

Lecture 3-d

OLS – Goodness of Fit, and Introduction to MLE

Brooks (4th edition): Chapters 3 & 4

© R. Susmel, 2022 (for private use, not to be posted/shared online).¹

Review: OLS – Summary

- *Classical linear regression model* (CLM) - Assumptions:

(A1) DGP: $\mathbf{y} = \mathbf{X} \boldsymbol{\beta} + \boldsymbol{\varepsilon}$ is correctly specified.

(A2) $E[\boldsymbol{\varepsilon} | \mathbf{X}] = 0$

(A3) $\text{Var}[\boldsymbol{\varepsilon} | \mathbf{X}] = \sigma^2 \mathbf{I}_T$

(A4) \mathbf{X} has full column rank $\rightarrow \text{rank}(\mathbf{X}) = k$, where $T \geq k$.

- Objective function: $S(\mathbf{x}_i, \boldsymbol{\beta}) = \sum_i \varepsilon_i^2 = \boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$

First order condition: $-2 \mathbf{X}'\mathbf{y} + 2 \mathbf{X}'\mathbf{X} \mathbf{b} = 0$

Solving for $\mathbf{b} \Rightarrow \mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{y}$ ($k \times 1$) vector

- Finite Properties for \mathbf{b} .

1) Unbiased: $E[\mathbf{b} | \mathbf{X}] = \boldsymbol{\beta}$

2) Efficiency (& BLUE): $\text{Var}[\mathbf{b} | \mathbf{X}] = \sigma^2 (\mathbf{X}'\mathbf{X})^{-1}$

3) If (A5) $\boldsymbol{\varepsilon} | \mathbf{X} \sim i.i.d. N(\mathbf{0}, \sigma^2 \mathbf{I}_T) \Rightarrow \mathbf{b} | \mathbf{X} \sim i.i.d. N(\boldsymbol{\beta}, \sigma^2 (\mathbf{X}'\mathbf{X})^{-1})$

Review: OLS – Summary

- Asymptotic properties for \mathbf{b} .

4) Consistent: $\mathbf{b} \xrightarrow{p} \boldsymbol{\beta}$

5) Asymptotic Normality: $\mathbf{b} \xrightarrow{a} N(\boldsymbol{\beta}, \sigma^2 (\mathbf{X}' \mathbf{X})^{-1})$

We use these asymptotic properties when we introduce more realistic assumptions about the data (\mathbf{X} is an RV) and (A5) does not apply.

Review: OLS – Testing One Parameter

- We want to test hypothesis about one parameter in our linear model:

$$\mathbf{y} = \mathbf{X} \boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

1. Set H_0 and H_1 (about only one parameter): $H_0: \beta_k = \beta_k^0$
 $H_1: \beta_k \neq \beta_k^0$

2. Appropriate $T(X)$: *t-statistic*. Under H_0 :

$$\text{If (A5),} \quad t_k = \frac{b_k - \beta_k^0}{s_{b,k}} \sim t_{T-k}$$

$$\text{Otherwise,} \quad t_k \xrightarrow{d} N(0, 1)$$

3. Compute t_k, \hat{t} , using b_k, β_k^0, s , and $(\mathbf{X}' \mathbf{X})^{-1}$. Get *p-value*(\hat{t}).

4. Rule: Set an α level. If $p\text{-value}(\hat{t}) < \alpha \Rightarrow \text{Reject } H_0: \beta_k = \beta_k^0$
Alternatively, if $|\hat{t}| > t_{T-k, 1-\alpha/2} \Rightarrow \text{Reject } H_0: \beta_k = \beta_k^0$.

Review: OLS – Testing One Parameter

- Special case: $H_0: \beta_k = 0$
 $H_1: \beta_k \neq 0$.

Then,

$$t_k = \frac{b_k}{SE[b_k]} = t\text{-value or } t\text{-ratio}.$$

- Usually, $\alpha = 5\%$, then if $|t_k| > 1.96 \approx 2$, we say the coefficient b_k is “*significant*.”

Review: OLS – Is IBM’s Beta equal to 1?

Example: Using the 1-factor CAPM for IBM returns, we test if IBM’s market $\beta = 1$, that is, if IBM bears the same market risk as the market. Using the *lm* function previous estimation:

```
SFX_da <-
read.csv("http://www.bauer.uh.edu/rsusmel/4397/Stocks_FX_1973.csv",head=TRUE,sep
=",")
x_ibm <- SFX_da$IBM           # Extract IBM price data
x_Mkt_RF <- SFX_da$Mkt_RF     # Extract Market excess returns (in %)
x_RF <- SFX_da$RF             # Extract risk free rate (in %)
T <- length(x_ibm)            # Sample size
lr_ibm <- log(x_ibm[-1]/x_ibm[-T]) # Log returns for IBM (lost one observation)
Mkt_RF <- x_Mkt_RF[-1]/100     # Adjust size (take one observation out )
RF <- x_RF[-1]/100             # Adjust size (take one observation out )
ibm_x <- lr_ibm - RF           # Define excess returns for IBM

fit_ibm_capm <- lm(ibm_x ~ Mkt_RF) # OLS estimation with lm package in R
```

Review: OLS – Is IBM's Beta equal to 1?

Example (continuation):

```
> summary(fit_ibm_capm)
```

Coefficients:

```
      Estimate Std. Error t value Pr(> |t|)
(Intercept) -0.005791  0.002487  -2.329  0.0202 *
xMkt_RF      0.895774  0.053867  16.629 <2e-16 ***
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```
b_ibm <- fit_ibm_capm$coefficients # Extract from lm function OLS coefficients
SE_ibm <- sqrt(diag(vcov(fit_ibm_ff3))) # SE from fit_ibm (also a kx1 vector)
t_beta1 <- (b_ibm[2] - 1)/SE_ibm[2] # t-stat for H0: Beta1 - 1
> t_beta1
[1] -1.934877 ⇒ |t̂1 = -1.934877| < 1.96 ⇒ Cannot reject H0 at 5% level
p_val <- (1 - pnorm(abs(t_beta1))) * 2 # pvalue for t_beta (adjusted b/c two sided test)
> p_val
[1] 0.0530 ⇒ cannot reject H0: βIBM = 1 at 5% level, but borderline!
```

Review: OLS – Is IBM's Beta equal to 1?

Example (continuation):

Conclusions: Cannot reject H_0 : $\beta_{IBM} = 1 \Rightarrow$ IBM has a one-to-one risk relation with the market, but borderline test decision!

Note: You can get same results using linear algebra. From last class:

```
b <- solve(t(x)%*% x)%*% t(x)%*% y # b = (X'X)-1X'y (OLS regression)
SE <- sqrt(diag(vcov(fit_ibm_capm))) # SE from fit_ibm (also a kx1 vector)
t_beta1 <- (b[2] - 1)/SE[2] # t-stat for H0: Beta1 - 1
> t(b)
      Mkt_RF
[1,] -0.005791039 0.895773564
> t(SE_b)
      Mkt_RF
[1,] 0.002487 0.053867
> t_beta1
[1] -1.934877
```

Review: Application – Testing the CAPM

Example: We test the CAPM for IBM using the time-series.

$$\text{CAPM: } E[r_{i,t} - r_f] = \beta_i E[(r_{m,t} - r_f)].$$

\Rightarrow equilibrium excess returns are only determined by excess market returns –i.e., the CAPM is a one factor model. There is no constant or extra factors besides the market.

