finsummarize — Financial summary statistics+

+This command is part of StataNow.

Description

`finsummarize` calculates six summary statistics for financial assets or portfolios: the mean, standard deviation, Sharpe ratio, Treynor index, Jensen's alpha, and the market beta. When a benchmark asset return is specified, `finsummarize` provides all statistics. Otherwise, the mean, standard deviation, and Sharpe ratio are provided.

Quick start

Summarize portfolio `myport` to obtain its mean, standard deviation, and Sharpe ratio

```
finsummarize myport
```

Same as above, but also use `mktrisk` as the return of the benchmark asset and the risk-free rate variable `rf`, and calculate the Treynor index, Jensen's alpha, and market beta

```
finsummarize myport, rfrate(rf) benchmark(mktrisk)
```

Same as above, but request only the Treynor index statistic

```
finsummarize myport, rfrate(rf) benchmark(mktrisk) statistics(treynor)
```

Menu

Statistics > Financial statistics > Financial summary statistics

Syntax

`finsummarize` *varlist* [*if*] [*in*] [*weight*] [, *options*]

options	Description
Main	
`rfrate(`*varname*`)`	specify variable containing the risk-free rate
`benchmark(`*varname*[, `adjust`]`)`	specify the benchmark asset return variable
`statistics(`*stats*`)`	specify a list of statistics; default is all available statistics
`detail`	request detailed summary statistics
* `nodfk`	do not adjust variance estimate for degrees of freedom
* `post`	post the results from option `detail` in `e()`
SE/Robust	
* `vce(`*vcetype*`)`	*vcetype* may be `robust`, `cluster` *clustvar*, or `hac` *hacspec*
Reporting	
`novarlabel`	display variable names rather than variable labels
* `coeflegend`	display legend instead of statistics

You must `tsset` your data before using `finsummarize`; see [TS] **tsset**.

varlist and *varname* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

`collect` is allowed; see [U] **11.1.10 Prefix commands**.

`aweight`s and `fweight`s are allowed; see [U] **11.1.6 weight**.

* `nodfk`, `post`, `vce()`, and `coeflegend` are meaningful only when the `detail` option is specified.

`coeflegend` does not appear in the dialog box.

stats	Description
`sharpe`	Sharpe ratio
`treynor`	Treynor index
`jensen`	Jensen's alpha
`beta`	market beta

Options

Main

rfrate(*varname*) specifies *varname* as the risk-free rate for computation of the Sharpe ratio, Treynor index, Jensen's alpha, and the market beta. If no variable is specified, finsummarize assumes the risk-free rate is equal to zero.

benchmark(*varname*[, adjust]) specifies *varname* as the benchmark asset return. A benchmark asset is required to compute the Treynor index, Jensen's alpha, and the market beta. If benchmark() is not specified, finsummarize will default to reporting only the mean, standard deviation, and Sharpe ratio. adjust specifies that excess returns of *varname* be used in computations (that is, returns in excess of the risk-free rate specified in rfrate()).

statistics(*stats*) specifies a list of statistics to be computed. *stats* may include sharpe, treynor, jensen, and beta. If statistics() is not specified, finsummarize reports all possible statistics.

detail produces standard error estimates in addition to point estimates. All asset summary statistics are estimated jointly, allowing for tests across performance measures and across assets.

nodfk specifies that a small-sample degrees-of-freedom adjustment not be made when estimating the variance of the returns. By default, a small-sample degrees-of-freedom adjustment is made. nodfk is applicable only when the detail option is specified.

post posts the results of the underlying regression run by the detail option as estimation results in e().

