FNCE611 Problem Set 3

Jessica Blum, Jordan Farrer, Marc Lanthemann

2017-02-24

Contents

1 Divisification

```
wfm_tsla <- gs_title("FNCE611") %>%  gs_read(ws = "3.1")
```

1.1 a

```
e_wfm <- mean(wfm_tsla$wfm) * 12
e_tsla <- mean(wfm_tsla$tsla) * 12

sd_wfm <- sd(wfm_tsla$wfm) * sqrt(12)
sd_tsla <- sd(wfm_tsla$tsla) * sqrt(12)

cc <- cor(wfm_tsla$wfm, wfm_tsla$tsla)
```

Estimate (annualized)	Value
Expected Return (Whole Foods)	0.128
Expected Return (Tesla)	0.514
Standard Deviation (Whole Foods)	0.2712
Standard Deviation (Tesla)	0.5875
Correlation Coefficient	0.3

1.2 b

```
w_wfm = 0.2
e_p <- w_wfm*e_wfm + (1-w_wfm)*e_tsla
```

If $w_{wfm} = 0.2$, then the expected return for the portfolio is

$$\begin{aligned} E[r_p] &= \omega_w E[r_w] + (1-\omega_w)E[r_t] && (1)\\ &= 0.2(0.128) + (1-0.2)(0.514) && (2)\\ &= 0.4368 && (3) \end{aligned}$$

1.3 c

$$
\begin{aligned}
Var[r_p] &= \omega_w^2\sigma_w^2 + \omega_t^2\sigma_t^2 + 2\omega_w\omega_t\rho_{w,t}\sigma_w\sigma_t && (4)\\
&= 0.2^2 \cdot 0.2712^2 + (1-0.2)^2 \cdot 0.5875^2 + 2 \cdot 0.2 \cdot (1-0.2) \cdot 0.3 \cdot 0.2712 \cdot 0.5875 && (5)\\
&= 0.2392 && (6)
\end{aligned}
$$

So the standard deviation of the portfolio is $\sqrt{Var[r_p]} = 0.4891$.

1.4 d

```
w_wfm_opt <- uniroot(function(w) w*e_wfm + (1-w)*e_tsla -.2, interval=c(0,1), tol=0.0001)$root
```

The weights of a portfolion showing an expected return of 20% would be $\omega_w = 0.8136$ and $\omega_t = 0.1864$.