Then, there are two ways to test the CAPM:

- 1) Check if a constant is significant
- 2) Check if other factors are significant.

In this example, we check if a constant is significant; later, with the 3-factor Fama-French model, we check if other factors are significant.

- Note: The CAPM was so well regarded, before the early 1980s, that any significant factor, beyond $r_{m,t} - r_f$, was considered an “*anomaly*.”

Review: Application – Testing the CAPM

Example (continuation):

CAPM DGP:

$$r_{i,t} - r_f = \alpha_i + \beta_i (r_{m,t} - r_f) + \varepsilon_{i,t}, \quad i = 1, \dots, N \quad \& \quad t = 1, \dots, T$$

Thus, we test the CAPM by testing H_0 (CAPM holds): $\alpha_{i=IBM} = 0$

H_1 (CAPM rejected): $\alpha_{i=IBM} \neq 0$

```
SFX_da <-
read.csv("http://www.bauer.uh.edu/rsusmel/4397/Stocks_FX_1973.csv",head=TRUE,sep=",")
x_ibm <- SFX_da$IBM           # Extract IBM price data
x_Mkt_RF <- SFX_da$Mkt_RF     # Extract Market excess returns (in %)
x_RF <- SFX_da$RF             # Extract risk free rate (in %)
T <- length(x_ibm)            # Sample size
lr_ibm <- log(x_ibm[-1]/x_ibm[-T]) # Log returns for IBM (lost one observation)
Mkt_RF <- x_Mkt_RF[-1]/100     # Adjust size (take one observation out)
RF <- x_RF[-1]/100
```

Review: Application – Testing the CAPM

Example (continuation):

```
ibm_x <- lr_ibm - RF # Define excess returns for IBM
fit_ibm_capm <- lm(ibm_x ~ Mkt_RF) # OLS estimation with lm package in R
> summary(fit_ibm_capm)
```

Coefficients:

```
      Estimate Std. Error t value Pr(> |t|)
(Intercept) -0.005791  0.002487  -2.329  0.0202 *
xMkt_RF      0.895774  0.053867  16.629 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Q: Is $\alpha_{i=IBM} = 0$? Compute the t-value of $\alpha_{i=IBM}$:

$$\hat{t}_\alpha = \frac{\alpha_{i=IBM}}{SE[\alpha_{i=IBM}]} = \frac{-0.005791}{0.002487} = -2.329$$

$$\Rightarrow |\hat{t}_\alpha| > 1.96 \quad \Rightarrow \text{Reject } H_0 \text{ (CAPM) at 5\% level}$$

Review: Application – Testing the CAPM

Example (continuation):

Note: You can also reject H_0 by looking at the *p-value* of intercept
 $p\text{-value}(|\hat{t}_\alpha| = 2.329) = 0.0202 < \alpha = 5\% \Rightarrow \text{Reject } H_0 \text{ at 5\% level}$

Conclusion: The CAPM is rejected for IBM at the 5% level.
 Sometimes, we say the CAPM is “**rejected in the time-series.**”

Interpretation: Given that the intercept is significant (& negative), IBM **underperformed** relative to what the CAPM expected:

$$\begin{aligned} -r_{IBM,t} - r_f &: \text{mean}(\text{ibm_x}) = -0.00073141 \\ -r_{IBM,t} - r_f \text{ (CAPM): } \beta_i * \text{mean}(\text{Mkt_RF}) &= 0.895774 * 0.0056489 \\ &= 0.0050601 \\ -\text{Ex-post difference: } -0.00073141 - 0.0050601 &= -0.00579151 (\approx \alpha_{IBM}) \end{aligned}$$

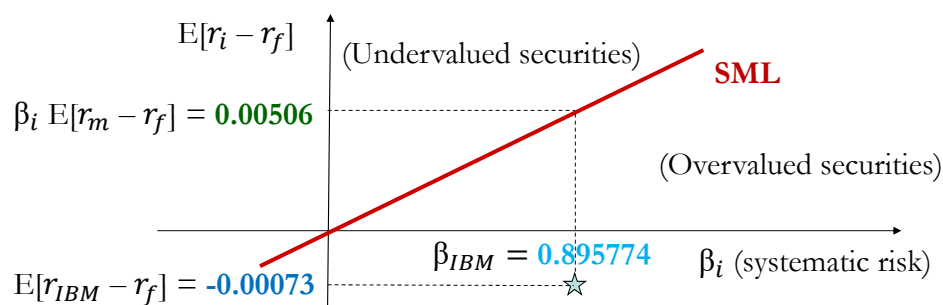
Review: Application – Testing the CAPM

Example (continuation):

IBM underperformed relative to what the CAPM expected by

$$\alpha_{i=IBM} = -0.005791$$

Then, according to the SML, IBM has been overvalued. The average, negative, performance (-0.00073) is the performance of a much safer asset, with a small, negative β ($=-0.1295$)!



OLS Estimation – Testing the CAPM: Remark

- We tested (& rejected) the CAPM for one asset only, IBM. But, the CAPM should apply to all assets. Suppose we have N assets. Then, a test for the CAPM involves testing N α_i 's:

$$H_0: \alpha_1 = \alpha_2 = \dots = \alpha_N = 0$$

$$H_1: \text{at least one } \alpha_i \neq 0.$$

- This test is a **joint** test. It requires a simultaneous estimation of N CAPM equations.
- There are different ways to approach the test of the CAPM in the cross-section. Two popular methods are the **2-step procedure of Fama-MacBeth** (1973) and the Likelihood Ratio test.

OLS Estimation – Testing the CAPM (CS)

Q: Which assets pay a higher return? The SML answers this question: Assets with a higher exposure to market risk –i.e., higher β_i .

CAPM DGP (in the cross-section):

$$(r_i - r_f) = \alpha + \beta_i \lambda + \varepsilon_i, \quad i = 1, \dots, N$$

Testing implication of the SML for the cross-section of stock returns:

H_0 (CAPM holds in the CS): $\alpha = 0$ & $\lambda = E[r_{m,t} - r_{f,t}] > 0$

H_1 (CAPM rejected in the CS): $\alpha \neq 0$ and/or $\lambda \neq E[r_{m,t} - r_{f,t}] > 0$

- Again, we have a **joint** test. There are different ways to approach this test, a common approach is a two-step estimation, popularly known as Fama-MacBeth (1973).

OLS Estimation – Testing the CAPM (CS)

- Fama-MacBeth (1973) propose a well-known two-step approach to test the CAPM in the cross-section:

(1) Estimate β_i using the time series (T observations) for each asset i .
 $r_{i,t} - r_{f,t} = \alpha_i + \beta_i (r_{M,t} - r_{f,t}) + \varepsilon_{i,t}, \quad t = 1, \dots, T \Rightarrow$ Get N β_i 's.

(2) Using the N β_i 's as regressors, estimate

$$(\bar{r}_i - \bar{r}_f) = \alpha + \beta_i \lambda + \varepsilon_i, \quad i = 1, \dots, N$$

where $(\bar{r}_i - \bar{r}_f)$ is the average excess return of asset i in our sample.

The usual execution of almost all 2-step test procedures involves:

- 1) Since returns are estimated with a lot of noise, portfolios are used.
- 2) The estimation takes into account the possible change over time of beta coefficients, by estimating the coefficients every 5 or 10 years.

OLS Estimation – Testing the CAPM (CS)

Example: We test the CAPM, in the cross-section, using the 2-step Fama-McBeth method. We use returns of **25 Fama-French portfolios** (sorted by Size (ME) and Book-to-Market), downloaded, along the 3-Fama-French factors from Ken French's website.

