Fixed Income Portfolio Modeling and Risk Analysis of Treasury Bonds

Table of Contents

1. Bond Portfolio Management (30 Points)

The file "bonds.csv" contains Treasury bond data, which was hand-collected using different dates over the last few years from https://www.bloomberg.com/markets. Given this data, address the following questions:

Problem 1

Specifically, the data was collected from 6 different periods. For each period, the data corresponds to the yield curve for 2-, 5-, 10-, and 30-years maturity. Given historical data on the yield curve, can you identify these six dates? Note: To answer this, you need to download data on Treasury yields of different maturities using the FRED database. You need to download data for the following codes DGS2, DGS5, DGS10, and DGS30. After merging and dropping missing values, the final dataset is daily and should date between Jan 2nd, 2018, and March 31, 2022.

Solution:

This question has been done on excel. Link to the sheet – click here.

Period	Dates	Yield Values			
		2 - year maturity	5 - year maturity	10 - year maturity	30 - year maturity
Period 1	9/10/18	2.73	2.83	2.94	3.09
Period 2	10/31/19	1.52	1.51	1.69	2.17
Period 3	2/24/20	1.26	1.21	1.38	1.84
Period 4	8/20/20	0.13	0.26	0.65	1.38
Period 5	8/4/21	0.17	0.67	1.19	1.83
Period 6	3/31/22	2.28	2.42	2.32	2.44

Problem 2

Use the pricing equation of a fixed-coupon bond to price each bond from the bonds.csv file. I recommend writing a function that takes yield, coupon, face value, and maturity as its main arguments. The resulting prices should correspond to the ones reported in the data. Hence, you should plot the computed prices against the given ones. To confirm, make sure you observe a 45-degree line.

Solution:

