III. Regression in Matrix Form and MVN

a. Multivariate Regression in Matrix Form
b. Gauss-Markov Theorem
c. MVN distribution
d. Linear combinations and Quadratic forms in MVN
e. Sampling distribution of least squares with normal errors
f. Prediction in Multiple Regression
g. General Linear Hypothesis

Appendix A. Proof of Gauss-Markov Thm
Appendix B. Joint and Conditional Distributions
Appendix C. Quadratic Forms in Normal Random Vectors

a. Multiple Regression in Matrix Form

$$\boxed{Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \cdots + \beta_k X_{i,k-1} + \varepsilon_i}$$

One way of writing this is in terms of vectors

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \beta_0 + \begin{bmatrix} X_{1,1} \\ X_{2,1} \\ \vdots \\ X_{N,1} \end{bmatrix} \beta_1 + \ldots + \begin{bmatrix} X_{1,k-1} \\ X_{2,k-1} \\ \vdots \\ X_{N,k-1} \end{bmatrix} \beta_{k-1} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{bmatrix}$$

$$y = X\beta + \varepsilon \qquad \underset{N \times k}{X} = \begin{bmatrix} \iota & X_1 & \cdots & X_{k-1} \end{bmatrix}$$

"iota"- a vector of ones

a. Multiple Regression in Matrix Form

Least Squares can be thought of picking the beta vector to make X orthogonal to the residual

$$\underset{k\times N}{X'}\left(y - Xb\right) = X'e = 0$$

Note: this is a set of k linear equations in b. ‘ means transpose.

$$X'y - X'Xb = 0$$

$$X'Xb = X'y$$

$$b = \left(X'X\right)^{-1} X'y$$

A matrix A (square) has an inverse if there is a unique solution to Ax=b

a. Multiple Regression in Matrix Form

Let's take a look at some of these quantities:

$$\underset{k\times N}{X'}\,\underset{N\times k}{X} = \begin{bmatrix} \iota' \\ x_1' \\ \vdots \\ x_{k-1}' \end{bmatrix} \begin{bmatrix} \iota & x_1 & \cdots & x_{k-1} \end{bmatrix}$$

$$= \begin{bmatrix} N & \iota' x_1 & \cdots & \iota' x_{k-1} \\ x_1'\iota & x_1' x_1 & & x_1' x_{k-1} \\ \vdots & & \ddots & \vdots \\ x_{k-1}'\iota & x_{k-1}' x_1 & \cdots & x_{k-1}' x_{k-1} \end{bmatrix}$$

So X'X has all of the info necessary to compute means, variances and covariances.

Same is true for X'y

a. Multiple Regression in Matrix Form

The least squares vector satisfies this equation:

$$X'\left(y - Xb\right) = 0$$

$$X'e = 0$$

$$\begin{bmatrix} \iota' \\ x_1' \\ \vdots \\ x_{k-1}' \end{bmatrix} e = \begin{bmatrix} \iota'e \\ x_1'e \\ \vdots \\ x_{k-1}'e \end{bmatrix}$$

The first element of this vector is the requirement that the sum of the residuals should be zero. The others that the residuals are uncorrelated with each x var.

a. Multiple Regression in Matrix Form

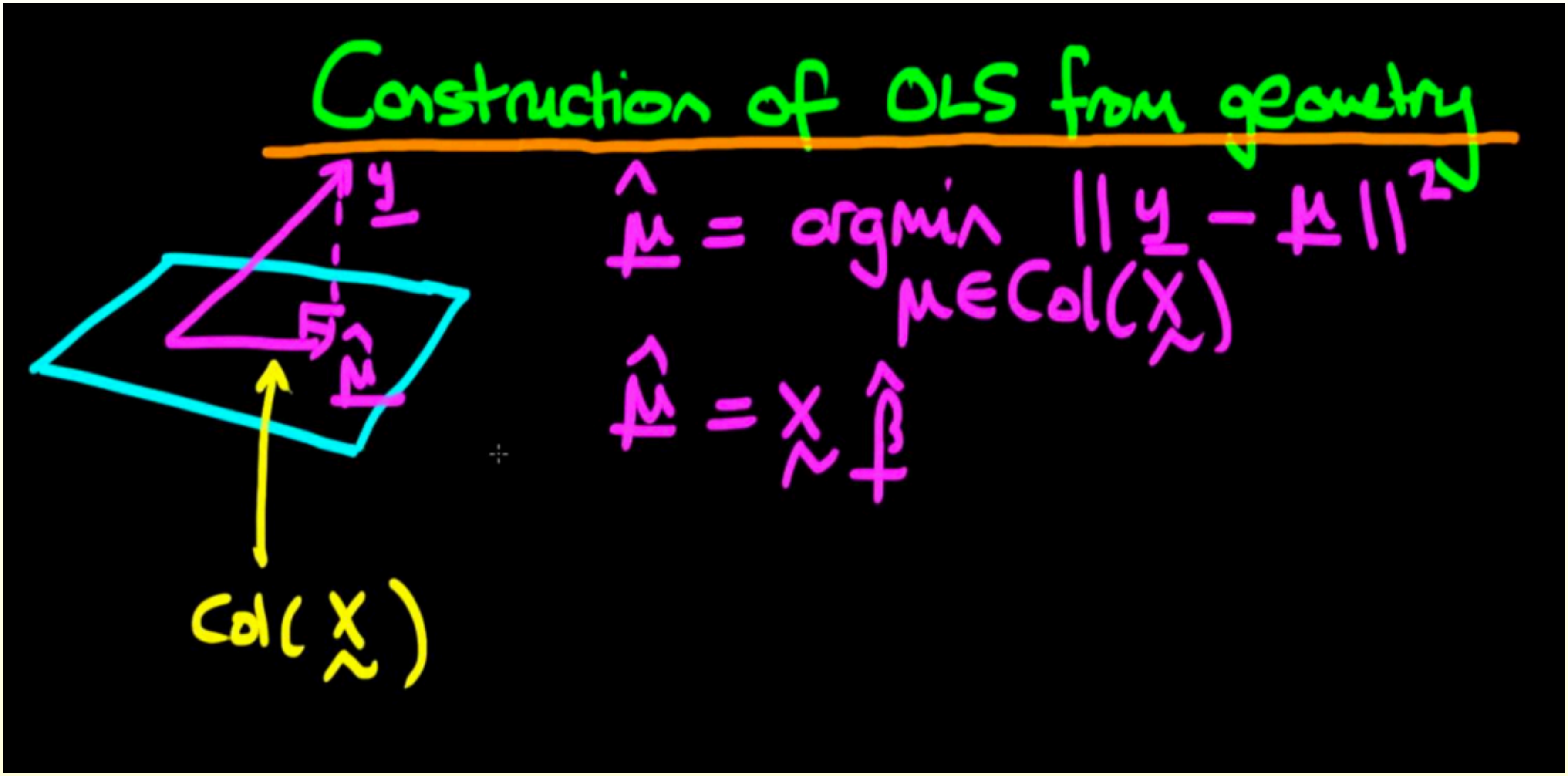
We have just seen that the least squares residual vector is uncorrelated with any of the “X” variables. This means that the fitted values are uncorrelated or orthogonal to the residuals

