -0.51    0.29
```

*# Compare with lm output*

```
fit <- lm(cbind(tear, gloss, opacity) ~ rate,
          data = Plastic)
coef(fit)
```

```
##           tear gloss opacity
## (Intercept) 6.49  9.57    3.79
## rateHigh    0.59 -0.51    0.29
```

# Geometry of LS i

- Let  $P = \mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T$ .
- $P$  is symmetric and *idempotent*:

$$P^2 = \mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T = \mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T = P.$$

- Let  $\hat{\mathbb{Y}} = \mathbb{X}\hat{B}$  be the fitted values, and  $\hat{\mathbb{E}} = \mathbb{Y} - \hat{\mathbb{Y}}$ , the residuals.
  - We have  $\hat{\mathbb{Y}} = P\mathbb{Y}$ .
  - We also have  $\hat{\mathbb{E}} = (I - P)\mathbb{Y}$ .

- Putting all this together, we get

$$\begin{aligned}\hat{\mathbf{Y}}^T \hat{\mathbf{E}} &= (P\mathbf{Y})^T (I - P)\mathbf{Y} \\ &= \mathbf{Y}^T P(I - P)\mathbf{Y} \\ &= \mathbf{Y}^T (P - P^2)\mathbf{Y} \\ &= 0.\end{aligned}$$

- In other words, the fitted values and the residuals are **orthogonal**.
- Similarly, we can see that  $\mathbf{X}^T \hat{\mathbf{E}} = 0$  and  $P\mathbf{X} = \mathbf{X}$ .

- **Interpretation:**  $\hat{\mathbb{Y}}$  is the orthogonal projection of  $\mathbb{Y}$  onto the column space of  $\mathbb{X}$ .



## Example (cont'd) i

```
Y_hat <- fitted(fit)
residuals <- residuals(fit)

crossprod(Y_hat, residuals)
```

##	tear	gloss	opacity
## tear	-9.489298e-16	2.959810e-15	-4.720135e-15
## gloss	-1.424461e-15	1.109357e-15	-1.150262e-14
## opacity	-7.268852e-16	1.211209e-15	1.648459e-16

## Example (cont'd) ii

```
crossprod(X, residuals)
```

##	tear	gloss	opacity
## (Intercept)	0 5.828671e-16	-4.440892e-16	
## rateHigh	0 1.387779e-16	4.440892e-16	

## Example (cont'd) iii

```
# Is this really zero?
isZero <- function(mat) {
  all.equal(mat, matrix(0, ncol = ncol(mat),
                        nrow = nrow(mat)),
            check.attributes = FALSE)
}

isZero(crossprod(Y_hat, residuals))

## [1] TRUE
```

## Example (cont'd) iv

```
isZero(crossprod(X, residuals))
```

```
## [1] TRUE
```

# Bootstrapped Confidence Intervals i

- We still have not made any assumption about the distribution of  $\mathbf{Y}$ , beyond the conditional mean and covariance function.
  - Let's see how much further we can go.
- We will use **bootstrap** to derive confidence intervals for our quantities of interest.
- Bootstrap is a resampling technique for estimating the sampling distribution of an estimator of interest.
  - Particularly useful when we think the usual assumptions may not hold, or when the sampling distribution would be difficult to derive.

## Bootstrapped Confidence Intervals ii

- Let's say we want to estimate the sampling distribution of the correlation coefficient.
- We have a sample of pairs  $(U_1, V_1), \dots, (U_n, V_n)$ , from which we estimated the correlation  $\hat{\rho}$ .
- The idea is to resample **with replacement** from our sample to mimic the process of “repeating the experiment”.

## Bootstrapped Confidence Intervals iii

- For each bootstrap sample  $(U_1^{(b)}, V_1^{(b)}), \dots, (U_n^{(b)}, V_n^{(b)})$ , we compute the sample correlation  $\hat{\rho}^{(b)}$ .
- We now have a whole sample of *correlation coefficients*  $\hat{\rho}^{(1)}, \dots, \hat{\rho}^{(B)}$ .
- From its quantiles, we can derive a confidence interval for  $\hat{\rho}$ .

## Example i

