Analysis of SP500 excess returns over bond markets over the years

Initial Setup

```
library(dplyr)
library(ggplot2)
library(lubridate)
library(quantmod)
library(xts)
library(psych)

data1 <- read.csv('~/Documents/Documents/Investments/gsw_yields.csv',
                  header = TRUE)
data2 <- read.csv('~/Documents/Documents/Investments/p5-sp500.csv',
                  header = TRUE)
data1$Date <- as.Date(data1$Date, format = "%m/%d/%Y")
data2$caldt <- as.Date(as.factor(as.numeric(data2$caldt)), format = "%Y%m%d")
colnames(data2) <- c("date", "vwretd")
## data <- merge(data1,data2,by="date");

nss <- function(t, b0, b1, b2, b3, t1, t2) {
    r <- b0 + b1*((1-exp(-t/t1))/(t/t1)) + (b2*(((1-exp(-t/t1))/(t/t1))-(exp(-t/t1)))) +
      (b3*(((1-exp(-t/t2))/(t/t2))-(exp(-t/t2))))
    return(r)
}

calculate_bond_price <- function(maturity, par, rate) {
  bond_price <- par/((1+(rate/100))^maturity)
  return(bond_price)
}
```

(a) For the (CRSP) S&P500 with dividends, what was the average arithmetic and geometric historical mean rate of return for daily returns, for monthly returns, for annual returns from 1/3/1972 through 12/31/2019, and for 5-year returns from 1/3/1972 through 12/30/2016? Make sure to annualize the returns you report.

Computation of Annualized Mean Returns:

For computing the average arithmetic return, the following steps are followed: 1. Annualized the daily returns 2. Arithmetic Mean Return and Geometric Mean Return are computed using the annualised daily return for each period(daily, monthly, yearly and 5-year) from 1/3/1972 through 12/31/2019

Arithmetic Mean Return is computed by:

$$R_{arithmetic} = \frac{\Sigma R_i}{n}$$

Geometric Mean Return is computed by:

$$\bar{R}_{geometric} = [(1+R_1)(1+R_2)...(1+R_T)]^{1/T}$$

```r
price <- rep(1,length(data2$date));
data2 <- data2 %>% mutate(vwretd = vwretd*365)
ts <- as.xts(data2$vwretd, data2$date)

temp_month <- 0;
temp_year <- 0;
temp_year_5 <- 1972;
monthly_vector <- 0;
annual_vector <- 0;
vector_5year <- 0;
sp_daily <- 0;
sp_monthly_arithmetic <- 0;
sp_annual_arithmetic <- 0;
sp_monthly_geometric <- 0;
sp_annual_geometric <- 0;
sp_5year_arithmetic <- 0;
sp_5year_geometric <- 0;
var_sp_monthly <- 0;
var_sp_annual <- 0;
var_sp_5year <- 0;
a <- 1;
b <- 21;
c <- 1;
d <- 1;
e <- 1;
f <- 1;

for (i in 1:length(data2$date)) {
  sp_daily[i] <- data2$vwretd[i]
  if(month(data2$date[i]) != temp_month) {
    monthly_vector <- data2$vwretd[a:b];
    sp_monthly_arithmetic[a] <- sum(monthly_vector) / length(monthly_vector);
    sp_monthly_geometric[a] <- geometric.mean(monthly_vector);
    var_sp_monthly[a] <- var(monthly_vector);
    a <- i;
    b <- a;
  } else {
    sp_monthly_geometric[i] <- NA;
    sp_monthly_arithmetic[i] <- NA;
    var_sp_monthly[i] <- NA;
    b <- b + 1;
  }

  if(year(data2$date[i]) != temp_year) {
    annual_vector <- data2$vwretd[c:d];
    sp_annual_arithmetic[c] <- sum(annual_vector) / length(annual_vector)
    sp_annual_geometric[c] <- geometric.mean(annual_vector)
    var_sp_annual[c] <- var(annual_vector);
    c <- i;
    d <- c;
  } else {
```

```r
    sp_annual_arithmetic[i] <- NA;
    sp_annual_geometric[i] <- NA;
    var_sp_annual[i] <- NA;
    d <- d + 1;
  }

  if(i == length(data2$date)) {
    monthly_vector <- data2$vwretd[a:b];
    sp_monthly_arithmetic[a] <- sum(monthly_vector) / length(monthly_vector);
    sp_monthly_geometric[a] <- geometric.mean(monthly_vector)
    var_sp_monthly[a] <- var(monthly_vector);
    annual_vector <- data2$vwretd[c:d];
    sp_annual_arithmetic[c] <- sum(annual_vector) / length(annual_vector)
    sp_annual_geometric[c] <- geometric.mean(annual_vector)
    var_sp_annual[c] <- var(annual_vector);
  }

  if(temp_year_5 - year(data2$date[i]) == 0) {
    temp_year_5 <- year(data2$date[i]) + 5;
    vector_5year <- data2$vwretd[e:f];
    sp_5year_arithmetic[e] <- sum(vector_5year) / length(vector_5year);
    sp_5year_geometric[e] <- geometric.mean(vector_5year);
    var_sp_5year[e] <- var(vector_5year);
    e <- i;
    f <- e;
  } else {
    sp_5year_arithmetic[i] <- NA;
    sp_5year_geometric[i] <- NA;
    var_sp_5year[i] <- NA;
    f <- f + 1;
  }

  temp_month <- month(data2$date[i]);
  temp_year <- year(data2$date[i]);
}

sp_500 <- data.frame("date"=data2$date, sp_daily, sp_monthly_arithmetic,
                     sp_annual_arithmetic, sp_monthly_geometric, sp_annual_geometric,
                     sp_5year_arithmetic, sp_5year_geometric, var_sp_monthly, var_sp_5year,
                     var_sp_annual)
# %>%
#   mutate(sp_monthly_arithmetic = sp_monthly_arithmetic*12,
#          sp_monthly_geometric = sp_monthly_geometric*12,
#          sp_5year_arithmetic = sp_5year_arithmetic/5,
#          sp_5year_geometric = sp_5year_geometric/5)
```