```
FF_p_da <- read.csv("https://www.bauer.uh.edu/rsusmel/4397/FF_25_portfolios.csv",
head=TRUE, sep=",")
FF_f_da <- read.csv("https://www.bauer.uh.edu/rsusmel/4397/FF_3_factors.csv", head=TRUE,
sep=",")

# Extract variables from imported data
Mkt_RF_fm <- FF_f_da$Mkt_RF      # extract Market excess returns (in %)
HML_fm <- FF_f_da$HML           # extract HML returns (in %)
SMB_fm <- FF_f_da$SMB           # extract HML returns (in %)
RF_fm <- FF_f_da$RF             # extract Risk-free rate (in %)
Y_p <- FF_p_da[,2:26] - RF_fm    # Compute excess returns of 25 portfolios

T <- length(HML_fm)              # Number of observations (1926:July on)
x0 <- matrix(1,T,1)             # Vector of ones, represents constant in X
```

OLS Estimation – Testing the CAPM (CS)

Example (continuation):

```
x <- cbind(x0, Mkt_RF_fm)      # Matrix X (T x 2)
k <- ncol(Y_p)

## First Pass
Allbs = NULL                  # Initialize empty (a space to put betas)
for (i in seq(1,k,1)) {
  y <- Y_p[,i]                # select Y (portfolio)
  b <- solve(t(x)%*% x)%*% t(x)%*% y  # OLS regression = (X'X)^(-1) X'y
  Allbs = cbind(Allbs,b)      # accumulate b as rows
}

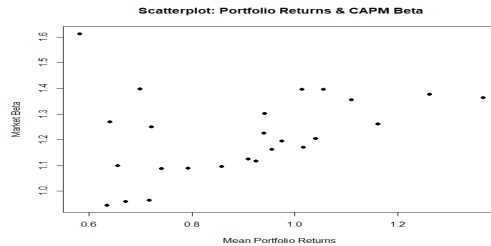
beta_ret <- cbind(colMeans(Y_p),t(Allbs)) # Mean portfolio returns along alpha & beta estimates
cor(beta_ret[,1], beta_ret[,3])          # Correlation of mean portfolio return & beta

> cor(beta_ret[,1], beta_ret[,3])
[1] 0.3326008

plot(beta_ret[,1], beta_ret[,3], main="Scatterplot: Portfolio Returns & CAPM Beta",
      xlab="Mean Portfolio Returns ", ylab="Market Beta", pch=19)
```

OLS Estimation – Testing the CAPM (CS)

Example (continuation):



Second Pass (CAPM)

```
fit_fm_capm_25 <- lm(beta_ret[,1] ~ beta_ret[,3])
```

```
> summary(fit_fm_capm_25)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	0.3728	0.3113	1.198	0.243	
beta_ret[, 3]	0.4289	0.2536	1.691	0.104	⇒ Not significant: Beta plays no role!

Conclusion: CAPM's beta does not seem to be useful to explain expected returns.

OLS Estimation – Testing the CAPM (CS)

- Fama and French (1992, 1993) estimated variations of the DGP with more factors. They found that β was weakly significant or not significant, even with the wrong sign, in explaining the C-S of stock returns, which created a big splash in the literature (“**Beta is dead**”). (See next slides.)

- Other researchers dispute the “Beta is dead” finding, criticizing the selection of estimation period, construction of portfolios, number of factors, statistical problems like measurement error and incorrect SE, etc.

- The debate about β & what (& how many) factors to include in the DGP continues.

The 3-Factor F-F Model

- The CAPM is routinely rejected. A popular alternative is the empirically derived 3-Factor Fama-French Model (1993), which adds to the CAPM two (macro-) factors, related to firm's characteristics:

a) *Size* factor (**SMB**): It is measured as the difference between the returns on diversified portfolios of small stocks (*small size portfolio*) and big stocks (*big size portfolio*) = long **S**mall & short **B**ig

b) *Value* factor or book-to-market factor (**HML**): It is measured as the difference between the returns on diversified portfolios of high B/M stocks (*high B/M portfolio*) and low B/M stock (*low B/M portfolio*) = long **H**igh & short **L**ow.

- With these 3 Fama-French factors, the DGP becomes:

$$r_{i,t} - r_f = \alpha_i + \beta_1 (r_{m,t} - r_f) + \beta_2 SMB_t + \beta_3 HML_t + \varepsilon_{i,t}.$$

The 3-Factor F-F Model: Descriptive Statistics

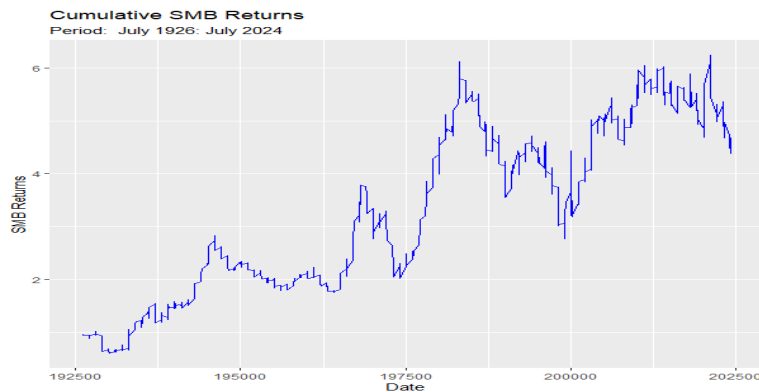
- Using monthly data from July 1927- July 2024 (98 years), we have the following returns for the macro-factors:

	$(r_{M,t} - r_{f,t})$	SMB_t	HML_t
Min	-29.1300	-17.2000	-13.8800
Median	1.0600	0.0700	0.1200
Mean	0.6845	0.1801	0.3449
Max	38.8500	36.5600	35.6100
SD	5.33431	3.176139	3.568389

- The annualized average returns (& SD) for Market, SMB, and HML are, respectively, **8.21%** (18.48%), **2.16%** (11%) and **4.14%** (12.36%). High premiums, but with high estimation uncertainty (high SD).