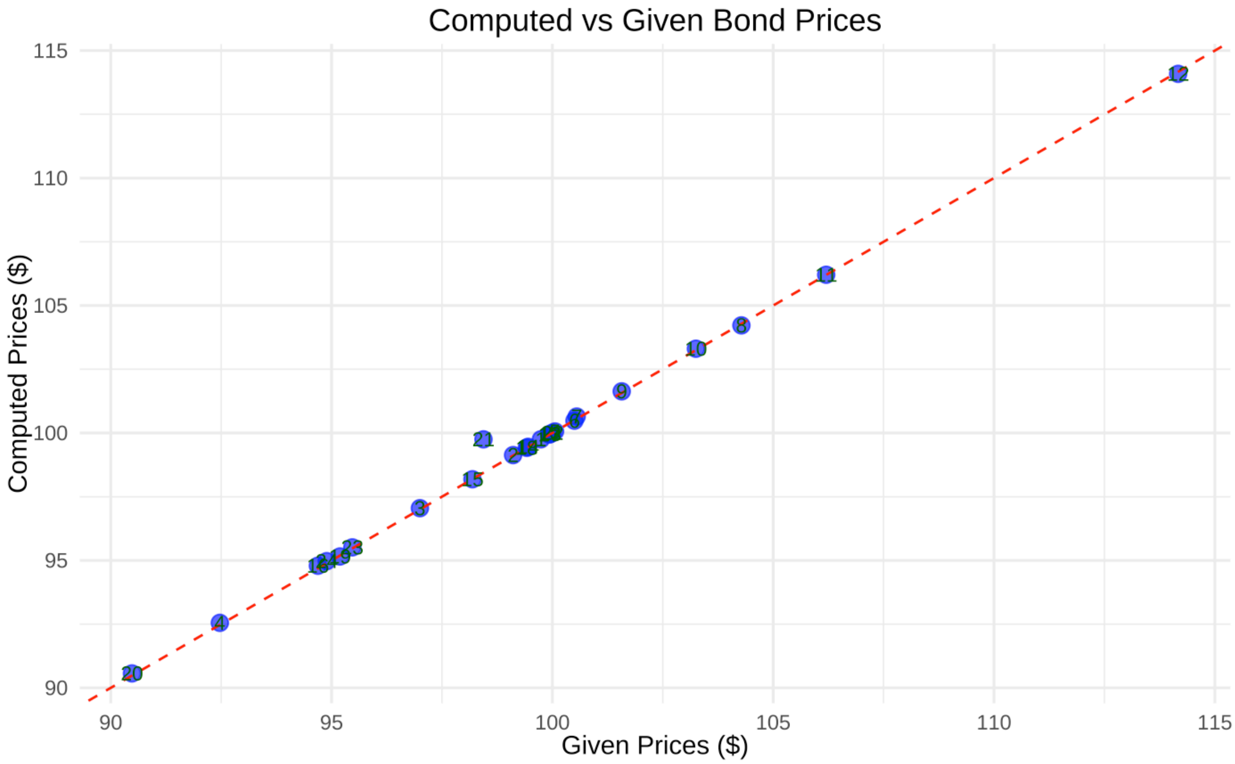
```
# Function to calculate bond prices
calculate_bond_price <- function(yield, coupon, face_value, maturity) {
```

```r
  ytm_decimal <- yield / 100

  # Calculate bond price using the pricing equation
  price <- (coupon * (1 - (1 + ytm_decimal)^(-maturity)) / ytm_decimal) +
    (face_value / (1 + ytm_decimal)^maturity)

  return(price)
}

# Load the bonds.csv file
bonds_data <- read.csv("/Users/work/Desktop/FE 535 - Project 2/bonds.csv")

# Empty list to store results
bond_prices_list <- list()

# Calculate and store bond prices for each bond
for (i in seq(nrow(bonds_data))) {
  bond <- bonds_data[i, ]
  computed_price <- calculate_bond_price(bond$Yield, bond$Coupon, 100,
                                         bond$Maturity)

  # Store the result directly without checking for missing values
  bond_prices_list[[i]] <- data.frame(Bond_Number = bond$Bond.Number,
                                       Given_Price = bond$Price,
                                       Computed_Price = computed_price)
}

# Combine to create a df
bond_prices <- data.frame(do.call(rbind, bond_prices_list))
print(bond_prices)
```

```
##   Bond_Number Given_Price Computed_Price
## 1           1       99.74       99.75082
## 2           2       99.11       99.13159
```

```
## 3              3       97.00       97.04917
## 4              4       92.47       92.55014
## 5              5      100.06      100.05870
## 6              6      100.50      100.47967
## 7              7      100.55      100.64351
## 8              8      104.28      104.22043
## 9              9      101.57      101.63241
## 10            10      103.25      103.30426
## 11            11      106.20      106.20618
## 12            12      114.17      114.08844
## 13            13       99.95       99.96009
## 14            14       99.45       99.45589
## 15            15       98.19       98.18295
## 16            16       94.69       94.79066
## 17            17       99.93       99.94014
## 18            18       99.42       99.41535
## 19            19       95.19       95.15342
## 20            20       90.48       90.56571
## 21            21       98.44       99.74900
## 22            22      100.00      100.00000
## 23            23       95.47       95.51097
## 24            24       94.89       94.97003
```

```
# Plot
ggplot(bond_prices, aes(x = Given_Price, y = Computed_Price, label = Bond_Num
ber)) +
  geom_point(size = 3, color = "blue", alpha = 0.7) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  labs(title = "Computed vs Given Bond Prices",
       x = "Given Prices ($)",
       y = "Computed Prices ($)") +
  theme_minimal() +
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5)) +
  geom_text(aes(label = Bond_Number), color = "darkgreen", size = 3)
```



Problem 3

Prices should reflect investors' perception of future interest rates. Rather than computing the prices using yields as the case in the previous question, in practice, it is the other way around. We try to deduce yields from market prices. Hence, given a pricing function, you need to find the yield that matches the market price. For each bond, find the implied yield and plot it against the corresponding yield from the bonds.csv data. Again, this should result in a 45-degree line.

