Basis Methods for Data Analysis

Roger D. Peng
Stephanie C. Hicks

Advanced Data Science
Term 2
2019

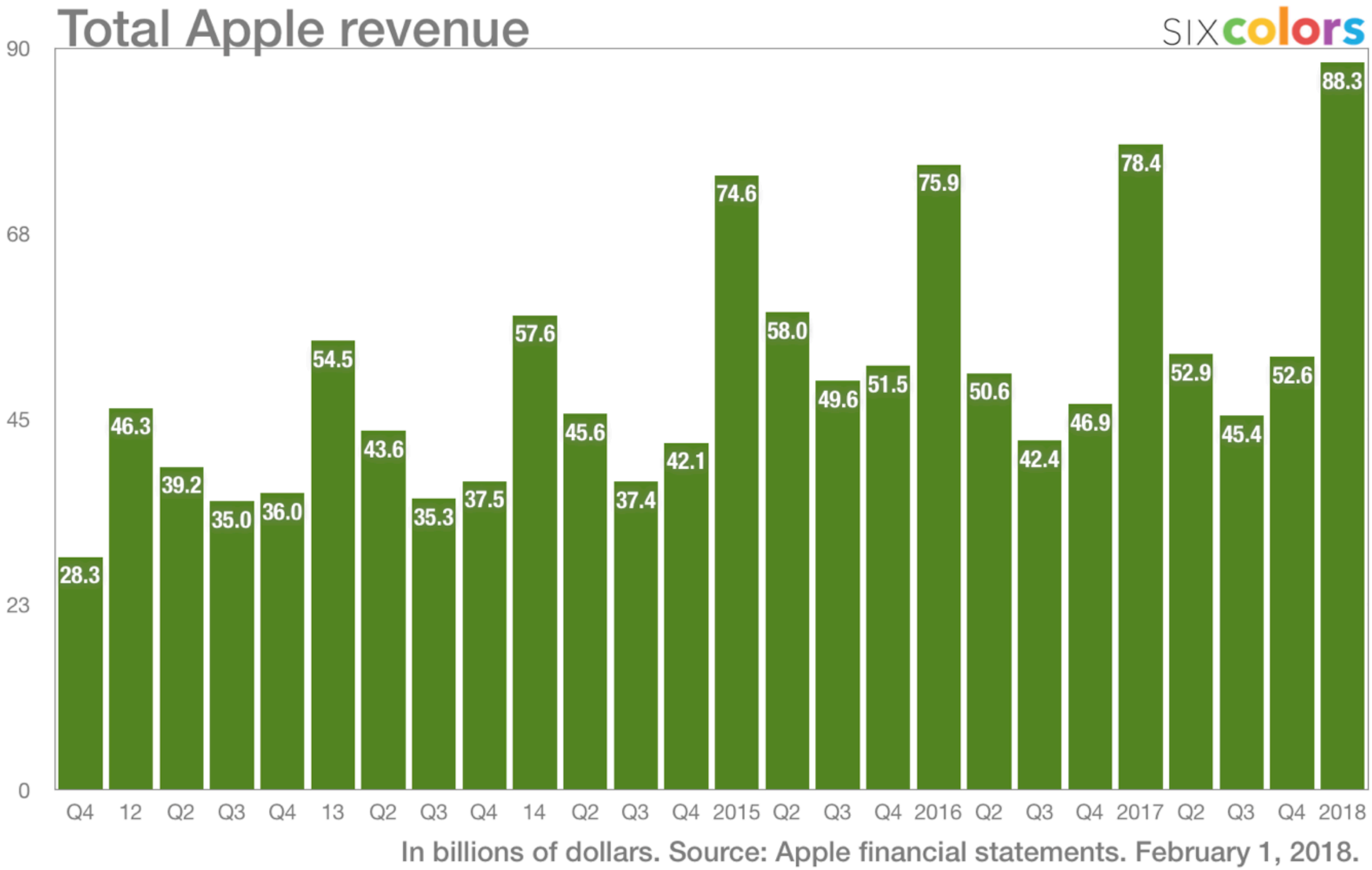
What is This?

What is This?

What is This?

What is This?

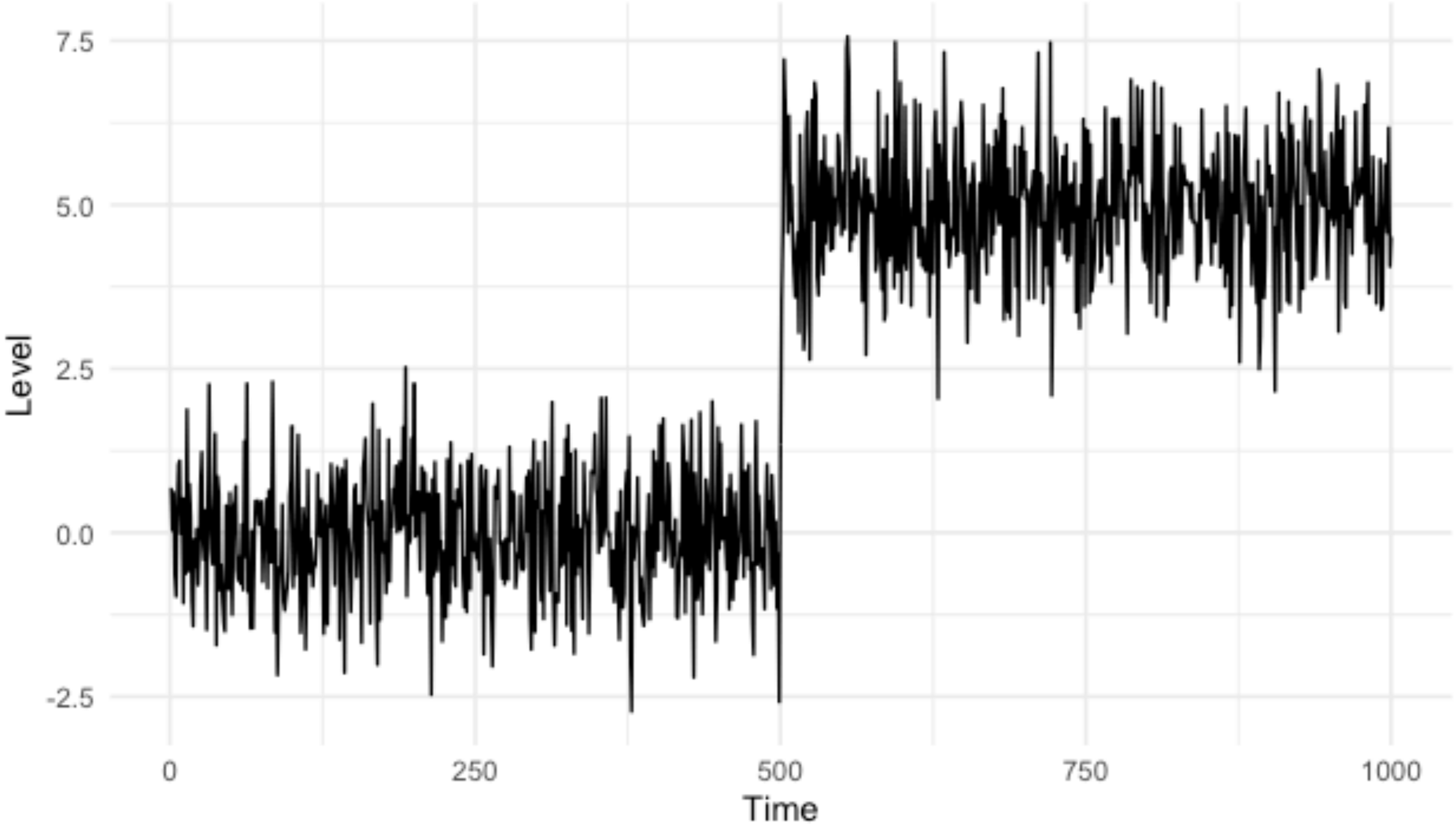
What is This?


Total Apple revenue
SIXcolors
90
68
45
23
0
28.3
46.3
39.2
35.0
36.0
54.5
43.6
35.3
37.5
57.6
45.6
37.4
42.1
74.6
58.0
49.6
51.5
75.9
50.6
42.4
46.9
78.4
52.9
45.4
52.6
88.3
Q4 12 Q2 Q3 Q4 13 Q2 Q3 Q4 14 Q2 Q3 Q4 2015 Q2 Q3 Q4 2016 Q2 Q3 Q4 2017 Q2 Q3 Q4 2018
In billions of dollars. Source: Apple financial statements. February 1, 2018.

What is This?


AAPL Adjusted Closing Price
240
200
160
120
80
2016
2018
2020
Date

What is This?



What is This?

https://apod.nasa.gov/apod/ap021017.html

What is This?

Basis Functions

- Typically known functions that are parametrized to form a dictionary of functions
- Often pre-defined: polynomial, spline, Fourier, wavelet
- Sometimes learned: PCA (but requires replicate data)
- Functions in the basis are orthonormal
- Usually massively overcomplete (require truncation/ penalization)

Basis for Bases

- If dictionary of basis functions is known, we can take it for granted
    - Data compression
    - Essentially just need to index into the library of basis functions
        - Serves a set of "surrogate covariates" that can be used anytime
- Basis functions usually have interpretable meaning
- Curve estimation or approximation
- Exploratory data analysis - feature extraction/identification
- Decompose data into "more data"

Basis Methods

- Define a dictionary of basis functions that is easily described
  - Polynomial: functions that are powers of x
  - B-Spline: cubic polynomials that are localized
  - Fourier: sines and cosines at given frequencies
  - Wavelets: localized functions that can be shrunk/stretched
- See if your data are correlated with any of the basis functions

Polynomial Basis

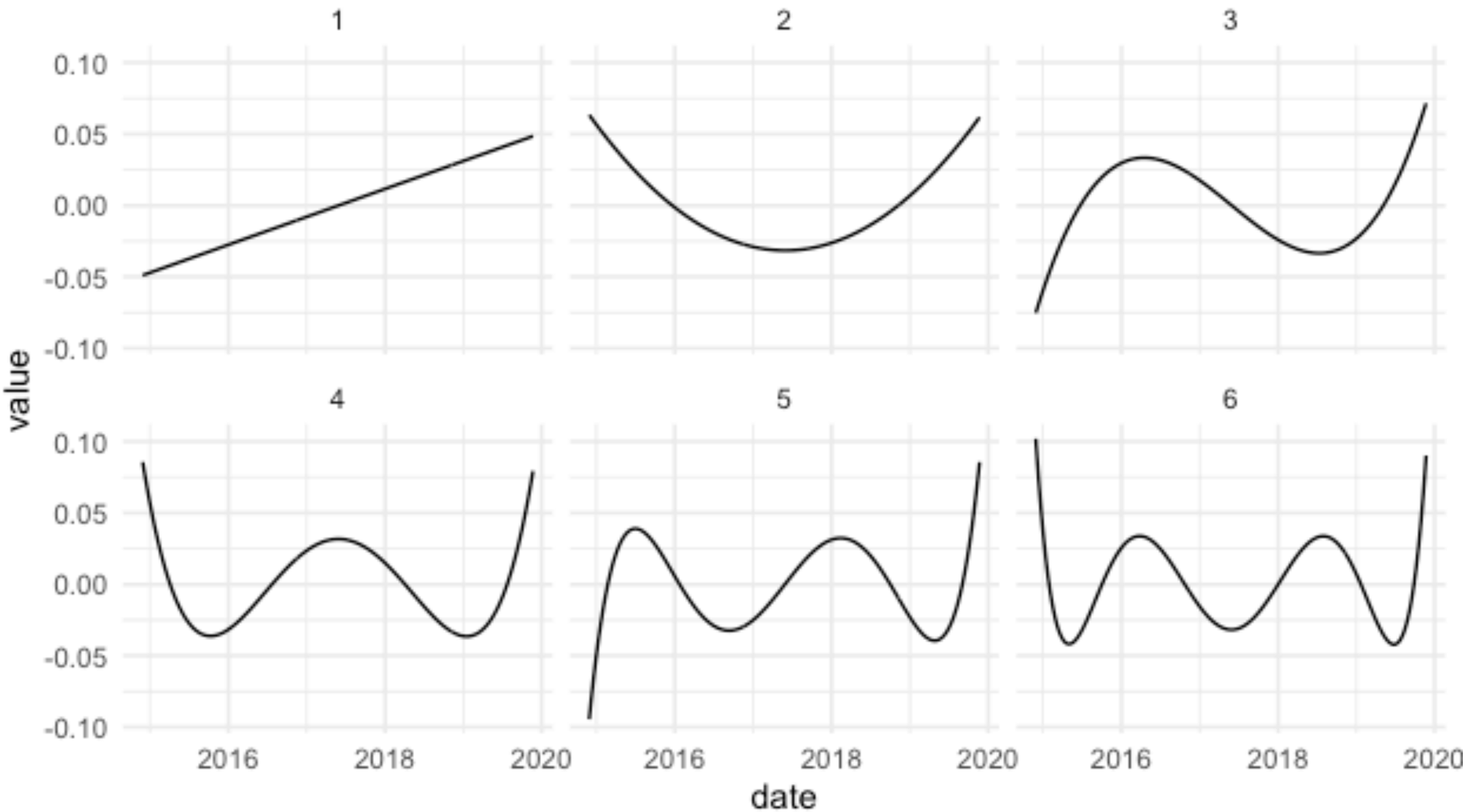
$$\mathbb{E}[y_t] = \alpha + \sum_{k=1}^{6} \beta_k \, x_t^k$$

Polynomial Basis

$$\mathbb{E}[y_t] = \alpha + \sum_{k=1}^{6} \beta_k \, x_t^k$$

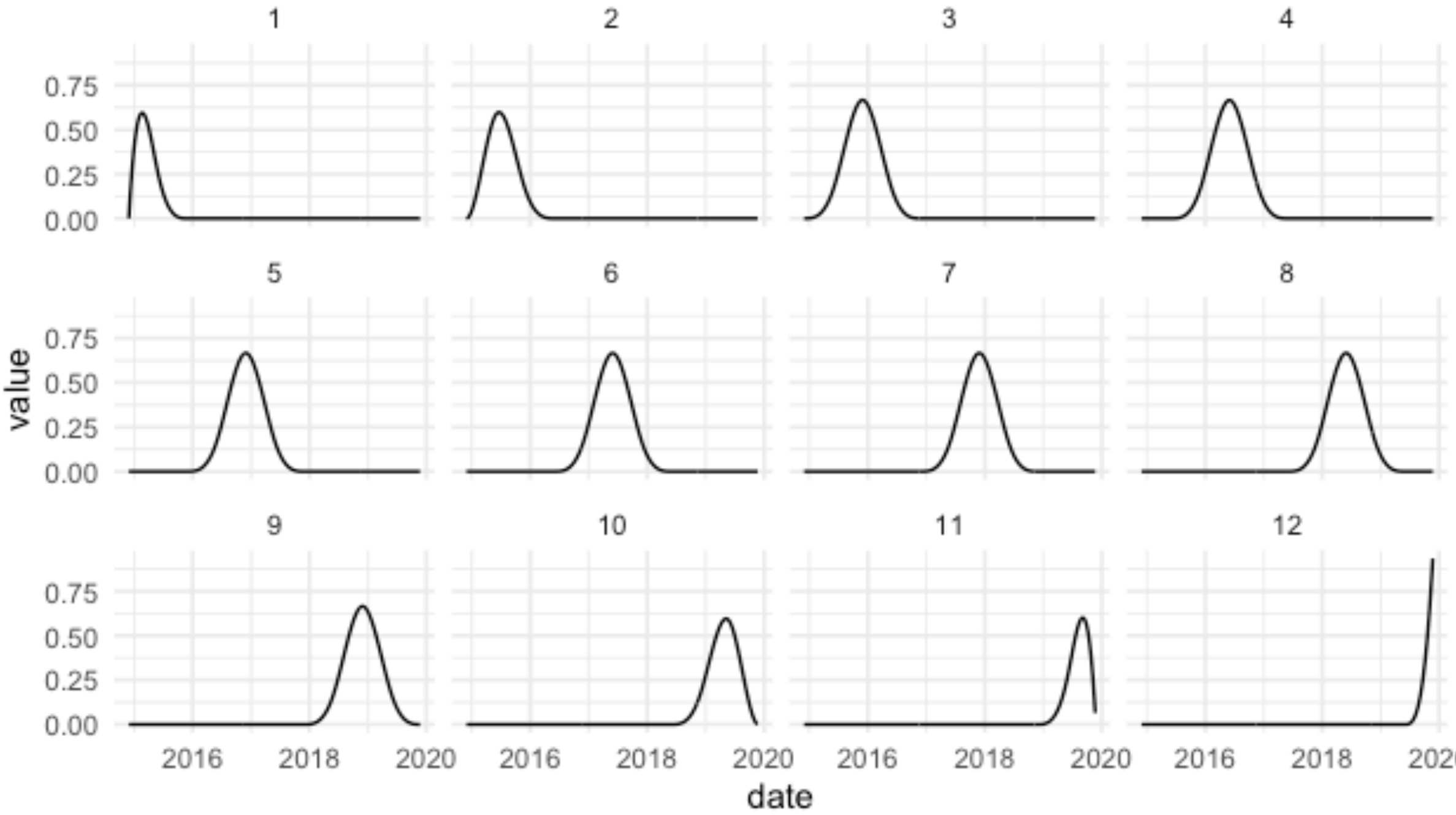
```
> fit <- lm(AAPL.Adjusted ~ poly(date, 6), g)
> tidy(fit)
# A tibble: 7 x 5
  term              estimate std.error statistic   p.value
  <chr>                <dbl>     <dbl>     <dbl>     <dbl>
1 (Intercept)          146.      0.309     473.   0.
2 poly(date, 6)1      1355.     11.0       124.   0.
3 poly(date, 6)2       349.     11.0        31.8  1.96e-163
4 poly(date, 6)3      -247.     11.0       -22.5  2.57e- 94
5 poly(date, 6)4       123.     11.0        11.2  8.12e- 28
6 poly(date, 6)5       380.     11.0        34.7  2.85e-185
7 poly(date, 6)6        41.4    11.0         3.77 1.70e-  4
```