$$\hat{y} = X\left(X'X\right)^{-1} X'y$$

$$y = \hat{y} + e$$

$$e'\hat{y} = e'X\left(X'X\right)^{-1} X'y = 0$$

Thus, the regression is a projection of the vector y on the space spanned by the columns of X.


Construction of OLS from geometry
$\underset{\sim}{y}$
$\hat{\underset{\sim}{\mu}}$
Col($\underset{\sim}{X}$)
$\hat{\underset{\sim}{\mu}} = \underset{\mu \in \text{Col}(\underset{\sim}{X})}{\text{argmin}} \|\underset{\sim}{y} - \underset{\sim}{\mu}\|^2$
$\hat{\underset{\sim}{\mu}} = \underset{\sim}{X}\hat{\underset{\sim}{\beta}}$

a. Multiple Regression in Matrix Form

In R, we don't directly invert any matrix. We use the Cholesky root of X'X or the QR decomposition of X.

A Cholesky root of a positive definite matrix is the generalization of a square root.

$$A = U'U$$

where U is an upper triangular matrix. To invert a matrix, we find the Cholesky root of the matrix and then invert the root.

$$A^{-1} = \left(U'U\right)^{-1} = U^{-1}\left(U^{-1}\right)'$$

a. Multiple Regression in Matrix Form

Let's do it in R. We will need following R functions:

```
%*%                << matrix multiplication
chol()             << compute Cholesky root
chol2inv(chol(A))  << invert a pos definite matrix
crossprod(A,B)     << compute A'B
```

```
> y=Van_risk$VWNFX
> X=cbind(rep(1,length(y)),Van_risk$RmRf,Van_risk$SMB,Van_risk$HML)
> # remove missing from X and y
> X=X[-which(is.na(y)),]
> y=y[-which(is.na(y))]
> b=chol2inv(chol(crossprod(X)))%*%crossprod(X,y)
> b
             [,1]
[1,]  0.002654552
[2,]  0.959048993
[3,] -0.193117789
[4,]  0.425266190
> outml$coef
 (Intercept)        RmRf         SMB         HML 
 0.002654552  0.959048993 -0.193117789  0.425266190 
```

a. Multiple Regression in Matrix Form

Recall the definition of an inner and outer product of two vectors.

Inner product:

$$w'z = z'w = \sum_{i=1}^{N} w_i z_i$$

Outer product:

$$wz' = \begin{bmatrix} w_1 z_1 & w_1 z_2 & \cdots & w_1 z_N \\ w_2 z_1 & w_2 z_2 & \cdots & w_2 z_N \\ \vdots & \vdots & \ddots & \vdots \\ w_N z_1 & w_N z_2 & \cdots & w_N z_N \end{bmatrix}$$

a. Multiple Regression in Matrix Form

A random vector, X, is a vector of random variables.

$$E\left[x\right] = \mu = \begin{bmatrix} E\left[x_1\right] \\ E\left[x_2\right] \\ \vdots \\ E\left[x_N\right] \end{bmatrix}$$

$$\mathrm{Var}\left(x\right) = E\left[\left(x-\mu\right)\left(x-\mu\right)'\right]$$

Variance-Covariance Matrix

$$= \begin{bmatrix} E\left[\left(x_1-\mu_1\right)^2\right] & \cdots & E\left[\left(x_1-\mu_1\right)\left(x_N-\mu_N\right)\right] \\ \vdots & \ddots & \vdots \\ E\left[\left(x_N-\mu_N\right)\left(x_1-\mu_1\right)\right] & \cdots & E\left[\left(x_N-\mu_N\right)^2\right] \end{bmatrix}$$

a. Multiple Regression in Matrix Form

Find mean and variance-covariance matrix of b.