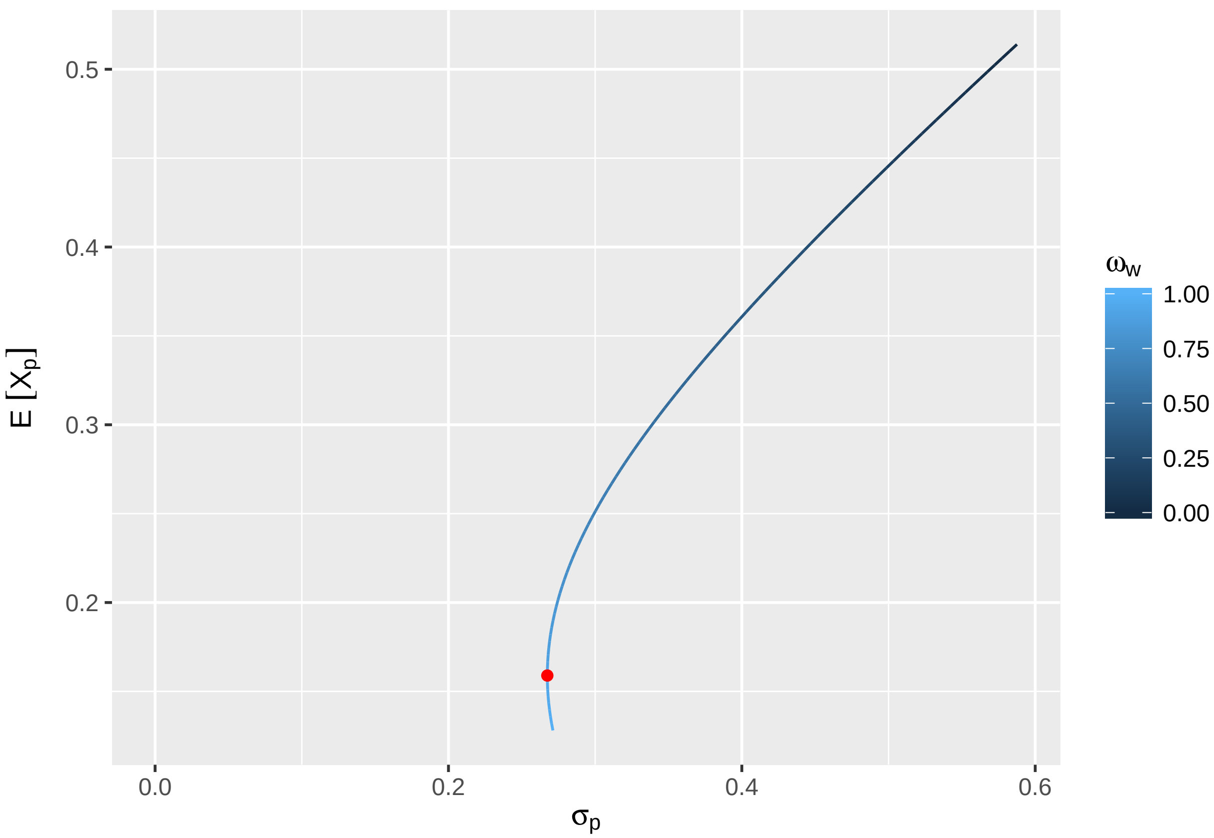
1.5 e

```
fn_mvp <- function(par) {
  w <- par[1]
  w^2 * sd_wfm^2 + (1-w)^2 * sd_tsla^2 + 2*w*(1 - w)*cc*sd_wfm*sd_tsla
}

w_wfm_mvp <- nlminb(.5, fn_mvp, lower = 0, upper = 1)$par[1]
```

The minimum variance portfolio (MVP) would have weights $\omega_w = 0.9203$ and $\omega_t = 0.0797$.

```
data_frame(w_wfm = seq(0, 1, by = 0.01)) %>%
  mutate(
    expected_std = sqrt(w_wfm^2 * sd_wfm^2 + (1-w_wfm)^2 *
                          sd_tsla^2 + 2*w_wfm*(1 - w_wfm)*cc*sd_wfm*sd_tsla)
    , expected_return = w_wfm*e_wfm + (1-w_wfm)*e_tsla
  ) %>%
  ggplot(aes(x = expected_std, y = expected_return, colour = w_wfm)) +
  geom_path() +
  labs(x = expression(sigma[p]), y = expression(E~group("[",X[p], "]")),
       colour = expression(omega[w])) +
  geom_point(data = . %>% top_n(-1, expected_std),
             aes(x = expected_std, y = expected_return), colour = 'red') +
  scale_x_continuous(limits = c(0, NA))
```