```
library(candisc)

dataset <- HSB[,c("math", "sci")]

(corr_est <- cor(dataset)[1,2])

## [1] 0.6495261
```



## Example ii

```
# Choose a number of bootstrap samples
B <- 5000
corr_boot <- replicate(B, {
  samp_boot <- sample(nrow(dataset),
                      replace = TRUE)
  dataset_boot <- dataset[samp_boot,]
  cor(dataset_boot)[1,2]
})

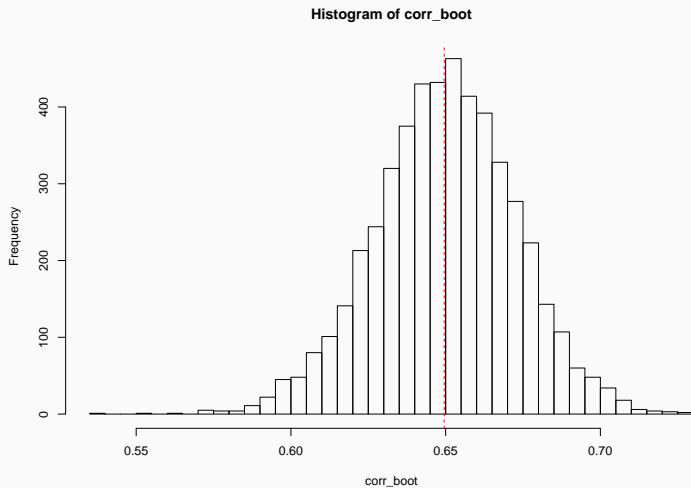
quantile(corr_boot,
         probs = c(0.025, 0.975))
```

## Example iii

```
##          2.5%      97.5%  
## 0.6029038 0.6936202
```

```
hist(corr_boot, breaks = 50)  
abline(v = corr_est, col = 'red',  
       lty = 2)
```

## Example iv



## Bootstrapped Confidence Intervals (cont'd) i

- Going back to our multivariate linear regression setting, we can bootstrap our estimate of the matrix of regression coefficients!
- We will sample with replacement the rows of  $\mathbb{Y}$  and  $\mathbb{X}$ 
  - It's important to sample the **same** rows in both matrices. We want to keep the relationship between  $\mathbf{Y}$  and  $\mathbf{X}$  intact.
- For each bootstrap sample, we can compute the estimate  $\hat{B}^{(b)}$ .
- From these samples, we can compute confidence intervals for each entry in  $B$ .

## Bootstrapped Confidence Intervals (cont'd) ii

- We can also technically compute confidence regions for multiple entries in  $B$ 
  - E.g. a whole column or a whole row
  - But multivariate quantiles are tricky...

## Example (cont'd) i

```
B_boot <- replicate(B, {  
  samp_boot <- sample(nrow(Y),  
                      replace = TRUE)  
  X_boot <- X[samp_boot,]  
  Y_boot <- Y[samp_boot,]  
  
  solve(crossprod(X_boot)) %*% t(X_boot) %*% Y_boot  
})  
  
# The output is a 3-dim array  
dim(B_boot)
```

## Example (cont'd) ii

```
## [1]      2      3 5000
```

```
B_boot[, ,1]
```

```
##                tear gloss opacity  
## (Intercept) 6.55  9.56    3.35  
## rateHigh    0.47 -0.34    0.47
```

```
# CI for effect of rate on tear
```

```
quantile(B_boot["rateHigh", "tear", ],  
         probs = c(0.025, 0.975))
```

## Example (cont'd) iii

```
##          2.5%      97.5%  
## 0.265636 0.900000
```

```
# CI for effect of rate on gloss  
quantile(B_boot["rateHigh", "gloss", ],  
         probs = c(0.025, 0.975))
```