Polynomial Basis



Polynomial Prediction

B-Spline Basis



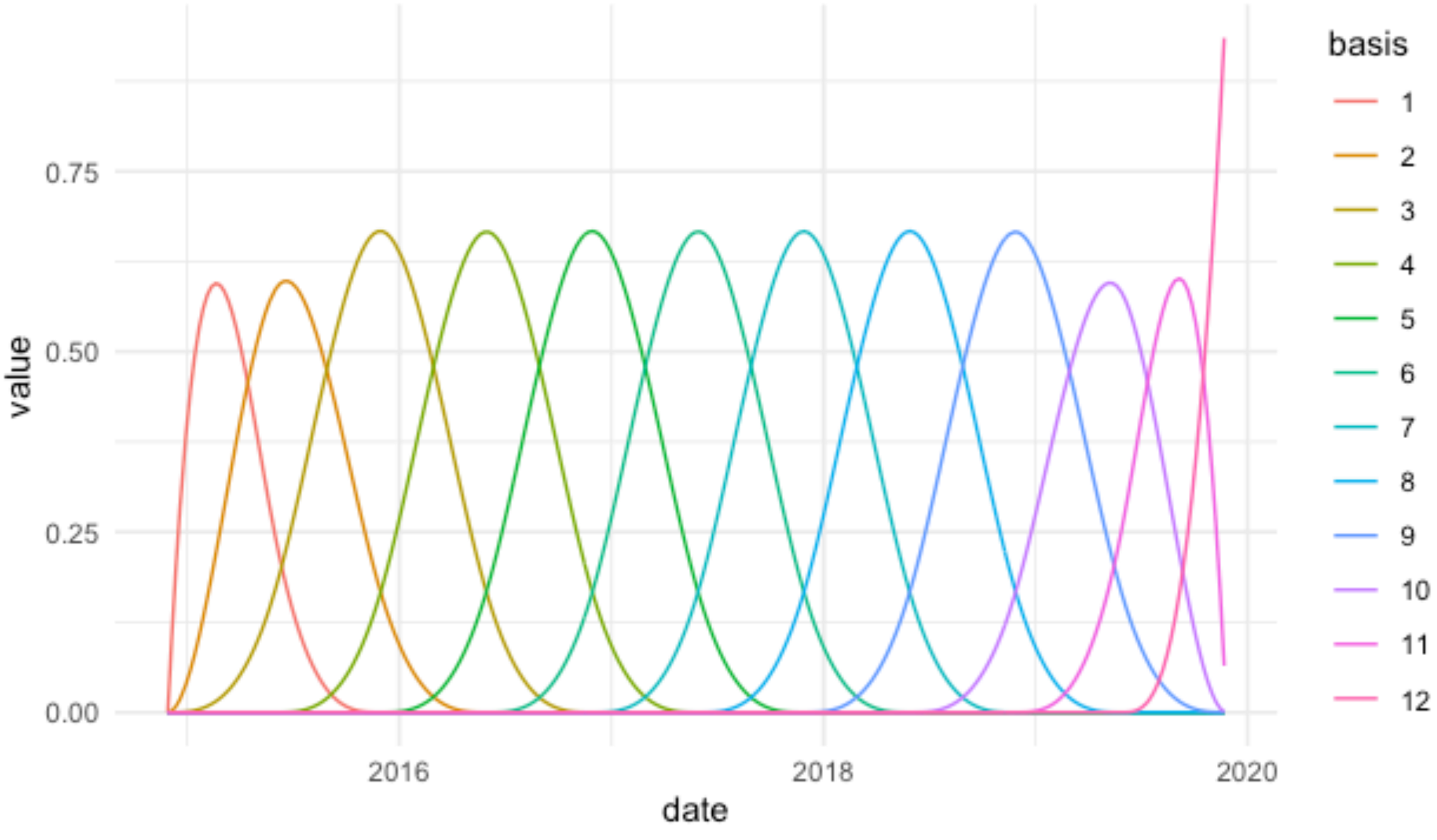
B-Spline Basis

$$\mathbb{E}[y_t] = \alpha + \sum_{k=1}^{12} \beta_k B_k(x_t)$$

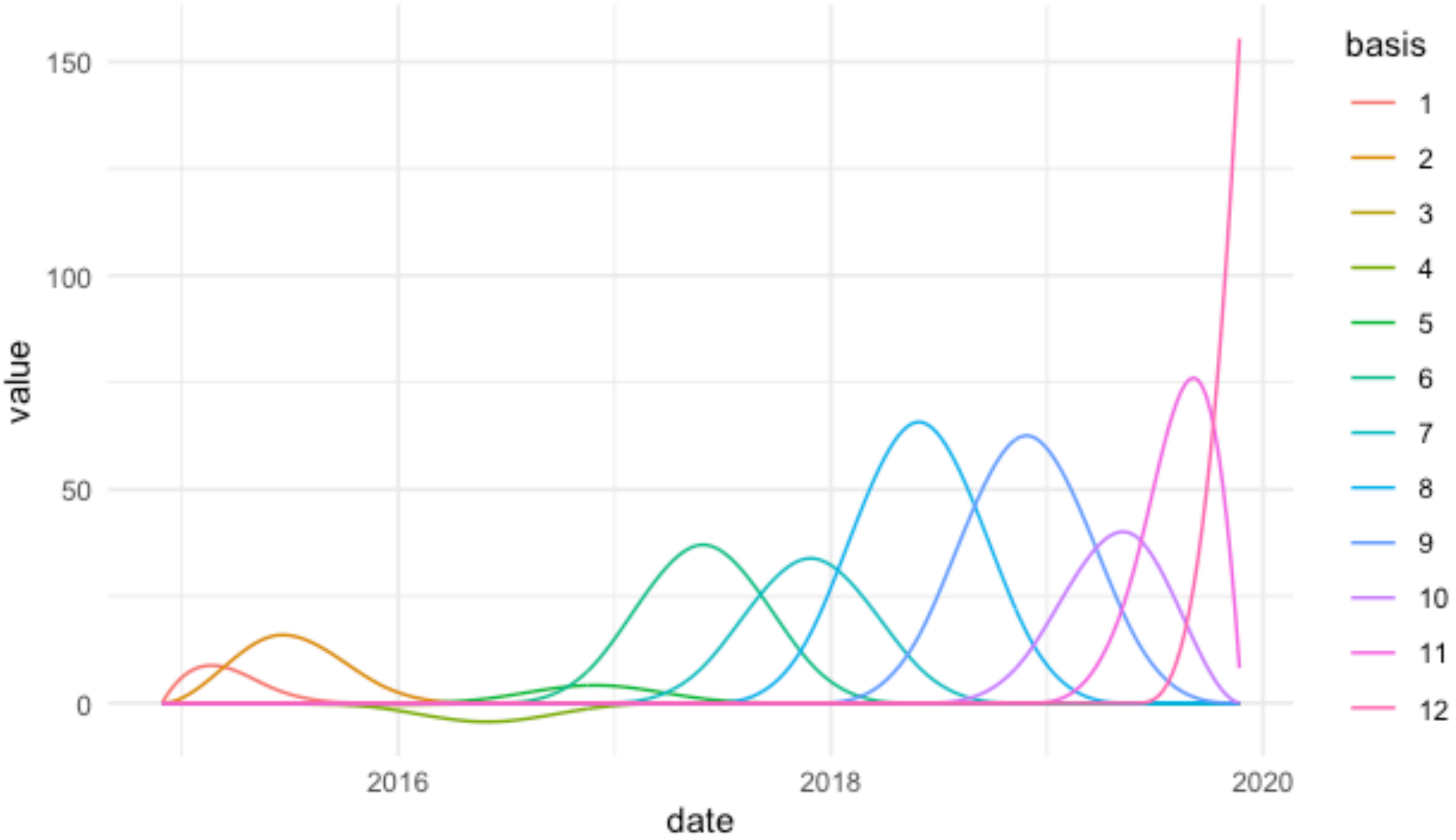
```
> fit <- lm(AAPL.Adjusted ~ bs(date, dfr), g)
> tidy(fit)
# A tibble: 13 x 5
   term              estimate std.error statistic   p.value
   <chr>                <dbl>     <dbl>     <dbl>     <dbl>
 1 (Intercept)         98.2       2.90     33.8   1.33e-178
 2 bs(date, dfr)1      14.8       5.32      2.79  5.41e-  3
 3 bs(date, dfr)2      26.7       3.36      7.95  4.15e- 15
 4 bs(date, dfr)3      -0.693     4.01     -0.173 8.63e-  1
 5 bs(date, dfr)4      -6.58      3.39     -1.94  5.21e-  2
 6 bs(date, dfr)5       6.25      3.64      1.72  8.62e-  2
 7 bs(date, dfr)6      55.6       3.49     15.9   3.27e- 52
 8 bs(date, dfr)7      50.8       3.58     14.2   1.46e- 42
 9 bs(date, dfr)8      98.6       3.56     27.7   3.42e-132
10 bs(date, dfr)9      94.0       3.69     25.4   2.39e-115
11 bs(date, dfr)10     67.3       4.09     16.5   2.34e- 55
12 bs(date, dfr)11    127.        4.23     29.9   1.01e-148
13 bs(date, dfr)12    166.        4.07     40.8   5.97e-232
```

B-Spline Prediction

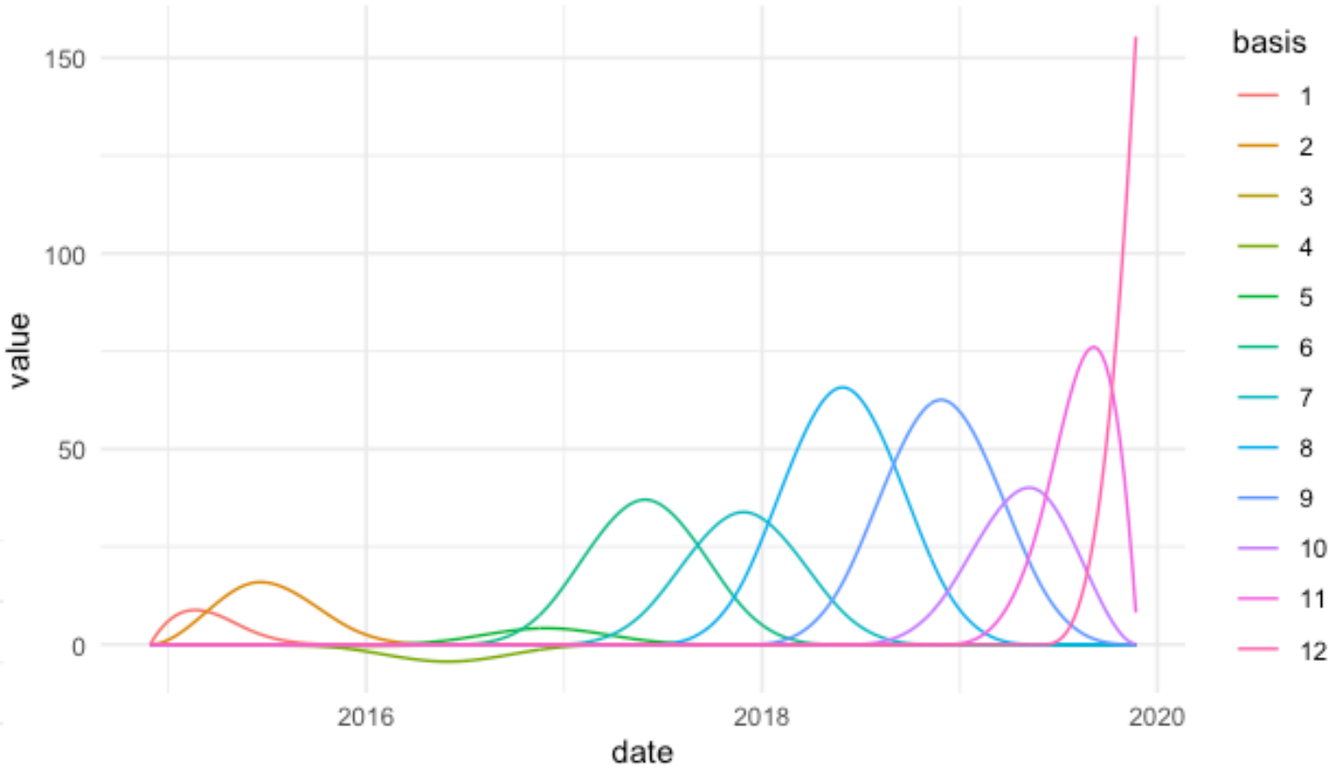
B-Spline Basis



Basis * Coef

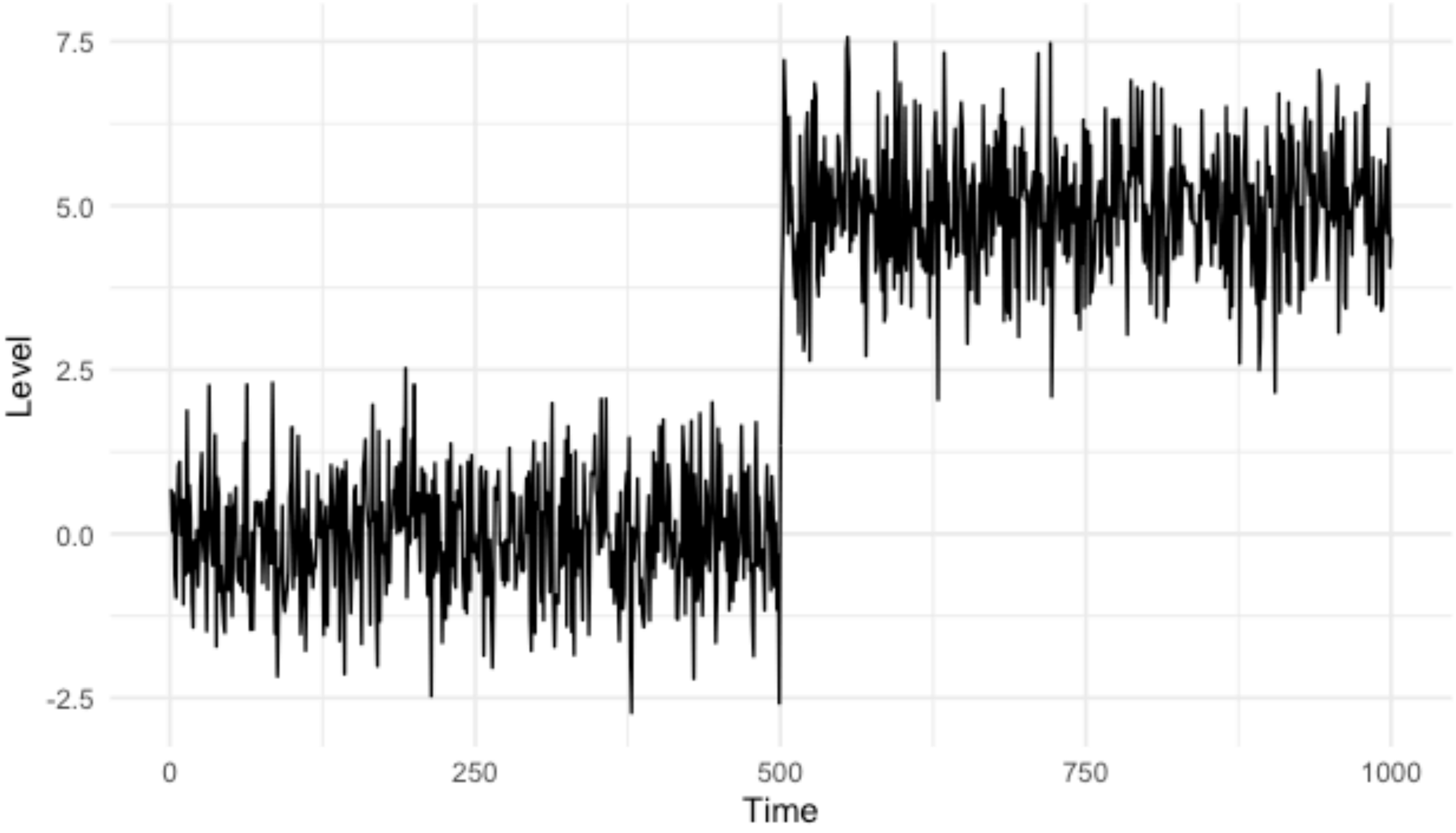

150
100
50
0
value
2016
2018
2020
date
basis
1
2
3
4
5
6
7
8
9
10
11
12

Basis * Coef

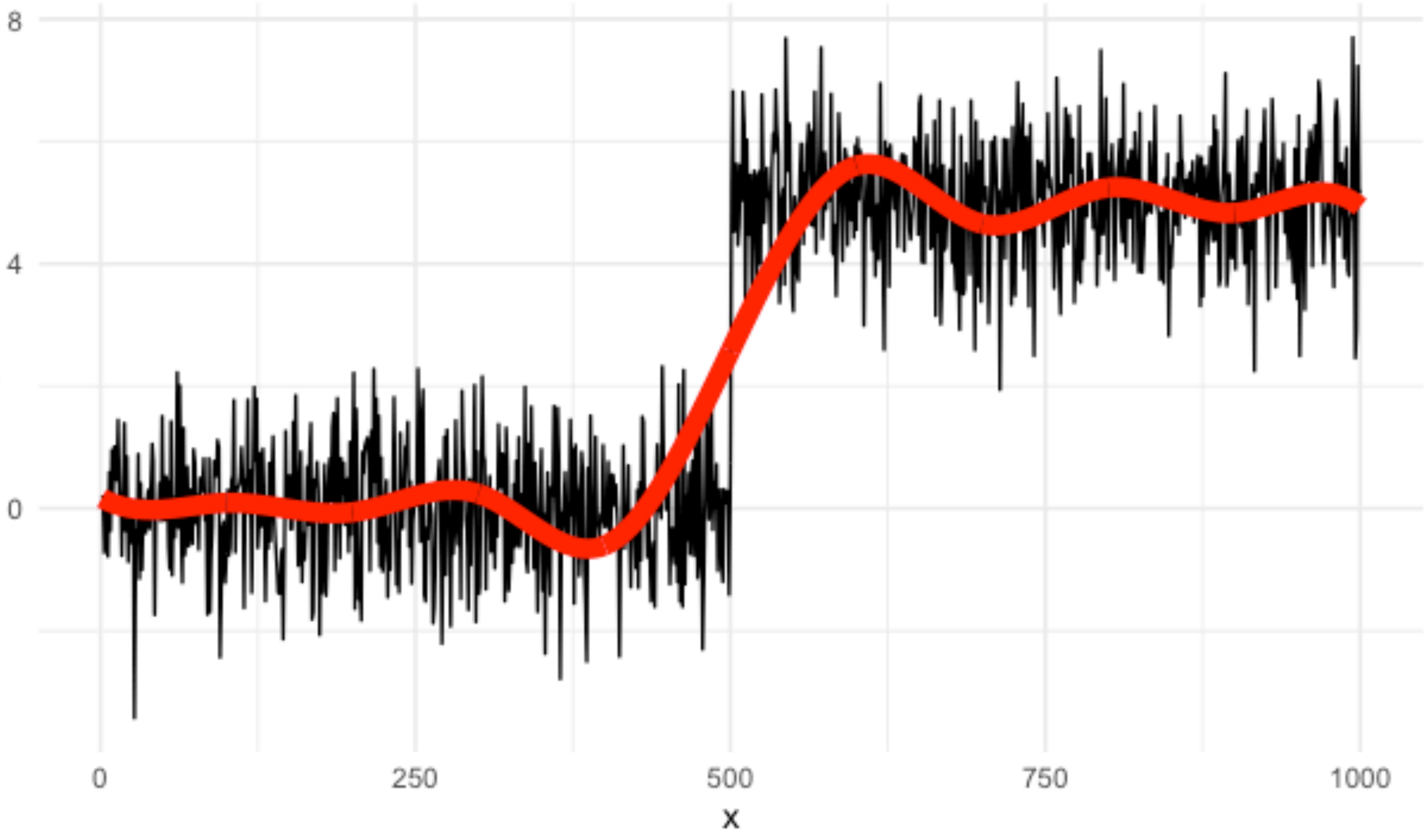



```
> fit <- lm(AAPL.Adjusted ~ bs(date, dfr), g)
> tidy(fit)
# A tibble: 13 x 5
   term            estimate std.error statistic   p.value
   <chr>              <dbl>     <dbl>     <dbl>     <dbl>
 1 (Intercept)        98.2       2.90     33.8   1.33e-178
 2 bs(date, dfr)1     14.8       5.32      2.79  5.41e-  3
 3 bs(date, dfr)2     26.7       3.36      7.95  4.15e- 15
 4 bs(date, dfr)3     -0.693     4.01     -0.173 8.63e-  1
 5 bs(date, dfr)4     -6.58      3.39     -1.94  5.21e-  2
 6 bs(date, dfr)5      6.25      3.64      1.72  8.62e-  2
 7 bs(date, dfr)6     55.6       3.49     15.9   3.27e- 52
 8 bs(date, dfr)7     50.8       3.58     14.2   1.46e- 42
 9 bs(date, dfr)8     98.6       3.56     27.7   3.42e-132
10 bs(date, dfr)9     94.0       3.69     25.4   2.39e-115
11 bs(date, dfr)10    67.3       4.09     16.5   2.34e- 55
12 bs(date, dfr)11   127.        4.23     29.9   1.01e-148
13 bs(date, dfr)12   166.        4.07     40.8   5.97e-232
```

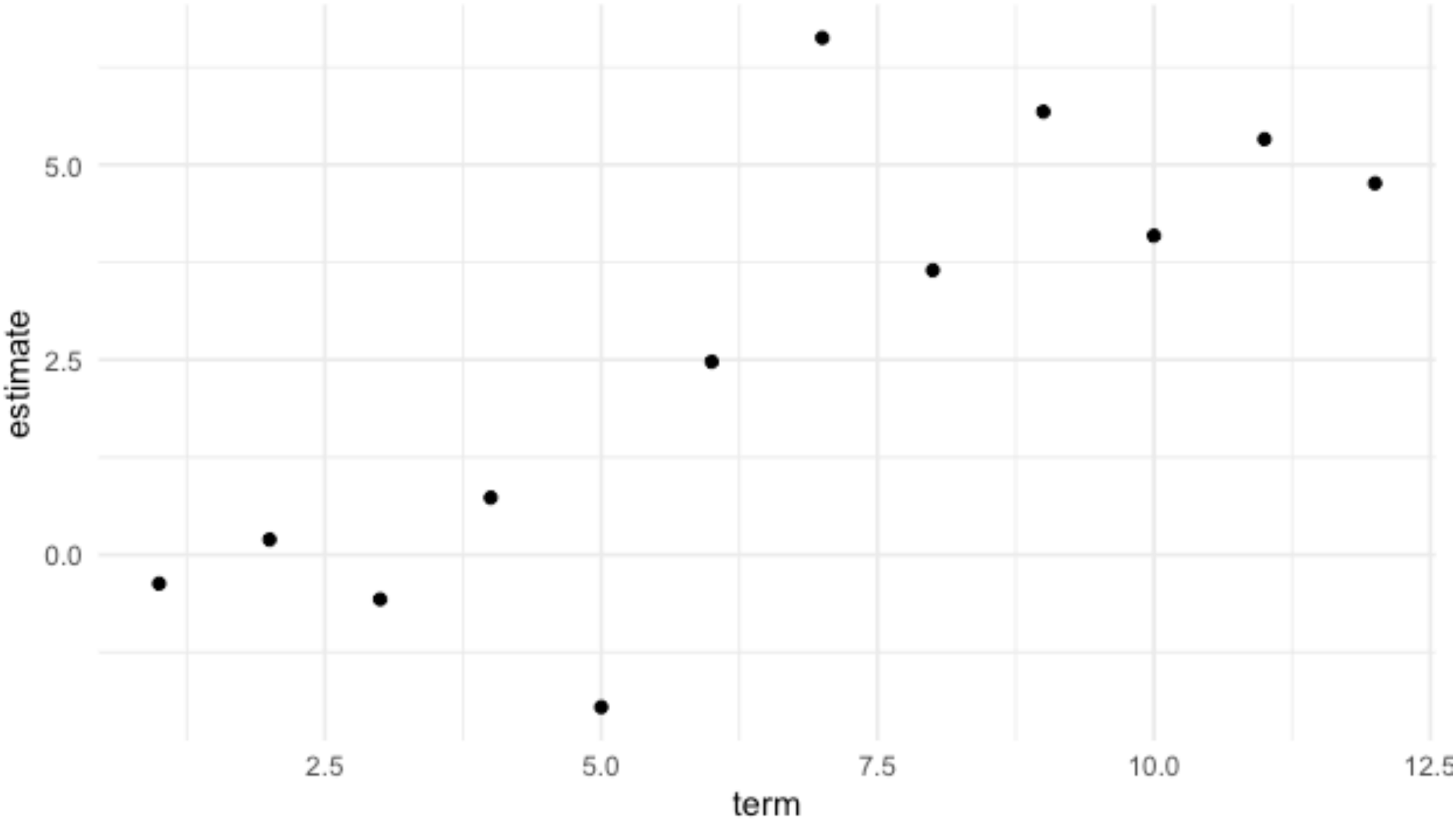
B-Splines (cont'd)



