

# Portfolios

BUSI 721: Data-Driven Finance I

Kerry Back, Rice University



# Portfolio returns

- Two assets, prices  $p_1$  and  $p_2$
- Own shares  $x_1$  and  $x_2$
- Portfolio value is  $p_1x_1 + p_2x_2$
- Fraction of value in asset  $i$  (**weight of asset**) is

$$w_i = \frac{p_i x_i}{p_1 x_1 + p_2 x_2}$$

- Future prices + dividends  $\Rightarrow$  returns  $r_1$  and  $r_2$
- Portfolio return is  $w_1 r_1 + w_2 r_2$ .



## Example

- \$200 in asset 1 and \$300 in asset 2
- asset 1 goes up 10% and asset 2 goes up 5%
- asset 1  $\mapsto$  \$220
- asset 2  $\mapsto$  \$315
- portfolio value is \$535
- This is a 7% gain and

$$\frac{2}{5} \times 0.1 + \frac{3}{5} \times 0.05 = 0.07$$



# Expected return

- Returns are random variables
- Expected = mean
- mean portfolio return  $w_1r_1 + w_2r_2$  is

$$w_1 \times \text{mean of } r_1 + w_2 \times \text{mean of } r_2$$



# Variance

- The variance of the portfolio return is

$$w_1^2\sigma_1^2 + w_2\sigma_2^2 + 2w_1w_2\rho\sigma_1\sigma_2$$

- where  $\sigma_i =$  std dev of  $r_i$  and  $\rho$  is the correlation of  $r_1$  and  $r_2$ .
- Lower correlation implies lower portfolio risk!



# Proof of variance formula

- Variance is expected squared deviation from mean
- Set  $r_p = w_1 r_1 + w_2 r_2$  and use overbars to denote means
- Variance of  $r_p$  is

$$\text{mean of } (r_p - \bar{r}_p)^2$$

- And

$$r_p - \bar{r}_p = w_1 r_1 + w_2 r_2 - (w_1 \bar{r}_1 + w_2 \bar{r}_2)$$

- So

$$r_p - \bar{r}_p = w_1 (r_1 - \bar{r}_1) + w_2 (r_2 - \bar{r}_2)$$

- To square this, use

$$(a + b)^2 = a^2$$

$$+ b^2 + 2ab$$



- So portfolio variance is

$$\begin{aligned} &w_1^2 \times \text{mean of } (r_1 - \bar{r}_1)^2 \\ &+ w_2^2 \times \text{mean of } (r_2 - \bar{r}_2)^2 \\ &+ 2w_1w_2 \times \text{mean of } (r_1 - \bar{r}_1)(r_2 - \bar{r}_2) \end{aligned}$$

- This is

$$w_1^2\sigma_1^2 + w_2\sigma_2^2 + 2w_1w_2\rho\sigma_1\sigma_2$$

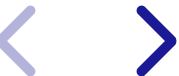


Example





```
In [2]: mu1, mu2 = 0.06, 0.1  
sigma1, sigma2 = 0.2, 0.3  
rho = 0.3  
w1, w2 = 0.4, 0.6
```



In [3]: `import numpy as np`

```
mn = w1*mu1 + w2*mu2
```

```
var = w1**2*sigma1**2 + w2**2*sigma2**2 + 2*w1*w2*rho*sigma1*sigma2
```

```
print(f"mean portfolio return is {mn:.2%}")
```

```
print(f"std dev of portfolio return is {np.sqrt(var):.2%}")
```

mean portfolio return is 8.40%

std dev of portfolio return is 21.78%



# Simulation



In [4]:

```
cov = [  
    [sigma1**2, rho*sigma1*sigma2],  
    [rho*sigma1*sigma2, sigma2**2]  
]  
  
from scipy.stats import multivariate_normal as multinorm  
rets = multinorm.rvs(  
    mean=[mu1, mu2],  
    cov=cov,  
    size=1000000  
)  
rp = w1*rets[:,0] + w2*rets[:,1]  
  
print(f"simulated mean is {np.mean(rp):.2%}")  
print(f"simulated std dev is {np.std(rp):.2%}")
```

```
simulated mean is 8.39%  
simulated std dev is 21.77%
```