```
##          2.5%      97.5%  
## -0.9022470 -0.1124375
```



## Example (cont'd) iv

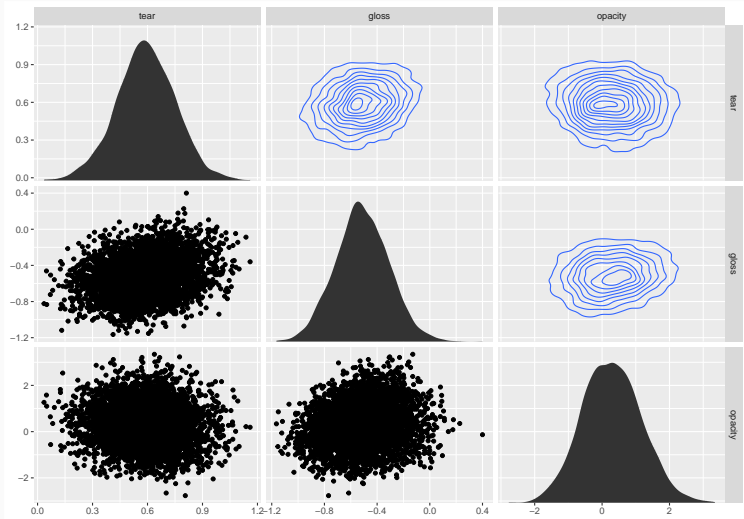
```
# CI for effect of rate on opacity  
quantile(B_boot["rateHigh", "opacity", ],  
         probs = c(0.025, 0.975))
```

```
##          2.5%      97.5%  
## -1.389077  2.044107
```

## Example (cont'd) v

```
library(ggforce)

B_boot["rateHigh",,] %>%
  t() %>%
  as.data.frame() %>%
  ggplot(aes(x = .panel_x, y = .panel_y)) +
  geom_point() +
  geom_autodensity() +
  geom_density2d() +
  facet_matrix(vars(everything()),
               layer.diag = 2,
               layer.upper = 3)
```



*# There is some correlation, but not much*