$$b-\beta=\left(X'X\right)^{-1}X'y-\beta=\left(X'X\right)^{-1}X'\left(X\beta+\varepsilon\right)-\beta$$

$$=\left(X'X\right)^{-1}X'\varepsilon$$

$$\Rightarrow E\left[b\right]=\beta$$

$$\bar{E}\left(X'X\right)^{-1}X'\varepsilon$$

① X fixed

② $\underline{X \perp \varepsilon}$

$$Var\left(b\right)=E\left[\left(b-\beta\right)\left(b-\beta\right)'\right]$$

$$=E\left[\left(\left(X'X\right)^{-1}X'\varepsilon\right)\left(\left(X'X\right)^{-1}X'\varepsilon\right)'\right]$$

$$=\left(X'X\right)^{-1}X'E\left[\varepsilon\varepsilon'\right]X\left(X'X\right)^{-1}$$

$$=\sigma^2\left(X'X\right)^{-1}$$

Note: $(AB)'=B'A'$

and

$Var\left(\varepsilon\right)=E\left[\varepsilon\varepsilon'\right]=\sigma^2 I_n$

a. Multiple Regression in Matrix Form

Let's compute it in R:

One new R function, `diag(A)` << get diagonal of A

```
> e=y-X%*%b
> ssq=sum(e*e)/(length(y)-ncol(X))
> Var_b=ssq*chol2inv(chol(crossprod(X)))
> std_err=sqrt(diag(Var_b))
> names(std_err)=c("intercept","RmRf","SMB","HML")
> std_err
   intercept         RmRf          SMB          HML
0.0006604903 0.0150102001 0.0216915778 0.0231962198
```

b. Gauss-Markov Theorem

Why Least Squares?

- It’s a projection (I like ‘em)
- It’s unbiased (not much)

Is Least Squares the “best” estimator among all unbiased estimators?

How should we score an unbiased estimator? Why not by its “variance”?

$$\mathrm{Var}\left(\tilde{b}\right) = E\left[\left(\tilde{b} - b\right)\left(\tilde{b} - b\right)'\right]$$

The Gauss-Markov Theorem says that least squares is the “best” linear unbiased estimator.

b. Gauss-Markov Theorem

This is cool, but is it persuasive?

Why restrict to unbiased estimators?

We might want to score estimators based on Mean Squared Error (MSE) rather than variance. In other words, I like estimators that put a lot of mass near the true value.

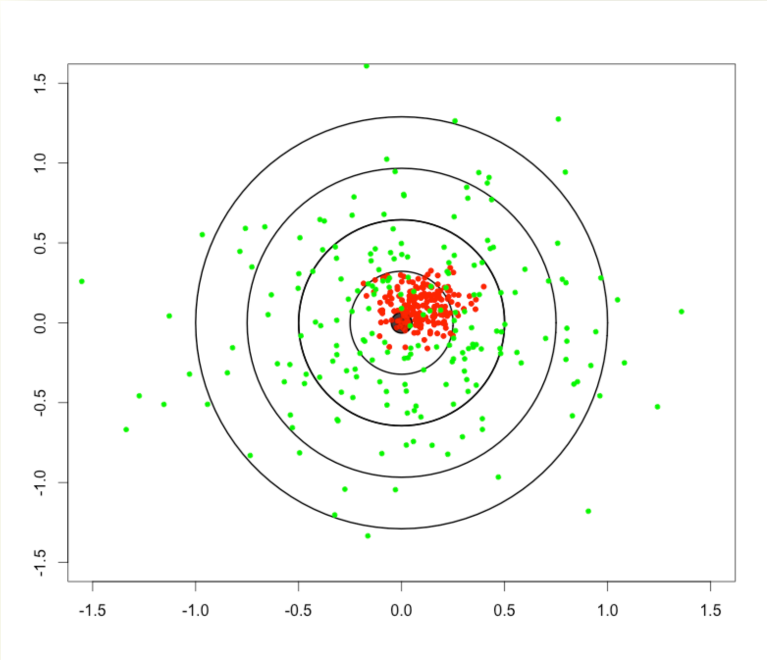
bias

$$\mathrm{MSE}\left(\tilde{b}\right) = E\left[\left(\tilde{b}-\beta\right)\left(\tilde{b}-\beta\right)\right] = \mathrm{Var}\left(\tilde{b}\right) + \left(E\left[\tilde{b}\right]-\beta\right)\left(E\left[\tilde{b}\right]-\beta\right)'$$

We want estimators with low MSE but it is possible these could be "biased."

b. Gauss-Markov Theorem

Consider the “green” and the “red” estimators. Note that the “bullseye” is the true beta.



Most of us would prefer the red to the green. Red has lower MSE but some bias.

c. MVN

The Multivariate Normal (MVN) distribution is a workhorse distribution both in finance and in econometrics.

Just as the univariate normal is a family of distributions indexed by the mean and variance, the MVN is indexed by the mean vector and Variance-Covariance Matrix.

