# Returns

Kevin Crotty
BUSI 448: Investments



RICE | BUSINESS

# Where are we?

**Last time**:

- Saving for retirement
- Real and nominal cash flows and rates

**Today**:

- Calculating returns
- Fetching data
- Summarizing returns

# Calculating Returns

# General definition

For an investment, a **gross return** is the value today scaled by the value in a prior period:

$$\text{Gross Return}_t = \frac{\text{Value}_t}{\text{Value}_{t-1}}.$$

A **net return** is the *change* in value, scaled by the value in a prior period:

$$\text{Net Return}_t = \frac{\Delta\text{Value}_t}{\text{Value}_{t-1}}.$$

We will usually use net returns in class.

# Stock returns

For stocks, the value today is measured by the current price and any dividends ($D$).

$$R_t = \frac{P_t + D_t - P_{t-1}}{P_{t-1}}$$

- Stock prices include future dividends until the **ex-dividend** date
- For computing returns, the dividend goes on the ex day.

# Stock splits

- If a company does an $x$-for-$y$ stock split, then each shareholder gets $x$ new shares for every $y$ of her existing shares. Shares are worth roughly $y/x$ as much.
- Data vendors routinely compute **split-adjusted prices**, scaling down old prices by the same factor for comparability to new prices.
- Example: finance.yahoo.com is a good source for data.
  - Yahoo's adjusted closing prices are adjusted for splits and also adjusted for dividends on each ex date.

# Bond returns

For bonds, the value today is measured by the current price and any accrued interest $(AI)$ or coupon payments $(C)$.

$$R_t = \frac{P_t + AI_t + C_t - (P_{t-1} + AI_{t-1})}{P_{t-1} + AI_{t-1}}$$

- Prices quoted as **clean prices**, which exclude accrued interest.
- Trades between coupon payments transact at **dirty price**
  - dirty price = clean price plus accrued interest (dashboard)

# Compounding returns

- We can compound daily returns to get weekly, monthly, or annual returns.

$$(1+r_1)(1+r_2)\ldots(1+r_T)-1$$

- The weekly, etc. returns are as if
  - the dividend was received on the ex day and reinvested in new shares for stocks, or
  - interest payments were reinvested in the bond.

# Fetching data

# Yahoo Finance package

## Install and import yfinance package

```
!pip install --upgrade yfinance
import yfinance as yf
```

# Daily equity returns from Yahoo Finance

- Yahoo provides daily data by default.
- Adjusted closing prices are adjusted for splits and also adjusted for dividends on each ex date.

```
price = yf.download('AAPL', start='2000-01-01', end='2020-12-31', progress=False)[
ret_daily = price.pct_change()
```

# Lower frequency returns from Yahoo

- Can get monthly or annual return as % change in monthly or annual Yahoo-adjusted closing prices
- Equivalent to compounding Yahoo daily returns.

```
price = yf.download('AAPL', start='2000-01-01', end='2020-12-31', progress=False)[
ret_monthly = price.resample('M').last().pct_change()
ret_annual  = price.resample('Y').last().pct_change()

# change index from datetime to period (optional)
ret_monthly.index = ret_monthly.index.to_period('M')
ret_annual.index  = ret_annual.index.to_period('Y')
```

# Pandas-datareader package

## Install and import pandas-datareader package

```
!pip install --upgrade pandas-datareader
import pandas as pd
from pandas_datareader import DataReader as pdr
```

# FRED data

- FRED at the St. Louis Fed
- A wide range of interest rate and macroeconomic series

# Market and various portfolio returns + RF

## Ken French Data Libary

- Ken French is a Dartmouth finance prof and board member of investment firm Dimensional
- Prolific researcher whose work we will be discussing later

Datasets include

- benchmark returns including VW market return
- portfolios sorted on characteristics
- industry portfolios

# Summarizing Returns

# Arithmetic Average Returns

- The **arithmetic average return** is sometimes used to estimate expected return:

$$\frac{1}{T}\sum_{t=1}^{T} r_t$$

- Assumption: each realized return was a draw from a single distribution.
  - We will talk later in the class about what to do if we think this is a bad assumption.

# Geometric Average Returns

- We may also be interested in summarizing the realized performance of an investment over a time horizon.
- The **geometric average return** is the constant return that would compound to the same compounded return experienced by an investor.
  - aka compound annual growth rate or time-weighted return.
- The geometric average return is **always less** than the arithmetic average return.
  - The difference is larger when returns are more volatile.

# Tesla

- Tesla went down 50% between Nov 2021 and May 2022.
- It then went up 50% between May 2022 and Aug 2022.
- Were Tesla shareholders back to even?

- For each $100 of Tesla stock, shareholders experienced $100 \to 50$
- and then $50 \to 75$.
- They lost 25%, even though the average return was zero.
- So, lose 50% and make 50% $\to$ lose 25%. Suppose you
  - make 50% and then lose 50%?
  - lose 50% and then make 100%?
  - make 100% and then lose 50%?

# Geometric Average Return

- Given returns $r_1, \cdots, r_n$, the geometric average return is the number $r$ such that

$$(1+r)^n = (1+r_1)\cdots(1+r_n)$$

- So earning $r$ each period produces the same accumulation as the actual returns $r_1, \cdots, r_n$. We solve for $r$ as

$$r = [(1+r_1)\cdots(1+r_n)]^{1/n} - 1$$

# Examples

- make 50% and lose 50% → geometric average is

$$\sqrt{1.5 \times 0.5} - 1 = -0.134$$

- make 100% and lose 50% → geometric average is

$$\sqrt{2 \times 0.5} - 1 = 0$$

Some historical data

# Average Returns in Python

Given a net return pandas data series `ret`, arithmetic average returns can be calculated:

```
ret.mean()
```

and the geometric averages can be calculated:

```
from scipy.stats import gmean
gmean(1+ret)-1
```

# Variance and Standard Deviation

- The dispersion in realized returns can be measured by either variance or standard deviation.

Sample standard deviation for sample mean $m$:

$$\sqrt{\sum_{t=1}^{T} \frac{(r_t - m)^2}{T-1}}$$

To estimate standard deviation in python from a pandas data series `ret`

```
ret.std()
```

# For next time: Returns of portfolios