(b) Do the same for excess rates of return over zero-coupon bonds. You should choose the maturity of the bonds to match the horizon of the returns so that the bond returns are risk-free (i.e. known at the beginning of each period). For example, one-month S&P 500 returns should be compared to the returns of a one-month zero-coupon bonds maturing at the end of each period. Again, make sure to annualize the returns you report.

Computation of Excess Returns:

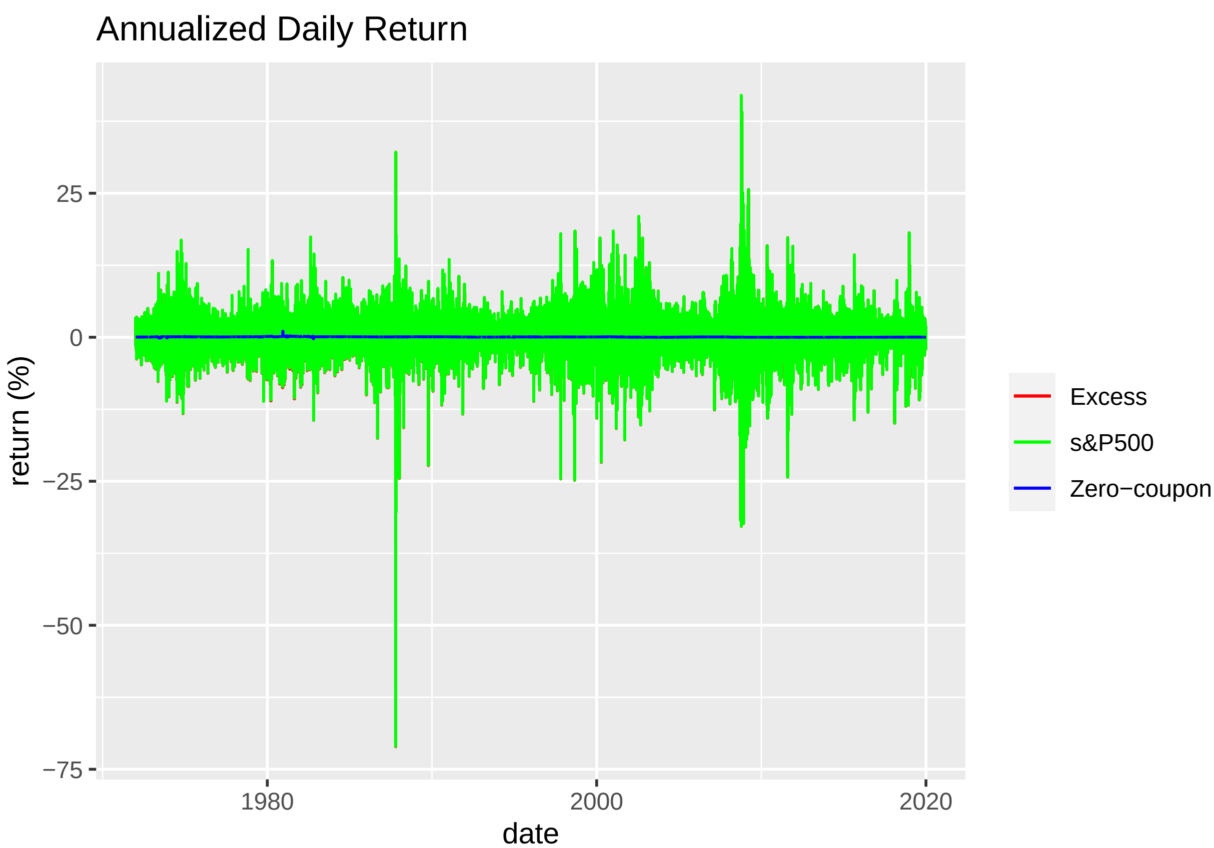
Using the `nss` function, returns from the zero coupon bonds are calculated for all the periods (daily, monthly, yearly and 5-year). One we have the zero coupon returns for each period, excess returns are computed by subtracting these zero coupon returns from S&P500 returns for each of the respective period.

```
temp_month <- 0;
temp_year <- 0;
temp_year_5 <- 1972;
zero_coupon_daily <- 0;
zero_coupon_monthly <- 0;
zero_coupon_annual <- 0;
zero_coupon_5year <- 0;
for (i in 1:length(data2$date)) {
  temp <- data1 %>% filter(Date == data2$date[i])
  zero_coupon_daily[i] <- nss(1/365, temp$BETA0, temp$BETA1, temp$BETA2, temp$BETA3,
                   temp$TAU1, temp$TAU2)/100

  if(month(data2$date[i]) != temp_month) {
    zero_coupon_monthly[i] <- nss(1/12, temp$BETA0, temp$BETA1, temp$BETA2, temp$BETA3, temp$TAU1,
                   temp$TAU2)/100
  } else {
    zero_coupon_monthly[i] <- NA;
  }

  if(year(data2$date[i]) != temp_year) {
    zero_coupon_annual[i] <- nss(1, temp$BETA0, temp$BETA1, temp$BETA2, temp$BETA3,
                    temp$TAU1, temp$TAU2)/100
  } else {
    zero_coupon_annual[i] <- NA;
  }

  if(temp_year_5 - year(data2$date[i]) == 0) {
    temp_year_5 <- year(data2$date[i]) + 5;
    zero_coupon_5year[i] <- nss(5, temp$BETA0, temp$BETA1, temp$BETA2, temp$BETA3, temp$TAU1,
                    temp$TAU2)/100
  } else {
    zero_coupon_5year[i] <- NA;
  }

  temp_month <- month(data2$date[i]);
  temp_year <- year(data2$date[i]);
}
zero_coupon <- data.frame("date" = data2$date, zero_coupon_daily, zero_coupon_monthly,
                          zero_coupon_annual, zero_coupon_5year)

combined <- merge(sp_500, zero_coupon, by="date") %>%
  mutate(excess_daily = sp_daily - zero_coupon_daily,
         excess_monthly_arithmetic = sp_monthly_arithmetic - zero_coupon_monthly,
         excess_monthly_geometric = sp_monthly_geometric - zero_coupon_monthly,
         excess_annual_arithmetic = sp_annual_arithmetic - zero_coupon_annual,
         excess_annual_geometric = sp_annual_geometric - zero_coupon_annual,
         excess_5year_arithmetic = sp_5year_arithmetic - zero_coupon_5year,
         excess_5year_geometric = sp_5year_geometric - zero_coupon_5year)
```