The MVN density is given by:

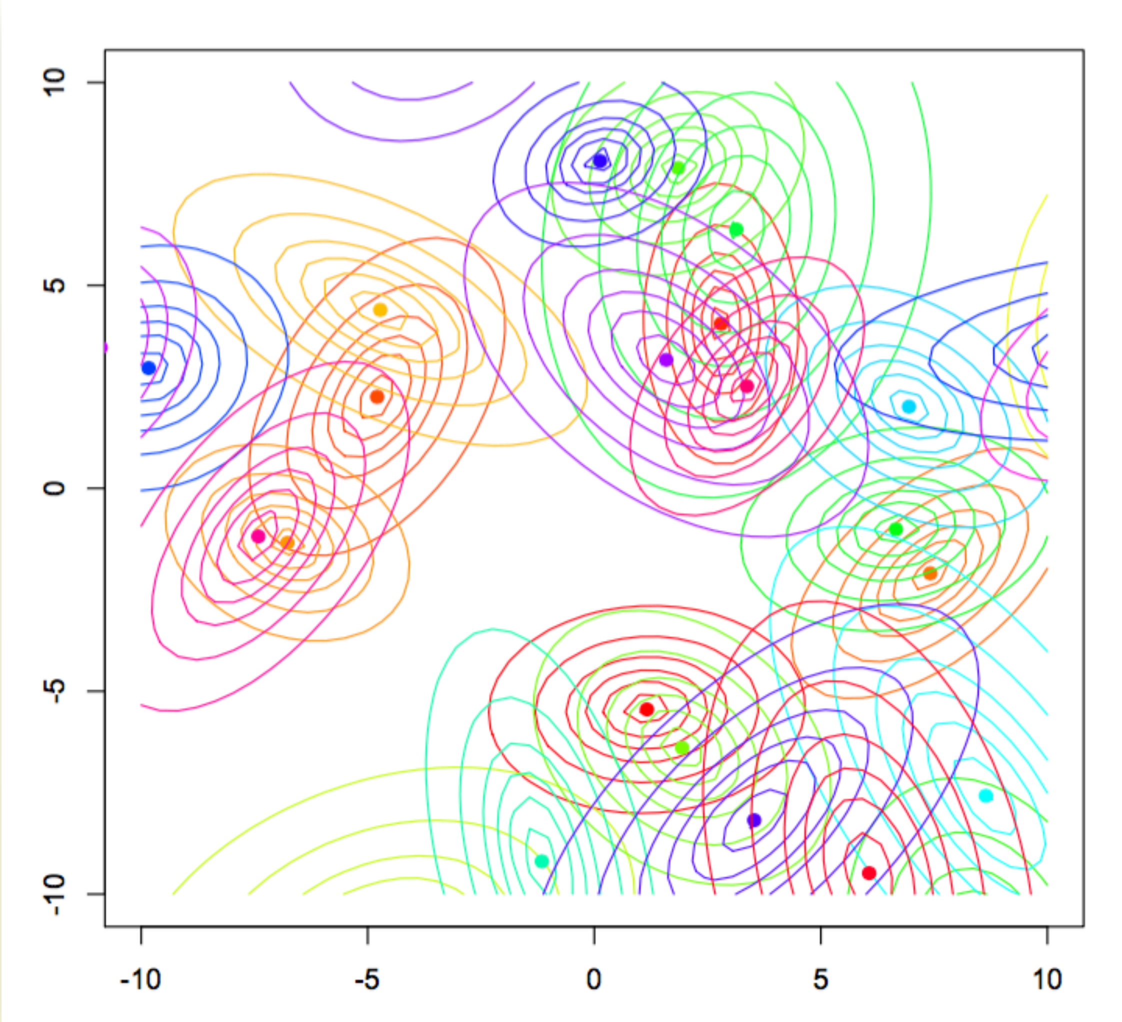
$$y \sim N(\mu, \Sigma)$$

$$p(y \mid \mu, \Sigma) = (2\pi)^{-\dim(y)/2} \left|\Sigma\right|^{-1/2} \exp\left(-\tfrac{1}{2}(y-\mu)'\Sigma^{-1}(y-\mu)\right)$$

c. MVN

The MVN density is a surface which is given by the density function (this expresses probability per volume element). The contours of this density are concentric ellipsoids centered on the mean vector.

Some Random Selected MVNs



c. MVN

To understand the MVN density, let's consider how to evaluate it and some special cases.

First, consider the case where

$$y' = \left(y_1,\ldots,y_n\right) \;\; y_i \sim iidN\left(\mu,\sigma^2\right)$$

$$y \sim N\left(\mu\iota,\sigma^2 I_n\right)$$

$$p\left(y \mid \mu\iota,\sigma^2\right) = \left(2\pi\right)^{-n/2}\left(\sigma^{2n}\right)^{-1/2}\exp\left(-\tfrac{1}{2\sigma^2}\left(y-\mu\iota\right)'\left(y-\mu\iota\right)\right)$$

$$= \left(2\pi\sigma^2\right)^{-n/2}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i-\mu\right)^2\right)$$

c. MVN- simulating from the distribution

We can use the Cholesky root idea to go back and forth between iid standard normals and MVN normals and simulate.

$$y \sim N(\mu, \Sigma) \quad \Sigma = U'U$$

$$z \sim N(0, I)$$

$$y = \mu + U'z \sim N(\mu, \Sigma)$$

$$E[y] = \mu$$

$$E[(y-\mu)(y-\mu)'] = E[U'z(U'z)'] = U'E[zz']U = \Sigma$$

c. MVN - simulating

In R,

```
> k=2
> n=1000
> mu=c(rep(1,k))
> Sigma=matrix(c(1,.8,.8,1),ncol=2)
> U=chol(Sigma)
> Z=matrix(rnorm(n*k),ncol=n)
> Y=crossprod(U,Z)+mu  # Y is k x n
> Y=t(Y)    #Y is n x k now
> # or
> dim(Z)=c(n,k)
> Y=Z%*%U + mu
```

c. MVN- Estimating Parameters of MVN

Suppose we have a sample of multivariate observations which we assume have a MVN distribution. How do we estimate the parameters?