The 3-Factor F-F Model: Size Factor (SMB)

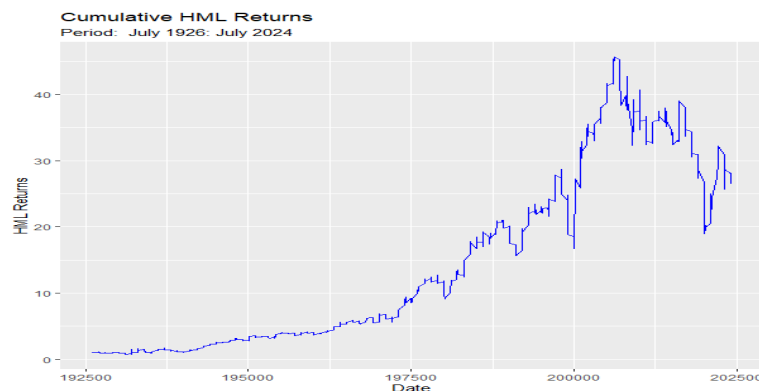
- Below, we plot the SMB and HML cumulative returns:



- The SMB or size factor performed very well up to about 1982, generating over 600% returns. Then, on average, the reverse occurred: large-cap stocks outdid small caps.

The 3-Factor F-F Model: Value Factor (HML)

- Below, we plot HML (value factor) returns:



- The value factor performed extremely well up to financial crisis 2008, generating over 4,500% returns. Value investing was very profitable. Since, then, on average, the pattern has changed.

The 3-Factor F-F Model: Factor Correlations

- The three (macro-) factors are, in theory, “*factor mimicking portfolios*,” that is, portfolios with exposure only to the factor in question (market, size, or value), and no exposure to any other factor.

The correlation matrix below show that, though not high, the correlations are not zero, especially with the market.

	Mkt_RF	SMB	HML
Mkt_RF	1.00000	0.31507	0.22697
SMB	0.31507	1.00000	0.12044
HML	0.22697	0.12044	1.00000

The 3-Factor F-F Model: Factor Construction

- Computation of factor portfolios in the Fama-French Models.

The portfolios are formed as follows:

Step 1. At the end of June of year t , sort the stock returns by attribute (size of Size, or B/M).

Step 2. Split the sorted assets by attribute into 3 equal/value-weighted portfolios (3 *tercile* portfolios). Split can be thinner (quintile portfolios) or based on more complicated sorts, for example, using 6 portfolios constructed by intersecting 2 size portfolios & 3 value portfolios.

Step 3. At the end of each month (week or day), from July of year t to June of year $t + 1$, based on the portfolios constructed in **Step 1**, compute the returns of each of the split portfolios.

Step 4 Form a “hedge portfolio”: long the top portfolio (say, top tercile) and short the bottom portfolio (say, bottom tercile).

The 3-Factor F-F Model: Interpretation

- **3-factor Fama-French DGP:**

$$r_{i,t} - r_f = \alpha_i + \beta_1 (r_{m,t} - r_f) + \beta_2 SMB_t + \beta_3 HML_t + \varepsilon_{i,t},$$

under this model, the main drivers of expected returns are sensitivity to the market, sensitivity to size, and sensitivity to value stocks, as measured by the book-to-market ratio.

- Interpretation of coefficients (also called “*factor loadings*”):
 - β_1 has the same interpretation as in the CAPM, it measures the relation between asset i risk and market risk.
 - β_2 measures how tilted asset i is towards small stock (in general, $\beta_2 > 0$ means that returns of asset i behaves like small stocks).
 - β_3 measures how tilted asset i is towards value stock (in general, $\beta_3 > 0$ means that returns of asset i behave like high book-to-market stocks).

The 3-Factor F-F Model: Expected Returns

- Like the CAPM, the 3-factor FF model also produces an expected excess return for asset i :

$$E[r_{i,t} - r_f] = \beta_1 E[r_{m,t} - r_f] + \beta_2 E[SMB_t] + \beta_3 E[HML_t]$$

- A significant constant, α_i , in the linear 3-factor FF DGP is evidence against this model: something is missing. Then, we test the FF model for asset i :

$$H_0 \text{ (3-factor FF model true)} = \alpha_i$$

$$H_1 \text{ (3-factor FF model not true)} = \alpha_i$$

Using asset i 's time series, this test can be implemented by a simple t-test on the estimated constant, α_i .

- A cross-section test of the 3-factor FF model is a **joint test**:

$$H_0: \alpha_1 = \alpha_2 = \dots = \alpha_N = 0$$

$$H_1: \text{at least one } \alpha_i \neq 0.$$

OLS Estimation – The 3-factor F-F Model: IBM

Example (continuation): Using the time-series, we test the significance of the 3-factor Fama-French model for IBM returns:

```
SFX_da <-
read.csv("http://www.bauer.uh.edu/rsusmel/4397/Stocks_FX_1973.csv",head=TRUE,
sep=",")
x_ibm <- SFX_da$IBM           # Extract IBM price data
x_Mkt_RF <- SFX_da$Mkt_RF     # Extract Market excess returns (in %)
x_SMB <- SFX_da$SMB           # Extract SMB factor returns (in %)
x_HML <- SFX_da$HML           # Extract HML factor returns (in %)
x_RF <- SFX_da$RF             # Extract Risk-free rate factor returns (in %)

Mkt_RF <- x_Mkt_RF[-1]/100     # Adjust size (take one observation out )
SMB <- x_SMB[-1]/100          # Adjust size (take one observation out )
HML <- x_HML[-1]/100          # Adjust size (take one observation out )
RF <- x_RF[-1]/100
```

OLS Estimation – The 3-factor F-F Model: IBM

Example (continuation):

```
T <- length(x_ibm)           # Sample size
lr_ibm <- log(x_ibm[-1]/x_ibm[-T]) # IBM log returns (lost one observation)
ibm_x <- lr_ibm - RF
```

```
fit_ibm_ff3 <- lm(ibm_x ~ Mkt_RF + SMB + HML)
> summary(fit_ibm_ff3)       # print lm results
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-0.005089	0.002488	-2.046	0.0412 *	
Mkt_RF	0.908299	0.056722	16.013	<2e-16 ***	
SMB	-0.212460	0.084112	-2.526	0.0118 *	⇒ significant at 5% level.
HML	-0.171500	0.084682	-2.025	0.0433 *	⇒ significant at 5% level.

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.05848 on 565 degrees of freedom

OLS Estimation – The 3-factor F-F Model: IBM

Example (continuation):

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-0.005089	0.002488	-2.046	0.0412 *	
Mkt_RF	0.908299	0.056722	16.013	<2e-16 ***	
SMB	-0.212460	0.084112	-2.526	0.0118 *	⇒ significant at 5% level.
HML	-0.171500	0.084682	-2.025	0.0433 *	⇒ significant at 5% level.

Conclusion: Consistent with the 3-factor Fama-French model, Mkt_RF, SMB and HML are drivers of the expected returns for IBM. The signs of β_2 & β_3 : IBM behaves like a large & low B/M firm.

Note 1: The constant is significant, that is, there is an “extra” component of expected returns not explained by the 3 F-F factors.

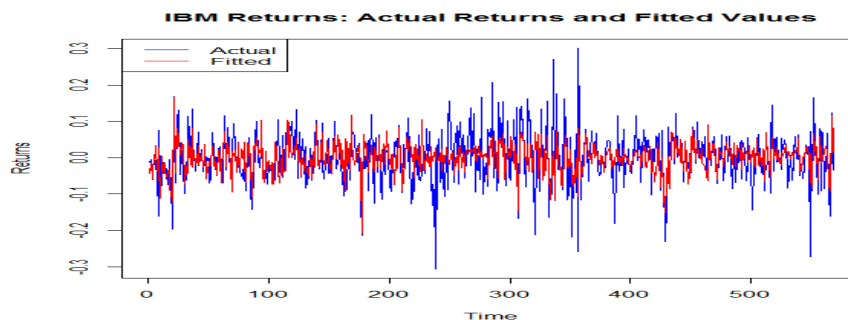
Note 2: The CAPM is also rejected for IBM, since there are, beyond the market factor, other significant factors.

OLS Estimation – The 3-factor F-F Model: IBM