Plots

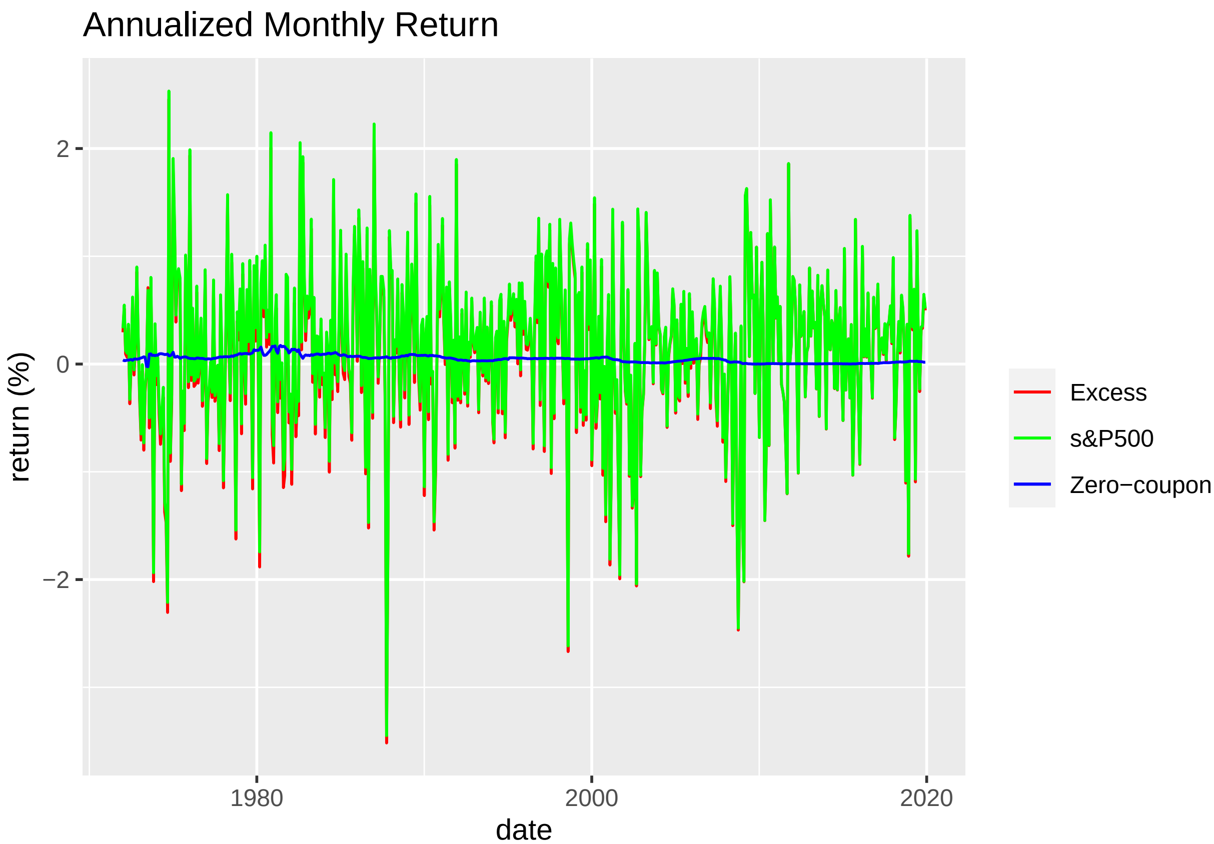
Plotted below is the annualized daily arithmetic mean return (green) vs date in comparison with the annualised daily return from zero coupon bond (blue). From the plot, we can see that the excess returns (red) are almost equal to the absolute value of annualized daily return since the annualised daily return is significantly high (or low) in comparison with the zero coupon return.

```r
ggplot(combined, aes(date), ylabel) +
  geom_line(aes(y = excess_daily, color = "Excess")) +
  geom_line(aes(y = sp_daily, color = "s&P500")) +
  geom_line(aes(y = zero_coupon_daily, color = "Zero-coupon")) +
  labs(title = "Annualized Daily Return") + ylab("return (%)") +
  scale_colour_manual("", breaks = c("Excess", "s&P500", "Zero-coupon"),
                      values = c("Excess"="red", "s&P500"="green",
                                 "Zero-coupon"="blue"))
```



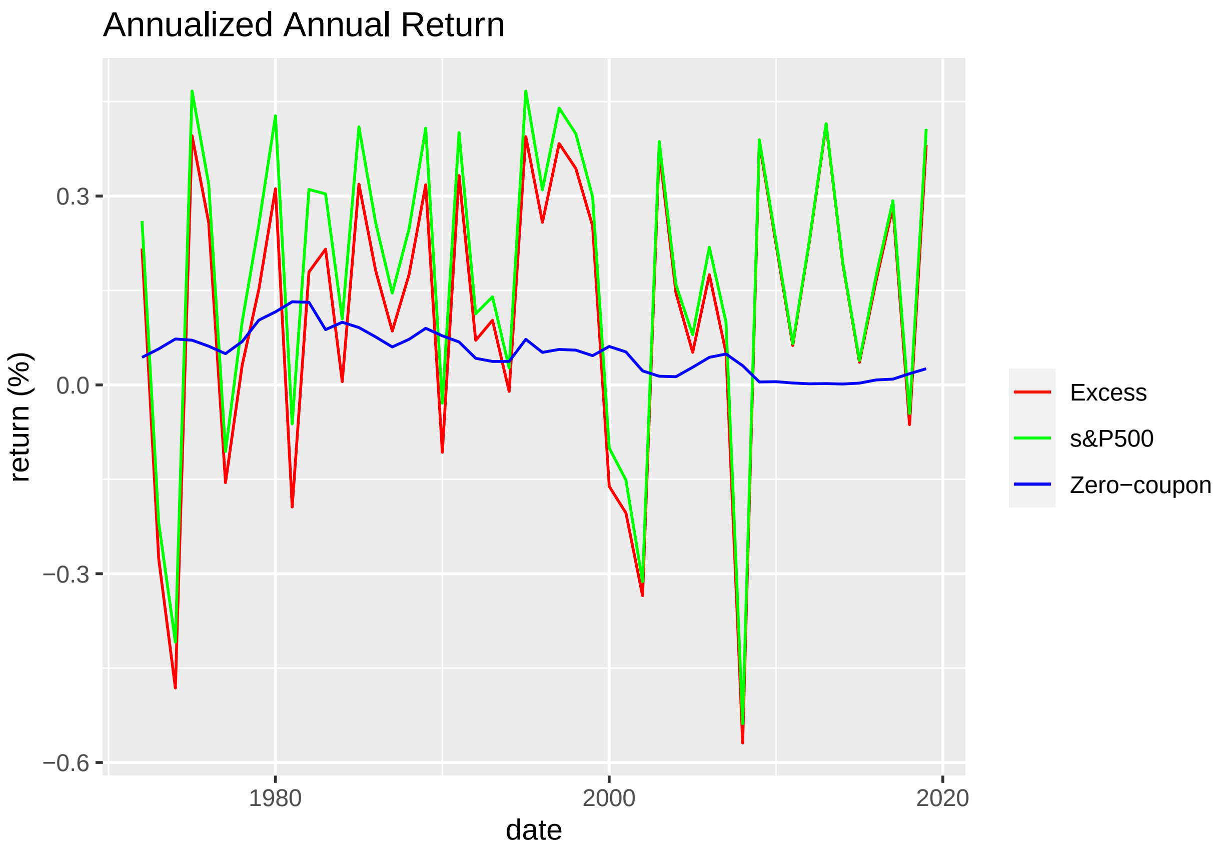
Plotted below is the annualized monthly arithmetic mean return (green) vs date in comparison with the annualised monthly return from zero coupon bond (blue). From the plot, we can see that the excess returns (red) are almost equal to the absolute value of annualized monthly return since the annualised monthly return is significantly high (or low) in comparison with the zero coupon return.

```r
ggplot(subset(combined, !is.na(combined$excess_monthly_arithmetic)), aes(date), ylabel) +
  geom_line(aes(y = excess_monthly_arithmetic, color = "Excess")) +
  geom_line(aes(y = sp_monthly_arithmetic, color = "s&P500")) +
  geom_line(aes(y = zero_coupon_monthly, color = "Zero-coupon")) +
  labs(title = "Annualized Monthly Return") + ylab("return (%)") +
  scale_colour_manual("", breaks = c("Excess", "s&P500", "Zero-coupon"),
                      values = c("Excess"="red", "s&P500"="green",
                                 "Zero-coupon"="blue"))
```