We can look at the data as organized by observations (rows) or variables (columns):

$$Y = \begin{bmatrix} y_1' \\ y_2' \\ \vdots \\ y_n' \end{bmatrix} = \begin{bmatrix} y_1 & y_2 & \cdots & y_k \end{bmatrix}$$

$$y_i \sim \mathrm{iidMVN}(\mu, \Sigma)$$

c. Estimating Parameters of MVN

The estimates of the means and variances are straightforward.

$$\hat{\mu} = \frac{1}{n} Y' \iota_n$$

$$\hat{\Sigma} = \frac{1}{n-1} \left(Y - \frac{1}{n} \iota \iota' Y \right)' \left(Y - \frac{1}{n} \iota \iota' Y \right)$$

or

$$\hat{\sigma}_{kj} = \frac{1}{n-1} \left(y_k - \hat{\mu}_k \iota \right) \left(y_j - \hat{\mu}_j \iota \right)'$$

d. Linear Combinations of MVNs

Suppose we take a linear combination of RVs.

$$p = c'y \quad y \sim N(\mu, \Sigma)$$

$$E[p] = c'\mu$$

$$\begin{aligned} Var(p) &= E\left[(c'y - c'\mu)(c'y - c'\mu)'\right] \\ &= E\left[c'(y-\mu)(y-\mu)'c\right] \\ &= c'\Sigma c \end{aligned}$$

e. Distribution of Least Squares with Normal errors

Suppose the errors in a regression equation are iid Normal.

$$y = X\beta + \varepsilon$$

$$\varepsilon \sim N\left(0, \sigma^2 I_n\right)$$

What is the distribution of the least squares estimator?

$$b = \left(X'X\right)^{-1} X'y = Wy$$

$$b \sim MVN\left(\beta, \sigma^2 \left(X'X\right)^{-1}\right)$$

f. Prediction in Multiple Regression

Let's derive the variance of forecast or prediction errors from multiple regression.

Here we are using the results about the variance of a linear combination of RVs.

$$e_f = y_f - x_f'b = y_f - x_f'\beta + x_f'\beta - x_f'b$$

$$\mathrm{Var}(e_f) = \mathrm{Var}(\varepsilon_f) + \underbrace{\mathrm{Var}\left(x_f'(b-\beta)\right)}_{\text{sampling error}}$$

$$= \sigma^2\left(1 + x_f'(X'X)^{-1}x_f\right)$$

g. General Linear Hypothesis

We have considered t-tests which are about individual coefficients and F-test which are about setting groups of coefficients simultaneously to zero.

But what about other hypotheses?

Consider a test for market timing ability. We regress the return on a portfolio on not the marketing return but the “positive” and “negative” parts.

$$R_t = \alpha + \beta_{dwn} R_{mt}^{-} + \beta_{up} R_{mt}^{+} + \varepsilon_t$$

g. General Linear Hypothesis

Here the “positive” and “negative” parts of the market return are defined as

$$R_{mt}^{-} = \begin{cases} R_{mt}, \text{ if } R_{mt} < 0 \\ 0, R_{mt} \geq 0 \end{cases} \qquad R_{mt}^{+} = \begin{cases} R_{mt}, \text{ if } R_{mt} > 0 \\ 0, R_{mt} \leq 0 \end{cases}$$

$H_0 : \beta_{\downarrow} = \beta_{\uparrow}$

$H_A : \neq$

In a linear model, $\beta_{dwn} = \beta_{up}$. No market-timing ability is the linear model. Thus, we are interested in testing this hypothesis.

How do we test this? These hypotheses are special cases of what is called the General Linear Hypothesis.

g. General Linear Hypothesis

The General Linear Hypothesis:

$$H_0 : R\beta = r$$
$$H_A : R\beta \neq r$$

Some special cases

1. R is a diagonal matrix with ones in some places and zeroes elsewhere (t and F tests considered so far)
2. Put 1, and -1 in the right places to test equalities.

g. General Linear Hypothesis

Two Examples.

1. Suppose there are two independent variables and we want to do the overall F-test. There we are imposing q = 2 "restrictions" on the beta vector.

$$\underset{q\times k}{R} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad r = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

2. Test equality of two slope coefficients (e.g. market timing).

$$\underset{q\times k}{R} = \begin{bmatrix} 0 & 1 & -1 \end{bmatrix} \qquad r = \begin{bmatrix} 0 \end{bmatrix}$$

g. General Linear Hypothesis

How should we test the GLH? Let's evaluate the restrictions at the least squares estimates and see how "close" they are to holding.

Compute $Rb - r$

What is the distribution of these discrepancies under the null hypothesis?

$$E\left[Rb - r\right] = R\beta - r = 0\left(\text{under null}\right)$$

$$Var\left(V = Rb - r\right) = E\left[\left(Rb - r\right)\left(Rb - r\right)'\right]$$

g. General Linear Hypothesis

$$\mathrm{Var}\left(V = Rb - r\right) = E\left[\left(Rb - r\right)\left(Rb - r\right)'\right]$$

$$= E\left[\left(Rb - R\beta + R\beta - r\right)\left(Rb - R\beta + R\beta - r\right)'\right]$$

$$= E\left[R\left(b - \beta\right)\left(b - \beta\right)' R'\right] = \sigma^2 R\left(X'X\right)^{-1} R'$$

We will estimate the error variance and use an F statistic.