```
B_boot["rateHigh",,] %>%
```

```
  t() %>%
```

```
  cor()
```

```
##           tear      gloss    opacity
## tear      1.0000000 0.2340506 -0.0760627
## gloss     0.2340506 1.0000000  0.1572758
## opacity  -0.0760627 0.1572758  1.0000000
```

# Maximum Likelihood Estimation i

- We now introduce distributional assumptions on  $\mathbf{Y}$ :

$$\mathbf{Y} \mid \mathbf{X} \sim N_p(B^T \mathbf{X}, \Sigma).$$

- This is the same conditions on the mean and covariance as above. The only difference is that we now assume the residuals are normally distributed.
- **Note:** The distribution above is conditional on  $\mathbf{X}$ . It could happen that the marginal distribution of  $\mathbf{Y}$  is not normal.

# Maximum Likelihood Estimation ii

- **Theorem:** Suppose  $\mathbb{X}$  has full rank  $q + 1$ , and assume that  $n \geq q + p + 1$ . Then the least-squares estimator  $\hat{B} = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \mathbb{Y}$  of  $B$  is also the *maximum likelihood estimator*. Moreover, we have
  1.  $\hat{B}$  is normally distributed.
  2. The maximum likelihood estimator for  $\Sigma$  is  $\hat{\Sigma} = \frac{1}{n} \hat{\mathbb{E}}^T \hat{\mathbb{E}}$ .
  3.  $n\hat{\Sigma}$  follows a Wishart distribution  $W_{n-q-1}(\Sigma)$  on  $n - q - 1$  degrees of freedom.
  4. The maximised likelihood is
$$L(\hat{B}, \hat{\Sigma}) = (2\pi)^{-np/2} |\hat{\Sigma}|^{-n/2} \exp(-pn/2).$$

- **Note:** Looking at the degrees of freedom of the Wishart distribution, we can infer that  $\hat{\Sigma}$  is a biased estimator of  $\Sigma$ . An *unbiased* estimator is

$$S = \frac{1}{n - q - 1} \hat{\mathbb{E}}^T \hat{\mathbb{E}}.$$

# Confidence and Prediction Regions i

- Suppose we have a new observation  $\mathbf{X}_0$ . We are interested in making predictions and inference about the corresponding outcome vector  $\mathbf{Y}_0$ .
- First, since  $\hat{B}$  is an unbiased estimator of  $B$ , we see that

$$E(\mathbf{X}_0^T \hat{B}) = \mathbf{X}_0^T E(\hat{B}) = \mathbf{X}_0^T B = E(\mathbf{Y}_0).$$

Therefore, it makes sense to estimate  $\mathbf{Y}_0$  using  $\mathbf{X}_0^T \hat{B}$ .



## Confidence and Prediction Regions ii

- *What is the estimation error?* Let's look at the covariance of  $\mathbf{X}_0^T \hat{\beta}_i$  and  $\mathbf{X}_0^T \hat{\beta}_j$

$$\begin{aligned}\text{Cov}(\mathbf{X}_0^T \hat{\beta}_i, \mathbf{X}_0^T \hat{\beta}_j) &= \mathbf{X}_0^T \text{Cov}(\hat{\beta}_i, \hat{\beta}_j) \mathbf{X}_0 \\ &= \sigma_{ij} \mathbf{X}_0^T (\mathbb{X}^T \mathbb{X})^{-1} \mathbf{X}_0.\end{aligned}$$

- *What is the forecasting error?* In that case, we also need to take into account the extra variation coming from the residuals.
- In other words, we also need to sample a new “error” term  $\mathbf{E}_0 = (E_{01}, \dots, E_{0p})$  independently of  $\mathbf{X}_0$ .

## Confidence and Prediction Regions iii

- Let  $\tilde{\mathbf{Y}}_0 = \mathbf{X}_0^T B + \mathbf{E}_0$  be the new value.
- The **forecast error** is given by

$$\tilde{\mathbf{Y}}_0 - \mathbf{X}_0^T \hat{B} = \mathbf{E}_0 - \mathbf{X}_0^T (\hat{B} - B).$$

- Since  $E(\tilde{\mathbf{Y}}_0 - \mathbf{X}_0^T \hat{B}) = 0$ , we can still deduce that  $\mathbf{X}_0^T \hat{B}$  is an unbiased predictor of  $\mathbf{Y}_0$ .

## Confidence and Prediction Regions iv

- Now let's look at the covariance of the forecast errors in each component:

$$\begin{aligned} & E \left[ \left( \tilde{Y}_{0i} - \mathbf{X}_0^T \hat{\beta}_i \right) \left( \tilde{Y}_{0j} - \mathbf{X}_0^T \hat{\beta}_j \right) \right] \\ &= E \left[ \left( E_{0i} - \mathbf{X}_0^T (\hat{\beta}_i - \beta_i) \right) \left( E_{0j} - \mathbf{X}_0^T (\hat{\beta}_j - \beta_j) \right) \right] \\ &= E(E_{0i} E_{0j}) + \mathbf{X}_0^T E \left[ (\hat{\beta}_i - \beta_i)(\hat{\beta}_j - \beta_j) \right] \mathbf{X}_0 \\ &= \sigma_{ij} + \sigma_{ij} \mathbf{X}_0^T (\mathbb{X}^T \mathbb{X})^{-1} \mathbf{X}_0 \\ &= \sigma_{ij} \left( 1 + \mathbf{X}_0^T (\mathbb{X}^T \mathbb{X})^{-1} \mathbf{X}_0 \right). \end{aligned}$$

- Therefore, we can see that the difference between the *estimation* error and the *forecasting* error is  $\sigma_{ij}$ .

## Example i

```
# Recall our model
```

```
fit <- lm(cbind(tear, gloss, opacity) ~ rate,  
         data = Plastic)
```

```
new_x <- data.frame(rate = factor("High",  
                                  levels = c("Low", "H  
(prediction <- predict(fit, newdata = new_x))
```

```
##   tear gloss opacity  
## 1 7.08  9.06    4.08
```

## Example ii

```
X <- model.matrix(fit)
S <- crossprod(resid(fit))/(nrow(Plastic) - ncol(X))
new_x <- model.matrix(~rate, new_x)

quad_form <- drop(new_x %*% solve(crossprod(X)) %*% t(

# Estimation covariance
(est_cov <- S * quad_form)
```

## Example iii

```
##                tear        gloss        opacity
## tear          0.014027778 0.003994444 -0.006083333
## gloss          0.003994444 0.021027778  0.014716667
## opacity -0.006083333 0.014716667  0.409916667
```

*# Forecasting covariance*