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (robust) and that allow for intragroup correlation (cluster *clustvar*); see [R] ***vce_option***. vce() is applicable only when the detail option is specified.

vce(hac *hacspec*) requests a heteroskedasticity- and autocorrelation-consistent (HAC) variance–covariance matrix. The full syntax of *hacspec* is one of the following:

vce(hac *kernel* [#]) requests a HAC variance–covariance matrix using the specified kernel (see below) with optional # lags. The bandwidth of a kernel is equal to # + 1. If # is not specified, a kernel with $N - 2$ lags is used, where N is the sample size.

vce(hac *kernel* opt [#]) requests a HAC variance–covariance matrix using the specified kernel (see below), and the lag order is selected using Newey and West's (1994) optimal lag-selection algorithm. # is an optional tuning parameter that affects the lag order selected; see the discussion in *Methods and formulas* in [R] **ivregress**.

kernel may be one of the following:

bartlett or nwest requests the Bartlett (Newey–West) kernel.

parzen or gallant requests the Parzen (Gallant 1987) kernel.

quadraticspectral or andrews requests the quadratic spectral (Andrews 1991) kernel.

Reporting

novarlabel specifies that the variable names be displayed instead of the variable labels. By default, variable labels are used to label assets in the output. Note that the variable names are displayed when the detail option is specified.

The following option is available with `finsummarize` but is not shown in the dialog box:

`coeflegend`; see [R] **Estimation options**. `coeflegend` is applicable only when the `detail` option is specified.

Remarks and examples

Financial assets and portfolios are summarized by the mean and variance of their returns, as well as simple functions of the mean and variance. `finsummarize` produces tables of these statistics.

The simplest characteristics of an asset or a portfolio's performance are its mean and standard deviation; the former captures average return and the latter captures return variability.

The Sharpe ratio is the return-to-risk ratio. If asset or portfolio i has mean return μ_i with standard deviation σ_i, and there exists a risk-free return r^f, then the Sharpe ratio is

$$S = \frac{\mu_i - r^f}{\sigma_i}$$

The numerator of this expression is the excess return of the asset. The denominator of this expression is the variability of the asset. Taken together, the Sharpe ratio expresses excess return in units of variability. A Sharpe ratio of zero indicates that the asset under consideration does not provide any excess return over the risk-free asset. A Sharpe ratio of one indicates that for every additional percentage point of variability, the asset returns one additional percentage point of return over the risk-free asset.

For a time-varying risk-free rate r_t^f, the Sharpe ratio is the average excess return divided by excess standard deviation. Thus, the Sharpe ratio becomes

$$S = \frac{\mu_i - \mu_r^f}{\sigma_e}$$

where μ_r^f is the average risk-free rate (so that $\mu_i - \mu_r^f$ is the average excess return) and σ_e is the standard deviation of excess returns, which are defined as $r_t - r_t^f$.

The market beta statistic and Jensen's alpha are closely related, because they are both parameters of a regression of excess returns to a given asset on excess returns to a benchmark asset. If asset i has return r_{it} in period t, the benchmark asset has return r_t^m, and the risk-free rate is r_t^f, then the regression

$$r_{it} - r_t^f = \alpha_i + \beta_i(r_t^m - r_t^f) + e_{it}$$

yields estimates of β_i and α_i. The statistic β_i, called the market beta, captures covariance of the asset with the benchmark, often termed systematic risk. The statistic α_i, called Jensen's alpha, measures average return in excess of market risk.

The formula above assumes that we have adjusted for the benchmark asset return by specifying the `adjust` suboption of the `benchmark()` option. For example, it is assumed that we typed

```
. finsummarize myport, rfrate(rf) benchmark(mktrisk, adjust)
```

However, without `adjust`, the regression above becomes

$$r_{it} - r_t^f = \alpha_i + \beta_i r_t^m + e_{it}$$

The Treynor index is similar to the Sharpe ratio in that it is a ratio of excess return to risk, but it uses β_i as its measure of risk:

$$T = \frac{\mu_i - r^f}{\beta_i}$$

Whereas the Sharpe ratio penalizes assets for their variability (σ_i), the Treynor index penalizes assets only for their covariance with the benchmark asset (β_i). According to the Treynor index, assets with high β_i are considered to be riskier.