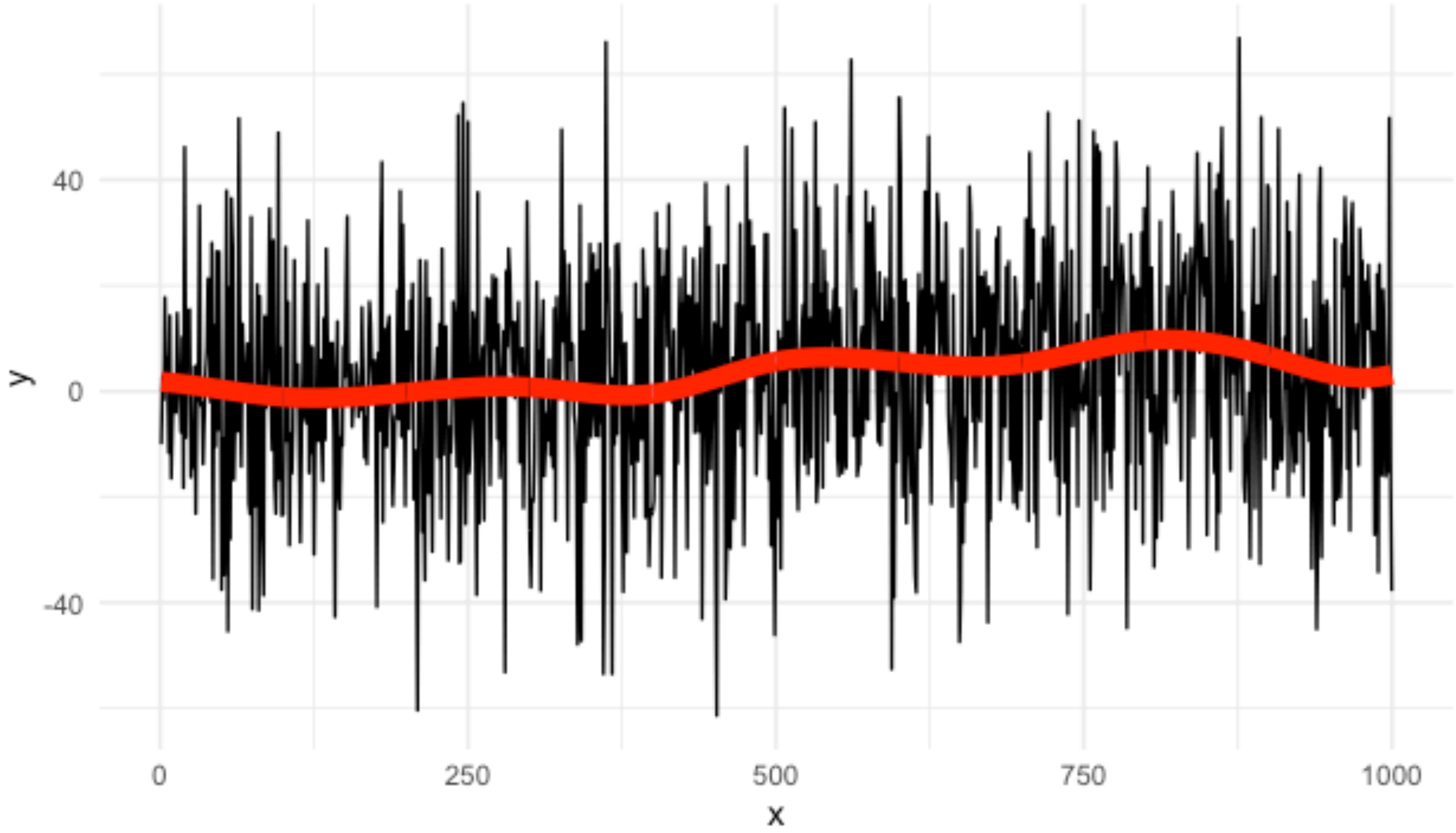
B-Splines (cont'd)



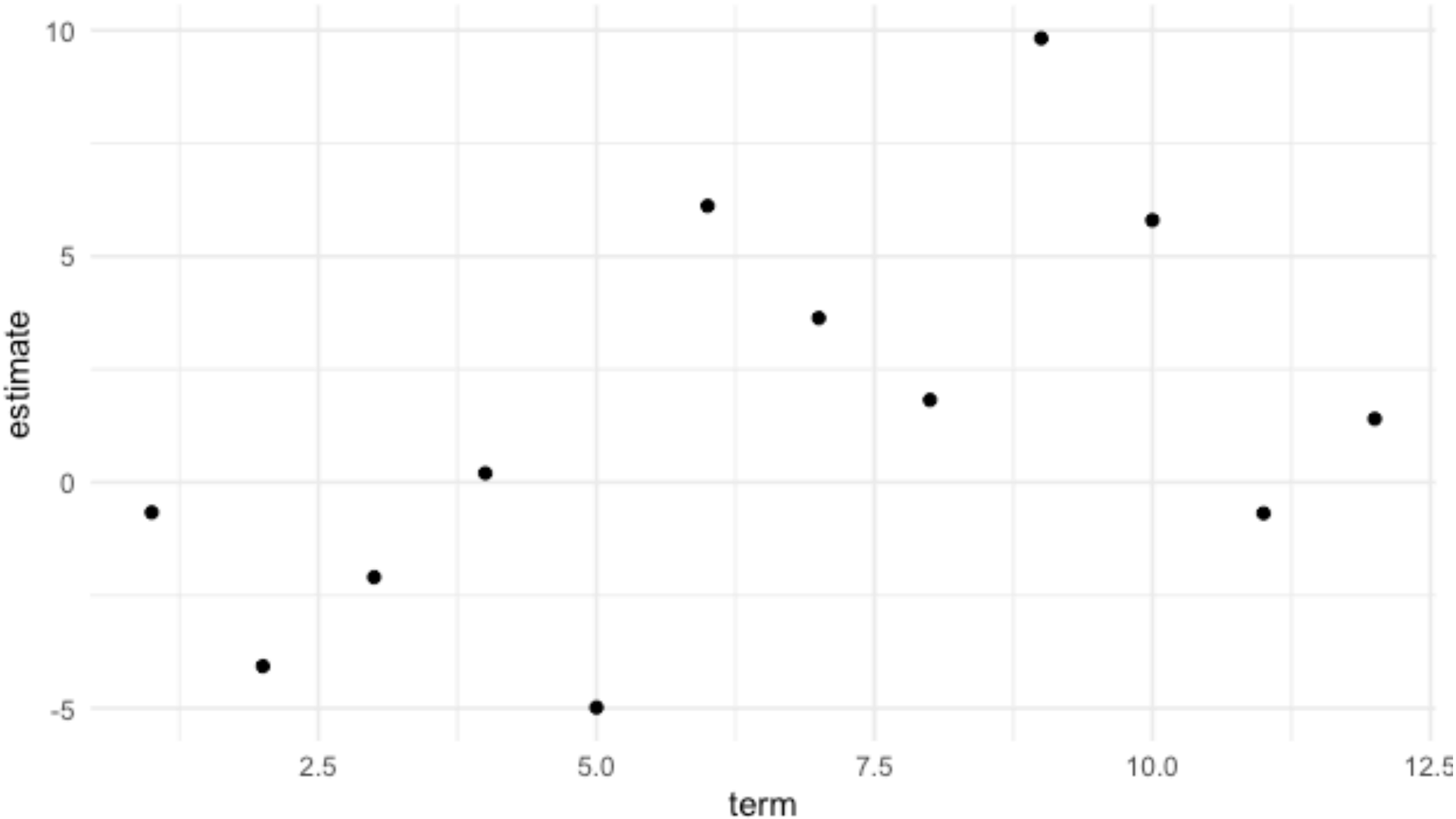
B-Spline Coefficients



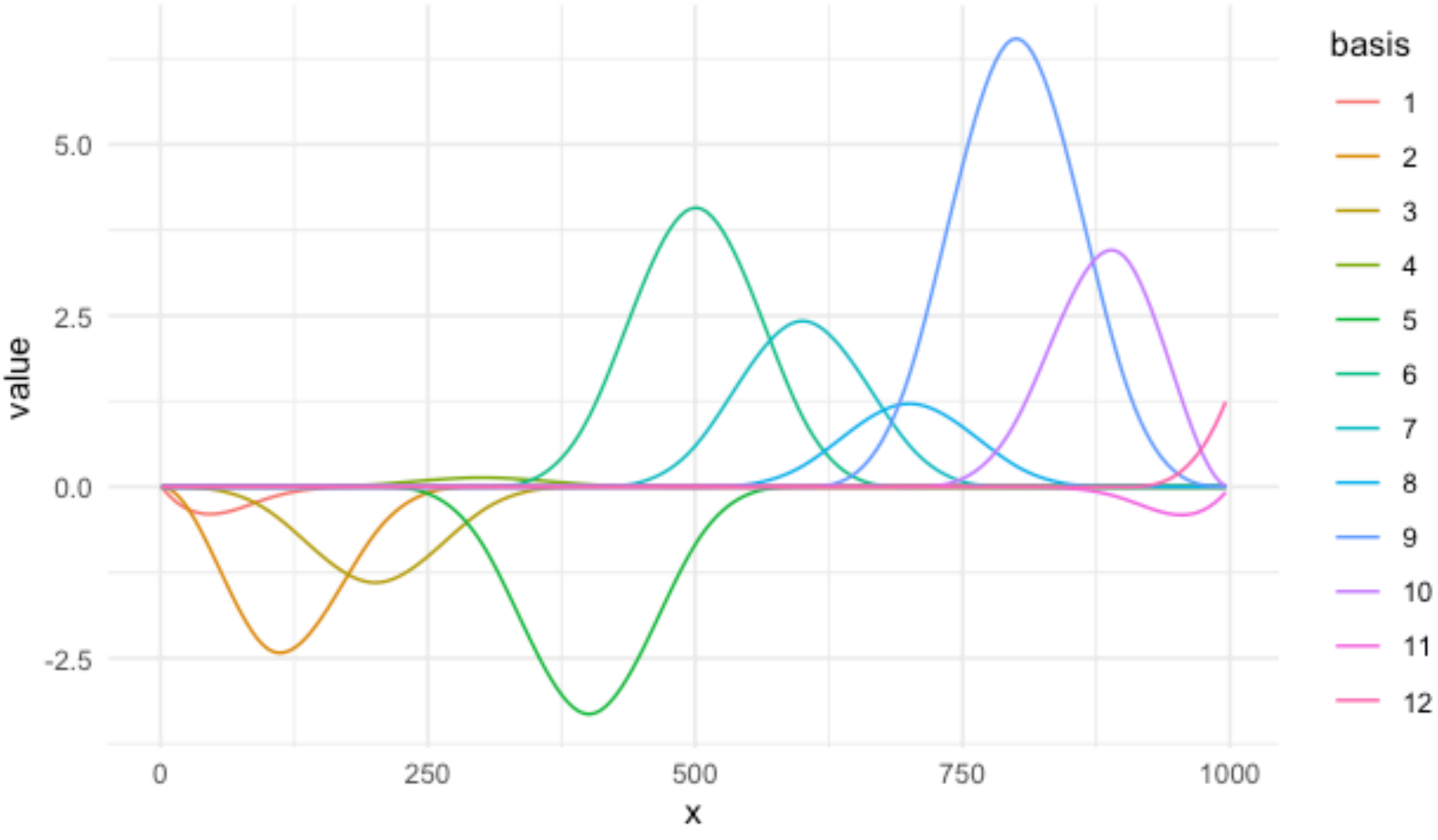
B-Splines



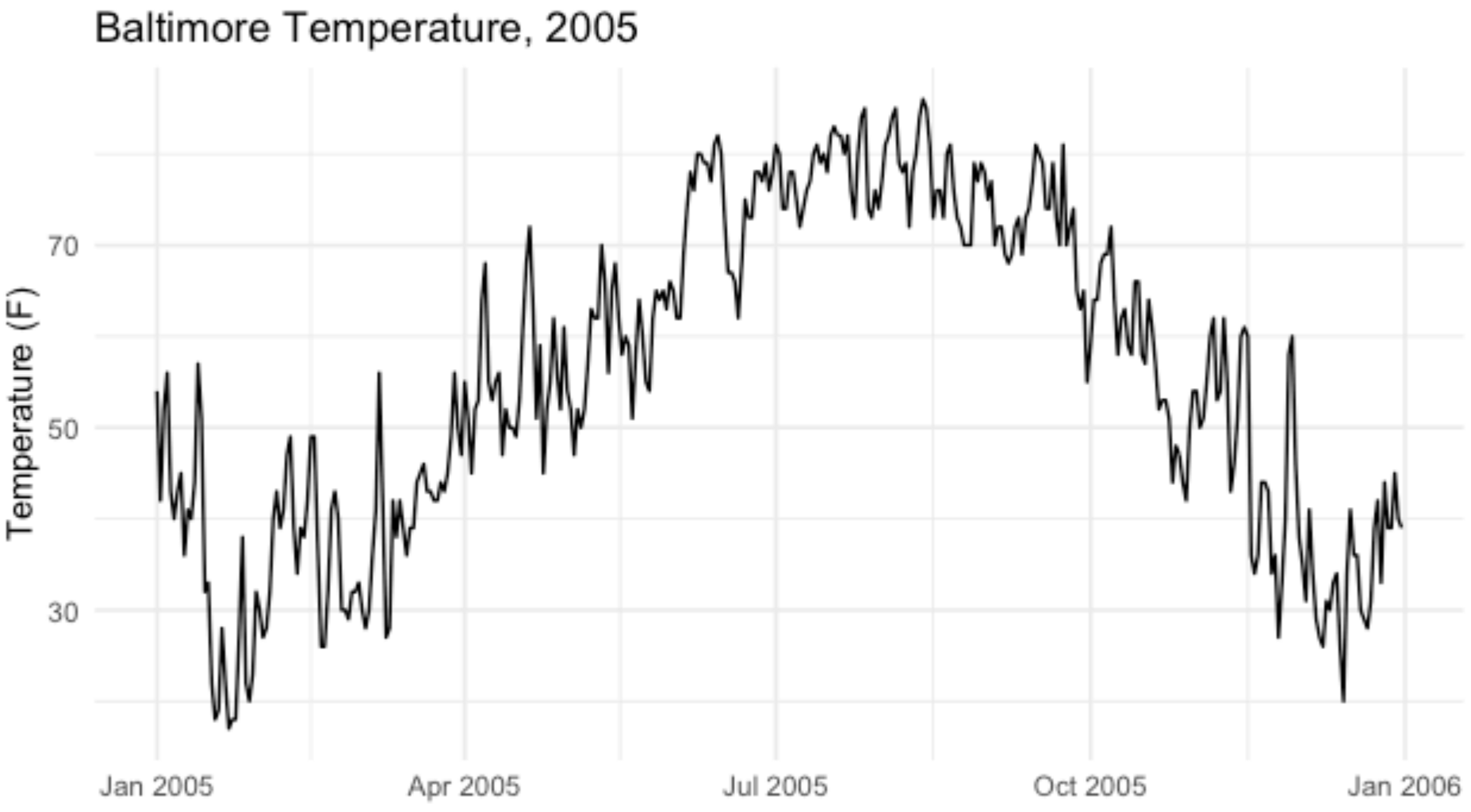
B-Spline Coefficients



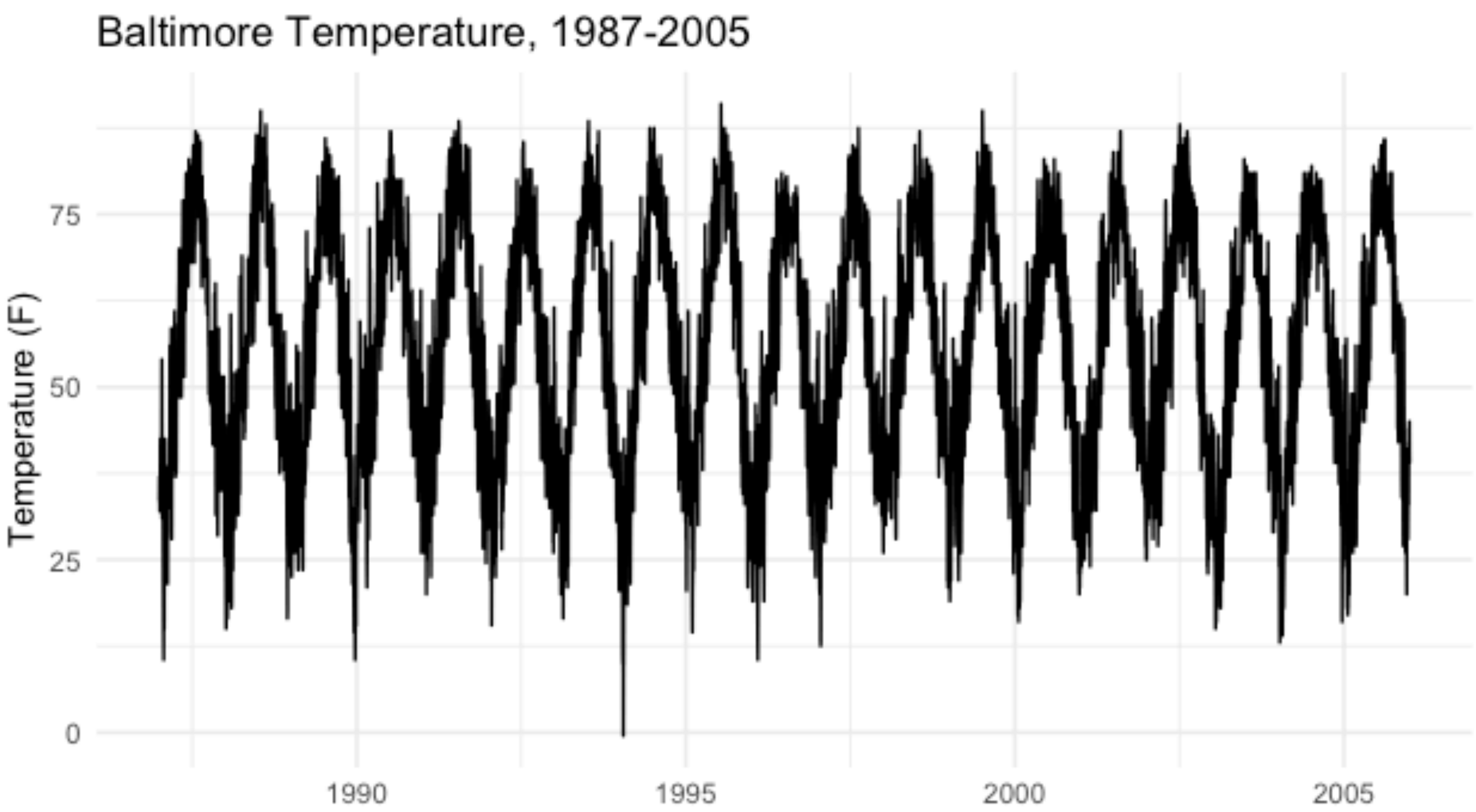
B-Spline * Coef



Baltimore Temperature



Baltimore Temperature



Fourier Basis

- How correlated are your data with sines and cosines of various frequencies?