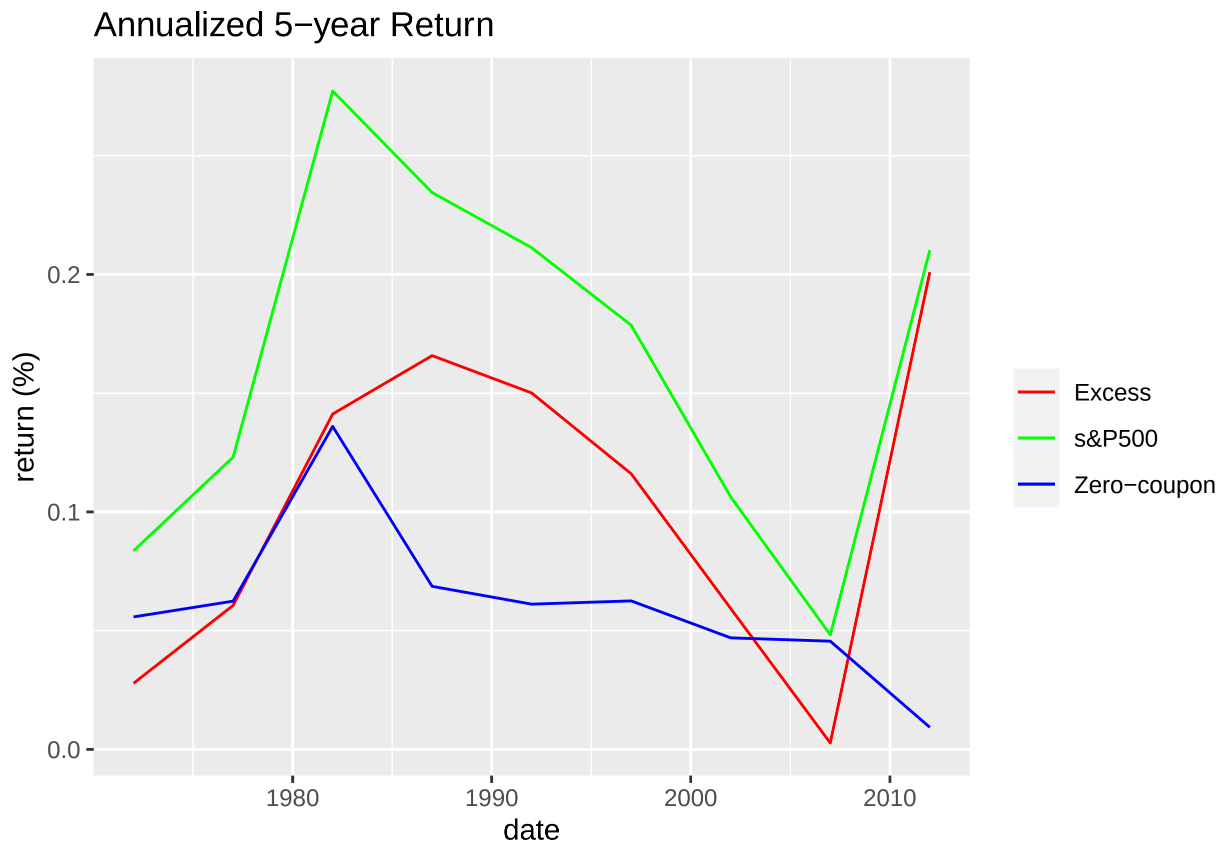
Plotted below is the annualized yearly arithmetic mean return (green) vs date in comparison with the annualised yearly return from zero coupon bond (blue). Excess returns (red) are plotted.

```
ggplot(subset(combined, !is.na(combined$excess_annual_arithmetic)), aes(date), ylabel) +
  geom_line(aes(y = excess_annual_arithmetic, color = "Excess")) +
  geom_line(aes(y = sp_annual_arithmetic, color = "s&P500")) +
  geom_line(aes(y = zero_coupon_annual, color = "Zero-coupon")) +
  labs(title = "Annualized Annual Return") + ylab("return (%)") +
  scale_colour_manual("", breaks = c("Excess", "s&P500", "Zero-coupon"),
                      values = c("Excess"="red", "s&P500"="green",
                                 "Zero-coupon"="blue"))
```