1.6 f

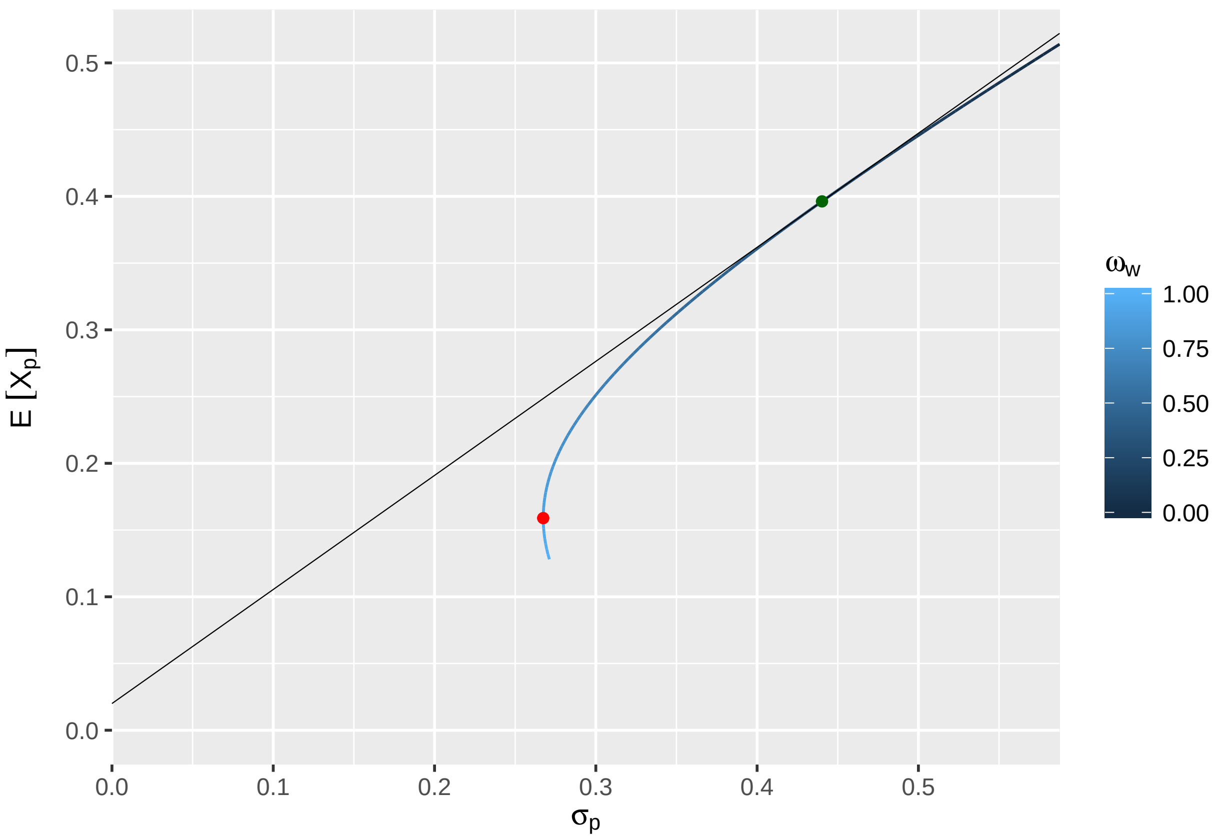
```
fn_rfr <- function(par) {
  w <- par[1]
  sharpe <- ((w*e_wfm + (1-w)*e_tsla) - 0.02) / sqrt(w^2 * sd_wfm^2 + (1-w)^2 * sd_tsla^2 +
                                                       2*w*(1 - w)*cc*sd_wfm*sd_tsla)
  return(-sharpe)
}
max_w <- nlminb(.5, fn_rfr, lower = 0, upper = 1)$par[1]
max_sharpe_ratio <- ((max_w*e_wfm + (1-max_w)*e_tsla) - 0.02) / sqrt(max_w^2 * sd_wfm^2 +
                      (1-max_w)^2 * sd_tsla^2 + 2*max_w*(1 - max_w)*cc*sd_wfm*sd_tsla)
```

A portfolio with the highest Sharpe ratio would have weights $\omega_w = 0.3051$ and $\omega_t = 0.6949$.

```
efficient_mix <-
  data_frame(expected_std =
         sqrt(max_w^2 * sd_wfm^2 + (1-max_w)^2 * sd_tsla^2 +
                2*max_w*(1 - max_w)*cc*sd_wfm*sd_tsla)
       , expected_return = max_w*e_wfm + (1-max_w)*e_tsla
  )

data_frame(w_wfm = seq(0, 1, by = 0.01)) %>%
  mutate(
    expected_std = sqrt(w_wfm^2 * sd_wfm^2 + (1-w_wfm)^2 * sd_tsla^2 +
                          2*w_wfm*(1 - w_wfm)*cc*sd_wfm*sd_tsla)
    , expected_return = w_wfm*e_wfm + (1-w_wfm)*e_tsla
  ) %>%
```

```r
ggplot(aes(x = expected_std, y = expected_return, colour = w_wfm)) +
geom_path() +
labs(x = expression(sigma[p]), y = expression(E~group("[",X[p], "]")),
     colour = expression(omega[w])) +
geom_point(data = . %>% top_n(-1, expected_std),
           aes(x = expected_std, y = expected_return), colour = 'red') +
scale_x_continuous(limits = c(0, NA), expand = c(0,0)) +
scale_y_continuous(limits = c(0, NA)) +
geom_abline(aes(slope = max_sharpe_ratio, intercept = 0.02), size = .2) +
geom_point(data = efficient_mix, aes(x = expected_std, y = expected_return),
           colour = 'dark green')
```