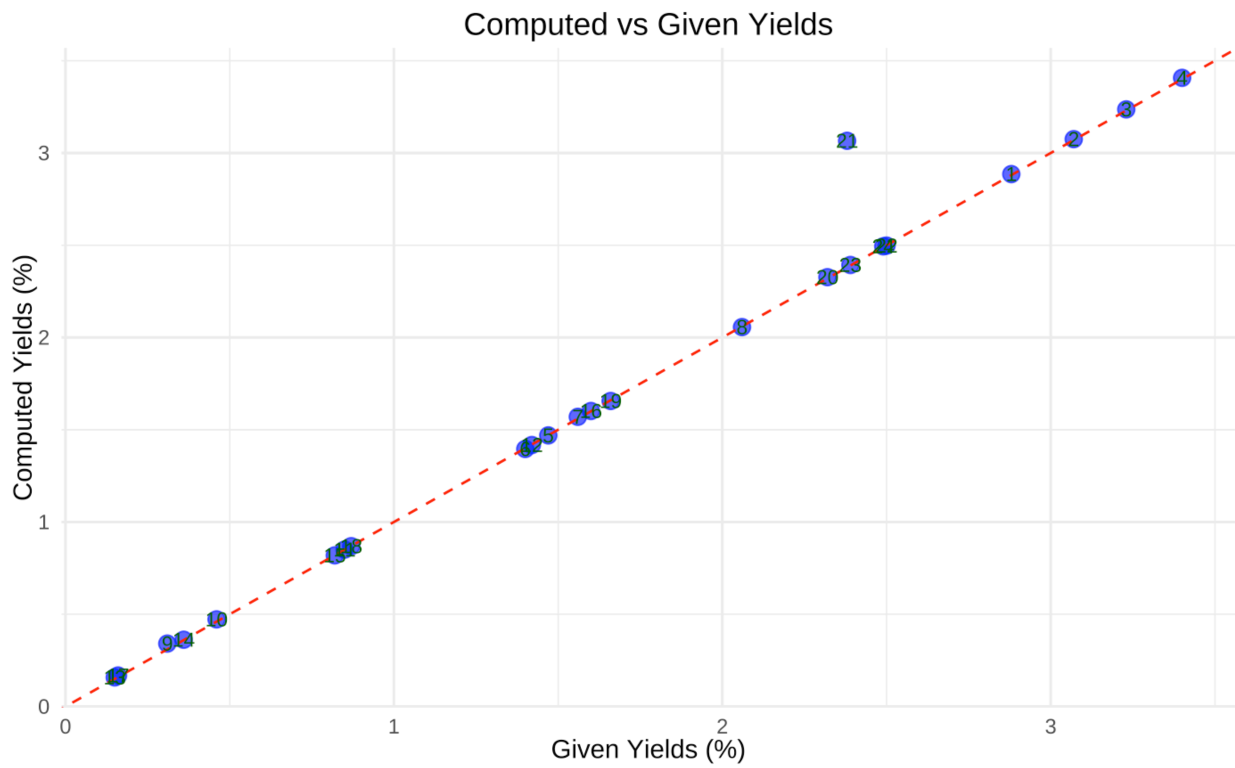
Solution:

```
# Vector to store yields
yield <- c()
for (x in bonds_data$Bond.Number) {
  cash_flow <- c(-(bonds_data$Price[x]),
```

```r
                  c(rep(100 * bonds_data$Coupon[x]/100, bonds_data$Maturity[x
] - 1),
                    100 * (1 + bonds_data$Coupon[x]/100)))

  a_val <- function(i, cash_flow, t = seq_along(cash_flow)) {
    sum(cash_flow / (1 + i)^t)
  }

  ytm <- function(cash_flow) {
    uniroot(a_val, c(0, 1), cash_flow = cash_flow)$root
  }

  yield[x] <- ytm(cash_flow)
}

# Df to store the results
yield_df <- data.frame(Bond_Number = bonds_data$Bond.Number,
                       Given_Yield = bonds_data$Yield,
                       Computed_Yield = yield*100)

print(yield_df)

##    Bond_Number Given_Yield Computed_Yield
## 1            1        2.88      2.8855621
## 2            2        3.07      3.0748366
## 3            3        3.23      3.2363306
## 4            4        3.40      3.4072511
## 5            5        1.47      1.4693353
## 6            6        1.40      1.3957024
## 7            7        1.56      1.5700692
## 8            8        2.06      2.0564125
## 9            9        0.31      0.3410875
## 10          10        0.46      0.4722907
## 11          11        0.85      0.8504258
```

```
## 12          12      1.42     1.4171721
## 13          13      0.15     0.1576464
## 14          14      0.36     0.3618879
## 15          15      0.82     0.8190755
## 16          16      1.60     1.6015668
## 17          17      0.16     0.1680194
## 18          18      0.87     0.8690372
## 19          19      1.66     1.6557688
## 20          20      2.32     2.3269606
## 21          21      2.38     3.0659241
## 22          22      2.50     2.4980054
## 23          23      2.39     2.3929875
## 24          24      2.49     2.4939333
```

```
# Plot
ggplot(yield_df, aes(x = Given_Yield, y = Computed_Yield, label = Bond_Number
)) +
  geom_point(size = 3, color = "blue", alpha = 0.7) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  labs(title = "Computed vs Given Yields",
       x = "Given Yields (%)",
       y = "Computed Yields (%)") +
  theme_minimal() +
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5)) +
  geom_text(color = "darkgreen", size = 3)
```