$$F = \frac{\left(Rb - r\right)'\left(R\left(X'X\right)^{-1}R'\right)^{-1}\left(Rb - r\right)}{qs^2}$$

$$\text{where } s^2 = \left(e'e\right) / \left(n - k\right)$$

g. General Linear Hypothesis

Under the null, this has an $F_{q,n-k}$ distribution. Let's try an example in R.

```
> library(reshape2)
> data(Vanguard)
> Van=Vanguard[,c(1,2,5)]   # grab relevant cols
> V_reshaped=dcast(Van,date~ticker,value.var="mret")
> data(marketRf)
> Van_mkt=merge(V_reshaped,marketRf,by="date")
> mkt_up=ifelse(Van_mkt$vwretd>0,1,0)
> Van_mkt$upvw=mkt_up*Van_mkt$vwretd
> Van_mkt$dwnvw=(1-mkt_up)*Van_mkt$vwretd
> mkt_timing=lm(VWNFX~upvw+dwnvw,data=Van_mkt)
```

g. General Linear Hypothesis

With results --

```
> lmSumm(mkt_timing)
Multiple Regression Analysis:
    3 regressors(including intercept) and 336 observations

lm(formula = VWNFX ~ upvw + dwnvw, data = Van_mkt)

Coefficients:
            Estimate Std Error t value p value
(Intercept) 0.001526  0.001486    1.03   0.305
upvw        0.876200  0.038740   22.62   0.000
dwnvw       0.901600  0.037830   23.83   0.000
---
Standard Error of the Regression:  0.017
Multiple R-squared:  0.846  Adjusted R-squared:  0.846
Overall F stat: 917.85 on 2 and 333 DF, pvalue= 0
```

g. General Linear Hypothesis

Now let's compute F-stat and p-value.

```
> R=matrix(c(0,1,-1),byrow=TRUE,nrow=1)
> r=c(0)
> X=cbind(c(rep(1,nrow(Van_mkt))),Van_mkt$upvw,Van_mkt$dwnvw)
> b=as.vector(mkt_timing$coef)
> QFmat=chol2inv(chol(crossprod(X)))
> QFmat=R%*%QFmat%*%t(R)
> Violation=R%*%b-matrix(r,ncol=1)
> fnum=t(Violation)%*%chol2inv(chol(QFmat))%*%Violation
> n_minus_k = nrow(Van_mkt)-length(b)
> fdenom=nrow(R)*sum(mkt_timing$resid**2)/n_minus_k
> f=fnum/fdenom
> f
          [,1]
[1,] 0.1708104
> pvalue=1-pf(f,df1=nrow(R),df2=n_minus_k)
> pvalue
          [,1]
[1,] 0.6796486
```

g. General Linear Hypothesis

If you don’t want to do this yourself, then you can use the package, `multicomp`. However, you might take longer to learn this package than you would to run our class code snippet!

Appendix A. Gauss-Markov Theorem

Let's consider a general unbiased estimator of the regression coefficients.

Express this estimator as a "change" from least squares.

$$\tilde{b} = (M + A)y \text{ where } M = (X'X)^{-1}X'$$

$$\tilde{b} = My + Ay = b + Ay$$

Since our new estimator must be unbiased, we can't use just any linear combinations of y!

$$E[\tilde{b}] = E[My + Ay] = E[b] + E[Ay]$$

But

$$E\left[Ay\right] = E\left[A\left(X\beta + \varepsilon\right)\right] = AX\beta + E\left[A\varepsilon\right]$$
$$= AX\beta + 0$$

This means that we require AX = 0 for unbiasedness.

Now let's compute the variance and remember that we are restricted to only estimators for which AX = 0.

$$Var\left(\tilde{b}\right) = E\left[\left(\left(M + A\right)y - \beta\right)\left(\left(M + A\right)y - \beta\right)'\right]$$

$$\begin{aligned}\left(M+A\right)y-\beta &= \left(M+A\right)\left(X\beta+\varepsilon\right)-\beta \\ &= MX\beta+AX\beta+\left(M+A\right)\varepsilon-\beta \\ &= \left(M+A\right)\varepsilon\end{aligned}$$

$$\begin{aligned} MX\beta &= \beta \\ AX &= 0 \end{aligned}$$

$$\begin{aligned}\mathrm{Var}\left(\tilde{b}\right) &= E\left[\left(M+A\right)\varepsilon\varepsilon'\left(M+A\right)'\right] \\ &= \left(M+A\right)E\left[\varepsilon\varepsilon'\right]\left(M+A\right)' \\ &= \sigma^2\left(M+A\right)\left(M'+A'\right) \\ &= \sigma^2\left(MM'+MA'+AM'+AA'\right)\end{aligned}$$

But,
$$MA' = \left(X'X\right)^{-1} X'A = 0$$
$$AM' = A\left(X\left(X'X\right)^{-1}\right) = 0$$

Therefore,

$$Var\left(\tilde{b}\right) == \sigma^2 \left(MM' + AA'\right)$$

The diagonal elements of this Variance-Covariance matrix can be minimized by taking A = 0.

Thus, least squares is the best unbiased, linear estimator.

Appendix A. Gauss-Markov Theorem

What about linear estimators? Not in the age of computers do we care!

However, it turns out that if the error terms in the regression are normally distributed, then least squares is the "optimal" or most efficient unbiased estimator. More later.