# Cash

- Adding cash (money market investment) to a portfolio has a simple effect on expected return and risk.
- Let asset 2 be cash. Its return has negligible risk. Call its return  $r_{mm}$ .
- Portfolio mean is  $w_1\mu_1 + w_2r_{mm}$
- Portfolio variance is  $w_1^2\sigma_1^2$
- Portfolio std dev is  $w_1\sigma_1$ .



# Margin loans

- You can have negative cash by borrowing from your broker.
- Example: put \$1,000 in an account, borrow \$200 and buy \$1,200 of a stock.
- Continue to call asset 2 cash.
- Your portfolio weights are  $w_1 = 1.2$  and  $w_2 = -0.2$ .
- Your expected return is  $1.2 \times \mu_1 - 0.2 \times r_{ml}$  where  $r_{ml}$  is the margin loan rate.
- Your std dev is  $1.2\sigma_1$ .



# Short selling

- You can have a negative weight on a stock by selling short.
- To sell short, your broker borrows shares on your behalf and sells them.
- You eventually have to buy the shares back in the market and return them.
- Profit by selling high and buying low.
- Example: short sell 100 shares \$100 stock.
  - Stock falls to \$90 and you cover the short (buy and return shares).
  - Paid \$90 and sold at \$100  $\Rightarrow$  profit \$10 per share on 100 shares.



## Long and short returns (simplified version)

- Assume we invest \$1,000, short sell \$400 of stock 2 and buy \$1,400 of stock 1.
- The weights are 1.4 in stock 1 and -0.4 in stock 2.
- Suppose stock 1 goes up 10% and stock 2 goes up 5%.
  - Stock 1  $\rightarrow$  \$1,540.
  - Stock 2 position  $\rightarrow$   $-\$420$ .
  - Portfolio value  $\rightarrow$  \$1,120 = 12% return
  - $w_1r_1 + w_2r_2 = 1.4 \times 0.10 - 0.4 \times 0.05$   
 $= 0.12$





## Long and short returns (practical version)

- Proceeds from short sales are retained as collateral
- Investor may get some interest on the proceeds while they are held (called short interest rebate)
- We can invest \$1,000, short sell \$400 of stock 2 and buy \$1,400 of stock 1 only if we take out a margin loan for \$400.
- Actual return in example is

$$1.4 \times 0.10 - 0.4 \times 0.05$$

$$-0.4 \times r_{ml} + 0.4 \times r_{sir} - 0.4 \times \text{short borrowing fee}$$

- where  $r_{sir}$  is the short interest rebate rate.



## Enhanced index return example

- Invest \$1,000. Borrow \$1,000 on margin loan.
- Buy \$1,000 of SPY and buy \$1,000 of CVX.
- Short sell \$1,000 of COP.
- Return is  $\text{SPY return} + \text{CVX return} - \text{COP return} - \text{margin loan/short interest rebate/short borrowing fee drag}$ .
- If CVX beats COP enough, you will beat SPY.



# More assets

- $n$  stocks
- weights  $w_1, \dots, w_n$
- expected returns  $\mu_1, \dots, \mu_n$
- covariance matrix  $\Sigma$ 
  - diagonal elements of covariance matrix are variances
  - off-diagonal elements are correlation  $\times$  std dev  $\times$  std dev.