```

y_hat <- fit_ibm_ff3$fitted.values          # Extract from fit_ibm_ff3 fitted values
plot(ibm_x, type = "l", col = "blue",      # Plot IBM returns
     main = "IBM Returns: Actual Returns and Fitted Values", ylab = "Returns", xlab = "Time")
lines(y_hat, type = "l", col = "red")      # Add fitted values to plot
legend("topleft",                          # Add legend to plot
     legend = c("Actual", "Fitted"),      col = c("blue", "red"), lty = 1)

```



⇒ Some periods with good fit –early & late periods- & some periods with poor fit –middle period.

OLS Estimation – Is IBM's Beta equal to 1?

Example: Using the estimated 3-factor F-F model for IBM returns, we test if IBM's market $\beta = 1$, that is, if IBM bears the same market risk as the market. Using the previous estimation:

```
fit_ibm_ff3 <- lm(ibm_x ~ Mkt_RF + SMB + HML)
b_ibm <- fit_ibm_ff3$coefficients      # Extract from lm function OLS coefficients
SE_ibm <- sqrt(diag(vcov(fit_ibm_ff3))) # SE from fit_ibm (also a kx1 vector)
t_beta1 <- (b_ibm[2] - 1)/SE_ibm[2]    # t-stat for H0: Beta1 - 1

> t_beta1
[1] -1.616674

p_val <- (1 - pnorm(abs(t_beta1))) * 2  # pvalue for t_beta (adjusted b/c two sided test)
> p_val
[1] 0.1059487    ⇒ Cannot reject H0 at 5% level
```

OLS Estimation – Is IBM's Beta equal to 1?

Example (continuation):

- Q: Is the market beta (β_1) equal to 1? That is,

$$H_0: \beta_1 = 1 \text{ vs.}$$

$$H_1: \beta_1 \neq 1$$

$$\text{Compute } t_k = \frac{b_k - \beta_k^0}{\text{Est. SE}[b_k]} \Rightarrow \hat{t}_1 = \frac{0.9082989 - 1}{0.05672206} = -1.6167$$

$$\Rightarrow |\hat{t}_1| < 1.96 \Rightarrow \text{Cannot reject } H_0 \text{ at 5\% level}$$

Conclusions: IBM has a one-to-one risk relation with the market.

The 3-Factor F-F Model: Two-step Test

- An alternative test can be based on the theoretical implications that assets with higher exposure to factor risk –i.e., higher β_i 's – should have higher returns.

3-factor FF DGP (in the cross-section):

$$(r_i - r_f) = \alpha + \beta_1 \lambda_1 + \beta_2 \lambda_2 + \beta_i \lambda_3 + \varepsilon_i, \quad i = 1, \dots, N$$

$$H_0 \text{ (FF holds in the CS): } \alpha = 0 \text{ \& } \lambda_j > 0, \quad j = 1, 2, k = 3$$

$$H_1 \text{ (FF rejected in the CS): } \alpha \neq 0 \text{ and/or}$$

Note: In equilibrium, $\lambda_1 = E[r_{m,t} - r_f]$, $\lambda_2 = E[SMB_t]$ & $\lambda_3 = E[HML_t]$.

- Again, a Fama-MacBeth (1973) two-step procedure is usually done:

1) First pass: Using the time series, estimate N betas (b_1, b_2, b_3).

2) Second pass: Using the cross section of returns, regress mean excess returns against a constant and the estimated betas.

The 3-Factor F-F Model: Two-step Test

- An alternative test can be based on the theoretical implications that assets with higher exposure to factor risk –i.e., higher β_i 's – should have higher returns.

3-factor FF DGP (in the cross-section):

$$(r_i - r_f) = \alpha + \beta_1 \lambda_1 + \beta_2 \lambda_2 + \beta_i \lambda_3 + \varepsilon_i, \quad i = 1, \dots, N$$

$$H_0 \text{ (FF holds in the CS): } \alpha = 0 \text{ \& } \lambda_j > 0, \quad j = 1, 2, k = 3$$

$$H_1 \text{ (FF rejected in the CS): } \alpha \neq 0 \text{ and/or } \lambda_j \leq 0,$$

Note: In equilibrium, $\lambda_1 = E[r_{m,t} - r_f]$, $\lambda_2 = E[SMB_t]$ & $\lambda_3 = E[HML_t]$.

- Again, a Fama-MacBeth (1973) two-step procedure is usually done:

1) First pass: Using the time series, estimate N betas (b_1, b_2, b_3).

2) Second pass: Using the cross section of returns, regress mean excess returns against a constant and the estimated betas.

The 3-Factor F-F Model: Testing Model (CS)

Example: We test the 3-factor Fama-French model, in the cross-section, using the 2-step Fama-McBeth method. We use returns of **25 Fama-French portfolios** (sorted by Size & BM), downloaded, along the 3-Fama-French factors from Ken French's website.

```
FF_p_da <- read.csv("https://www.bauer.uh.edu/rsusmel/4397/FF_25_portfolios.csv",
head=TRUE, sep=",")
FF_f_da <- read.csv("https://www.bauer.uh.edu/rsusmel/4397/FF_3_factors.csv", head=TRUE,
sep=",")

# Extract variables from imported data
Mkt_RF_fm <- FF_f_da$Mkt_RF      # extract Market excess returns (in %)
HML_fm <- FF_f_da$HML           # extract HML returns (in %)
SMB_fm <- FF_f_da$SMB           # extract HML returns (in %)
RF_fm <- FF_f_da$RF             # extract Risk-free rate (in %)
Y_p <- FF_p_da[,2:26] - RF_fm    # Compute excess returns of 25 portfolios

T <- length(HML_fm)             # Number of observations (1926:July on)
x0 <- matrix(1,T,1)             # Vector of ones, represents constant in X
x <- cbind(x0, Mkt_RF, SMB, HML) # Regressors (vector of ones + 3 Factors)
```

OLS Estimation – Testing the CAPM (CS)

Example (continuation):

```
k <- ncol(Y_p)

## First Pass
Allbs = NULL                      # Initialize empty (a space to put betas)
for (i in seq(1,k,1)){
  y <- Y_p[,i]  # select Y (portfolio)
  b <- solve(t(x)%*% x)%*% t(x)%*% y  # OLS regression = (X'X)^(-1) X'y
  Allbs = cbind(Allbs,b)          # accumulate b as rows
}