Appendix B. Joint, Marginal and Conditional Distributions

When we are considering the joint distribution of a set of k RVs, we can characterize this by the joint density or the set of marginals and conditionals.

y is k dim random vector

Joint: $p(y)$ or $p(y_1, y_2) = p(y_1)p(y_2 \mid y_1)$

Marginal: $p(y_1) = \int p(y_1, y_2)\,dy_2$

Conditional: $p(y_2 \mid y_1) = \dfrac{p(y_1, y_2)}{p(y_1)}$

Appendix B. Marginals and Conditionals for the MVN

Both marginals and conditionals are normal! But different means, variances.

$$y \sim N(\mu, \Sigma)$$

$$y' = (y_1', y_2')$$

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

$$y_1 \sim N(\mu_1, \Sigma_{11})$$

$$y_2 \mid y_1 \sim N\left(\mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(y_1 - \mu_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right)$$

Appendix B. Marginals and Conditionals for the MVN

To understand this better, consider a simplified bivariate normal distribution.

$$y \sim N(\mu, \Sigma)$$

$$\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

$$y_1 \sim N(\mu_1, 1)$$

$$y_2 \mid y_1 \sim N\left(\mu_2 + \rho\left(y_1 - \mu_1\right), 1 - \rho^2\right)$$

If rho = 0, then we have independence. If y_1 = mu_1, then we don't update! $Var(Y_2|Y_1) < Var(Y_2)$.

Appendix C. Quadratic Forms in MVNs

In other contexts, we might be interested in determining the distribution of the sum of squares of Normal RVs. If we start with the fact that

$$Y = \sum_{i=1}^{\nu} Z_i^2 \sim \chi_\nu^2$$

where Z_i are N(0,1), then

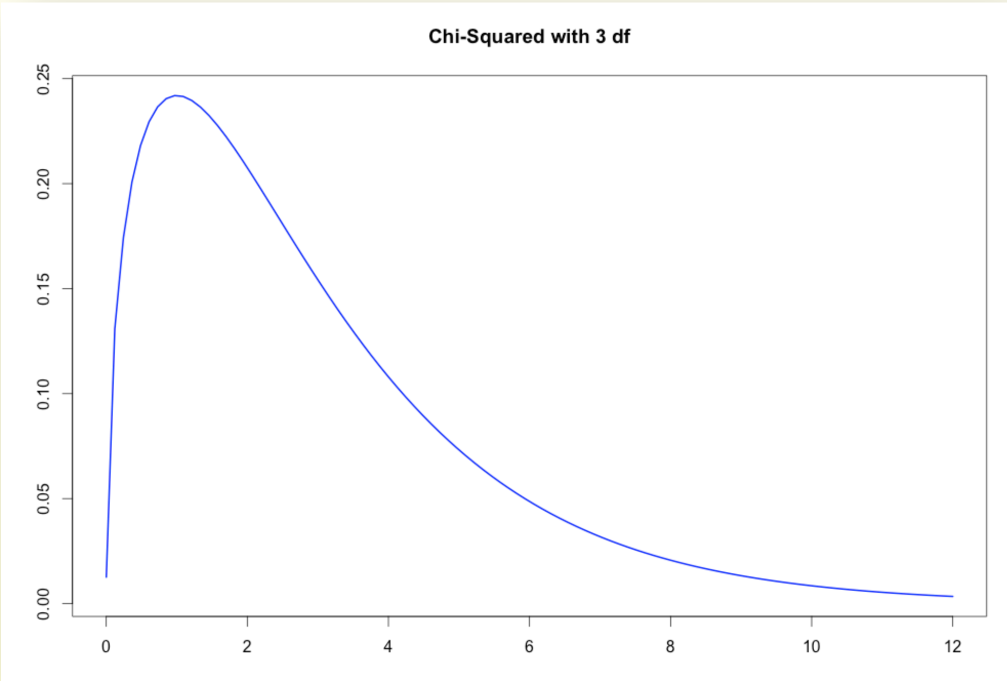
$$y = (x - \mu)' \Sigma^{-1} (x - \mu) \sim \chi_\nu^2 \qquad \nu = \dim(x)$$

Why is this true?

Appendix C. Sidebar on Chi-squared Distribution

The Chi-squared distribution is a right skewed distribution with on parameter, called the degrees of freedom parameter. If

$$Y \sim \chi^2_\nu,\ E\left[Y\right] = \nu,\ \mathrm{Var}\left(Y\right) = 2\nu$$



Appendix C. Quadratic Forms in MVNs

$$\Sigma = U'U$$

$$\Sigma^{-1} = U^{-1}\left(U'\right)^{-1} = U^{-1}\left(U^{-1}\right)'$$

$$\text{define } v = \left(U^{-1}\right)'\left(x-\mu\right), \text{ then } \operatorname{Var}\left(v\right) = \left(U^{-1}\right)'\Sigma U^{-1}$$

$$\left(U^{-1}\right)'\Sigma U^{-1} = \left(U^{-1}\right)'U'UU^{-1} = I$$

$$\text{so } y = \left(x-\mu\right)'\Sigma^{-1}\left(x-\mu\right)$$

$$= v'v \qquad v \sim N\left(0,I\right)$$

Appendix C. Quadratic Forms in MVNs

Another commonly encountered problem is the distribution of a quadratic form in what is called an Idempotent matrix.

What is the distribution of

$$Y = \sum_{i=1}^{n} \left(X_i - \bar{X} \right)^2 = x'Mx$$

where

$$M = I_n - \frac{\iota\iota'}{n}$$

and

$$\left\{ X_i \right\} \sim iidN\left(\mu, \sigma^2 \right)$$

Appendix C. Quadratic Forms in MVNs

What does M do? It “demeans” each of the values of X_i by subtracting the mean. This is equivalent to projecting the entire X vector down on the sub-space spanned by the vector of ones or

$$\min_c \left(x - c\iota\right)^2$$

Thus, “demeaning” is a projection operator. Note that once you project something, it stays projected. In fact, this is one way of defining a projection. Or

$$M\left(Mx\right) = Mx \text{ or } M^2 = M$$

Also, what will M do to a vector of the same scalar? It will annihilate it!

$$M(\mu\iota) = 0$$

Thus,

$$x'Mx = (x - \mu\iota)' M (x - \mu\iota)$$

$$= \sigma^2 \left(\frac{x - \mu\iota}{\sigma}\right)' M \left(\frac{x - \mu\iota}{\sigma}\right) = \sigma^2 z'Mz$$

Appendix C. Quadratic Forms in MVNs

It turns out that a quadratic form in independent standard normal rvs has Chi-squared distribution with degrees of freedom equal to the trace of the idempotent matrix.