- Fourier coefficient for frequency *f* is proportional to the covariance of the data with sine/cosine at frequency *f*

- Imagine the linear model (for a given *f* cycles/period)

$$y_t = \beta_0 + \beta_1 \cos(2\pi tf/N) + \beta_2 \sin(2\pi tf/N) + \varepsilon_t$$

where $t = 1,\ldots,N$

Fourier Basis

$$\beta = (\beta_0, \beta_1, \beta_2) \qquad Y = (Y_1 \cdots Y_N)$$

$$X = \begin{bmatrix} 1 & \cos(2\pi f 1/N) & \sin(2\pi f 1/N) \\ 1 & \cos(2\pi f 2/N) & \sin(2\pi f 2/N) \\ \vdots & \vdots & \vdots \\ 1 & \cos(2\pi f N/N) & \sin(2\pi f N/N) \end{bmatrix}$$

$$\hat{\beta} = (X'X)^{-1}X'Y$$

Fourier Basis

- The Fourier transform of a series $y_1, \ldots, y_N$ at frequency *f* is

$$a_f = \sum_{t=1}^{N} y_t \cos(2\pi ft/N)$$

$$b_f = \sum_{i=1}^{N} y_t \sin(2\pi ft/N)$$

Shorthand:

$$(a_f, b_f) = \sum_{t=1}^{N} y_t \exp(-i2\pi ft/N)$$

- We can do this from $f = 1$ up to $f = N/2$, the **Nyquist Frequency**

Fourier Basis

- The "power" of a series at frequency *f* can be computed as $R_f^2 = a_f^2 + b_f^2$

- We can plot R_f^2 to see where the power is concentrated

Parseval's Theorem: $\sum_{t=1}^{N} (y_t - \bar{y})^2 = \sum_{f=1}^{N/2} R_f^2$

Fourier Basis

- Running a regression for each frequency is a computationally expensive $O(N^2)$ operation

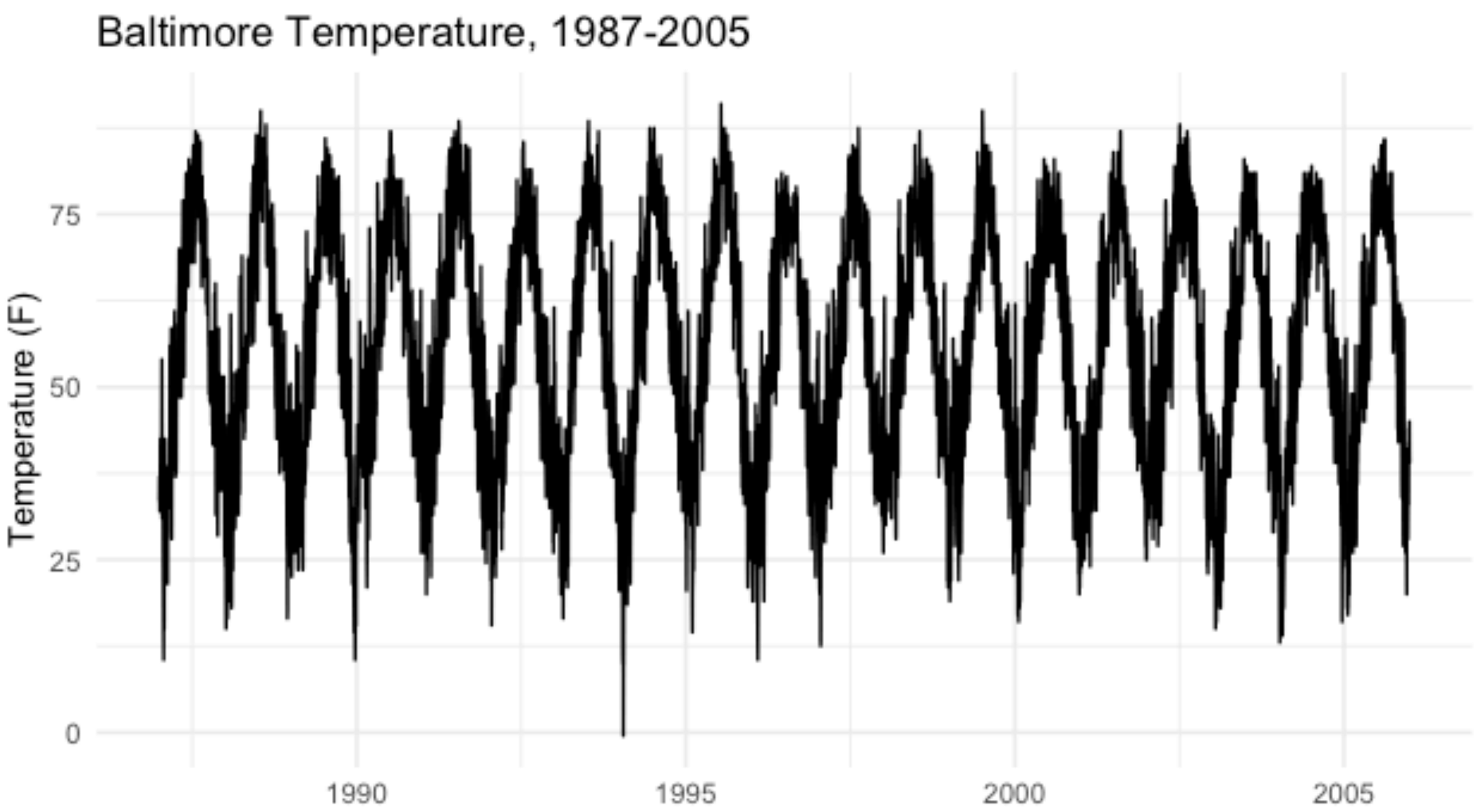
- The Fast Fourier Transform (Tukey & Cooley) can compute the Fourier coefficients in $O(N \log N)$ time

- We can compute the FFT using the `fft()` function in R

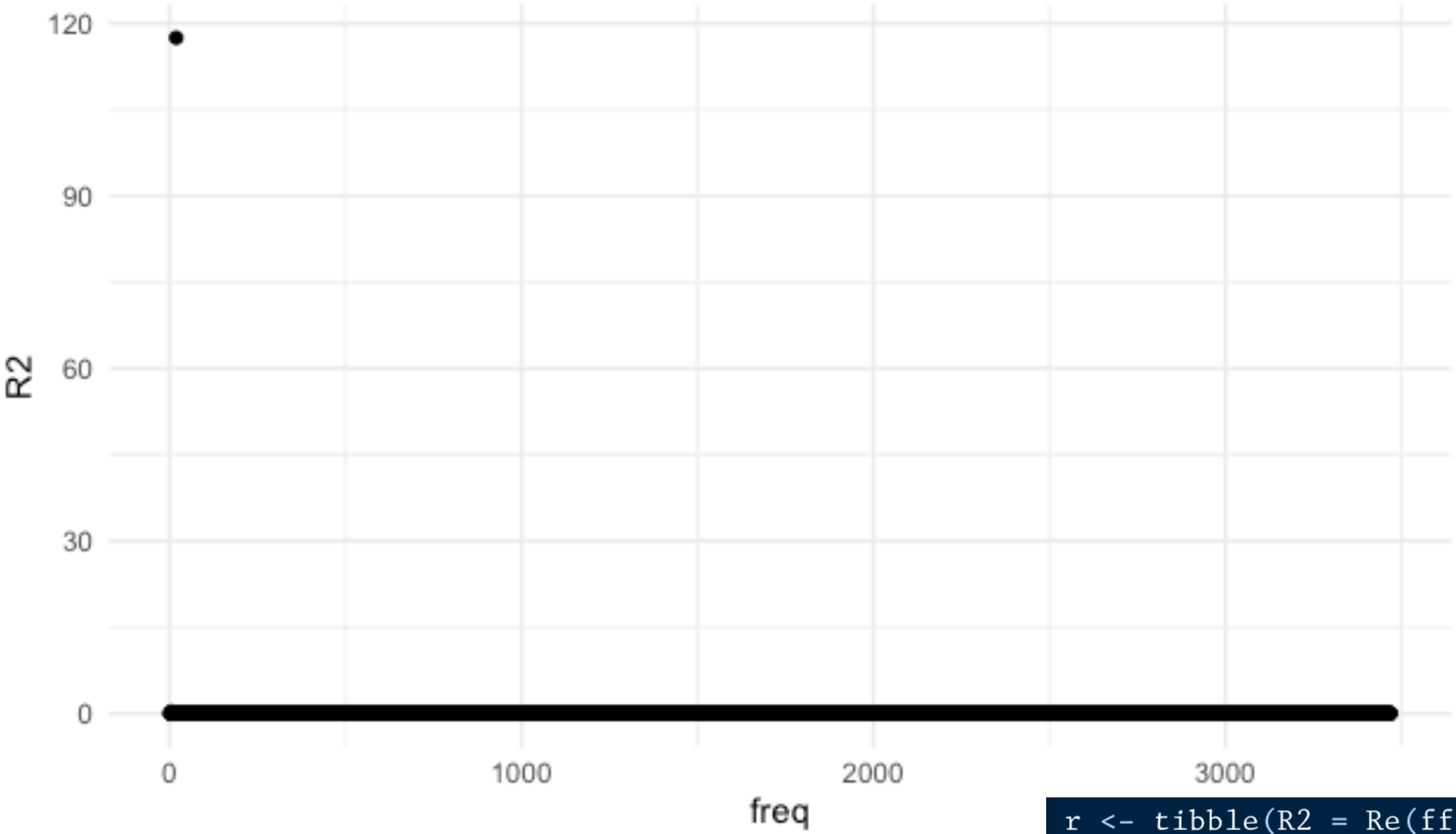
FFT

```
> f <- fft(d$tempF) / n
> f1 <- f[1]  ## Mean of the series
> ff <- f[2:(1 + floor(length(f) / 2))]
> head(ff)
[1] -0.03680245-0.11496786i -0.07530623-0.15053905i
[3] -0.01781281+0.02845252i -0.02253988+0.26135707i
[5]  0.55993628-0.15818797i -0.24614022+0.13761281i
```

Baltimore Temperature

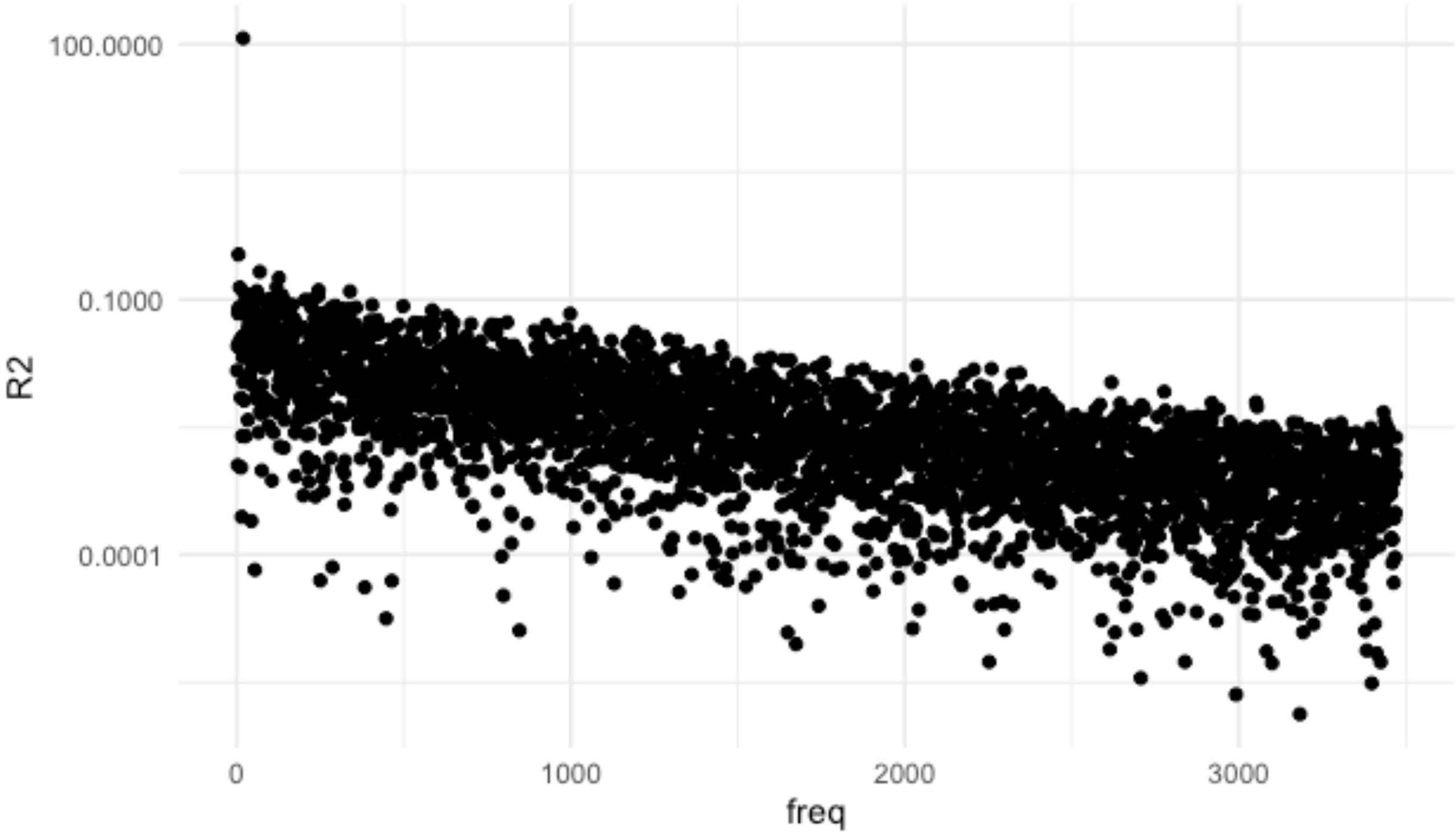


Power Spectrum

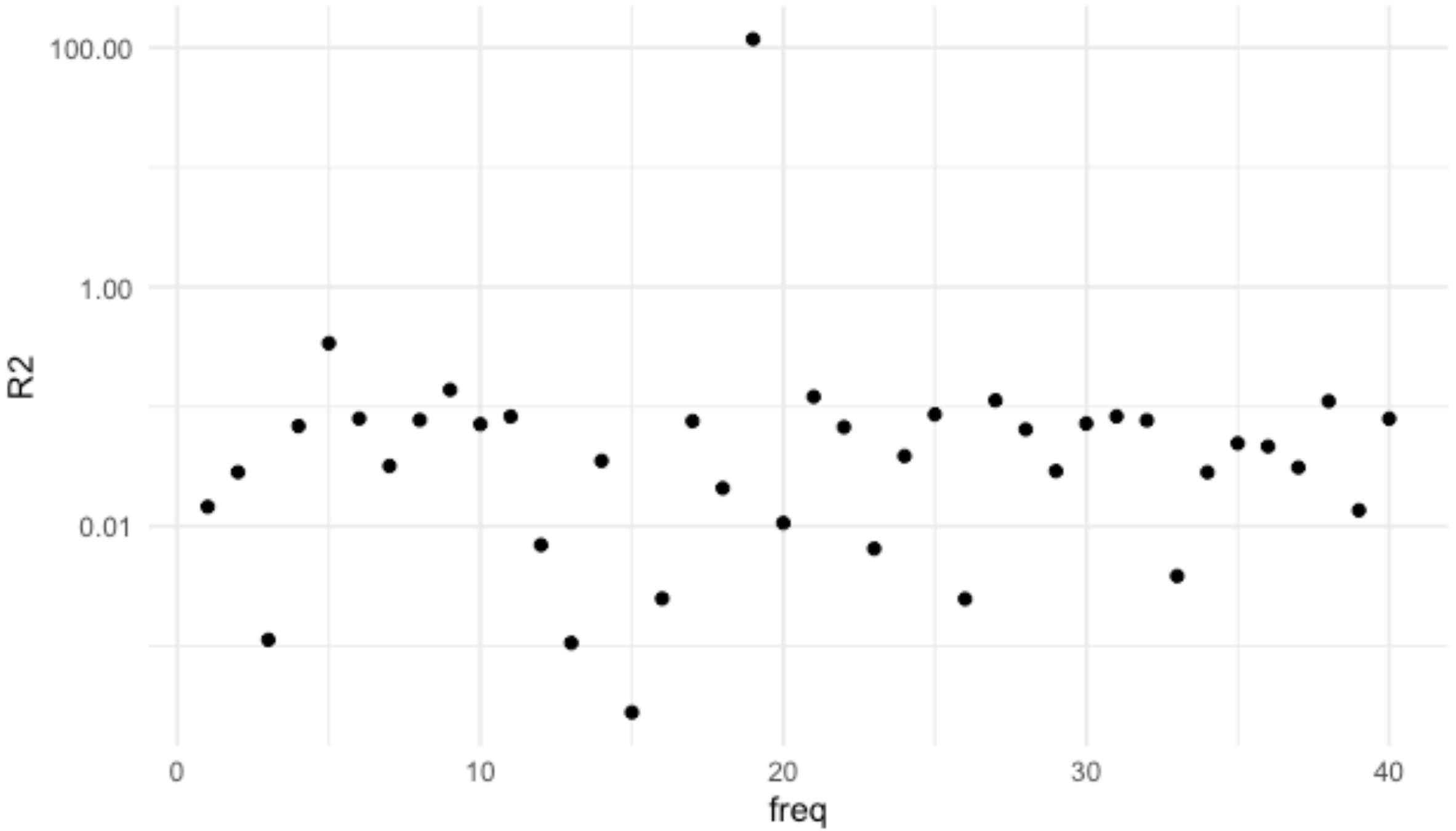



```
r <- tibble(R2 = Re(ff)^2 + Im(ff)^2,
            freq = 1:length(ff))
r %>%
        ggplot(aes(freq, R2)) +
        geom_point()
```