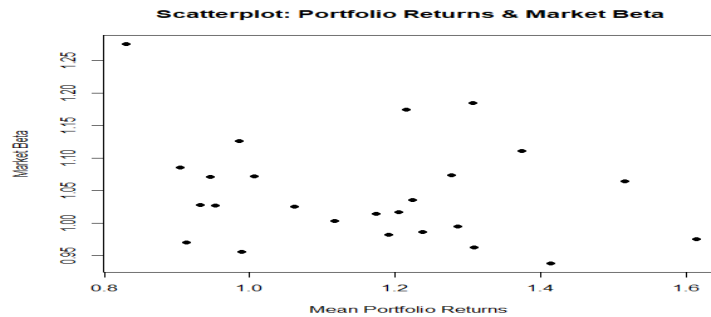
beta_ret <- cbind(colMeans(Y_p),t(Allbs)) # Mean portfolio returns along alpha & beta estimates
> cor(beta_ret)
```

			Mkt_RF	SMB	HML
	1.000000	0.5242033	-0.2502789	0.26183205	0.69403377
	0.524203	1.0000000	-0.7533964	-0.3289295	-0.02888856
Mkt_RF	-0.250279	-0.7533967	1.0000000	0.1747401	0.11693857
SMB	0.261832	-0.328929	0.1747402	1.0000000	0.03922282
HML	0.694034	-0.028888	0.1169386	0.0392228	1.00000000

OLS Estimation – Testing the CAPM (CS)

Example (continuation):

```
plot(beta_ret[,1], beta_ret[,3], main="Scatterplot: Portfolio Returns & Market Beta",
     xlab="Mean Portfolio Returns ", ylab="Market Beta", pch=19)
```



OLS Estimation – Testing the CAPM (CS)

Example (continuation):

Second Pass (CAPM)

```
fit_fm_ff3_25 <- lm(beta_ret[,1] ~ beta_ret[, 3:5])
```

```
> summary(fit_fm_ff3_25)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	1.98661	0.33046	6.012	5.76e-06 ***	⇒ Significant!
beta_ret[, 3:5]Mkt_RF	-0.98579	0.32011	-3.080	0.00568 **	⇒ Negative! & significant
beta_ret[, 3:5]SMB	0.11140	0.04633	2.405	0.02550 *	⇒ Positive as expected
beta_ret[, 3:5]HML	0.37471	0.06393	5.861	8.11e-06 ***	⇒ Positive as expected

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Conclusion: The 3 factor FF model has a significant constant, which strong evidence against the model –i.e., something is missing. The negative coefficient of the market beta goes against theory. Question: Is Beta dead or the model is incorrect?

The 3-Factor F-F Model: Remarks

- Questions:

- How were these factors determined to be drivers of stock returns?
By looking at data characteristics, not theory.

As seen in the previous graphs, by the mid-1990s there was evidence that small firms outperformed big firms and that high BM firms outperformed low BM firms. Thus, data mining issues are likely present.

- Are these 3 (or 5) factors the definitive number of factors?

No. There have been over 200 factors proposed! Big number, likely due to data mining. Feng, Giglio and Xiu (2020), who try to propose a method to select factors, call their paper “Taming the **Factor Zoo**.”

The 5-Factor F-F Model: RMW & CMA

- In 2014, Fama and French added two additional factors to their 3-factor model: RMW & CMA.

- RMW measures the return of the portfolio of most profitable firms (“robust”) minus the returns of a portfolio least profitable (“weak”).

- CMA measures the return of a portfolio of firms that invest “conservatively” minus a portfolio of firms that invest “aggressively”.

- Again, the 5-factor FF model produces expected excess returns:

$$E[r_{i,t} - r_f] = \beta_1 E[r_{m,t} - r_f] + \beta_2 E[SMB_t] + \beta_3 E[HML_t] + \beta_4 E[RMW_t] + \beta_5 E[CMA_t]$$

- There is debate regarding the validity of this extension, especially, outside the U.S. market.

OLS Estimation – Linear Algebra Interpretation

- Disturbances and Residuals

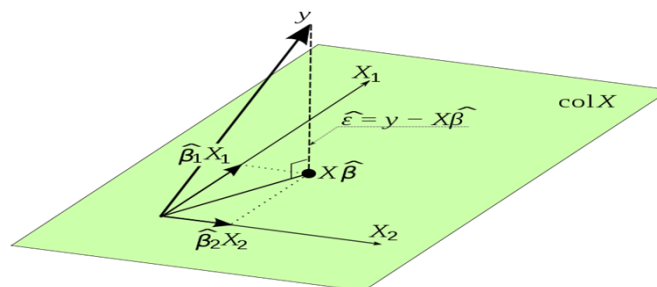
In the population: $E[\mathbf{X}' \boldsymbol{\varepsilon}] = 0$.

In the sample: $\mathbf{X}'\mathbf{e} = \mathbf{X}'(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}) = \mathbf{X}'\mathbf{y} - \mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = 0$.

- We have two ways to look at \mathbf{y} :

$\mathbf{y} = E[\mathbf{y} | \mathbf{X}] + \boldsymbol{\varepsilon} = \text{Conditional mean} + \text{disturbance}$

$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e} = \text{Projection ("fitted values")} + \text{residual}$



Results when X Contains a Constant Term

- Let the first column of \mathbf{X} be a column of ones ($\mathbf{x}_1 = \mathbf{i}$). That is,

$$\mathbf{X} = \begin{bmatrix} 1 & x_{21} & \cdots & x_{k1} \\ 1 & x_{22} & \cdots & x_{k2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{2T} & \cdots & x_{kT} \end{bmatrix}$$

- Recall $\mathbf{i}'\mathbf{z} = \sum_i^T z_i$, where \mathbf{z} and \mathbf{i} are $T \times 1$ vectors. Then,

(1) Residuals sum to zero.

Since $\mathbf{X}'\mathbf{e} = \mathbf{0}$

$$= \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_{21} & x_{22} & \cdots & x_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k1} & x_{k2} & \cdots & x_{kT} \end{bmatrix} * \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_T \end{bmatrix} = \mathbf{0} \quad \Rightarrow \sum_i^T e_i = 0$$

$$\Rightarrow \mathbf{x}_1'\mathbf{e} = \mathbf{i}'\mathbf{e} = \sum_i^T e_i = 0 \quad \text{--the residuals sum to zero.}$$

Results when X Contains a Constant Term

(2) Regression line passes through the means

Recall we can write \mathbf{y} = fitted values + residuals

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e}$$

After some algebra, it can be shown that:

$$\bar{y} = \bar{\mathbf{x}}'\mathbf{b}$$

- That is, the regression line passes through the means.

Goodness of Fit of the Regression

- After estimating the model (**A1**), we would like to judge the adequacy of the model. There are two ways to do this:
 - Visual: Plots of fitted values and residuals, histograms of residuals.
 - Numerical measures: R^2 , Adjusted R^2 , AIC, BIC, etc.

- Numerical measures. In general, they are simple and easy to compute. We call them *goodness-of-fit* measures. Most popular: R^2 .

- Definition: Variation

In the context of a model, we consider the *variation* of a variable as the movement of the variable, usually associated with movement of another variable.

Goodness of Fit of the Regression

- Total variation = Total sum of squares (TSS) = $\sum_i (y_i - \bar{y})^2$.