# Portfolio risk

- Portfolio variance is  $w'\Sigma w$
- Portfolio std dev is  $\sqrt{w'\Sigma w}$



```
In [5]: # example

w = np.array([0.2, 0.2, 0.4])
sigma1, sigma2, sigma3 = 0.2, 0.3, 0.1
rho12, rho13, rho23 = 0.3, 0.5, 0.4

cov = np.array([
    [sigma1**2, rho12*sigma1*sigma2, rho13*sigma1*sigma3],
    [rho12*sigma1*sigma2, sigma2**2, rho23*sigma2*sigma3],
    [rho13*sigma1*sigma3, rho23*sigma2*sigma3, sigma3**2]
])
stdev = np.sqrt(w @ cov @ w)

print(f"portfolio std dev is {stdev:.2%}")
```

portfolio std dev is 10.84%



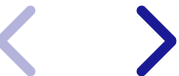
## Portfolio expected return

- If  $w_i \geq 0$  and sum to 1, then portfolio mean is  $w' \mu$ .
- If  $w_i \geq 0$  and sum to less than 1, then portfolio mean is

$$w' \mu + \left(1 - \sum w_i\right) r_{mm}$$

- If  $w_i \geq 0$  and sum to more than 1, then portfolio mean is

$$w' \mu + \left(1 - \sum w_i\right) r_{ml}$$



- So we can say that if  $w_i \geq 0$ , then portfolio mean is

$$w' \mu + \left(1 - \sum w_i\right) r_f$$

- where  $r_f = r_{mm}$  if cash  $> 0$  and  $r_m = r_{ml}$  if cash  $< 0$ .
- With short sales, portfolio mean is also

$$w' \mu + \left(1 - \sum w_i\right) r_f$$

- minus the drag from difference between margin loan and short interest rebate rates and minus short borrowing fees.



In [6]: *# continuing prior example*

```
rf = 0.04
mu1, mu2, mu3 = 0.1, 0.12, 0.08
mu = np.array([mu1, mu2, mu3])

port_mean = w @ mu + (1-np.sum(w))*rf
print(f"portfolio mean is {port_mean:.2%}")
```

portfolio mean is 8.40%





# Estimating from historical returns

- Example:
  - SPY = S&P 500
  - IEF = Treasury bonds
  - GLD = gold
- Get adjusted closing prices from Yahoo
- Compute returns as percent changes



In [7]: `import yfinance as yf`

```
tickers = ["SPY", "IEF", "GLD"]
prices = yf.download(tickers, start="1970-01-01")["Adj Close"]
rets = prices.pct_change().dropna()
rets.head(3)
```

[\*\*\*\*\*100%\*\*\*\*\*] 3 of 3 completed

Out[7]:

	<b>GLD</b>	<b>IEF</b>	<b>SPY</b>
<b>Date</b>			
<b>2004-11-19</b>	0.009013	-0.005480	-0.011117
<b>2004-11-22</b>	0.003796	0.000704	0.004769
<b>2004-11-23</b>	-0.004449	-0.000938	0.001526



In [8]: `rets.corr()`

Out[8]:

	<b>GLD</b>	<b>IEF</b>	<b>SPY</b>
<b>GLD</b>	1.000000	0.208371	0.049425
<b>IEF</b>	0.208371	1.000000	-0.320228
<b>SPY</b>	0.049425	-0.320228	1.000000



# Annualizing

- May be easier to interpret means and std devs when expressed in annual terms
- Annualize daily means by multiplying by 252 (# of trading days in a year)
- Annualize daily variances by multiplying by 252
- Annualize daily std devs by multiplying by square root of 252



```
In [9]: print(f"annualized means are \n{252*rets.mean()}")
print(f"\nannualized std devs are \n{np.sqrt(252)*rets.std()}")
```

annualized means are

GLD 0.090313

IEF 0.031442

SPY 0.106698

dtype: float64

annualized std devs are

GLD 0.176744

IEF 0.068225

SPY 0.193171

dtype: float64



# Portfolio returns



```
In [10]: w = np.array([0.2, 0.3, 0.5])
mean = w @ rets.mean()
var = w @ rets.cov() @ w
stdev = np.sqrt(var)

print(f"annualized mean portfolio return is {252*mean:.2%}")
print(f"annualized std dev is {np.sqrt(252)*stdev:.2%}")
```

```
annualized mean portfolio return is 8.08%
annualized std dev is 10.18%
```