A trace of a matrix is the sum of diagonal elements.

$$\mathrm{tr}\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} = \sum_{i=1}^{n} a_{ii}$$

Appendix C. Quadratic Forms in MVNs

$$x'Mx = \left(x - \mu\iota\right)' M\left(x - \mu\iota\right) = \sigma^2 z'Mz$$

$$x'Mx \sim \sigma^2 \chi^2_{tr(M)}$$

$$\frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2 \sim \frac{\sigma^2}{n-1}\chi^2_{tr(M)}$$

$$tr\left(M\right) = tr\left(I_n - \frac{\iota\iota'}{n}\right) = n-1$$

Appendix C. Quadratic Forms in MVNs

Recall that

$$E\left[\chi^2_\nu\right] = \nu$$

This implies that the standard variance estimator is unbiased and "justifies" dividing by n-1.

$$E\left[s^2\right] = E\left[\frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2\right] = \frac{\sigma^2}{n-1}\left(n-1\right) = \sigma^2$$

Appendix C. Quadratic Forms in MVNs

The same logic applies to the residual sum of squares from a regression model.

$$s^2 = \frac{1}{n-k} e'e = \frac{1}{n-k}\left(y - Xb\right)'\left(y - Xb\right)$$

The residual vector, e, is formed from an idempotent matrix which projects y onto the column space of X.

$$\begin{aligned} e &= y - Xb = X\beta + \varepsilon - X\left(X'X\right)^{-1}X'\left(X\beta + \varepsilon\right) \\ &= \left(I - X\left(X'X\right)^{-1}X\right)\varepsilon = M_X\varepsilon \end{aligned}$$

Appendix C. Quadratic Forms in MVNs

$$s^2 = \frac{1}{n-k} e'e = \left(M_X \varepsilon\right)' \left(M_X \varepsilon\right) = \varepsilon' M_X' M_X \varepsilon$$

$$= \frac{1}{n-k} \varepsilon' M_X \varepsilon = \frac{\sigma^2}{n-k} \left(\frac{\varepsilon}{\sigma}\right)' M_X \left(\frac{\varepsilon}{\sigma}\right)$$

$$\mathrm{tr}\left(M_X\right) = \mathrm{tr}\left(I_n - X\left(X'X\right)^{-1} X'\right) = n - \mathrm{tr}\left(X\left(X'X\right)^{-1} X'\right)$$

$$= n - \mathrm{tr}\left(X'X\left(X'X\right)^{-1}\right) = n - \mathrm{tr}\left(I_k\right) = n - k$$

Therefore,

$$E\left[s^2\right] = \frac{\sigma^2}{n-k} \mathrm{tr}\left(M_X\right) = \frac{\sigma^2}{n-k} \times \left(n-k\right) = \sigma^2$$

Why is n-k called the "degrees" of freedom?

What M_X does is to project the n dimensional vector, y, on to the space spanned by the k columns of X and form the residual.

$$M_X y = e$$

Therefore, what is left is only n-k linear independent quantities.

In other words, if I tell you (n-k) of the residuals, you can find all of the other k residuals!

Glossary of Symbols

$F_{\upsilon_1,\upsilon_2}$ - F distribution with υ_1 df in the numerator, υ_2 df in the denominator

f - value of F statistic

$\bar{R}^2$ - adjusted R-squared

Important Equations

$$\boxed{Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \varepsilon}$$

Multiple Regression Model

$$\boxed{f = \frac{R^2 / k}{\left(1 - R^2\right) / (N - k - 1)} = \frac{SSR / k}{SSE / (N - k - 1)}}$$

Overall F-test

$$\boxed{\overline{R}^2 = 1 - \frac{SSE / (N - k - 1)}{SST / (N - 1)} = 1 - \frac{s^2}{s_y^2}}$$

Adjusted R-squared

Important Equations

$$f = \frac{\Delta R^2 / k_2}{\left(1 - R^2_{full}\right) / \left(N - (k_1 + k_2) - 1\right)} \sim F_{k_2, N-k_1-k_2-1} \quad \text{under } H_0$$

Partial or Inclusion/
Exclusion F-test

Glossary of R Commands

- `pf(f value, df1=5,df2=54)`: Returns the probability left of value under the F distribution with df of numerator as 5, and df of denominator as 54.
- `qf(prob,df1=5,df2=54)`: Returns the critical value of the probability under the F distribution with df of numerator as 5, and df of denominator as 54.
- `matrix(vec,ncol=x)`: creates an array or matrix from the vector, vec, with x columns.
- `cbind(col1,col2)`: creates an array or matrix by combining col1 and col2 side by side or columnwise
- `chol(A)`: finds "square root" of a positive definite matrix, A

Glossary of R Commands

- `chol2inv(chol(A))`: finds the inverse of the matrix A
- `crossprod(A,B)`: computes A'B efficiently.
- `diag(A)`: fetches diagonal of A
- `A%*%B`: multiplies matrix A by matrix B
- `t(A)`: tranposes the matrix A