Econ 424
Descriptive Statistics for Financial Time Series

Eric Zivot

January 26, 2015

Covariance Stationarity

$$\{\ldots, X_1, \ldots, X_T, \ldots\} = \{X_t\}$$

is a covariance stationary stochastic process, and each X_t is identically distributed with unknown pdf $f(x)$.

Recall,

$$\begin{aligned} E[X_t] &= \mu \text{ indep of } t \\ \text{var}(X_t) &= \sigma^2 \text{ indep of } t \\ \text{cov}(X_t, X_{t-j}) &= \gamma_j \text{ indep of } t \\ \text{cor}(X_t, X_{t-j}) &= \rho_j \text{ indep of } t \end{aligned}$$

Observed Sample:

$$\{X_1 = x_1, \ldots, X_T = x_T\} = \{x_t\}_{t=1}^{T}$$

are observations generated by the stochastic process

Descriptive Statistics

Data summaries (statistics) to describe certain features of the data, to learn about the unknown pdf, $f(x)$, and to capture the observed dependencies in the data

Time Plots

Line plot of time series data with time/dates on horizontal axis

- Visualization of data - uncover trends, assess stationarity and time dependence
- Spot unusual behavior
- Plotting multiple time series can reveal commonality across series

Histograms

Goal: Describe the shape of the distribution of the data $\{x_t\}_{t=1}^{T}$

Hisogram Construction:

1. Order data from smallest to largest values

2. Divide range into N equally spaced bins

$$[-|-|-|\cdots|-|-|-]$$

3. Count number of observations in each bin

4. Create bar chart (optionally normalize area to equal 1)

R Functions

Function	Description
`hist()`	compute histogram
`density()`	compute smoothed histogram

Note: The `density()` function computes a smoothed (kernel density) estimate of the unknown pdf at the point x using the formula

$$\hat{f}(x) = \frac{1}{Tb}\sum_{t=1}^{T} k\left(\frac{x - x_t}{b}\right)$$
$$k(\cdot) = \text{kernel function}$$
$$b = \text{ bandwidth (smoothing) parameter}$$

where $k(\cdot)$ is a pdf symmetric about zero (typically the standard normal distribution). See Ruppert Chapter 4 for details.

Empirical Quantiles/Percentiles

Percentiles:

For $\alpha \in [0, 1]$, the $100 \times \alpha^{th}$ percentile (empirical quantile) of a sample of data is the data value $\hat{q}_\alpha$ such that $\alpha \cdot 100\%$ of the data are less than $\hat{q}_\alpha$.

Quartiles

$$\begin{aligned} \hat{q}_{.25} &= \text{ first quartile} \\ \hat{q}_{.50} &= \text{second quartile (median)} \\ \hat{q}_{.75} &= \text{third quartile} \\ \hat{q}_{.75} - \hat{q}_{.25} &= \text{interquartile range (IQR)} \end{aligned}$$

R functions

Function	Description
`sort()`	sort elements of data vector
`min()`	compute minimum value of data vector
`max()`	compute maximum value of data vector
`range()`	compute **min** and **max** of a data vector
`quantile()`	compute empirical quantiles
`median()`	compute median
`IQR()`	compute inter-quartile range
`summary()`	compute summary statistics

Historical Value-at-Risk

Let $\{R_t\}_{t=1}^T$ denote a sample of T simple monthly returns on an investment, and let $\$W_0$ be the initial value of an investment. For $\alpha \in (0,1)$, the historical VaR_α is

$$\begin{aligned} &\$W_0 \times \hat{q}_\alpha^R \\ \hat{q}_\alpha^R &= \text{empirical } \alpha \cdot 100\% \text{ quantile of } \{R_t\}_{t=1}^T \end{aligned}$$

Note: For continuously compounded returns $\{r_t\}_{t=1}^T$ use

$$\begin{aligned} &\$W_0 \times (\exp(\hat{q}_\alpha^r) - 1) \\ \hat{q}_\alpha^r &= \text{empirical } \alpha \cdot 100\% \text{ quantile of } \{r_t\}_{t=1}^T \end{aligned}$$

Sample Statistics

Plug-In Principle: Estimate population quantities using sample statistics

Sample Average (Mean)

$$\frac{1}{T}\sum_{t=1}^{T} x_t = \bar{x} = \hat{\mu}_x$$

Sample Variance

$$\frac{1}{T-1}\sum_{t=1}^{T}(x_t - \bar{x})^2 = s_x^2 = \hat{\sigma}_x^2$$

Sample Standard Deviation

$$\sqrt{s_x^2} = s_x = \hat{\sigma}_x$$

Sample Skewness

$$\frac{1}{T-1}\sum_{t=1}^{T}(x_t-\bar{x})^3/s_x^3=\widehat{skew}$$

Sample Kurtosis

$$\frac{1}{T-1}\sum_{t=1}^{T}(x_t-\bar{x})^4/s_x^4=\widehat{kurt}$$

Sample Excess Kurtosis

$$\widehat{kurt}-3$$

R Functions

Function	Package	Description
`mean()`	base	compute sample mean
`colMeans()`	base	compute column means of matrix
`var()`	stats	compute sample variance
`sd()`	stats	compute sample standard deviation
`skewness()`	PerformanceAnalytics	compute sample skewness
`kurtosis()`	PerformanceAnalytics	compute sample excess kurtosis