We want to decompose TSS in two parts: one explained by the regression and one unexplained by the regression.

$$\begin{aligned} \text{TSS} &= \sum_i (y_i - \bar{y})^2 = \sum_i (y_i - \hat{y}_i + \hat{y}_i - \bar{y})^2 \\ &= \sum_i (y_i - \hat{y}_i)^2 + \sum_i (\hat{y}_i - \bar{y})^2 + 2 \sum_i (y_i - \hat{y}_i)(\hat{y}_i - \bar{y}) \\ &= \sum_i e_i^2 + \sum_i (\hat{y}_i - \bar{y})^2 \end{aligned}$$

$$\text{since } \sum_i (y_i - \hat{y}_i)(\hat{y}_i - \bar{y}) = \sum_i e_i (\hat{y}_i - \bar{y}) = 0$$

$$\text{Or } \text{TSS} = \text{RSS} + \text{SSR}$$

RSS: Residual Sum of Squares (also called SSE: SS of errors)

SSR: Regression Sum of Squares (also called ESS: *explained* SS)

A Goodness of Fit Measure

- $\text{TSS} = \text{SSR} + \text{RSS}$

- We want to have a measure that describes the fit of a regression.
Simplest measure: the standard error of the regression (SER)

$$\text{SER} = \sqrt{\text{RSS}/(T - k)} \Rightarrow \text{SER depends on units. Not good!}$$

- R-squared (R^2)

$$1 = \text{SSR}/\text{TSS} + \text{RSS}/\text{TSS}$$

$$R^2 = \text{SSR}/\text{TSS} = \text{Regression variation}/\text{Total variation}$$

$$R^2 = 1 - \text{RSS}/\text{TSS}$$

As introduced here, R^2 lies between 0 and 1 (& it is independent of units of measurement!). It measures how much of total variation (TSS) is explained by regression (SSR): the higher R^2 , the better.

A Goodness of Fit Measure

- $R^2 = SSR/TSS$

Interpretation: The percentage of total variation (TSS) explained by the variation of regressors.

Note: R^2 is bounded by zero and one only if:

- (a) There is a constant term in \mathbf{X} .
- (b) The line is computed by OLS.

- Main problem with R^2 : Adding regressors

It can be shown that R^2 never falls when regressors (say \mathbf{z}) are added to the regression. This occurs because RSS decreases with more “information” (in the sense of more regressors).

Problem: Judging a model based on R^2 tends to over-fitting.

A Goodness of Fit Measure

- Comparing Regressions

- Make sure the denominator in R^2 is the same - i.e., same left hand side variable. For example, when modeling sales, it is common to use $\log(\text{Sales})$. Cannot compare R^2 to the one with Sales. Loglinear will almost always appear to fit better, taking logs reduces variation.

- Linear Transformation of data does not change R^2 .

- Based on \mathbf{X} , $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$.

Suppose we work with $\mathbf{X}^* = c\mathbf{X}$, instead (c is a constant).

$$\begin{aligned}\hat{\mathbf{y}}^* &= \mathbf{X}^* \mathbf{b}^* = c\mathbf{X} (c\mathbf{X}' c\mathbf{X})^{-1} c\mathbf{X}' \mathbf{y} \\ &= c\mathbf{X} (c^2 \mathbf{X}'\mathbf{X})^{-1} c\mathbf{X}' \mathbf{y} \\ &= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}' \mathbf{y} = \mathbf{X}\mathbf{b} = \hat{\mathbf{y}}\end{aligned}$$

\Rightarrow same fit, same residuals, same R^2 !

Adjusted R-squared

- R^2 is modified with a penalty for number of parameters: *Adjusted- R^2*

$$\bar{R}^2 = 1 - \frac{(T-1)}{(T-k)} (1 - R^2) = 1 - \frac{(T-1)}{(T-k)} \frac{\text{RSS}}{\text{TSS}} = 1 - \frac{s^2}{\text{TSS}/(T-1)}$$

$$\Rightarrow \text{maximizing } \bar{R}^2 \Leftrightarrow \text{minimizing } [\text{RSS}/(T - k)] = s^2$$

- *Degrees of freedom* –i.e., $(T - k)$ – adjustment assumes something about “unbiasedness.”
- \bar{R}^2 includes a penalty for variables that do not add much fit. Can fall when a variable is added to the equation.
- It will rise when a variable, say \mathbf{z} , is added to the regression if and only if the *t-ratio* on \mathbf{z} is larger than one in absolute value.

Adjusted R-squared

- Theil (1957) shows that, under certain assumptions (an important one: the true model is being considered), if we consider several linear models:

$$M_1: \mathbf{y} = \mathbf{X}_1\boldsymbol{\beta}_1 + \varepsilon_1 \quad \text{- true model}$$

$$M_2: \mathbf{y} = \mathbf{X}_2\boldsymbol{\beta}_2 + \varepsilon_2$$

$$M_3: \mathbf{y} = \mathbf{X}_3\boldsymbol{\beta}_3 + \varepsilon_3$$

& choose the model with smaller s^2 (or, larger Adjusted R^2), we select the true model, M_1 , on average.

- In this sense, we say that “maximizing Adjusted R^2 ” is an *unbiased* model-selection criterion.

Other Goodness of Fit Measures

- There are other goodness-of-fit measures that also incorporate penalties for number of parameters (degrees of freedom). We minimize these measures.

- Information Criteria (IC)

- *Amemiya*: $[\mathbf{e}'\mathbf{e}/(T-k)] * (1 + k/T) = s^2 * (1 + k/T)$

- *Akaike Information Criterion* (AIC)

$$\text{AIC} = -2/T(\ln L - k) \quad L: \text{Likelihood}$$

$$\Rightarrow \text{if normality } \text{AIC} = \ln(\mathbf{e}'\mathbf{e}/T) + (2/T)k \quad (+\text{constants})$$

- *Bayes-Schwarz Information Criterion* (BIC)

$$\text{BIC} = -(2/T \ln L - [\ln(T)/T]k)$$

$$\Rightarrow \text{if normality } \text{AIC} = \ln(\mathbf{e}'\mathbf{e}/T) + [\ln(T)/T]k \quad (+\text{constants})$$

Goodness of Fit Measures – Example

Example: 3 Factor F-F Model (continuation) for IBM returns:

```

b <- solve(t(x)%*% x)%*% t(x)%*% y          # b = (X'X)-1X'y (OLS regression)
e <- y - x%*%b                               # regression residuals, e
k <- ncol(x)                                  # Number of parameters estimated
RSS <- as.numeric(t(e)%*%e)                  # RSS
R2 <- 1 - as.numeric(RSS)/as.numeric(t(y)%*%y) # R-squared w/ TSS approximation
Adj_R2 <- 1 - (T-1)/(T-k)*(1-R2)              # Adjusted R-squared
AIC <- log(RSS/T) + 2*k/T                    # AIC under N(.,) –i.e., under (A5)

> R2
[1] 0.338985      ⇒ The 3 F-F factors explain 34% of the variability of IBM returns.
> Adj_R2
[1] 0.3354752
> AIC
[1] -5.671036

```

Maximum Likelihood Estimation (MLE)

- Idea: Assume a particular distribution with unknown parameters. Maximum likelihood (ML) estimation chooses the set of parameters that maximize the likelihood of drawing a particular sample.