Plotted below is the annualized 5-year arithmetic mean return (green) vs date in comparison with the annualised 5-year return from zero coupon bond (blue). Excess returns (red) are plotted.

```
ggplot(subset(combined, !is.na(combined$excess_5year_arithmetic)), aes(date), ylabel) +
  geom_line(aes(y = excess_5year_arithmetic, color = "Excess")) +
  geom_line(aes(y = sp_5year_arithmetic, color = "s&P500")) +
  geom_line(aes(y = zero_coupon_5year, color = "Zero-coupon")) +
  labs(title = "Annualized 5-year Return") + ylab("return (%)") +
  scale_colour_manual("", breaks = c("Excess", "s&P500", "Zero-coupon"),
                      values = c("Excess"="red", "s&P500"="green",
                                 "Zero-coupon"="blue"))
```



(c) Assume that the your coefficient of risk aversion, A, is equal to 4. How will you allocate your capital between bonds and the index? HINT: Arithmetic returns are more appropriate than geometric returns for this question.

Capital Allocation:

Assuming that the utility follows the function:

$$U(R_{t+1}, w_t) = w_t(E_t(R_{r,t+1}) - R_{f,t}) + R_{f,t} - \frac{A}{2} w_t^2 V_t(R_{r,t+1})$$

Maximizing this utility, we get the weight of the portfolio in S&P500 as:

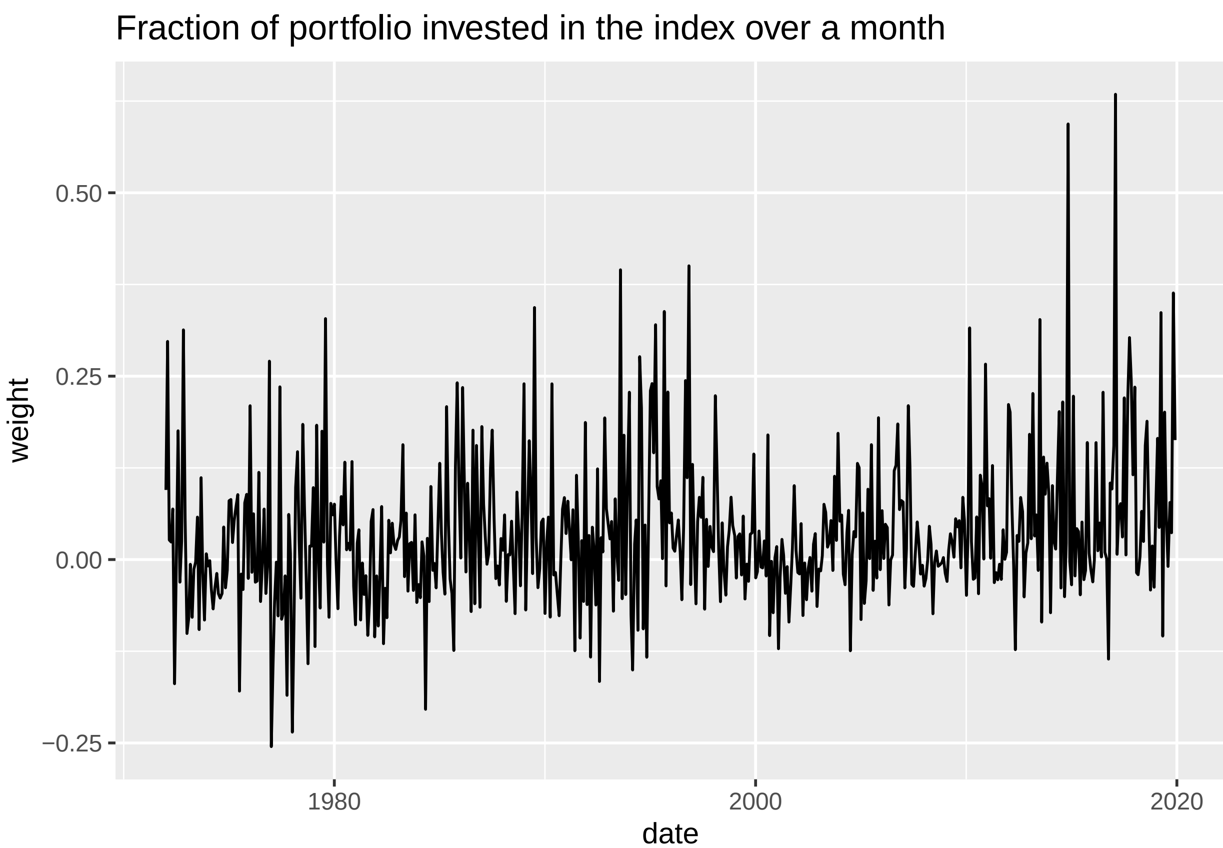
$$w_t^* = \frac{E_t(R_{r,t+1}) - R_{f,t}}{A.V_t(R_{r,t+1})}$$