```
(fct_cov <- S *(1 + quad_form))
```

```
##                tear        gloss        opacity
## tear          0.15430556 0.04393889 -0.06691667
## gloss          0.04393889 0.23130556  0.16188333
## opacity -0.06691667 0.16188333  4.50908333
```

## Example iv

```
# Estimation CIs
```

```
cbind(drop(prediction) - 1.96*diag(est_cov),  
      drop(prediction) + 1.96*diag(est_cov))
```

```
##           [,1]      [,2]  
## tear      7.052506 7.107494  
## gloss     9.018786 9.101214  
## opacity   3.276563 4.883437
```

## Example v

*# Forecasting CIs*

```
cbind(drop(prediction) - 1.96*diag(fct_cov),  
      drop(prediction) + 1.96*diag(fct_cov))
```

```
##           [,1]      [,2]  
## tear      6.777561  7.382439  
## gloss     8.606641  9.513359  
## opacity  -4.757803 12.917803
```



# Likelihood Ratio Tests i

- We can use a Likelihood Ratio test to assess the evidence in support of two nested models.
- Write

$$B = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}, \quad \mathbb{X} = \begin{pmatrix} \mathbb{X}_1 & \mathbb{X}_2 \end{pmatrix},$$

where  $B_1$  is  $(r + 1) \times p$ ,  $B_2$  is  $(q - r) \times p$ ,  $\mathbb{X}_1$  is  $n \times (r + 1)$ ,  $\mathbb{X}_2$  is  $n \times (q - r)$ , and  $r \geq 0$  is a non-negative integer.

## Likelihood Ratio Tests ii

- We want to compare the following models:

$$\text{Full model : } E(\mathbf{Y} \mid \mathbf{X}) = B^T \mathbf{X}$$

$$\text{Nested model : } E(\mathbf{Y} \mid \mathbf{X}_1) = B_1^T \mathbf{X}_1$$

- According to our previous theorem, the corresponding maximised likelihoods are

## Likelihood Ratio Tests iii

Full model :  $L(\hat{B}, \hat{\Sigma}) = (2\pi)^{-np/2} |\hat{\Sigma}|^{-n/2} \exp(-pn/2)$

Nested model :  $L(\hat{B}_1, \hat{\Sigma}_1) = (2\pi)^{-np/2} |\hat{\Sigma}_1|^{-n/2} \exp(-pn/2)$

- Therefore, taking the ratio of the likelihoods of the nested model to the full model, we get

$$\Lambda = \frac{L(\hat{B}_1, \hat{\Sigma}_1)}{L(\hat{B}, \hat{\Sigma})} = \left( \frac{|\hat{\Sigma}|}{|\hat{\Sigma}_1|} \right)^{n/2}.$$

# Likelihood Ratio Tests iv

- Or equivalently, we get *Wilks' lambda statistic*:

$$\Lambda^{2/n} = \frac{|\hat{\Sigma}|}{|\hat{\Sigma}_1|}.$$

- As discussed in the lecture on MANOVA, there is no closed-form solution for the distribution of this statistic under the null hypothesis  $H_0 : B_2 = 0$ , but there are many approximations.
- **Two important special cases:**
  - When  $r = 0$ , we are testing the full model against the empty model (i.e. only the intercept).

# Likelihood Ratio Tests v

- When  $\mathbb{X}_2$  only contains one covariate, we are testing the full model against a simpler model without that covariate. In other words, we are testing for the *significance* of that covariate.

## Other Multivariate Test Statistics i

- The Wilks' lambda statistic can actually be expressed in terms of the (generalized) eigenvalues of a pair of matrices  $(H, E)$ :
  - $E = n\hat{\Sigma}$  is the **error** matrix.
  - $H = n(\hat{\Sigma}_1 - \hat{\Sigma})$  is the **hypothesis** matrix.
- Under our assumptions about the rank of  $\mathbb{X}$  and the sample size,  $E$  is (almost surely) invertible, and therefore we can look at the nonzero eigenvalues of  $HE^{-1}$ :
  - Let  $\eta_1 \geq \dots \geq \eta_s$  be those nonzero eigenvalues, where  $s = \min(p, q - r)$ .

## Other Multivariate Test Statistics ii

- Equivalently, these eigenvalues are the nonzero roots of the determinantal equation  $\det \left( (\hat{\Sigma}_1 - \hat{\Sigma}) - \eta \hat{\Sigma} \right) = 0$ .
- The four classical multivariate test statistics are:

$$\text{Wilks' lambda : } \prod_{i=1}^s \frac{1}{1 + \eta_i} = \frac{|E|}{|E + H|}$$

$$\text{Pillai's trace : } \sum_{i=1}^s \frac{\eta_i}{1 + \eta_i} = \text{tr} \left( H(H + E)^{-1} \right)$$

$$\text{Hotelling-Lawley trace : } \sum_{i=1}^s \eta_i = \text{tr} \left( H E^{-1} \right)$$

$$\text{Roy's largest root : } \frac{\eta_1}{1 + \eta_1}$$

## Other Multivariate Test Statistics iii

- Under the null hypothesis  $H_0 : B_2 = 0$ , all four statistics can be well-approximated using the  $F$  distribution.
- **Note:** When  $r = q - 1$ , all four tests are equivalent.
- In general, as the sample size increases, all four tests give similar results. For finite sample size, Roy's largest root has good power only if there the leading eigenvalue  $\eta_1$  is significantly larger than the other ones.



## Example i