▷ Example 1: Asset characteristics

We have data on 25 fictional companies, along with true data on the S&P 500 index and the federal funds rate. We describe the first few variables.

```
. use https://www.stata-press.com/data/r19/finex
(Fictional stock price data)

. describe datestr-wgt

Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
datestr         str11   %11s                  String date
datem           int     %tm                   Monthly date
sp500           double  %10.0g                S&P 500
vol             float   %9.0g                 Volatility index
fedfunds        float   %9.0g                 Federal funds rate
acme            float   %9.0g                 Aciron Medical, Inc.
bat             float   %9.0g                 Boron Advanced Technologies
iron            float   %9.0g                 Industrial Operations Network
dune            float   %9.0g                 Digital Urban Network Enterprise
tyr             float   %9.0g                 Tyndale Resources Group
glo             float   %9.0g                 Green Logistics, Inc.
spa             float   %9.0g                 Space Rocket MFG
wgt             float   %9.0g                 Widget Gadgets
```

Our dataset records prices, but portfolios are constructed on the basis of returns. We use `finreturns` to generate 25 returns series.

```
. quietly finreturns acme-tks, simple(r_) multiply(100)
```

The `simple(r_)` option in `finreturns` created simple, one-period returns for each of the assets in the variable list. The resulting returns are stored in variables named `r_acme` through `r_tks`. The `multiply(100)` option multiplied the resulting returns by 100, so that they can be interpreted as percentage changes.

From here, we can use `finsummarize` to display summary statistics for the returns of each asset.

```
. finsummarize r_acme-r_tyr
Financial summary statistics

Number of obs = 779
Sample: 1955m2 thru 2019m12
```

	Mean	Std. dev.	Sharpe ratio
ACME	0.4517	1.1856	0.3810
BAT	0.9052	5.4073	0.1674
IRON	1.0214	6.5625	0.1556
DUNE	1.0593	6.6088	0.1603
TYR	0.7028	3.9642	0.1773

The header provides information about the number of observations and the sample range. The data are monthly, so we have 779 observations from the second month of 1955 through the final month of 2019.

Because the data are monthly and we requested simple returns, the `Mean` column displays the average monthly return on the selected assets. The `Std. dev.` column displays the standard deviation of the returns. Because we did not specify a risk-free asset, the implied risk-free rate is 0 in every period, so the Sharpe ratio is just the ratio of the mean of an asset's return to the standard deviation of that asset's return.

◁

▷ Example 2: Adding a risk-free rate and benchmark return

Next we add a risk-free rate and a benchmark asset return to the computation of summary statistics. First, we divide the annual federal funds rate by 12 to get a monthly risk-free interest rate.

```
. generate double rf = fedfunds / 12
```

Next, we use `finreturns` to compute the monthly simple return on our benchmark asset (`sp500`).

```
. quietly finreturns sp500, simple(rmkt) multiply(100)
```

The above `finreturns` command creates a new variable, `rmkt`, with the simple monthly return on `sp500`, and, as before, the result is multiplied by 100 so that the resulting return can be interpreted in monthly percentage change.

With this setup complete, we compute summary statistics for our five assets:

```
. finsummarize r_acme-r_tyr, rfrate(rf) benchmark(rmkt, adjust)
Financial summary statistics
Number of obs = 779
Sample: 1955m2 thru 2019m12
```

	Mean	Std. dev.	Sharpe ratio	Treynor index	Jensen's alpha	beta
ACME	0.4517	1.1856	0.0443	0.3816	0.0192	0.1340
BAT	0.9052	5.4073	0.0927	0.3316	0.1421	1.5217
IRON	1.0214	6.5625	0.0941	0.3351	0.1794	1.8526
DUNE	1.0593	6.6088	0.0991	0.3534	0.2147	1.8640
TYR	0.7028	3.9642	0.0756	0.2780	0.0433	1.0870

```
Notes: Variables adjusted for risk-free rate rf.
       Variable rmkt used as benchmark asset and adjusted for risk-free rate
       rf.
```

By default, `finsummarize` computes all the statistics it is able to, given the information supplied in the `rfrate()` and `benchmark()` options. Because a variable was specified in `benchmark()`, the beta statistic, Jensen's alpha, and Treynor index can be calculated.