Note: Use the R function `apply()`, to apply functions over rows or columns of a matrix or data.frame

Empirical Cumulative Distribution Function

Recall, the CDF of a random variable X is

$$F_X(x) = \mathsf{Pr}(X \leq x)$$

The empirical CDF of a random sample is

$$\begin{aligned}\hat{F}_X(x) &= \frac{1}{n}(\# x_i \leq x) \\ &= \frac{\text{number of } x_i \text{ values } \leq x}{\text{sample size}}\end{aligned}$$

How to compute and plot $\hat{F}_X(x)$ for a sample $\{x_1, \ldots, x_n\}$

- Sort data from smallest to largest values: $\{x_{(1)}, \ldots, x_{(n)}\}$ and compute $\hat{F}_X(x)$ at these points

- Plot $\hat{F}_X(x)$ against sorted data $\{x_{(1)}, \ldots, x_{(n)}\}$

- Use the R function `ecdf()`

Note: $x_{(1)}, \ldots, x_{(n)}$ are called the *order statistics.* In particular, $x_{(1)} = \min(x_1, \ldots, x_n)$ and $x_{(n)} = \max(x_1, \ldots, x_n)$.

Quantile-Quantile (QQ) Plots

A QQ plot is useful for comparing your data with the quantiles of a distribution (usually the normal distribution) that you think is appropriate for your data. You interpret the QQ plot in the following way:

- If the points fall close to a straight line, your conjectured distribution is appropriate

- If the points do not fall close to a straight line, your conjectured distribution is not appropriate and you should consider a different distribution

R functions

Function	Package	Description
`qqnorm()`	stats	QQ-plot against normal distribution
`qqline()`	stats	draw straight line on QQ-plot
`qqPlot()`	car	QQ-plot against specified distribution

Outliers

- Extremely large or small values are called "outliers"

- Outliers can greatly influence the values of common descriptive statistics. In particular, the sample mean, variance, standard deviation, skewness and kurtosis

- Percentile measures are more robust to outliers: outliers do not greatly influence these measures (e.g. median instead of mean; IQR instead of SD)

IQR (interquartile range) - outlier robust measure of spread

$$IQR = q_{.75} - q_{.25}$$

Moderate Outlier

$$\hat{q}_{.75} + 1.5 \cdot IQR < x < \hat{q}_{.75} + 3 \cdot IQR$$
$$\hat{q}_{.25} - 3 \cdot IQR < x < \hat{q}_{.25} - 1.5 \cdot IQR$$

Extreme Outlier

$$x > \hat{q}_{.75} + 3 \cdot IQR$$
$$x < \hat{q}_{.25} - 3 \cdot IQR$$

Boxplots

A box plot displays the locations of the basic features of the distribution of one-dimensional data—the median, the upper and lower quartiles, outer fences that indicate the extent of your data beyond the quartiles, and outliers, if any.

R functions

Function	Package	Description
`boxplot()`	graphics	box plots for multiple series
`chart.Boxplot()`	PerformanceAnalytics	box plots for asset returns

Bivariate Descriptive Statistics

$$\{\ldots, (X_1, Y_1), (X_2, Y_2), \ldots (X_T, Y_T), \ldots\} = \{(X_t, Y_t)\}$$

covariance stationary bivariate stochastic process with realized values

$$\{(x_1, y_1), (x_2, y_2), \ldots (x_T, y_T)\} = \{(x_t, y_t)\}_{t=1}^{T}$$

Scatterplot

XY plot of bivariate data

R functions: `plot()`, `pairs()`

Sample Covariance

$$\frac{1}{T-1}\sum_{t=1}^{T}(x_t-\bar{x})(y_t-\bar{y}) = s_{xy} = \hat{\sigma}_{xy}$$

Sample Correlation

$$\frac{s_{xy}}{s_x s_y} = r_{xy} = \hat{\rho}_{xy}$$

R functions

Function	Package	Description
`var()`	stats	compute sample variance-covariance matrix
`cov()`	stats	compute sample variance-covariance matrix
`cor()`	stats	compute sample correlation matrix

Time Series Descriptive Statistics

Sample Autocovariance

$$\hat{\gamma}_j = \frac{1}{T-1}\sum_{t=j+1}^{T}(x_t - \bar{x})(x_{t-j} - \bar{x}),\ j = 1, 2, \ldots$$

Sample Autocorrelation

$$\hat{\rho}_j = \frac{\hat{\gamma}_j}{\hat{\sigma}^2},\ j = 1, 2, \ldots$$

Sample Autocorrelation Function (SACF)

Plot $\hat{\rho}_j$ against j

R functions

Function	Package	Description
`acf()`	stats	compute and plot sample autocorrelations
`chart.ACF()`	PerformanceAnalytics	plot sample autocorrelations
`chart.ACFplus()`	PerformanceAnalytics	plot sample autocorrelations