```
# Going back to our example  
full_model <- lm(cbind(tear, gloss,  
                      opacity) ~ rate*additive,  
                data = Plastic)  
  
anova(full_model, test = "Wilks") %>%  
  broom::tidy() %>%  
  knitr::kable(digits = 3)
```

## Example ii

term	df	Wilks	approx.F	num.Df	den.Df	p.value
(Intercept)	1	0.001	5950.906	3	14	0.000
rate	1	0.382	7.554	3	14	0.003
additive	1	0.523	4.256	3	14	0.025
rate:additive	1	0.777	1.339	3	14	0.302
Residuals	16	-	-	-	-	-

## Example iii

```
anova(full_model, test = "Roy") %>%  
  broom::tidy() %>%  
  knitr::kable(digits = 3)
```

term	df	Roy	approx.F	num.Df	den.Df	p.value
(Intercept)	1	1275.194	5950.906	3	14	0.000
rate	1	1.619	7.554	3	14	0.003
additive	1	0.912	4.256	3	14	0.025
rate:additive	1	0.287	1.339	3	14	0.302
Residuals	16	-	-	-	-	-

## Example iv

```
# Fit a model with only rate  
rate_model <- lm(cbind(tear, gloss,  
                      opacity) ~ rate,  
                data = Plastic)  
  
# Removing the dfs from approx  
anova(full_model, rate_model,  
       test = "Wilks") %>%  
  broom::tidy() %>%  
  dplyr::select(-num.Df, -den.Df) %>%  
  knitr::kable(digits = 3)
```

## Example v

res.df	df	Gen.var.	Wilks	approx.F	p.value
16	-	0.407	-	-	-
18	2	0.479	0.43	2.447	0.05

## Example vi

```
anova(full_model, rate_model,  
      test = "Roy") %>%  
broom::tidy() %>%  
dplyr::select(-num.Df, -den.Df) %>%  
knitr::kable(digits = 3)
```

res.df	df	Gen.var.	Roy	approx.F	p.value
16	-	0.407	-	-	-
18	2	0.479	1.084	5.418	0.01

## Example vii

```
# Let's look at the eigenvalues
E <- crossprod(residuals(full_model))
H <- crossprod(residuals(rate_model)) - E

result <- eigen(H %*% solve(E),
                only.values = TRUE)
result$values[seq_len(2)]