1.7 g

$$E[r_p] = \omega_{riskless\ asset} E[r_{riskless\ asset}] + (1 - \omega_{riskless\ asset}) E[r_{mix}] \tag{7}$$

$$0.25 = \omega_{riskless\ asset} 0.02 + (1 - \omega_{riskless\ asset}) 0.3962 \tag{8}$$

```r
w_investor_rf <- uniroot(function(w) 0.02*w + (1-w)*efficient_mix$expected_return[1] - 0.25,
                         interval=c(0,1), tol=0.0001)$root
```

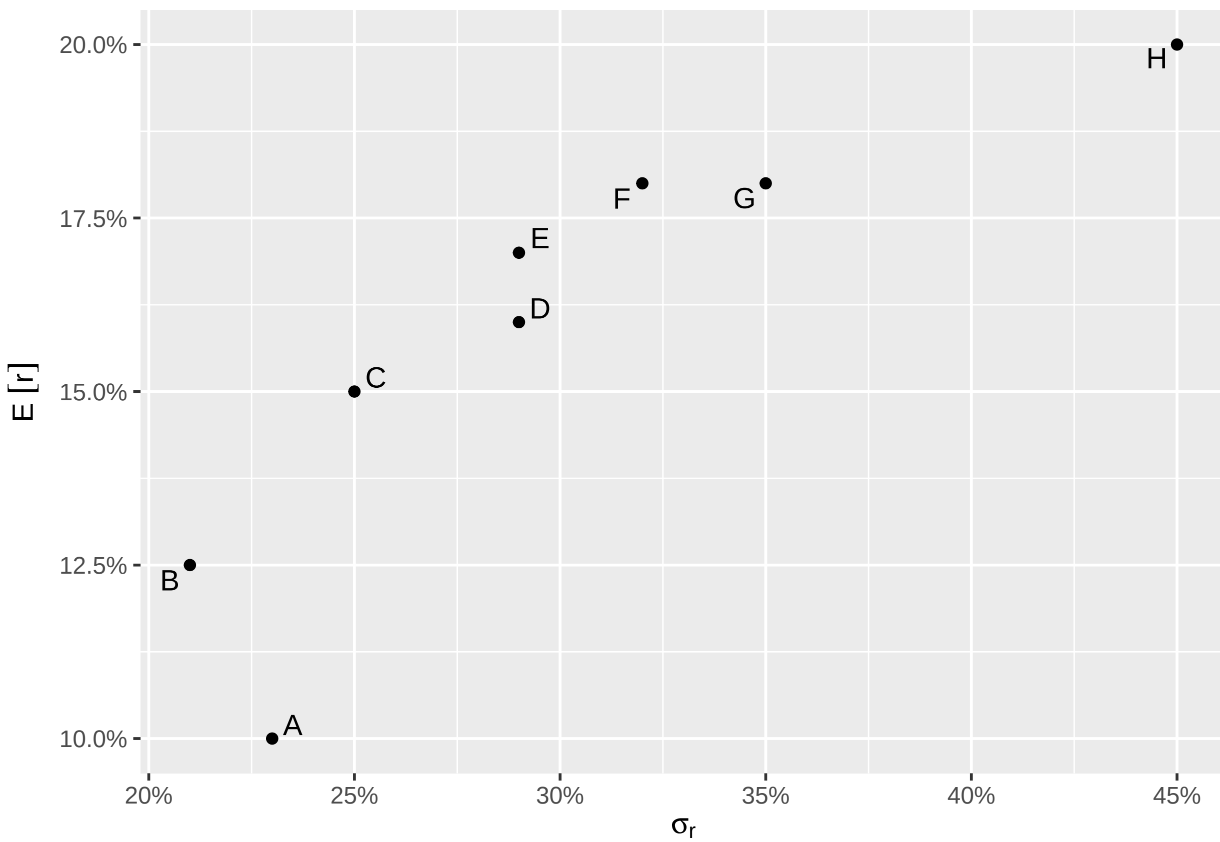
For a portfolio with an expected return of 25%, $\omega_{riskless\ asset} = 0.3887$ and $\omega_{mix} = 0.6113$. Thus, $\omega_w = 0.1865$ and $\omega_t = 0.4248$.