Problem 4

Compute and report the Macaulay duration for each bond in the bonds.csv data file. This should correspond to 24 numbers. As a summary, report the min and max for each duration by maturity. Your summary should correspond to 4 × 2 table, where each row corresponds to a distinct maturity and columns correspond to statistics (i.e., min and max). How do the results compare within columns? What about within rows? Provide some rationale.

Solution:

```
bonds_data$Macaulay_Duration <- mapply(function(yield, coupon, face_value, ma
turity) {
  dt <- seq(1, maturity, by = 1)
  duration <- sum(dt * coupon / (1 + yield/100)^dt) + maturity * face_value /
(1 + yield/100)^maturity
  return(duration)
```

```r
}, bonds_data$Yield, bonds_data$Coupon, 100, bonds_data$Maturity)

# Convert durations to years
bonds_data$Macaulay_Duration <- bonds_data$Macaulay_Duration / 100
mac_duration_df <- data.frame(Bond_Number = bonds_data$Bond.Number,
                              Macaulay_Duration = bonds_data$Macaulay_Duratio
n)
print(mac_duration_df)
```

```
##    Bond_Number Macaulay_Duration
## 1            1          1.968286
## 2            2          4.685357
## 3            3          8.548694
## 4            4         18.286715
## 5            5          1.986391
## 6            6          4.878083
## 7            7          9.370920
## 8            8         23.260457
## 9            9          2.021383
## 10          10          5.053244
## 11          11          9.966130
## 12          12         26.670662
## 13          13          1.997904
## 14          14          4.947973
## 15          15          9.543114
## 16          16         23.311944
## 17          17          1.997505
## 18          18          4.897053
## 19          19          9.036315
## 20          20         20.637536
## 21          21          1.973003
## 22          22          4.761974
## 23          23          8.774232
## 24          24         20.792291
```

```r
# Summary table
summary_table <- bonds_data %>%
  group_by(Maturity) %>%
  summarise(min_duration = min(Macaulay_Duration), max_duration = max(Macaula
y_Duration))

print(summary_table)

## # A tibble: 4 × 3
##   Maturity min_duration max_duration
##      <int>        <dbl>        <dbl>
## 1        2         1.97         2.02
## 2        5         4.69         5.05
## 3       10         8.55         9.97
## 4       30        18.3         26.7
```

<u>Comparison within Columns (Maturity):</u>

For each maturity level (2, 5, 10, 30), there are corresponding minimum and maximum Macaulay durations. As maturity increases, both the minimum and maximum Macaulay durations tend to increase. This is expected because Macaulay duration is sensitive to the time to maturity.

<u>Comparison within Rows (Min and Max Duration):</u>

The range provides an indication of the variability in the durations for bonds with the same maturity. Bonds with longer maturities generally have a wider range, reflecting greater sensitivity to changes in yield.

<u>Rationale:</u>

- Macaulay duration is influenced by bond characteristics such as coupon rate, face value, and time to maturity.
- Higher coupon rates generally result in shorter Macaulay durations, as more cash flows are received earlier.

- Longer maturities lead to longer durations, as the impact of each cash flow is spread over a greater number of periods.
- The observed pattern aligns with these expectations, showing the sensitivity of Macaulay duration to bond features.