Note that $V_t(R_{r,t+1})$ is the variance of returns for the period under consideration.

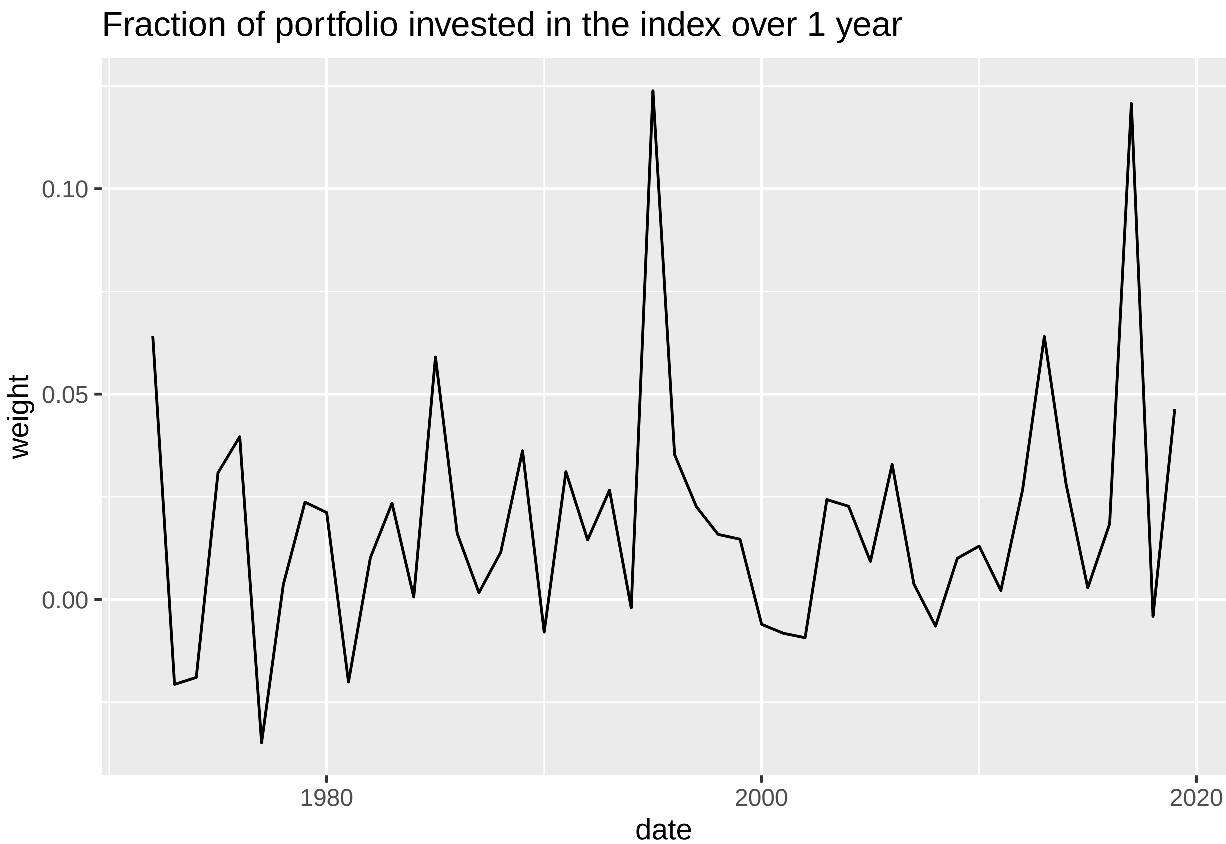
From the weights obtained, we can infer that the weight of the portfolio in index and stock differs as the risk aversion differs and also with the investment horizon. Plotted below are the weights of the portfolio in the S&P500 index that an investor with risk aversion A = 4, should be investing, based on his investment horizon in a particular time period between 1/3/1972 through 12/31/2019.

```
a <- 1;
ep <- endpoints(ts, on ='years', k=5)
combined <- combined %>%
  mutate(w_monthly = excess_monthly_arithmetic/(a*var_sp_monthly),
         w_annual = excess_annual_arithmetic/(a*var_sp_annual),
         w_5year = excess_5year_arithmetic/(a*var_sp_5year))
```