## [1] 1.083657 0.115087
```

# Information Criteria i

- We can use hypothesis testing for model building:
  - Add covariates that significantly improve the model (*forward selection*);
  - Remove non-significant covariates (*backward elimination*).
- Another approach is to use *Information Criteria*.
- The general form of Akaike's information criterion:

$$-2 \log L(\hat{B}, \hat{\Sigma}) + 2d,$$

where  $d$  is the number of parameters to estimate.



## Information Criteria ii

- In multivariate regression, this would be
$$d = (q + 1)p + p(p + 1)/2.$$
- Therefore, we get (up to a constant):

$$AIC = n \log |\hat{\Sigma}| + 2(q + 1)p + p(p + 1).$$

- The intuition behind AIC is that it estimates the Kullback-Leibler divergence between the posited model and the true data-generating mechanism.
  - So smaller is better.
- Model selection using information criteria proceeds as follows:

## Information Criteria iii

1. Select models of interest  $\{M_1, \dots, M_K\}$ . They do not need to be nested, and they do not need to involve the same variables.
  2. Compute the AIC for each model.
  3. Select the model with the smallest AIC.
- The set of interesting models should be selected using domain-specific knowledge when possible.
    - If it is not feasible, you can look at all possible models between the empty model and the full model.
  - There are many variants of AIC, each with their own trade-offs.
    - For more details, see Timm (2002) Section 4.2.d.

## Example (cont'd) i

```
## AIC(full_model)
# Error in logLik.lm(full_model) :
# 'logLik.lm' does not support multiple responses
class(full_model)

## [1] "mlm" "lm"
```

## Example (cont'd) ii

```
logLik.mlm <- function(object, ...) {  
  resids <- residuals(object)  
  Sigma_ML <- crossprod(resids)/nrow(resids)  
  ans <- sum(mvtnorm::dmvnorm(resids,  
                                sigma = Sigma_ML,  
                                log = TRUE))  
  
  df <- prod(dim(coef(object))) +  
    choose(ncol(Sigma_ML) + 1, 2)  
  attr(ans, c("nobs", "df")) <- c(nrow(resids), df)  
  class(ans) <- "logLik"  
  return(ans)  
}
```

```
logLik(full_model)
```

```
## 'log Lik.' -51.45783 (df=NULL)
```

```
AIC(full_model)
```

```
## numeric(0)
```

```
AIC(rate_model)
```

```
## numeric(0)
```

# Multivariate Influence Measures i

- Earlier we introduced the projection matrix

$$P = \mathbb{X}(\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T$$

and we noted that  $\hat{\mathbf{Y}} = P\mathbf{Y}$ .

- Looking at one row at a time, we can see that

$$\begin{aligned} \mathbf{Y}_i &= \sum_{j=1}^n P_{ij} \mathbf{Y}_j \\ &= P_{ii} \mathbf{Y}_i + \sum_{j \neq i} P_{ij} \mathbf{Y}_j, \end{aligned}$$

where  $P_{ij}$  is the  $(i, j)$ -th entry of  $P$ .

## Multivariate Influence Measures ii

- In other words, the diagonal element  $P_{ii}$  represents the *leverage* (or influence) of observation  $\mathbf{Y}_i$  on the fitted value  $\hat{\mathbf{Y}}_i$ .
  - Observation  $\mathbf{Y}_i$  is said to have a **high leverage** if  $P_{ii}$  is large compared to the other element on the diagonal.
- Let  $S = \frac{1}{n-q-1} \hat{\mathbf{E}}^T \hat{\mathbf{E}}$  be the unbiased estimator of  $\Sigma$ , and let  $\hat{\mathbf{E}}_i$  be the  $i$ -th row of  $\hat{\mathbf{E}}$ .
- We define the multivariate **internally Studentized residuals** as follows:

$$r_i = \frac{\hat{\mathbf{E}}_i^T S^{-1} \hat{\mathbf{E}}_i}{1 - P_{ii}}.$$

## Multivariate Influence Measures iii

- If we let  $S_{(i)}$  be the estimator of  $\Sigma$  where we have removed row  $i$  from the residual matrix  $\hat{\mathbf{E}}$ , we define the multivariate **externally Studentized residuals** as follows:

$$T_i^2 = \frac{\hat{\mathbf{E}}_i^T S_{(i)}^{-1} \hat{\mathbf{E}}_i}{1 - P_{ii}}.$$

- An observation  $\mathbf{Y}_i$  may be considered a potential outlier if

$$\left( \frac{n - q - p - 1}{p(n - q - 2)} \right) T_i^2 > F_\alpha(p, n - q - 2).$$



- Yet another measure of influence is the multivariate **Cook's distance**.

$$C_i = \frac{P_{ii}}{(1 - P_{ii})^2} \hat{\mathbf{E}}_i^T S^{-1} \hat{\mathbf{E}}_i / (q + 1).$$

- An observation  $\mathbf{Y}_i$  may be considered a potential outlier if  $C_i$  is larger than the median of a chi square distribution with  $\nu = p(n - q - 1)$  degrees of freedom.

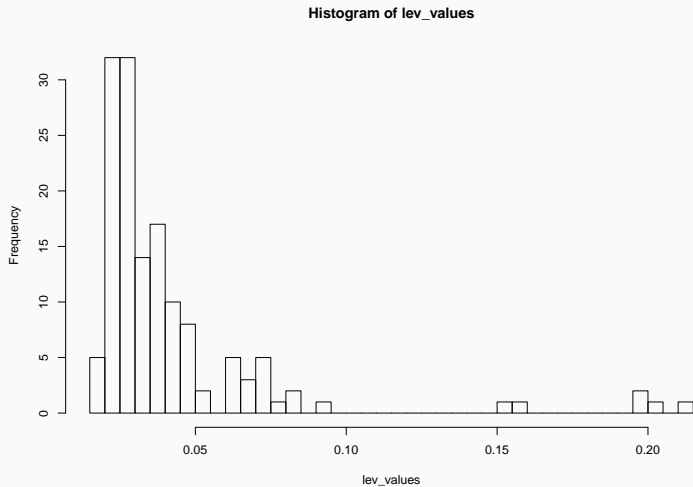
## Example i