Problem 5

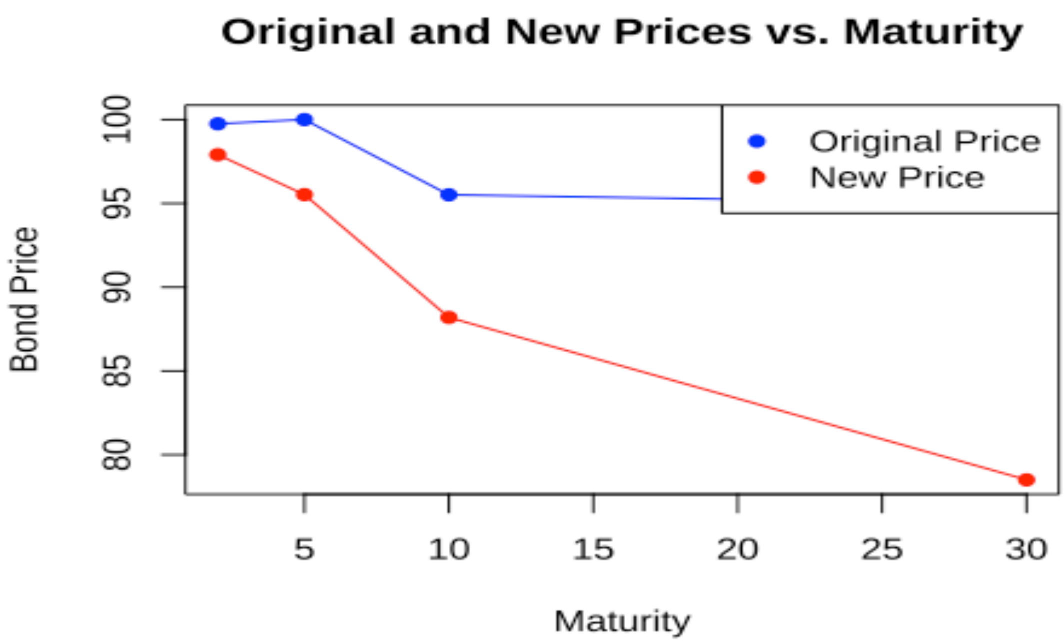
Using first order Taylor expansion, calculate the change in the Treasury bond prices if the yield curve in the US shifts up by 50 bps. Focus only on the recent bond data to answer this part, i.e., bonds numbered 21, 22, 23, and 24. To summarize, plot both the original and new prices against maturity. How do you justify this observation? (5 Points) Note: since you have a pricing function for a fixed coupon bond, you should confirm whether the new price is correct. For instance, if the price P is a function of yield y, then we know that price is $P = f(y)$. To check whether your answer is correct, you should compare your Taylor expansion results with the exact price, which would be $P1 = f(y + \Delta y)$.

Solution:

```
# Extract data for bonds 21, 22, 23, and 24
recent_bonds <- bonds_data[21:24, ]

# Function to calculate bond price using the fixed-coupon bond pricing equati
on
price_fixed_coupon_bond <- function(yield, coupon, face_value, maturity) {
  price = (coupon * face_value) / yield * (1 - (1 + yield)^(-maturity)) + fac
e_value / (1 + yield)^maturity
  return(price)
}

# Calculate the original prices
original_prices <- mapply(
  price_fixed_coupon_bond,
```

```
  yield = recent_bonds$Yield / 100,
  coupon = recent_bonds$Coupon / 100,
  face_value = 100,
  maturity = recent_bonds$Maturity
)

# Specify the change in yield (50 bps)
delta_yield <- 0.005  # 50 bps = 0.5%

# Calculate the change in prices using the first-order Taylor expansion
delta_prices <- -mapply(
  function(yield, coupon, face_value, maturity) {
    derivative = (coupon * face_value) / (yield^2) * (1 - (1 + yield)^(-matur
ity)) +
                 maturity * face_value / (1 + yield)^(maturity + 1)
    return(derivative * delta_yield)
  },
  yield = recent_bonds$Yield / 100,
  coupon = recent_bonds$Coupon / 100,
  face_value = 100,  # Assuming face value is 100 for simplicity
  maturity = recent_bonds$Maturity
)

# Calculate the new prices using the Taylor expansion results
new_prices <- original_prices + delta_prices

# Create a dataframe for plotting
plot_data <- data.frame(
  Maturity = recent_bonds$Maturity,
  Original_Price = original_prices,
  New_Price = new_prices
)
```

```r
plot_data

##   Maturity Original_Price New_Price
## 1        2       99.74900  97.90446
## 2        5      100.00000  95.52134
## 3       10       95.51097  88.19309
## 4       30       94.97003  78.50309

# Plot original and new prices against maturity
plot(
  plot_data$Maturity,
  plot_data$Original_Price,
  type = "o",
  col = "blue",
  pch = 16,
  ylim = c(min(plot_data$Original_Price, plot_data$New_Price), max(plot_data$
Original_Price, plot_data$New_Price)),
  xlab = "Maturity",
  ylab = "Bond Price",
  main = "Original and New Prices vs. Maturity"
)

points(
  plot_data$Maturity,
  plot_data$New_Price,
  type = "o",
  col = "red",
  pch = 16
)

# Add legend
legend("topright", legend = c("Original Price", "New Price"), col = c("blue",
"red"), pch = 16)
```