Power Spectrum



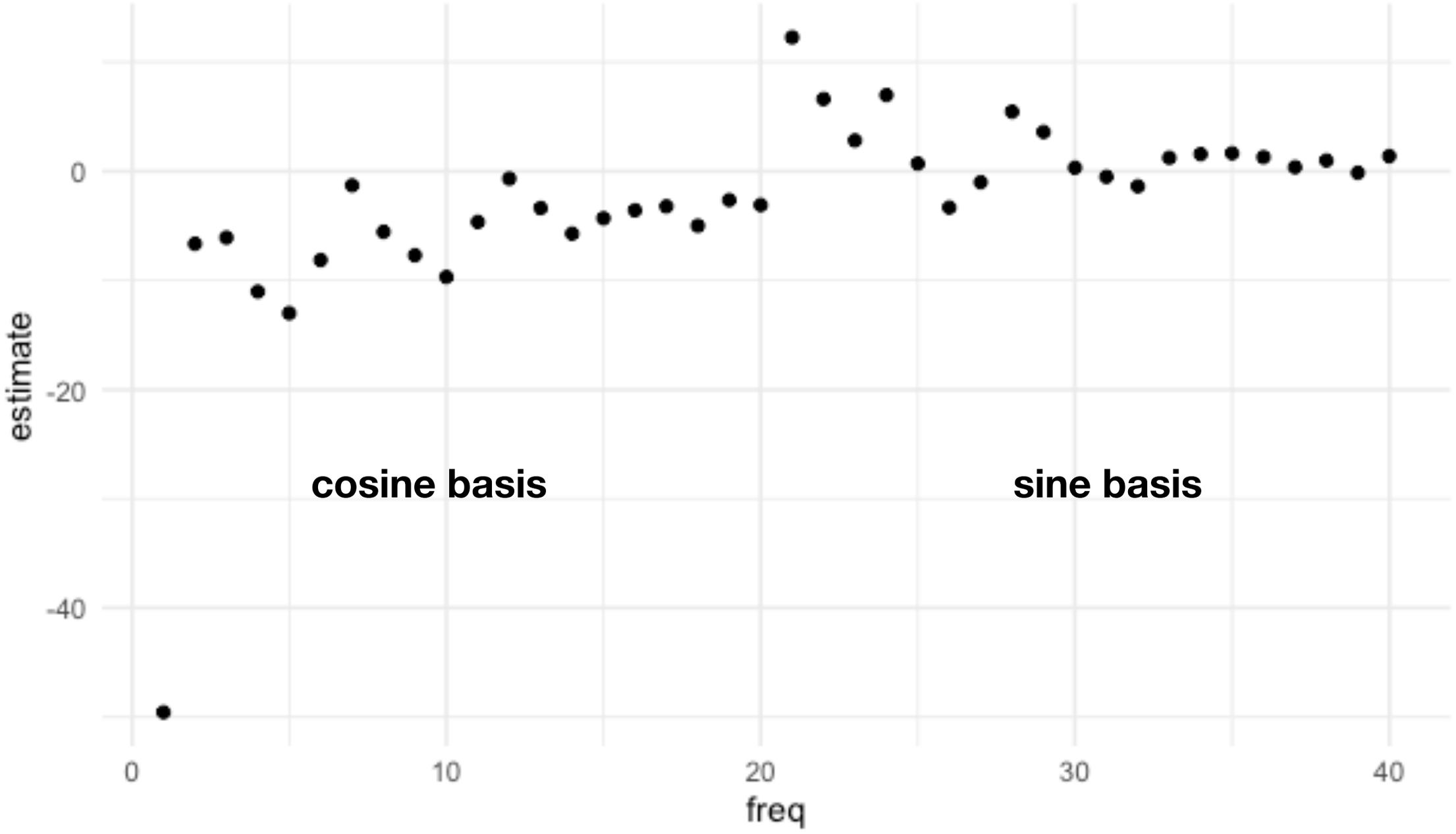
Power Spectrum



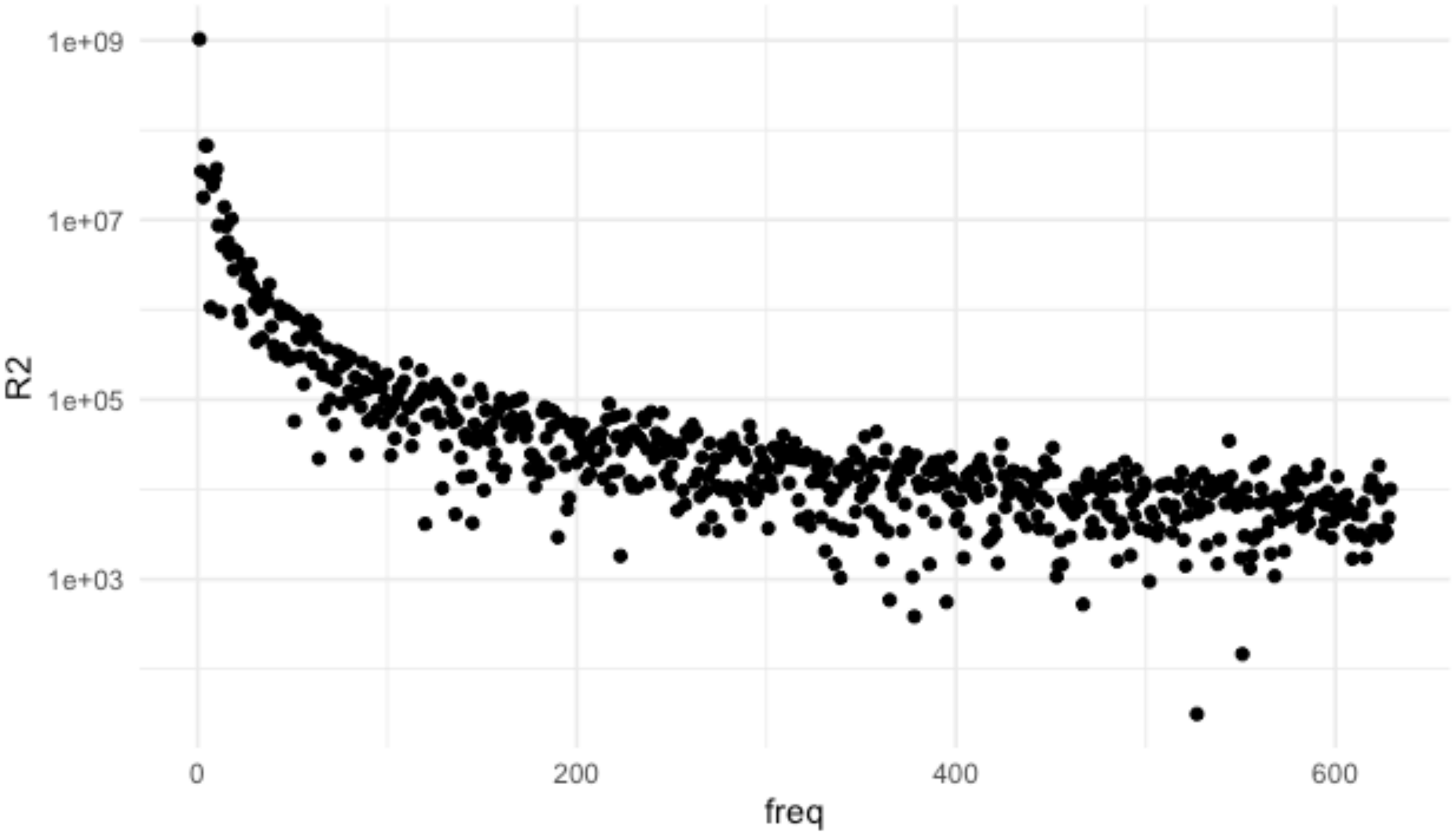
Fourier Basis

"Fourier" Coefficients

$$\mathbb{E}[y_t] = \beta_0 + \sum_{f=1}^{20} a_f \cos(2\pi f t/N) + b_f \sin(2\pi f t/N)$$

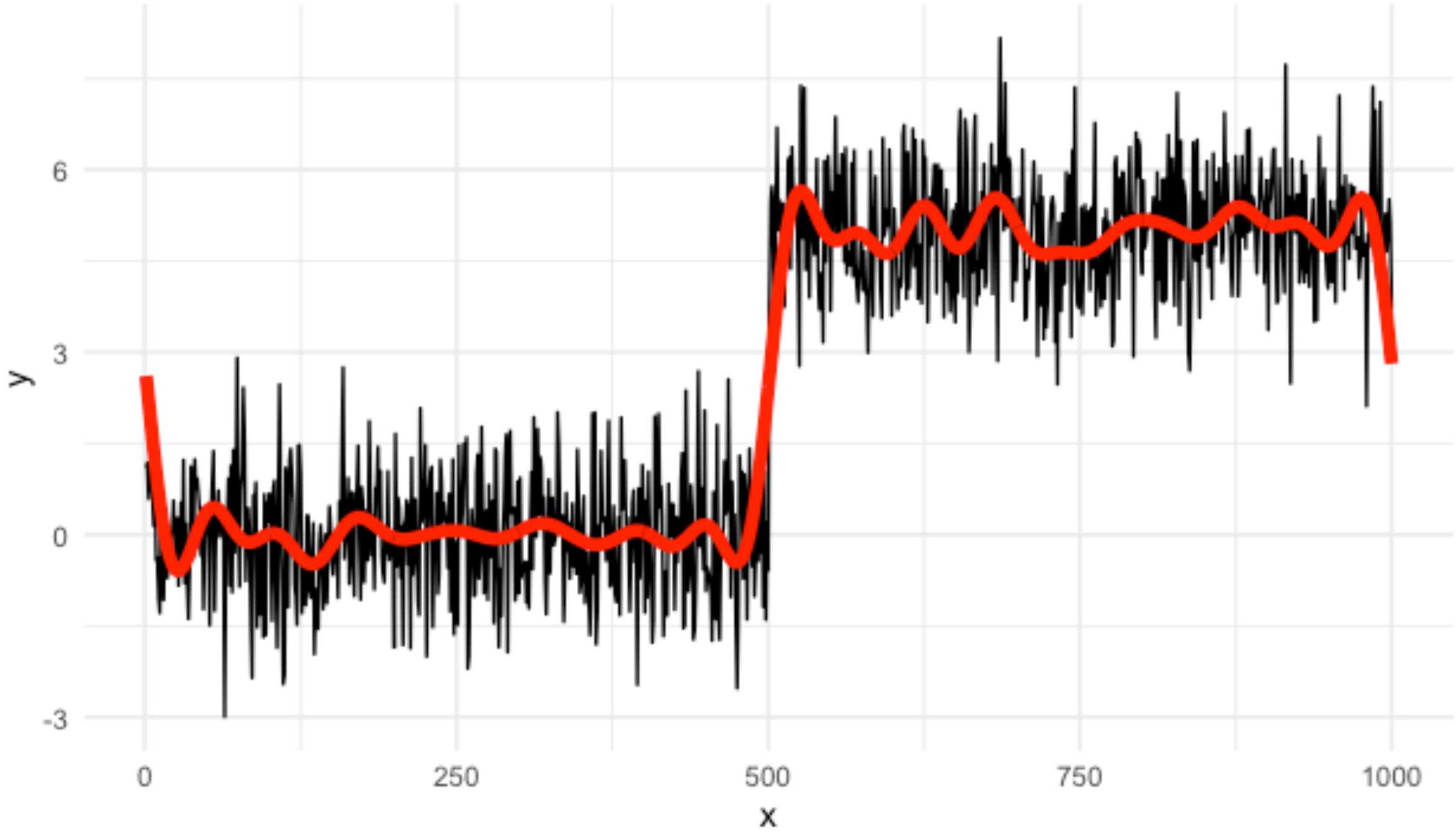


Power Spectrum



Fourier Prediction

Fourier Prediction



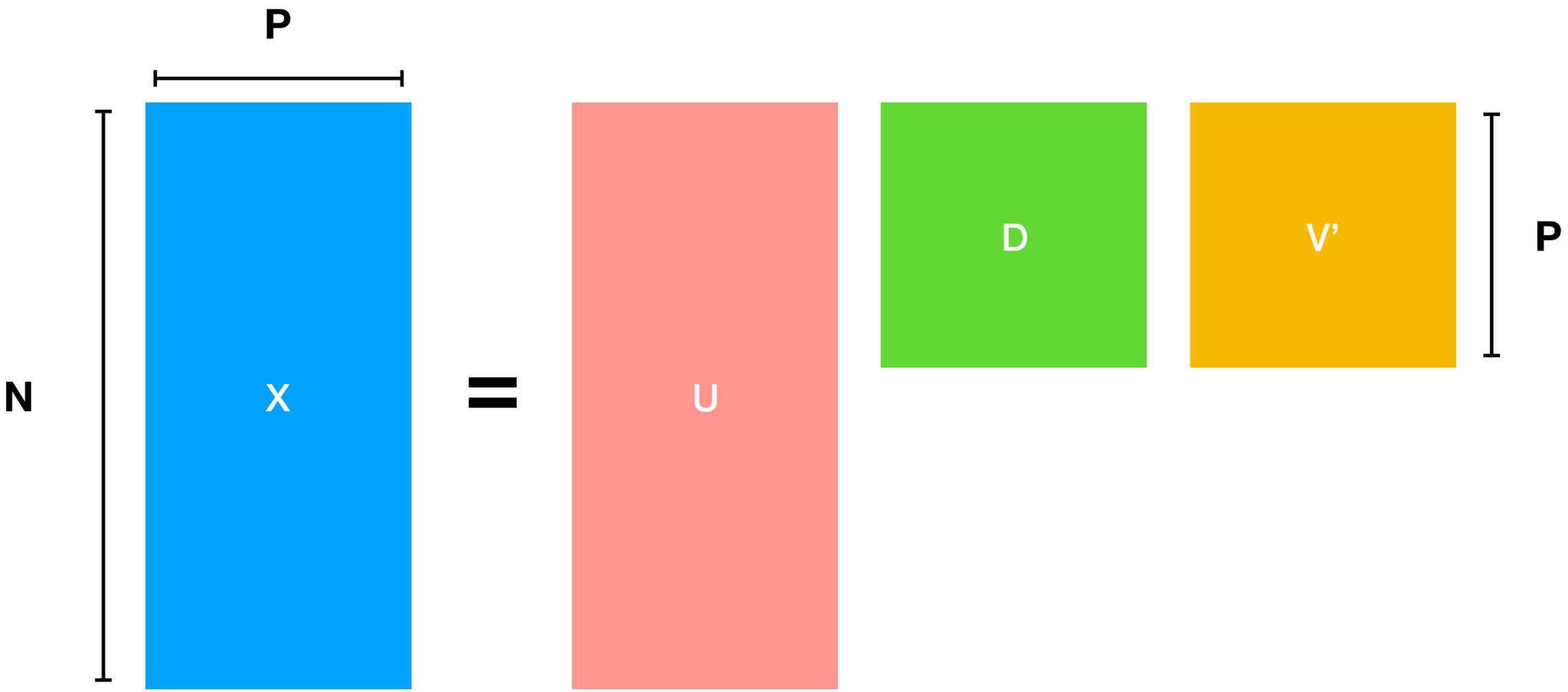
Learning the Basis

- Sometimes there is no obvious basis to apply to a dataset
- We can attempt to learn the basis using the SVD (and other approaches)
- Or we can just use other approaches
- Often, it will be useful to have some replication

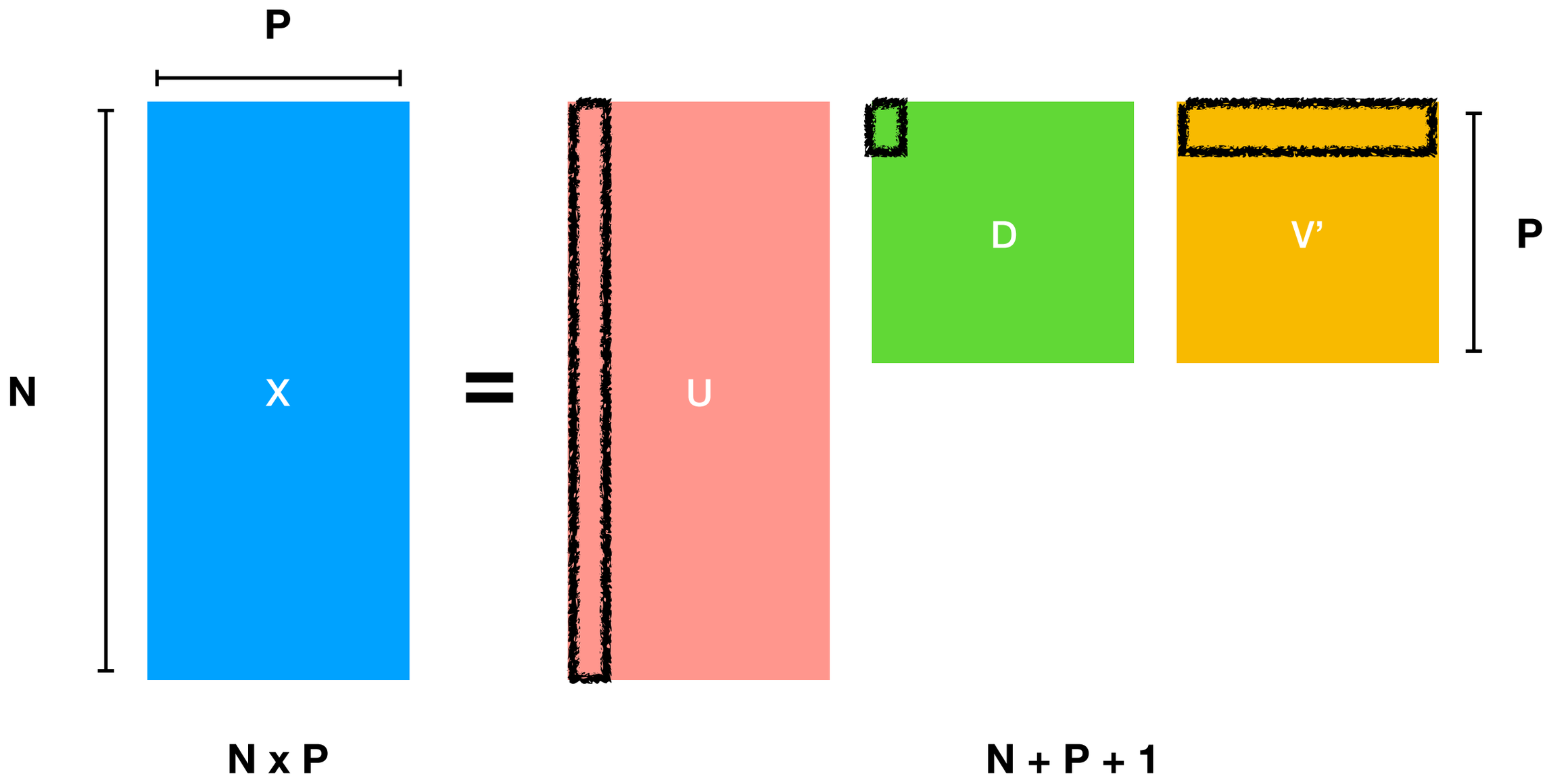
SVD in a Nutshell

- Takes a matrix X and solves for $X = UDV'$
- U is the matrix of **left singular vectors**
- V is the matrix of **right singular vectors**
- D is the diagonal matrix of **singular values**