These assets show some variation in average returns, volatility, and summary indices. Beta is the coefficient on the benchmark asset in a regression. It measures comovement with the benchmark. A value near one indicates that the asset rises approximately in proportion to the benchmark. A value greater than one indicates that the asset moves more than one for one with the benchmark. A value near zero indicates that the asset does not have a tendency to comove with the benchmark. Negative values indicate assets that move against the benchmark: they tend to have good days when the benchmark has bad days, and vice versa, thus acting as a hedge.

For the assets above, ACME returns have fairly low beta risk. The three assets BAT, IRON, and DUNE have beta risk greater than one and thus are characterized as aggressive. They earn greater returns during "good" times when the benchmark is also delivering good returns, but they also earn even lower returns than the benchmark when the benchmark is earning poor returns. TYR has beta risk near one, indicating that its returns follow the market closely in magnitude.

Jensen's alpha is the intercept in a regression of the asset's return on the benchmark return. It has the interpretation of the return an asset provides, independently of the benchmark asset. For instance, in the above table, the ACME stock has an alpha of about 0.02. By contrast, the DUNE asset has a relatively high alpha of 0.22.

Treynor's index is the ratio of the excess return to beta risk and measures the "gain" in return an asset enjoys for each unit of beta risk it bears. For instance, most of the values in the above table are around 0.33, indicating that as beta risk increases by one unit, excess return rises by 0.33, or about one-third of a percent, per month.

The beta statistic and Jensen's alpha are identical to what would be obtained in a `finregress capm` financial regression using the same variables. Treynor's index uses the beta statistic to compute a return-to-market-risk ratio.

◁

▷ Example 3: Controlling the displayed statistics

The `statistics()` option allows finer control over which statistics are displayed. The mean and standard deviation are always reported. The option takes one or more of four keywords (`sharpe`, `jensen`, `treynor`, and `beta`) and displays only those statistics.

Using the same assets as the previous example, we can restrict the output to display only the Sharpe ratio and Jensen's alpha by typing

```
. finsummarize r_acme-r_tyr, rfrate(rf) benchmark(rmkt, adjust)
> statistics(sharpe jensen)
Financial summary statistics

Number of obs = 779
Sample: 1955m2 thru 2019m12
```

	Mean	Std. dev.	Sharpe ratio	Jensen's alpha
ACME	0.4517	1.1856	0.0443	0.0192
BAT	0.9052	5.4073	0.0927	0.1421
IRON	1.0214	6.5625	0.0941	0.1794
DUNE	1.0593	6.6088	0.0991	0.2147
TYR	0.7028	3.9642	0.0756	0.0433

```
Notes: Variables adjusted for risk-free rate rf.
       Variable rmkt used as benchmark asset and adjusted
       for risk-free rate rf.
```

◁

You may wish to compute standard errors and confidence intervals for the financial summary statistics. The `detail` option provides these statistics. The `post` option can then be used to post the results to `e()`, after which you can perform the usual tests using `test`, `testnl`, `lincom`, and `nlcom`.

▷ Example 4: Computing detailed statistics

We compute detailed summary statistics for two asset returns:

```
. finsummarize r_bat r_iron, rfrate(rf) benchmark(rmkt, adjust) detail
Financial summary statistics

Number of obs = 779
Sample: 1955m2 thru 2019m12
```

	Statistic	Robust std. err.	z	P>|z|	[95% conf.	interval]
r_bat						
mean	.9051807	.1936109	4.68	0.000	.5257104	1.284651
sd	5.407262	.1974776	27.38	0.000	5.020213	5.794311
sharpe	.0927026	.0369553	2.51	0.012	.0202714	.1651337
treynor	.3316203	.1276876	2.60	0.009	.0813572	.5818835
alpha	.1421061	.0420959	3.38	0.001	.0595997	.2246125
beta	1.521697	.0124204	122.52	0.000	1.497353	1.54604
r_iron						
mean	1.021351	.2349746	4.35	0.000	.5608094	1.481893
sd	6.562488	.2391546	27.44	0.000	6.093754	7.031223
sharpe	.0940612	.0368828	2.55	0.011	.0217724	.1663501
treynor	.3350978	.1268855	2.64	0.008	.0864068	.5837887
alpha	.1794485	.0464301	3.86	0.000	.0884472	.2704498
beta	1.852581	.0149286	124.10	0.000	1.823322	1.881841

```
Notes: Variables adjusted for risk-free rate rf.
       Variable rmkt used as benchmark asset and adjusted for risk-free rate
       rf.
```