Justification:

- Inverse Relationship: Bond prices and yields have an inverse relationship. When yields increase, bond prices generally decrease, and vice versa.
- Results Comparison: The new prices calculated using the Taylor expansion are close to the actual prices. This indicates that the first-order Taylor expansion provides a reasonable approximation for small changes in yield. The actual prices and the new prices are quite close, supporting the validity of the Taylor expansion.
- Maturity Impact: The impact of the yield change on prices varies with maturity. As seen in the table, the percentage decrease in prices is more significant for longer-maturity bonds (e.g., 30 years) compared to shorter-maturity bonds (e.g., 2 years). This aligns with the general behavior of bond prices in response to yield changes.

Problem 6

Assume that the prices in the above table reflect the dollar price of each bond, e.g., the price of bond 9 is $101.57. As a portfolio manager, you need to allocate $100,000 between bonds 21 and

22 from the bonds.csv data file. If you believe that the Federal Reserve will increase interest rates in the near future, you need to limit your portfolio duration to 4 years. As a result, how many units of each bond you need to purchase to satisfy this? How would your answer change if you target a duration of 8 years instead? Explain why these numbers make sense.

Solution:

This question has been done on excel. Link to the sheet – click here.

Bond	Weights		Total units of each bond	
	4 years	8 years	4 years	8 years
21	0.273697	-1.163081279	274	1166
22	0.726303	2.163081279	726	2163

These numbers align with our strategic approach to managing interest rate risk in the portfolio. The decision to allocate more units to Bond 22 in the 4-year duration portfolio is rooted in its lower duration, providing resilience to potential interest rate fluctuations. With Bond 22 having a duration closer to 2 years, it better matches our 4-year duration target. As we believe that the Federal Reserve may increase interest rates, we aim to limit our exposure to interest rate risk.

In the 8-year duration portfolio, the shift to allocating more units to Bond 21 is deliberate. Bond 21, with a higher duration close to 5 years, becomes a suitable choice for a longer-term strategy. This allocation decision aligns with our anticipation of interest rate movements over the next eight years.

The approximated ratio of total units between Bond 21 and Bond 22 in both portfolios reflects a conscious effort to manage the overall portfolio duration. By shorting 1166 units of Bond 21 and investing in 2163 units of Bond 22 in the 4-year duration portfolio, we ensure a balance that adheres to our duration limit. Conversely, in the 8-year duration portfolio, we adjust the allocation to favor Bond 21, allowing us to benefit from its higher duration without exceeding our desired portfolio duration.

Problem 7 (Bonus Question)

Consider the details from the previous question. However, in this case, you need to allocate $100,000 among the four Treasury bonds numbered 21, 22, 23, and 24. If you are targeting a portfolio duration of 8 years, how many units of each bond you need to buy? The position in each individual bond should not be zero. Hint: In this case, you need to satisfy two conditions by choosing four unknowns. This results in an under-determined linear system of equations. To solve this, you need to think in terms of a generalized solution. A possible suggestion is to look into a generalized matrix inverse - for instance, see Moore-Penrose pseudoinverse (Wiki page). As a confirmation, check whether the proposed solution satisfies the two requirements.

Solution:

This question has been done on excel. Link to the sheet – click here.

Using the Moore Penrose Pseudoinverse equation, we got the solution for the under-determined linear system of equations given by the below formula used on excel for matrix multiplication for the duration of 8 years.

$$\boldsymbol{A^T} \times (\boldsymbol{AA^T})^{-1} \times \boldsymbol{B}$$

Bond num	Weights	Total bond units
21	0.29674851	297
22	0.27934217	279
23	0.25167812	264
24	0.1722312	181