2 Efficient Portfolios, Capital Market Line

2.1 a

```r
question2 <-
  data_frame(
    label = LETTERS[1:8]
    , e_r = c(.1, .125, .15, .16, .17, .18, .18, .2)
    , sd_r = c(.23, .21, .25, .29, .29, .32, .35, .45)
  )

question2 %>%
  ggplot(aes(x = sd_r, y = e_r)) +
  geom_point() +
  labs(x = expression(sigma[r]), y = expression(E~group("[",r, "]")),
       colour = expression(omega[w])) +
  ggrepel::geom_text_repel(aes(x = sd_r, y = e_r, label = label)) +
  scale_y_continuous(labels = scales::percent) +
  scale_x_continuous(labels = scales::percent)
```



2.2 b

Portfolios A and D are inefficient. A is inefficient because for less risk you can get more return with portfolio B. D is inefficient because for the same risk you can get a greater return with E.

2.3 c

We see from the table below, that if you can borrow and lend at an interest rate of 12%, **portfolio F** has the highest Sharpe Ratio.

```
question2 %>%
  mutate(
    sharpe_ratio = (e_r - 0.12) / sd_r
  ) %>%
  arrange(desc(sharpe_ratio)) %>%
  pander()
```

label	e_r	sd_r	sharpe_ratio
F	0.18	0.32	0.1875
H	0.2	0.45	0.1778
E	0.17	0.29	0.1724
G	0.18	0.35	0.1714
D	0.16	0.29	0.1379
C	0.15	0.25	0.12
B	0.125	0.21	0.02381
A	0.1	0.23	-0.08696

2.4 d

```
q4 <-
  question2 %>%
    mutate(
      w = .25 / sd_r
      , r_p = w*e_r +(1-e_r)*0.12
      ) %>%
    arrange(desc(r_p))

q4 %>% pander()
```

label	e_r	sd_r	w	r_p
B	0.125	0.21	1.19	0.2538
C	0.15	0.25	1	0.252
E	0.17	0.29	0.8621	0.2462
F	0.18	0.32	0.7812	0.239
D	0.16	0.29	0.8621	0.2387
G	0.18	0.35	0.7143	0.227
A	0.1	0.23	1.087	0.2167
H	0.2	0.45	0.5556	0.2071

If you are willing to tolerate a standard deviation of 25% you would borrow -19.0476% of your investment at the 12% interest rate and invest 119.0476% of your investment in portfolio B.

3 CAPM, Security Market Line (1)

- By definition we know that the Beta of the market portfolio M must be 1.
- We find the $sd(X) = \sqrt{\frac{\rho}{\beta}} = \sqrt{\frac{.4}{1.2}}$

- We use the fact that $r - r_f = \beta(r_m - r_f)$ to find $r_x = 9$ and $r_Q = 13$

```r
data_frame(
  `Title` = c("Stock X", "Efficient Portfolio Q","Market Portfolio M","Riskless Asset")
  , `Expected Return` = c("9%", "13%", "8%", "3%")
  , Beta = c(1.2, 2.0, 1, 0.0)
  , `Standard Deviation` = c("57.74%", NA, "20%", "0%")
) %>%
  pander(missing = "")
```

Title	Expected Return	Beta	Standard Deviation
Stock X	9%	1.2	57.74%
Efficient Portfolio Q	13%	2	
Market Portfolio M	8%	1	20%
Riskless Asset	3%	0	0%

4 CAPM, Security Market Line (2)