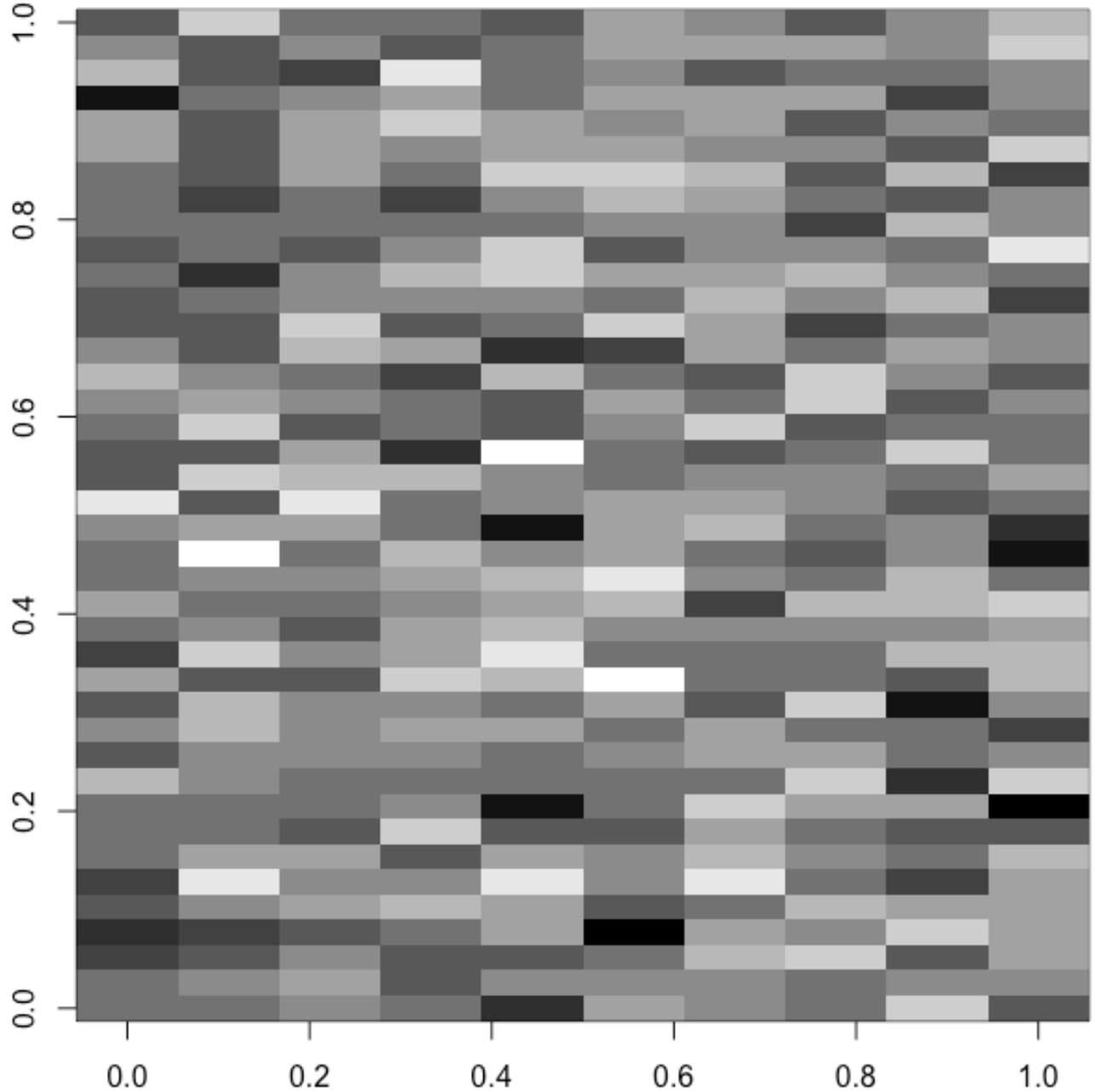
SVD in a Nutshell



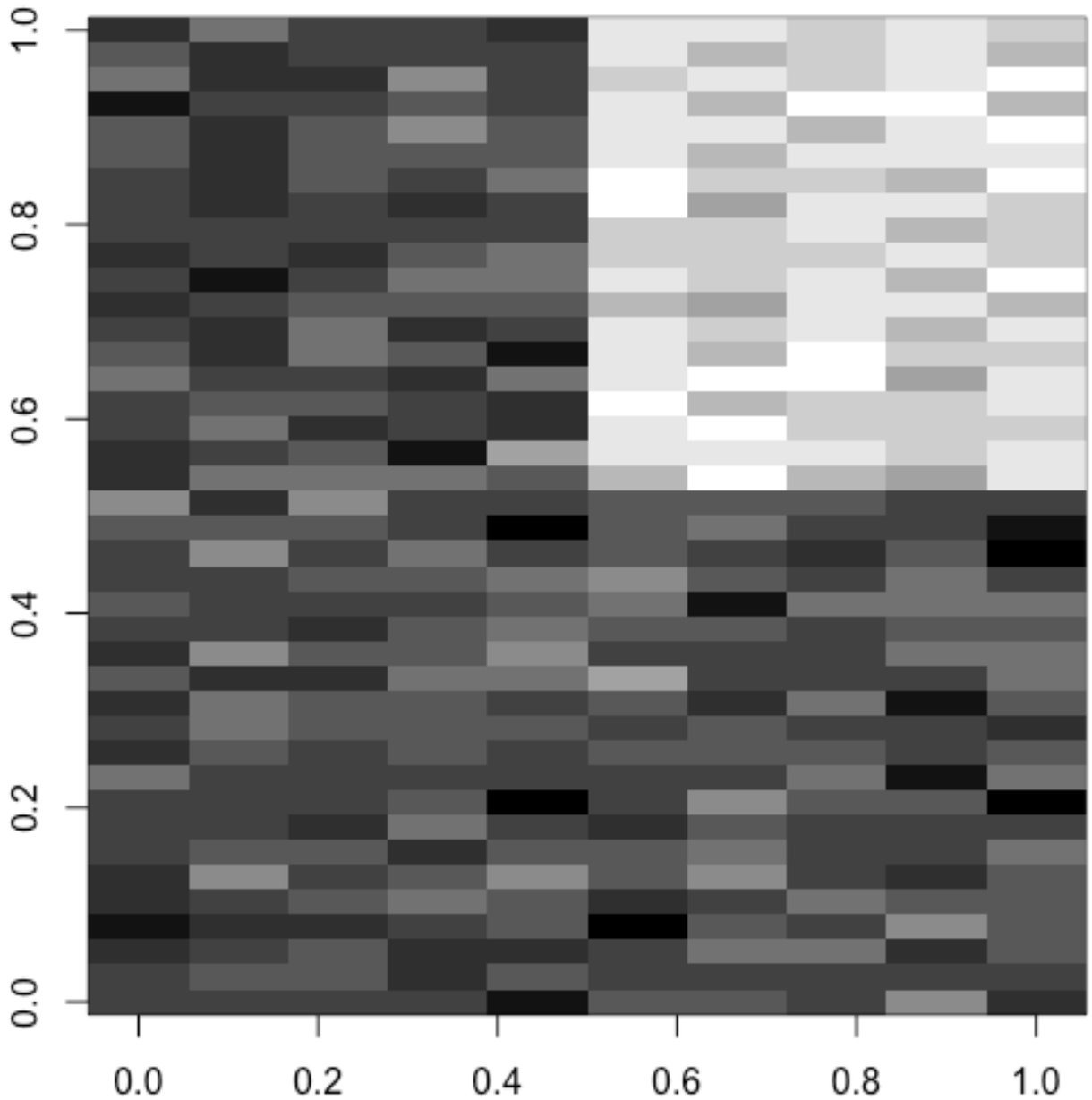
SVD in a Nutshell



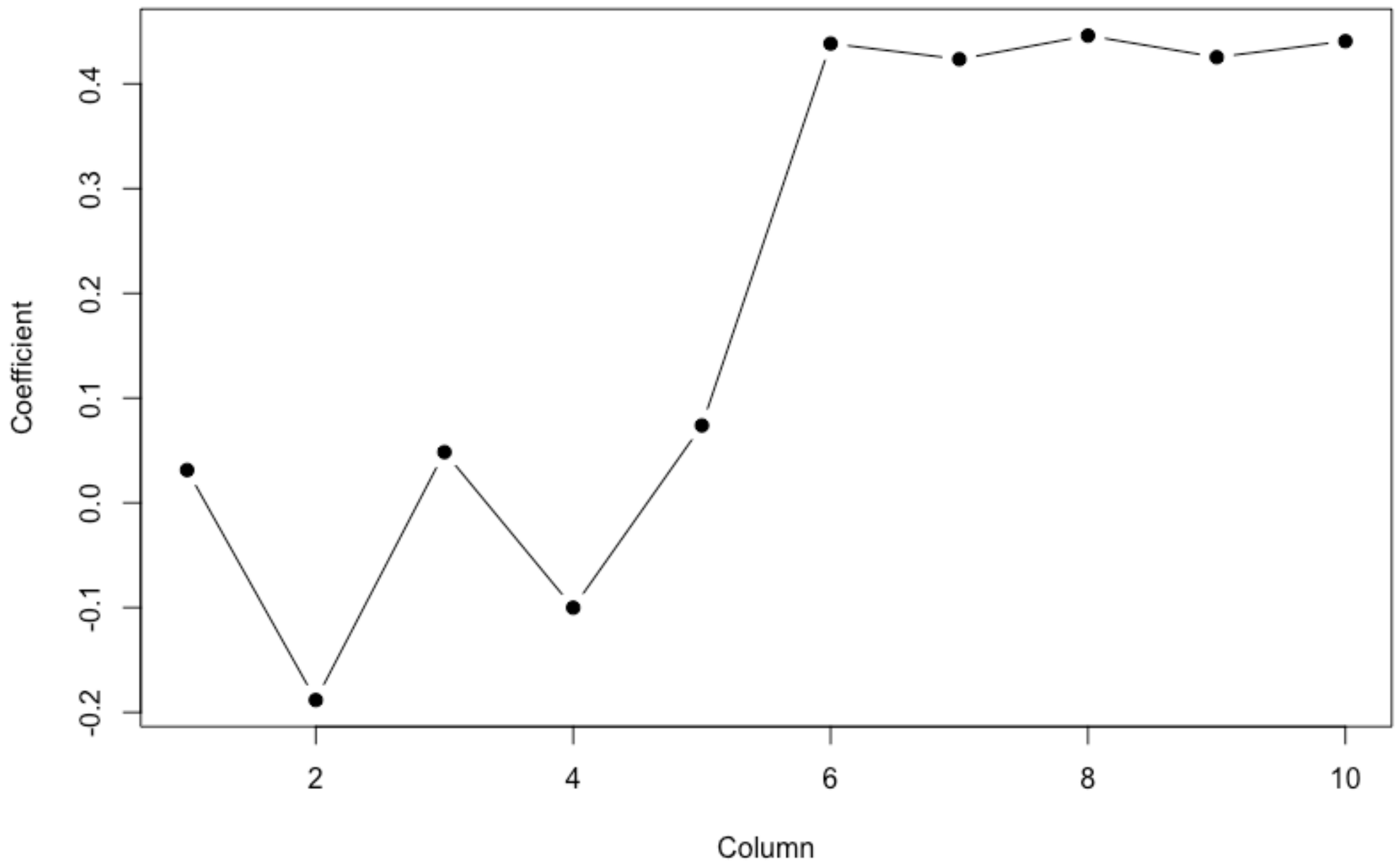
A (Real) Matrix



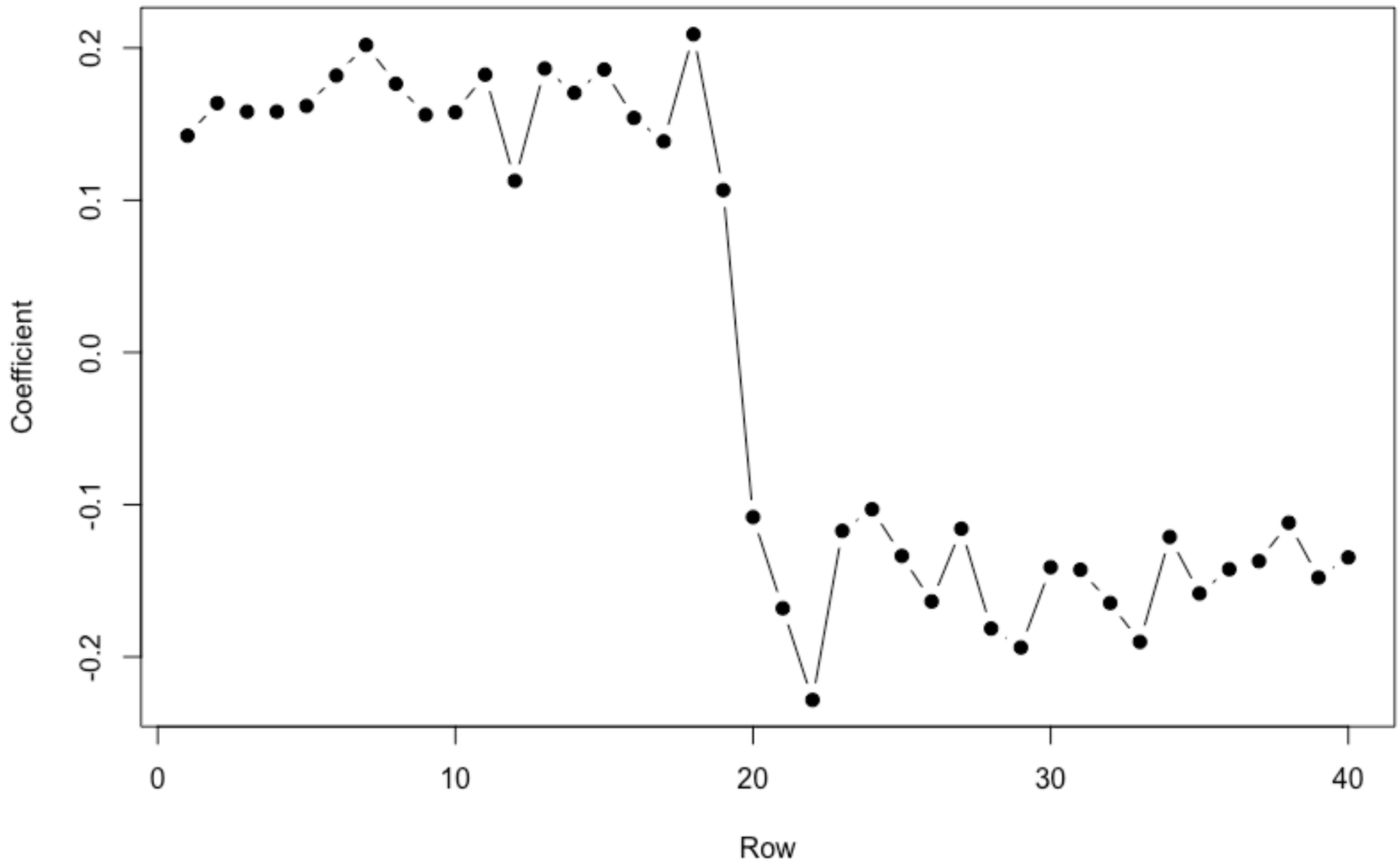
Matrix + Effect



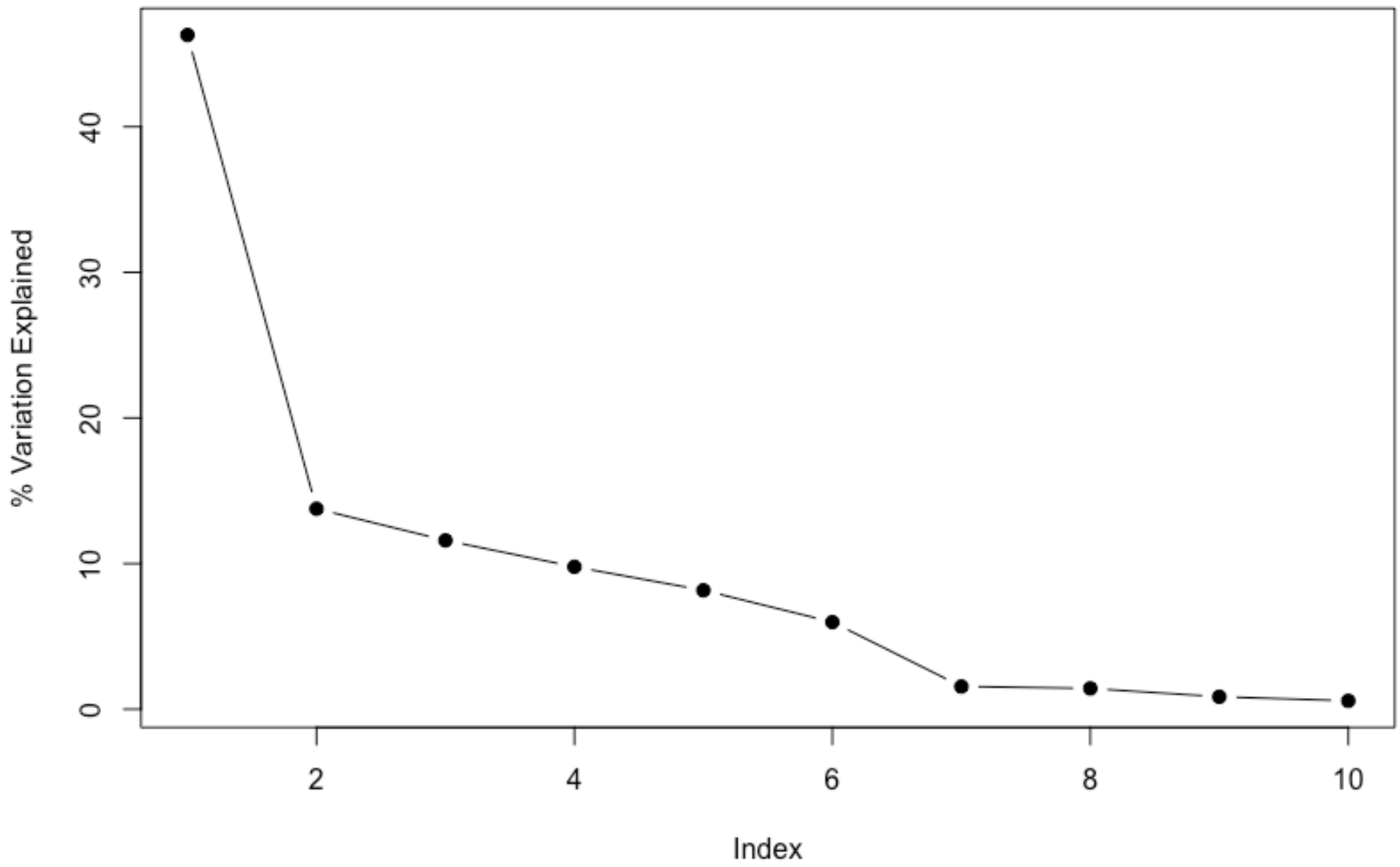
Column Effect



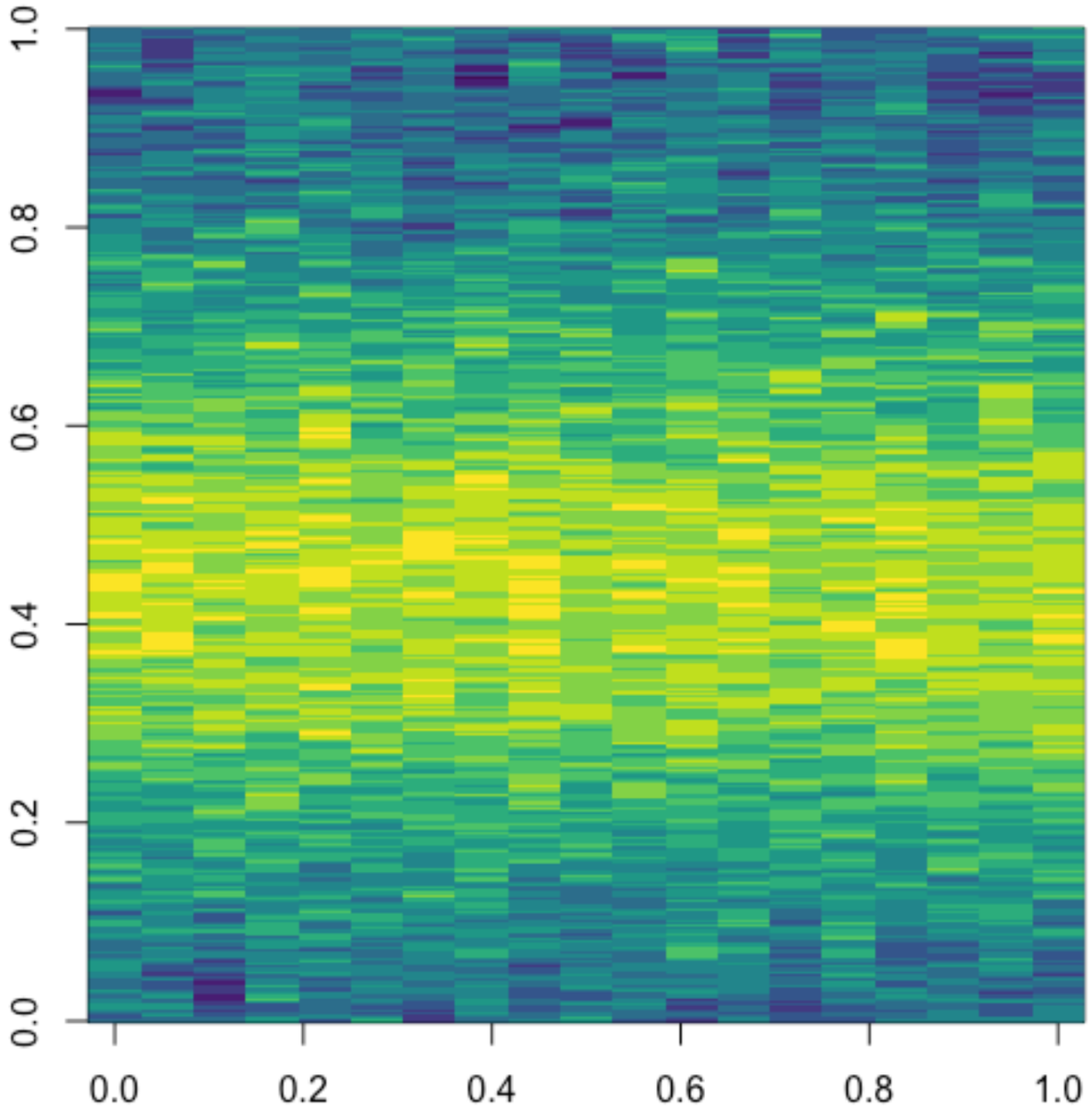
Row Effect



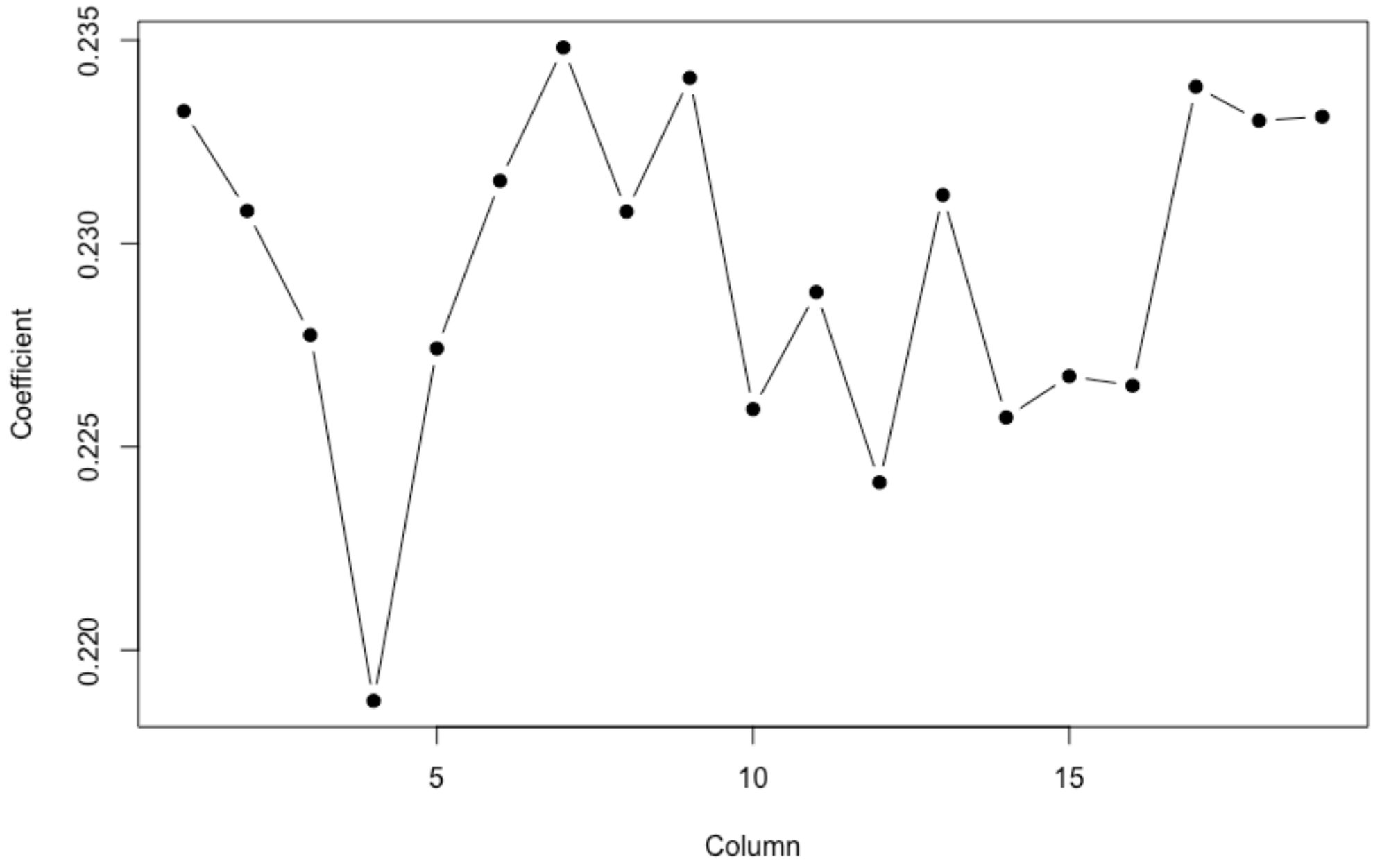
Variation Explained



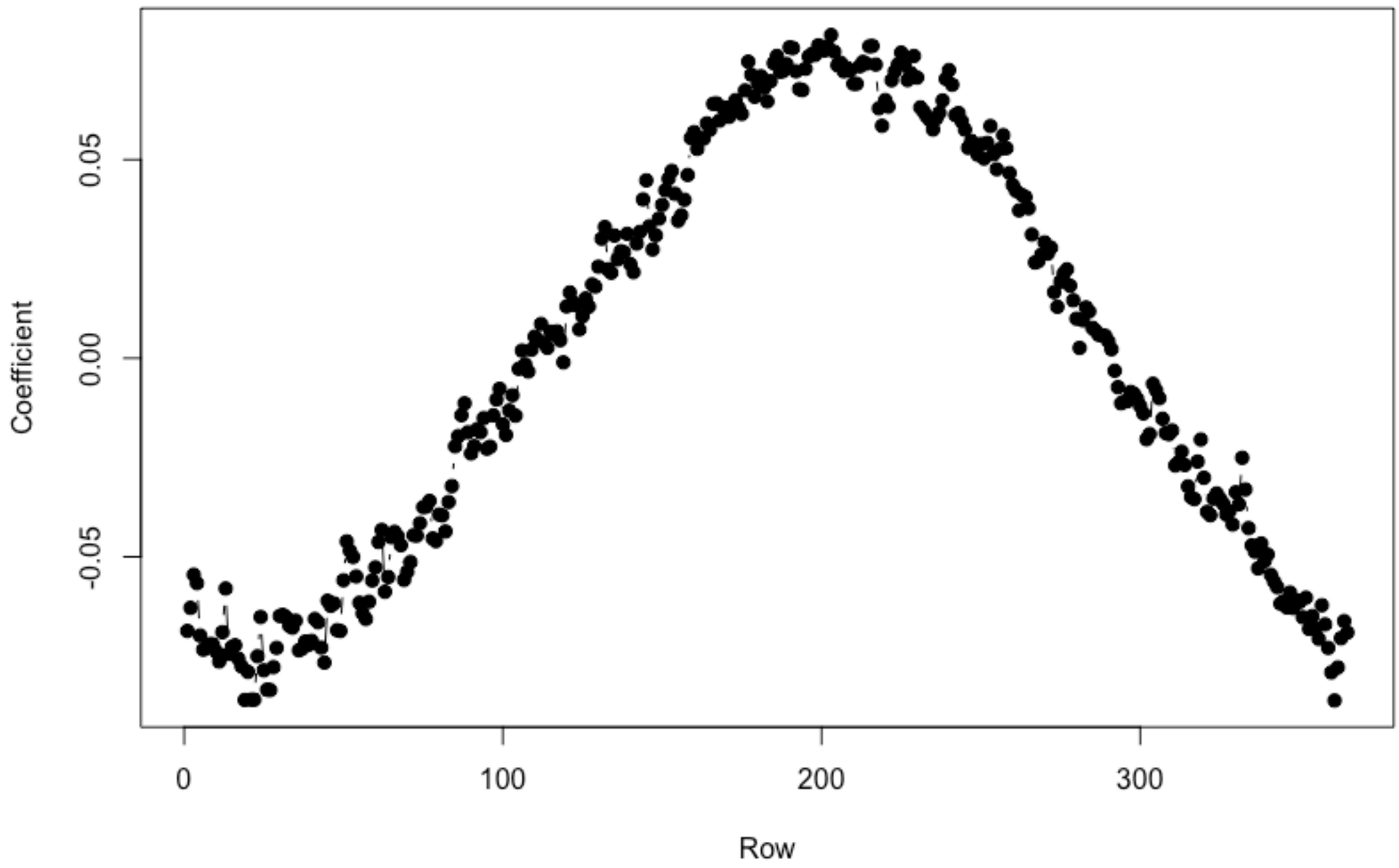
Daily Temperature, Baltimore, MD 1987—2005