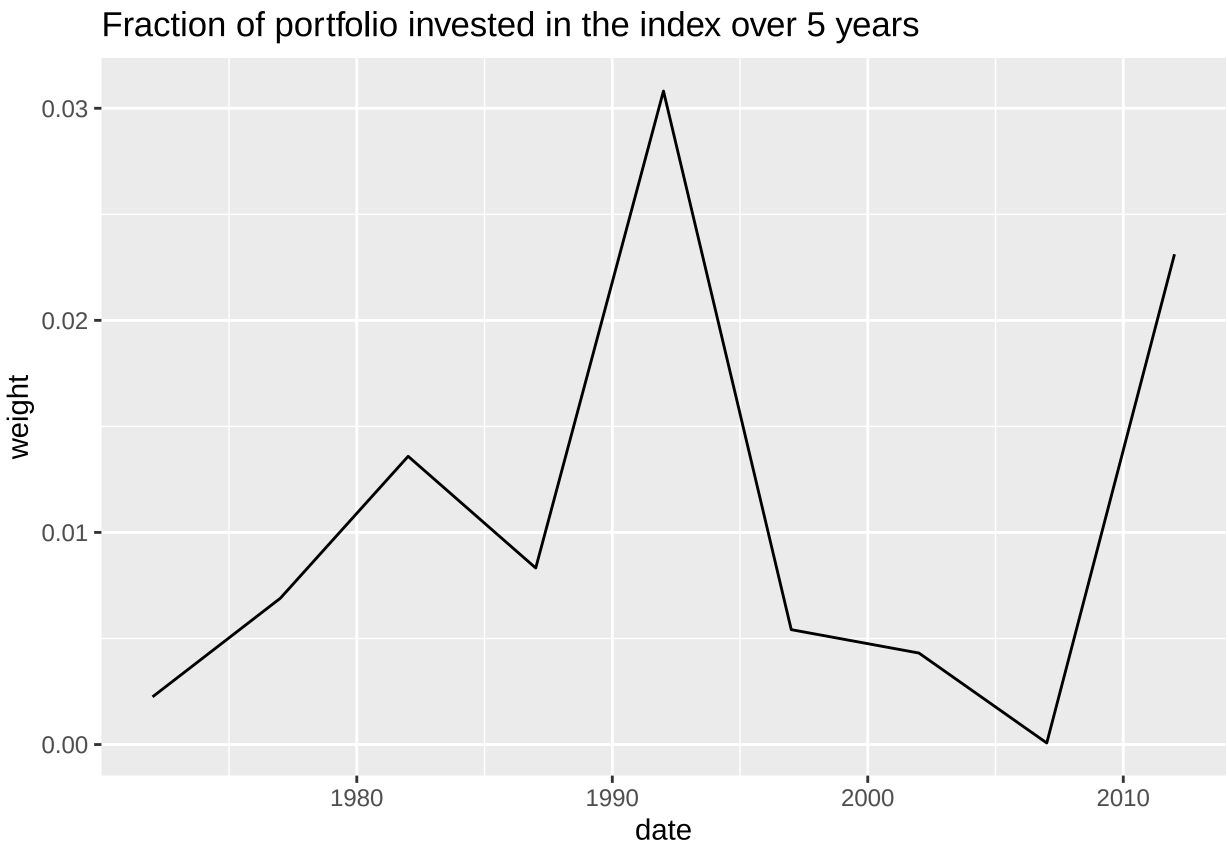
```r
ggplot(subset(combined, !is.na(combined$excess_monthly_arithmetic)), aes(date), ylabel) +
  geom_line(aes(y = w_monthly)) +
  labs(title = "Fraction of portfolio invested in the index over a month") + ylab("weight")
```




```r
ggplot(subset(combined, !is.na(combined$excess_annual_arithmetic)), aes(date), ylabel) +
  geom_line(aes(y = w_annual)) +
  labs(title = "Fraction of portfolio invested in the index over 1 year") + ylab("weight")
```




```
ggplot(subset(combined, !is.na(combined$excess_5year_arithmetic)), aes(date), ylabel) +
  geom_line(aes(y = w_5year)) +
  labs(title = "Fraction of portfolio invested in the index over 5 years") + ylab("weight")
```



Based on the weight of the portfolio in the S&P500 index, the weight of portfolio in bonds is obtained by computing $1 - w_t^*$.