```
library(openintro)
model <- lm(cbind(startPr, totalPr) ~
            nBids + cond + sellerRate +
            wheels + stockPhoto,
            data = marioKart)

X <- model.matrix(model)
P <- X %*% solve(crossprod(X)) %*% t(X)
lev_values <- diag(P)

hist(lev_values, 50)
```

## Example ii



## Example iii

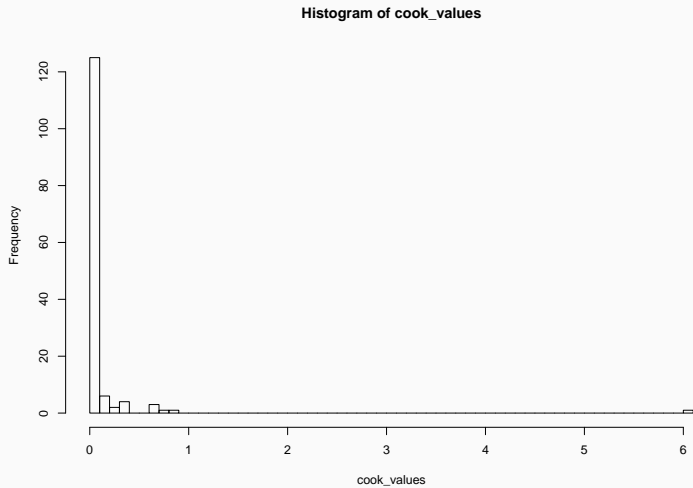
```
n <- nrow(HSB)
resids <- residuals(model)
S <- crossprod(resids)/(n - ncol(X))

S_inv <- solve(S)

const <- lev_values/((1 - lev_values)^2*ncol(X))
cook_values <- const * diag(resids %*% S_inv
                             %*% t(resids))

hist(cook_values, 50)
```

# Example iv



## Example v

```
# Cut-off value
```

```
(cutoff <- qchisq(0.5, ncol(S)*(n - ncol(X))))
```

```
## [1] 1187.333
```

```
which(cook_values > cutoff)
```

```
## named integer(0)
```

# Strategy for Multivariate Model Building

1. Try to identify outliers.
  - This should be done graphically at first.
  - Once the model is fitted, you can also look at influence measures.
2. Perform a multivariate test of hypothesis.
3. If there is evidence of a multivariate difference, calculate Bonferroni confidence intervals and investigate component-wise differences.
  - The projection of the confidence region onto each variable generally leads to confidence intervals that are too large.