The `detail` option provides estimates of the uncertainty around the financial summary statistics. ◁

Stored results

`finsummarize` stores the following in `r()`:

Scalars

`r(N)`	number of observations
`r(adjust)`	1 if benchmark asset return was adjusted, 0 otherwise

Macros

`r(stats)`	financial statistics list
`r(rfrate)`	risk-free rate
`r(benchmark)`	benchmark asset return variable
`r(wtype)`	weight type
`r(wexp)`	weight expression

Matrices

`r(table)`	table with all financial summary statistics

finsummarize with the detail option stores the following in r():

Scalars	
r(adjust)	1 if benchmark asset return was adjusted, 0 otherwise
r(N_clust)	number of clusters
r(hac_lag)	HAC lag
Macros	
r(rfrate)	risk-free rate
r(benchmark)	benchmark asset return variable
r(wtype)	weight type
r(wexp)	weight expression
r(vce)	*vcetype* specified in vce()
r(vcetype)	title used to label Std. err.
r(clustvar)	name of cluster variable
r(hac_kernel)	HAC kernel
r(nodfk)	nodfk, if specified
Matrices	
r(b)	estimates
r(V)	variance–covariance matrix of the estimates

finsummarize with the post option stores the following in e():

Macros	
e(properties)	b V
Matrices	
e(b)	estimates vector
e(V)	variance–covariance matrix of the estimators

Methods and formulas

For each of the financial asset returns specified in finsummarize, the mean and standard deviation are computed using summarize. The Sharpe ratio, sharpe, for a given return r_t and a possibly time-varying risk-free rate r_t^f is computed by first forming the excess return series,

$$r_t^e = r_t - r_t^f$$

and then by computing the ratio of mean excess return μ_e to excess standard deviation σ_e,

$$\text{Sharpe} = \frac{\mu_e}{\sigma_e}$$

When the risk-free rate is zero, this expression reduces to the return's mean divided by its standard deviation. When the risk-free rate is constant, this expression becomes the excess return of the asset divided by the standard deviation of the asset.

The value reported as beta is the slope coefficient in a regression of the return and the asset specified in benchmark(). Jensen's alpha is the constant term in this regression. These computations are performed by finregress capm. When adjust is specified in the benchmark() option, the benchmark asset is also adjusted by the risk-free rate specified in rfrate().

The Treynor index, treynor, uses the slope coefficient β from finregress capm (see *Remarks and examples* in [FIN] **finregress capm** for more details), the mean of the returns μ_r, and the mean of the risk-free rate μ_r^f to get

$$\text{Treynor} = \frac{\mu_r - \mu_r^f}{\beta}$$

The `detail` option estimates all financial summary statistics jointly by the generalized method of moments and supports robust, cluster–robust, and HAC standard errors.

References

Andrews, D. W. K. 1991. Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica* 59: 817–858. https://doi.org/10.2307/2938229.

Gallant, A. R. 1987. *Nonlinear Statistical Models*. New York: Wiley. https://doi.org/10.1002/9780470316719.

Newey, W. K., and K. D. West. 1994. Automatic lag selection in covariance matrix estimation. *Review of Economic Studies* 61: 631–653. https://doi.org/10.2307/2297912.

Also see

[FIN] **finreturns** — Generate financial returns+

[FIN] **finregress capm** — Capital asset pricing model (CAPM)+

[TS] **tsset** — Declare data to be time-series data

For suggested citations, see the FAQ on citing Stata documentation.