- Suppose we have a sample with N realizations of a coin flip. The coin was flipped $N=100$ times: 60 heads (H) & 40 tails (T).

The RV (X = number of H & T from N coin flips) follows a binomial distribution, with parameter p , the probability of a head. MLE estimates p as the probability that maximizes what we observed in our particular sample. In our case, MLE set $p = 0.60$.

- Formally speaking, we create a function that describes the probability of the RV $X = 60$: $L(X = 60|p) = \binom{N}{x} p^x (1 - p)^{N-x}$
Then, we maximize $L(X = 60|p)$ with respect to p .

Maximum Likelihood Estimation (MLE)

- More general, consider a sample (X_1, X_2, \dots, X_N) which is drawn from a pdf $f(\mathbf{X}|\theta)$ where θ are k parameters. Then, each X_i 's has a pdf $f(X_i|\theta)$.

If the X_i 's are *independent* with $f(X_i|\theta)$, the joint pdf for the whole sample (X_1, X_2, \dots, X_N) is:

$$\begin{aligned} L(\mathbf{X}|\theta) &= f(X_1, X_2, \dots, X_N|\theta) = f(X_1|\theta) * f(X_2|\theta) * \dots * f(X_N|\theta) \\ &= \prod_{i=1}^N f(X_i|\theta) \end{aligned}$$

The function $L(\mathbf{X}|\theta)$ is called the *likelihood function*. It represents how likely it is to get a particular sample from the model.

This function $L(\mathbf{X}|\theta)$ can be maximized with respect to θ to produce maximum likelihood estimates: $\hat{\theta}_{MLE}$.

Maximum Likelihood Estimation (MLE)

- Assuming the X_i 's are *independent* with $f(X_i|\theta)$, the joint pdf is:

$$L(\mathbf{X}|\theta) = f(X_1, X_2, \dots, X_N|\theta) = f(X_1|\theta) * f(X_2|\theta) * \dots * f(X_N|\theta) \\ = \prod_{i=1}^N f(X_i|\theta)$$

This function $L(\mathbf{X}|\theta)$ can be maximized with respect to θ to produce maximum likelihood estimates: $\hat{\theta}_{MLE}$.

It is often easier to work with the *Log of the likelihood* function. That is,

$$\ln L(\mathbf{X}|\theta) = \sum_{i=1}^N \ln f(X_i|\theta)$$

Then, we maximize as usual:

$$\text{1st-derivative} \Rightarrow \frac{\partial \ln L(\mathbf{X}|\theta)}{\partial \theta} = \sum_{i=1}^N \frac{\partial \ln f(X_i|\theta)}{\partial \theta}$$

$$\text{f.o.c.} \Rightarrow \frac{\partial \ln L(\mathbf{X}|\hat{\theta}_{MLE})}{\partial \theta} = 0 \Rightarrow \hat{\theta}_{MLE}.$$

MLE: Example I

Let the sample be $\mathbf{X} = \{5, 6, 7, 8, 9, 10\}$ drawn from a $\text{Normal}(\mu, 1)$. The probability of each of these points based on the unknown mean, μ , can be written as:

$$f(5|\mu) = \frac{1}{\sqrt{2\pi}} \exp \left[-\frac{(5-\mu)^2}{2} \right] \\ f(6|\mu) = \frac{1}{\sqrt{2\pi}} \exp \left[-\frac{(6-\mu)^2}{2} \right] \\ \vdots \\ f(10|\mu) = \frac{1}{\sqrt{2\pi}} \exp \left[-\frac{(10-\mu)^2}{2} \right]$$

Assume that the sample is *independent*. Then, the joint pdf is given by:

$$L(\mathbf{X}|\mu) = f(5|\mu) * f(6|\mu) * \dots * f(10|\mu)$$

MLE: Example I

Then, the joint pdf function can be written as:

$$L(X|\mu) = \frac{1}{(2\pi)^{6/2}} \exp \left[-\frac{(5-\mu)^2}{2} - \frac{(6-\mu)^2}{2} - \dots - \frac{(10-\mu)^2}{2} \right]$$

The value of μ that maximizes the likelihood function of the sample can then be defined by $\max_{\mu} L(X|\mu)$.

It easier to maximize the *Log likelihood*, $\ln L(X|\mu)$:

$$\max_{\mu} \ln(L(X|\mu)) = -6/2 \ln(2\pi) + \left[-\frac{(5-\mu)^2}{2} - \frac{(6-\mu)^2}{2} - \dots - \frac{(10-\mu)^2}{2} \right]$$

$$1^{\text{st}}\text{-derivative} \Rightarrow \frac{\partial}{\partial \mu} \left[K - \frac{(5-\mu)^2}{2} - \frac{(6-\mu)^2}{2} - \dots - \frac{(10-\mu)^2}{2} \right]$$

$$f.o.c. \Rightarrow (5 - \hat{\mu}_{MLE}) + (6 - \hat{\mu}_{MLE}) + \dots + (10 - \hat{\mu}_{MLE}) = 0$$

MLE: Example I

Then, the first order conditions:

$$(5 - \hat{\mu}_{MLE}) + (6 - \hat{\mu}_{MLE}) + \dots + (10 - \hat{\mu}_{MLE}) = 0$$

Solving for $\hat{\mu}_{MLE}$:

$$\hat{\mu}_{MLE} = \frac{5 + 6 + 7 + 8 + 9 + 10}{6} = 7.5 = \bar{x}$$

That is, the MLE estimator $\hat{\mu}_{MLE}$ is equal to the sample mean. This is good for the sample mean: MLE has very good properties!

MLE: Remarks

- ML estimation approach is general. All we need is to assume that the data we have follow a distribution. In our CLM context, we need a model (say, **A1**) and a pdf for the errors (say, normal) to use MLE. MLE picks the betas that maximize the likelihood.

Remark: We like MLE because its estimators, $\hat{\theta}_{MLE}$, have very good properties.

- A lot of applications in finance and economics: Time series, volatility (GARCH and stochastic volatility) models, factor models of the term structure, switching models, option pricing, logistic models (mergers and acquisitions, default, etc.), trading models, etc.
- In general, we rely on numerical optimization to get MLEs.

Ronald A. Fisher, England (1890 – 1962)



MLE: Example I – Code in R

Example (continuation):

```
negative_likelihood_n <- function(mu){ # R uses a minimization algorithm, change sign
  sum(log(dnorm(x_6, mu, sd=1)))) * (-1)
}
> negative_likelihood_n(mu)
[1] 183.0136

# Step 2 - Maximize (or Minimize negative Likelihood function)
results_n <- nlm(negative_likelihood_n, mu, stepmax=2) # nlm minimizes the function
> results_n # Show nlm results
$minimum
[1] 14.26363 <= The minimized value of function (-14.26363 is the max)
$estimate
[1] 7.5 <= The MLE for  $\mu$  ( $=\hat{\mu}_{MLE}$ ).
$gradient
[1] -4.736952e-12 <= Should be very close to zero if we're at a minimum
```

MLE: Example I – Code in R

Example (continuation):

```
mu_max <- results_n$estimate      # Extract estimates
> mu_max                        # Should be equal to mean
[1] 7.5
> likelihood_n(mu_max)           # Check max value at mu_max
[1] -14.26363
```