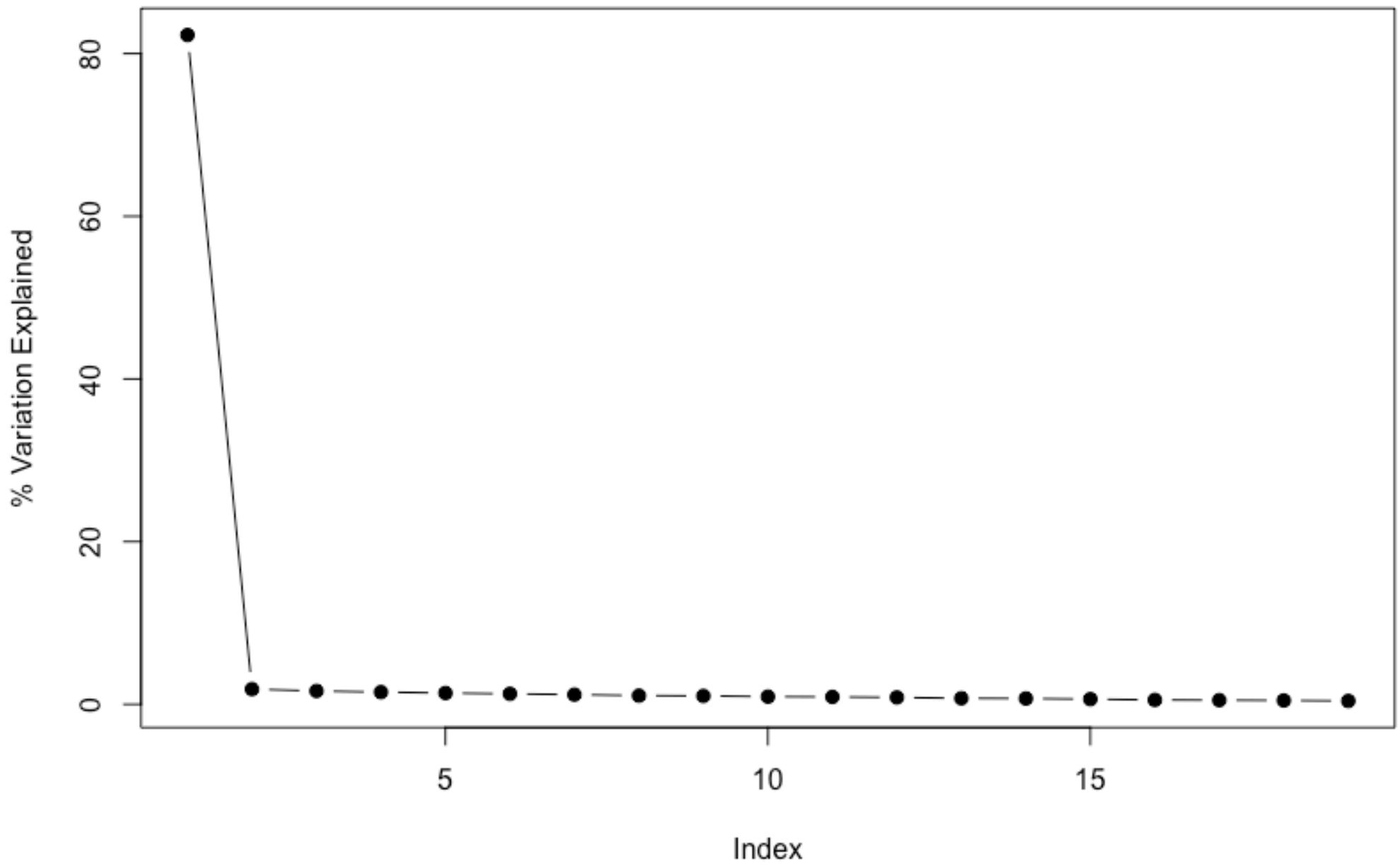
Yearly Effect



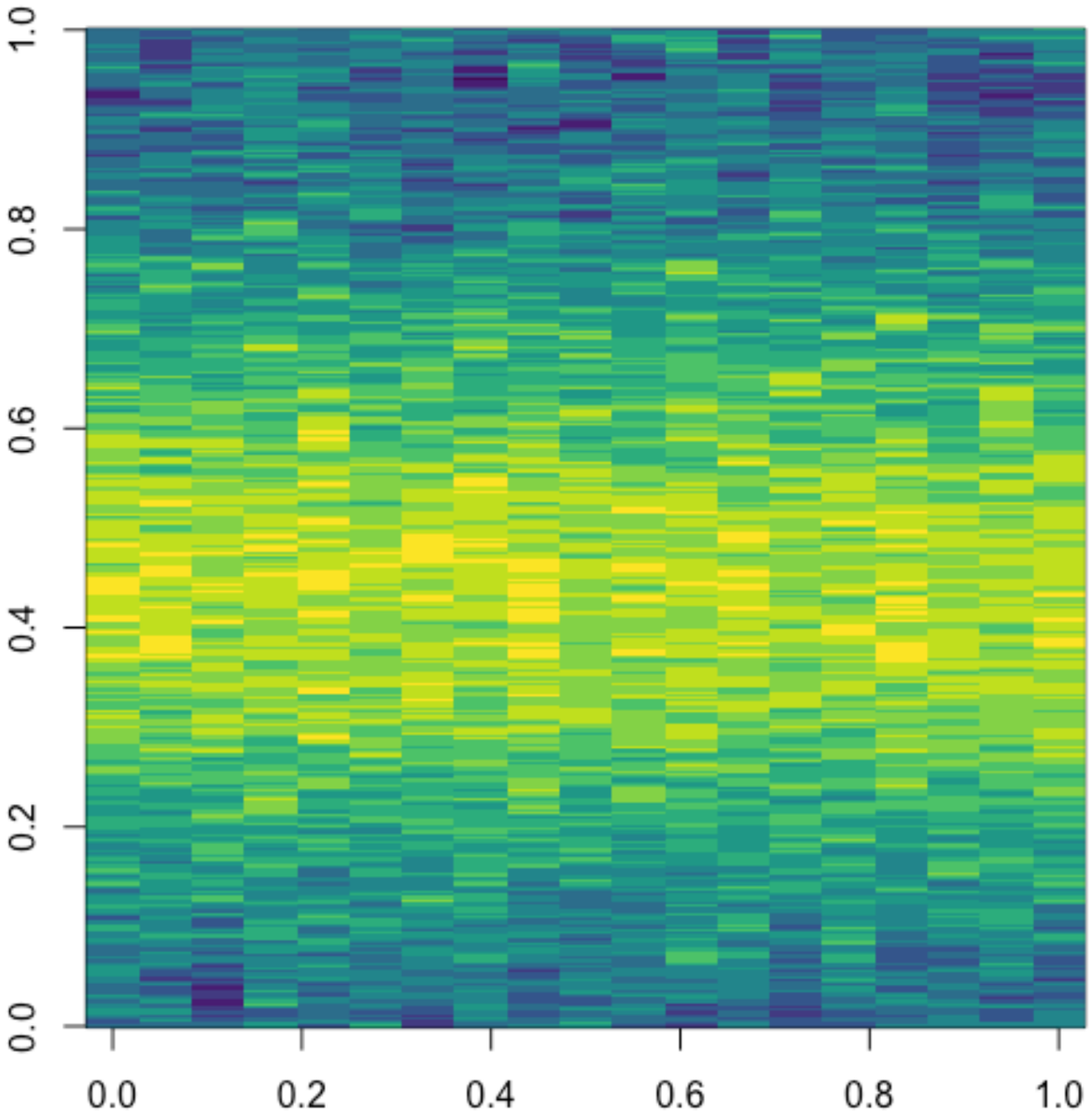
Seasonal Effect

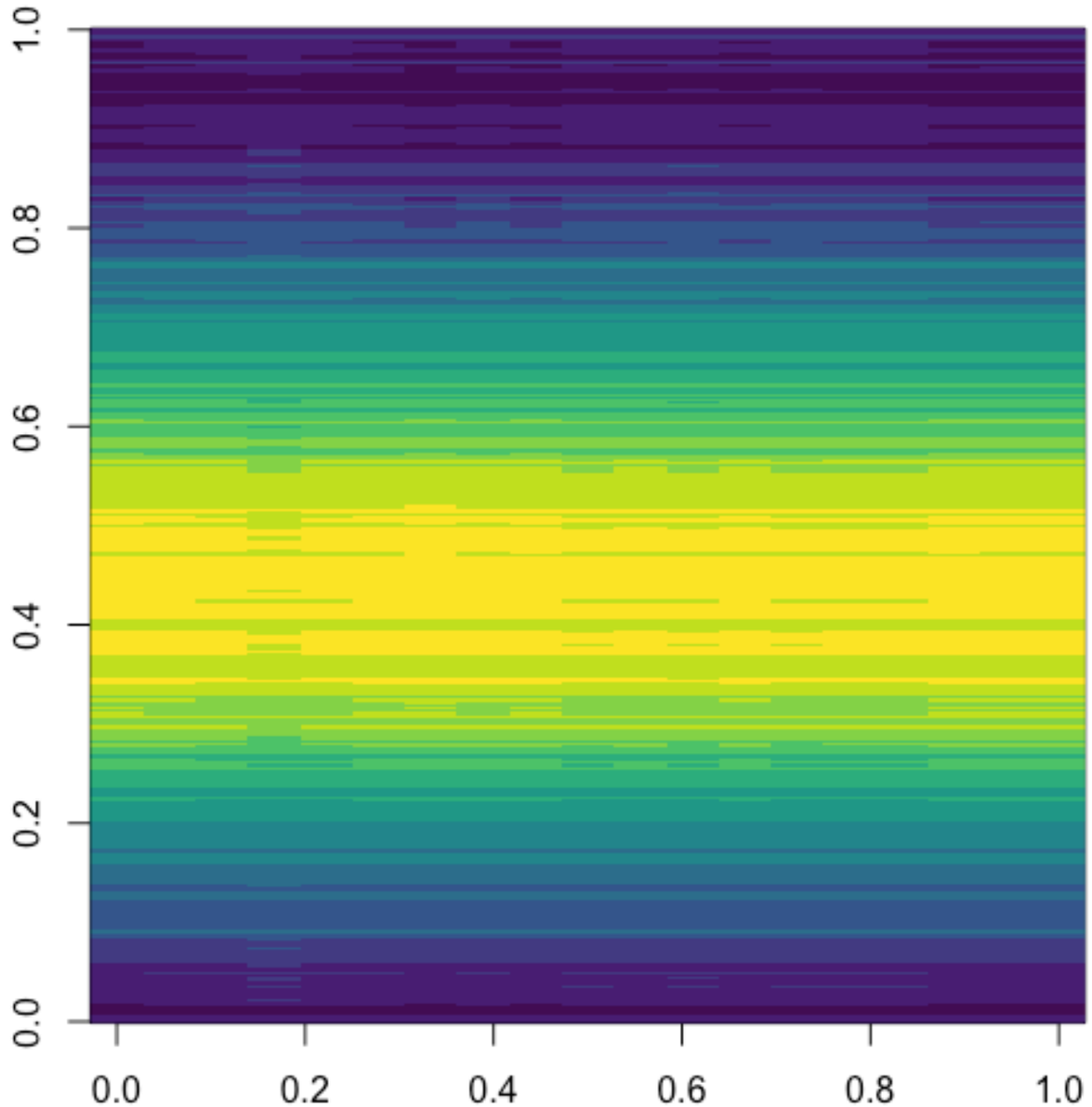


Variance Explained

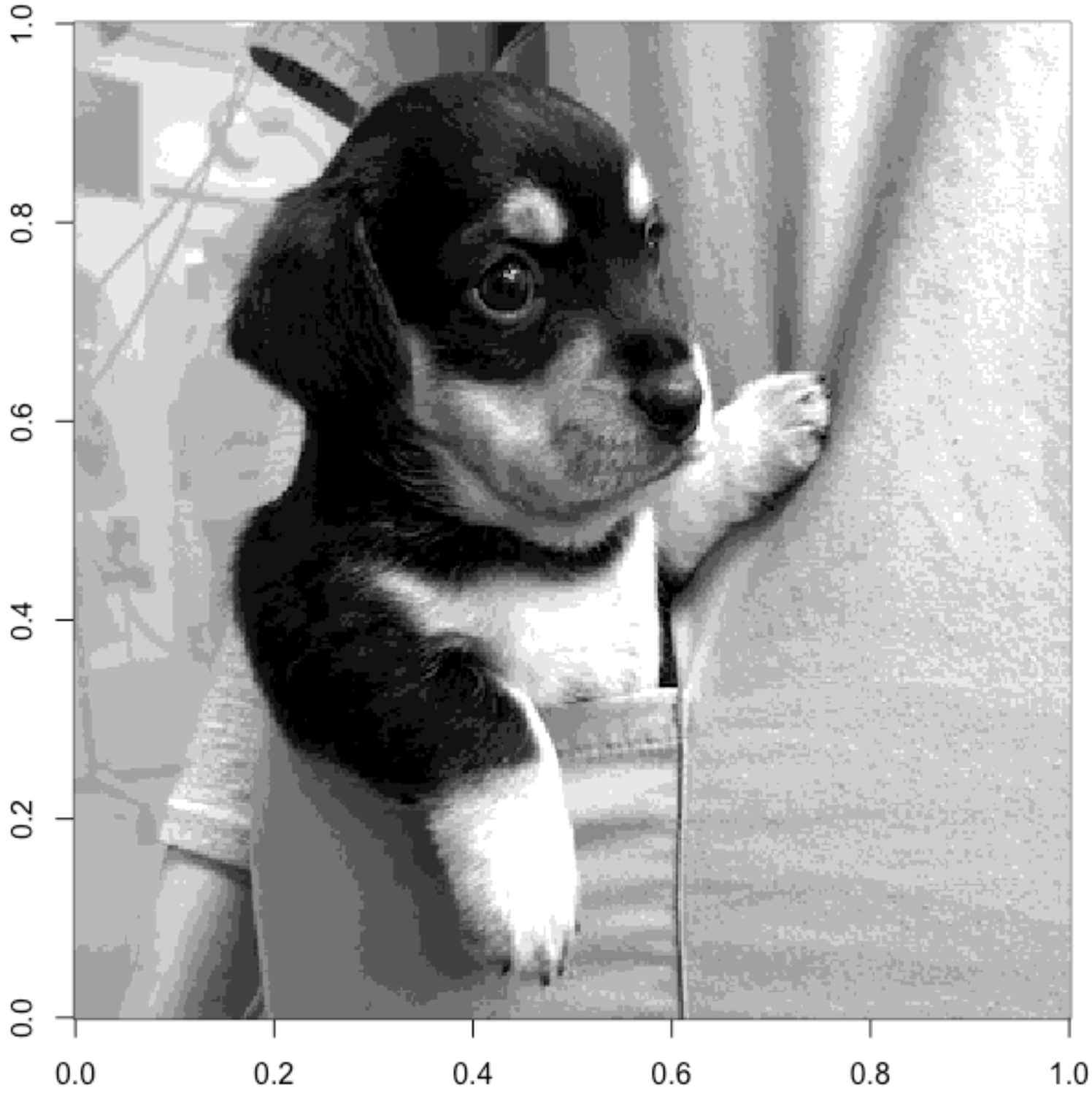


First Approximation

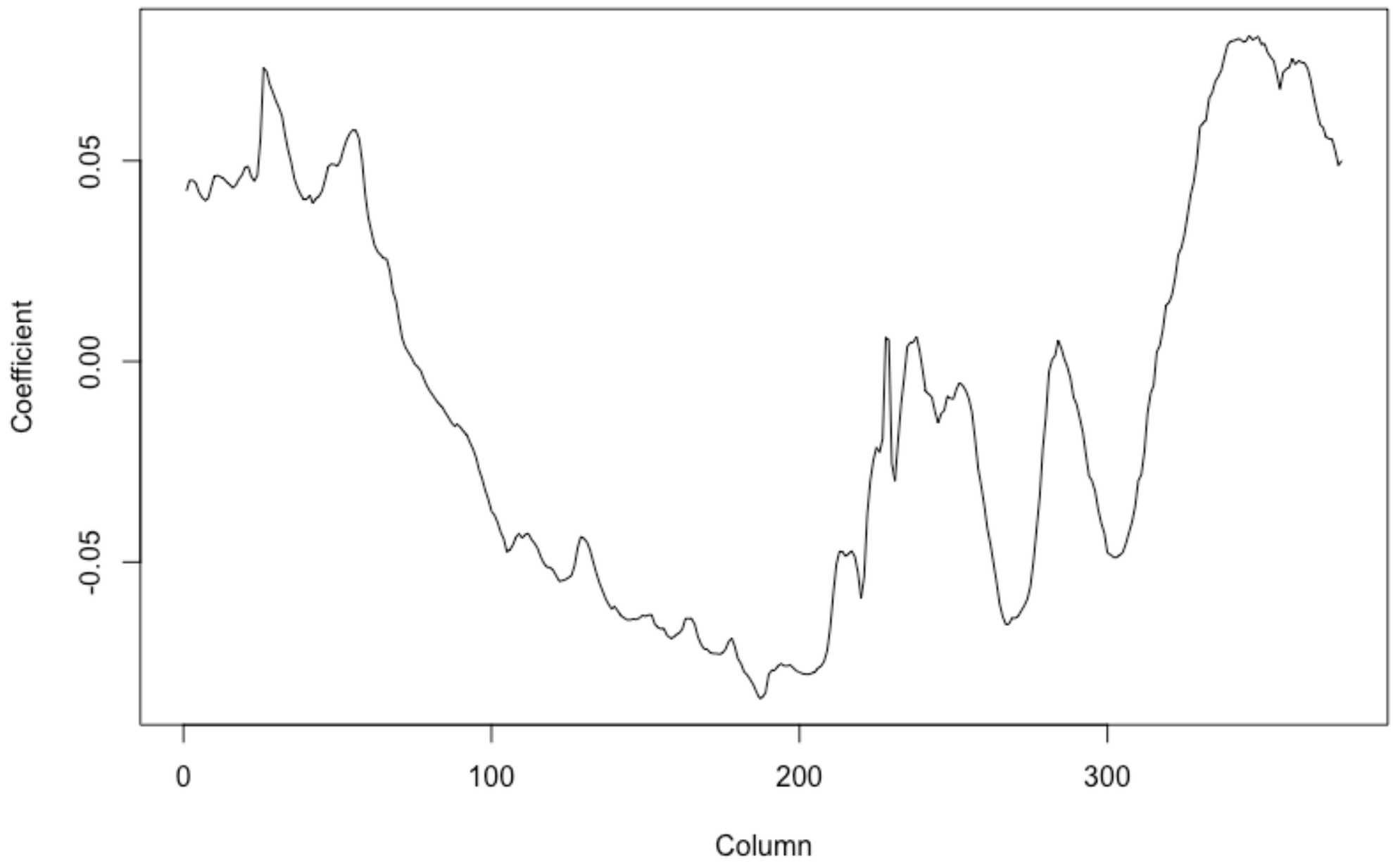




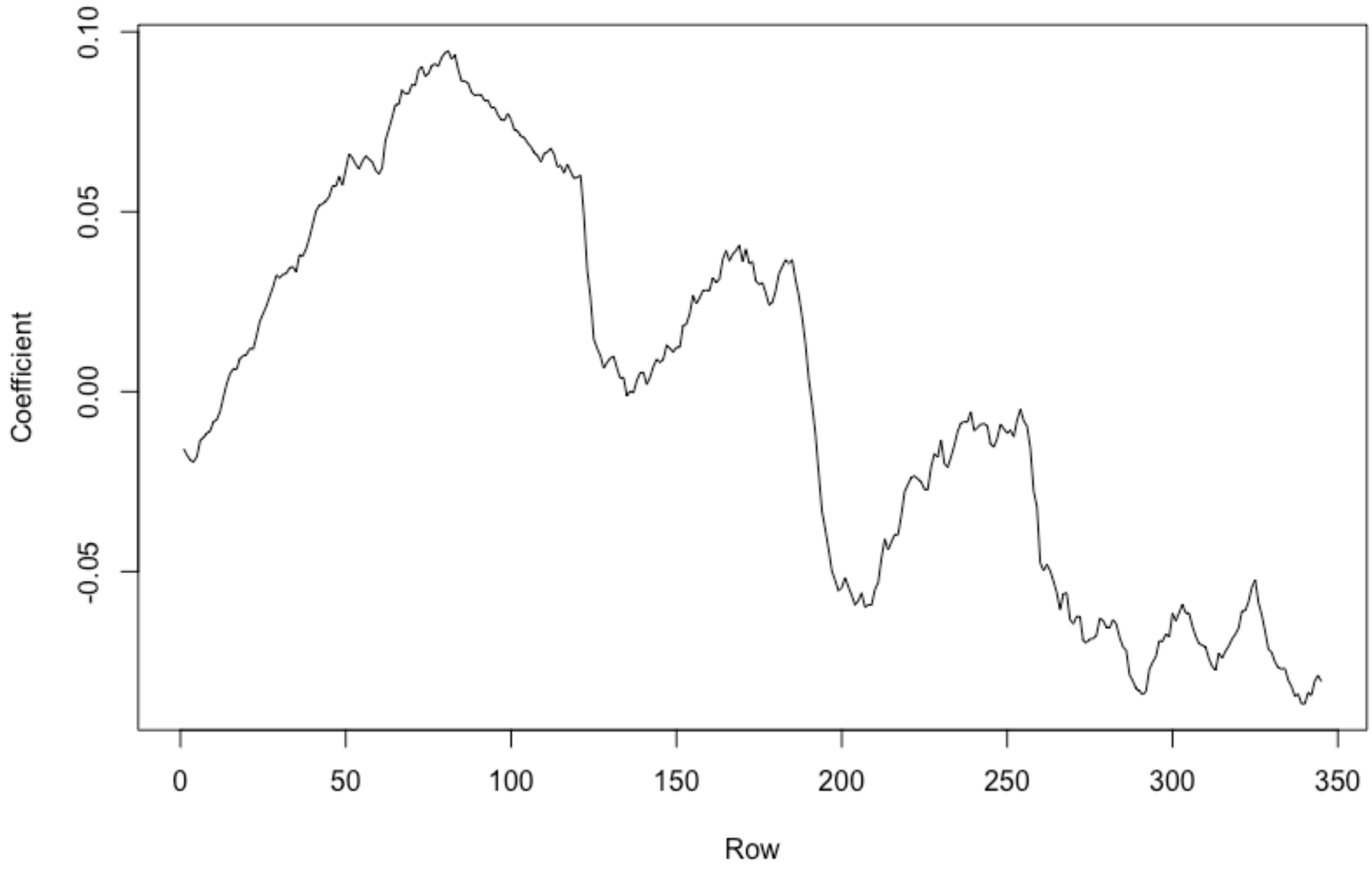
A Matrix



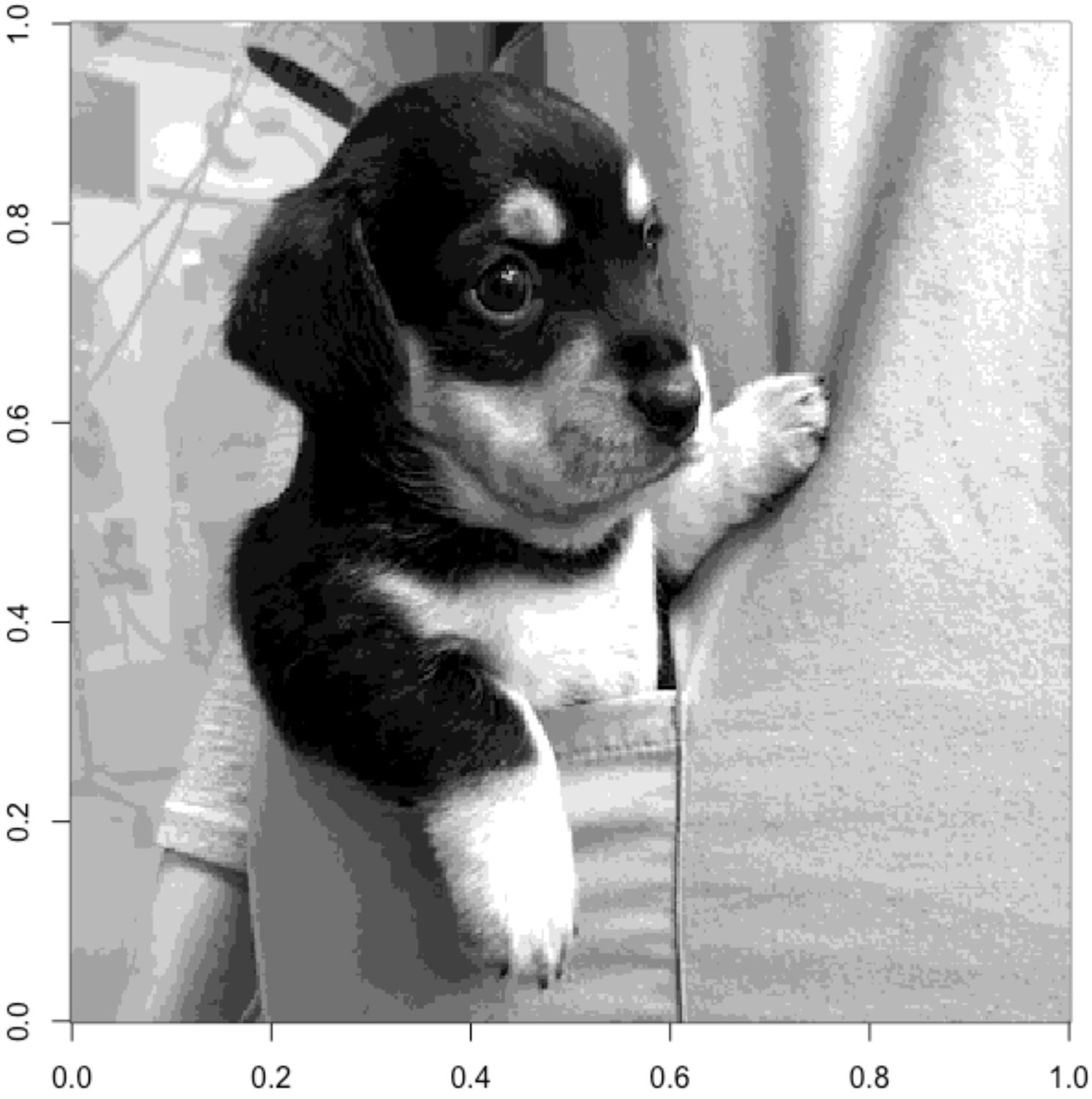
Column Basis



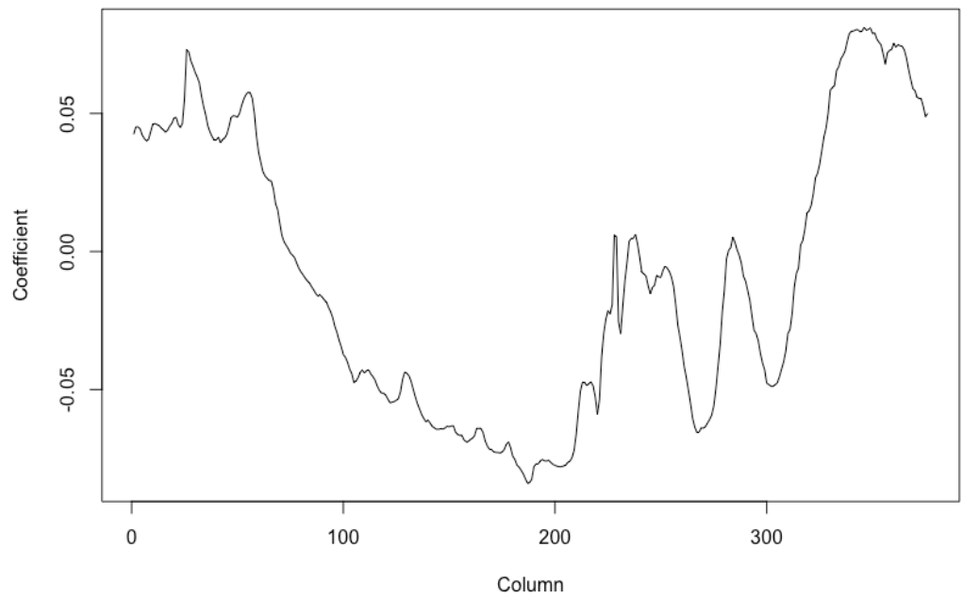
Row Basis



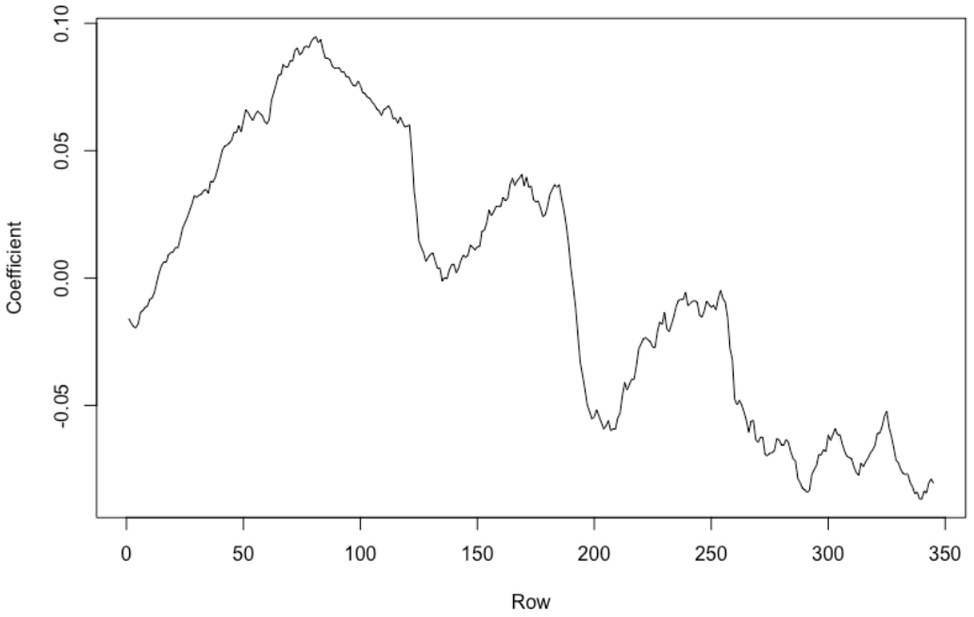
A Matrix



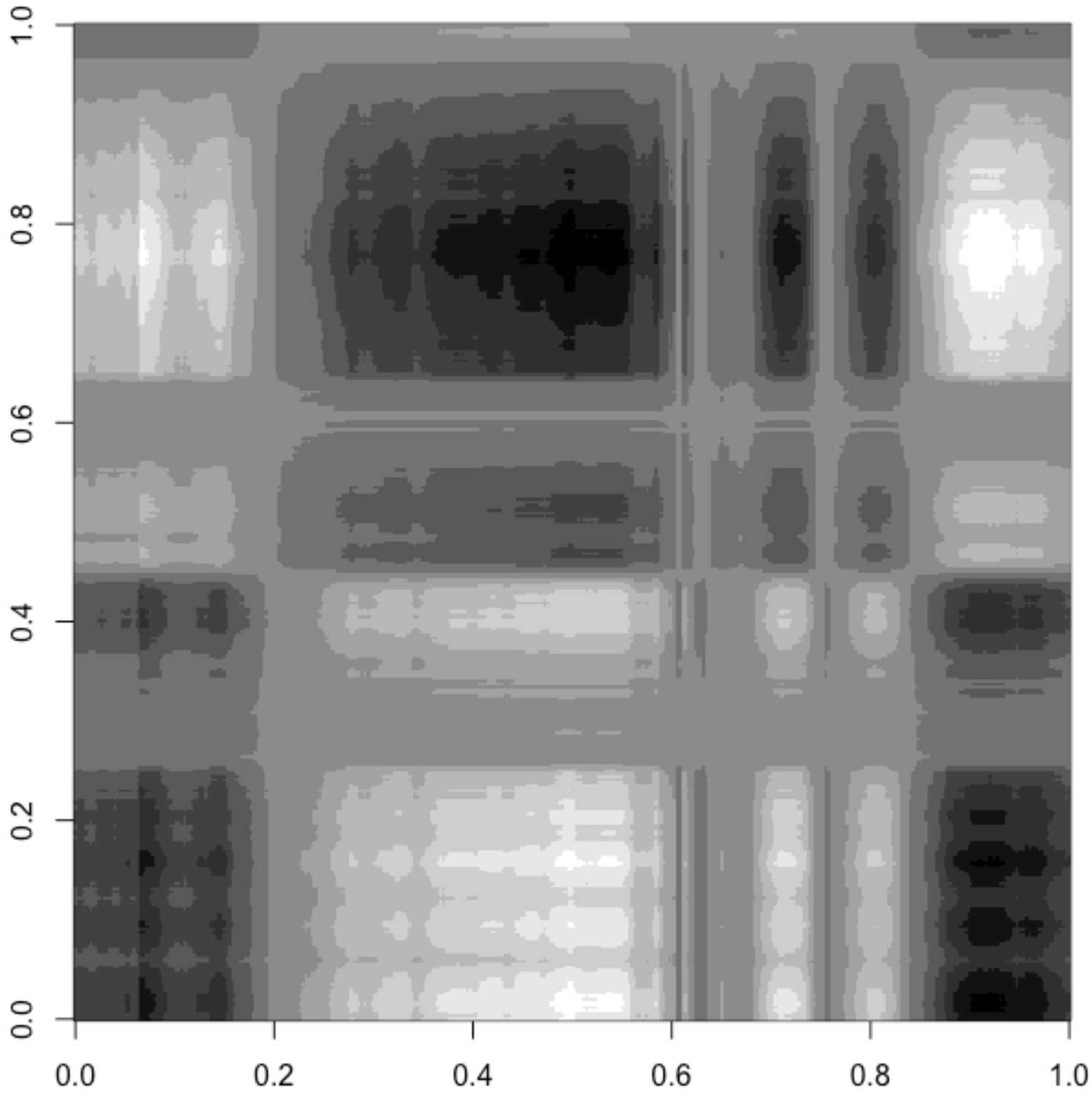
Column Basis



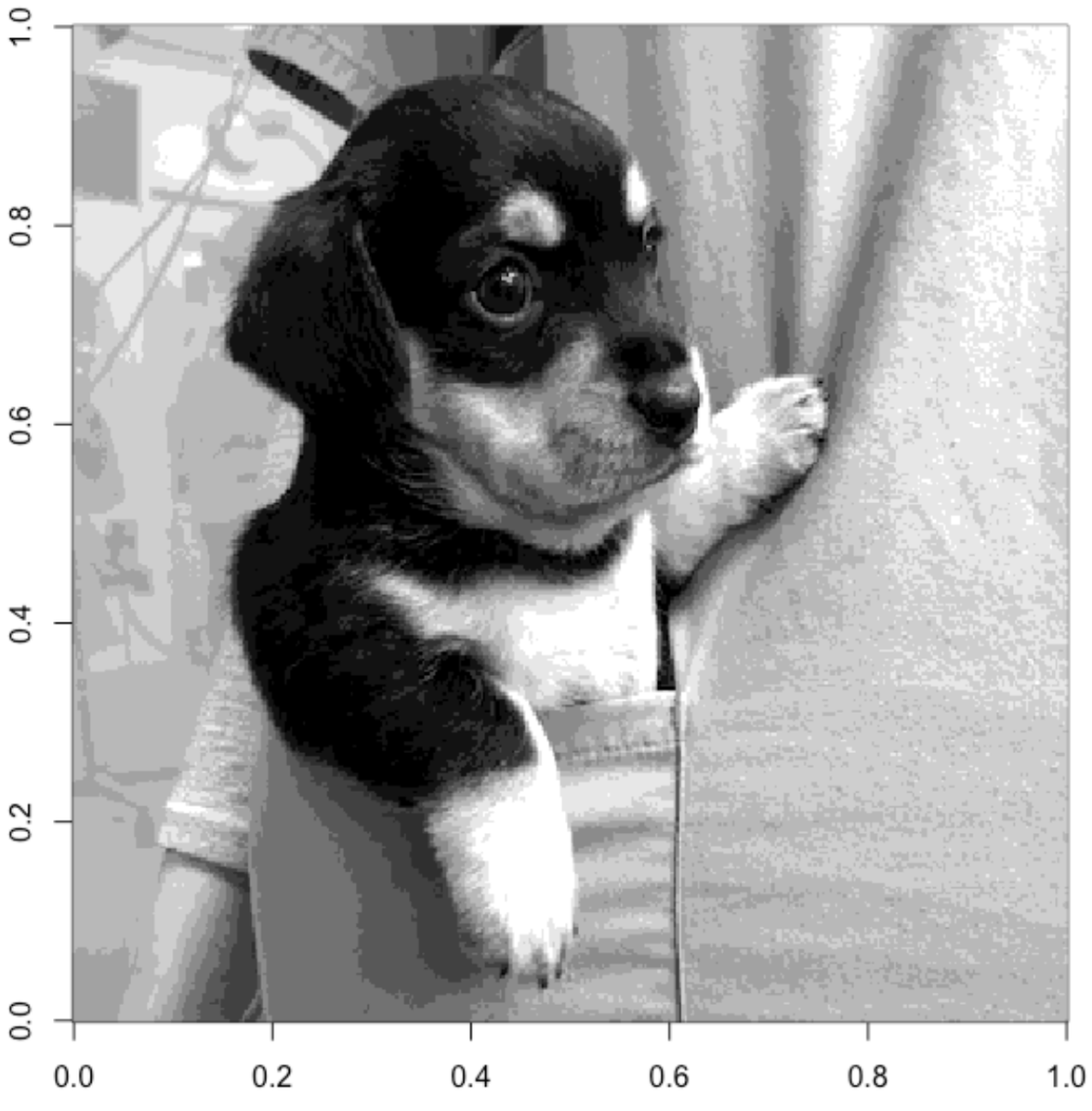
Row Basis

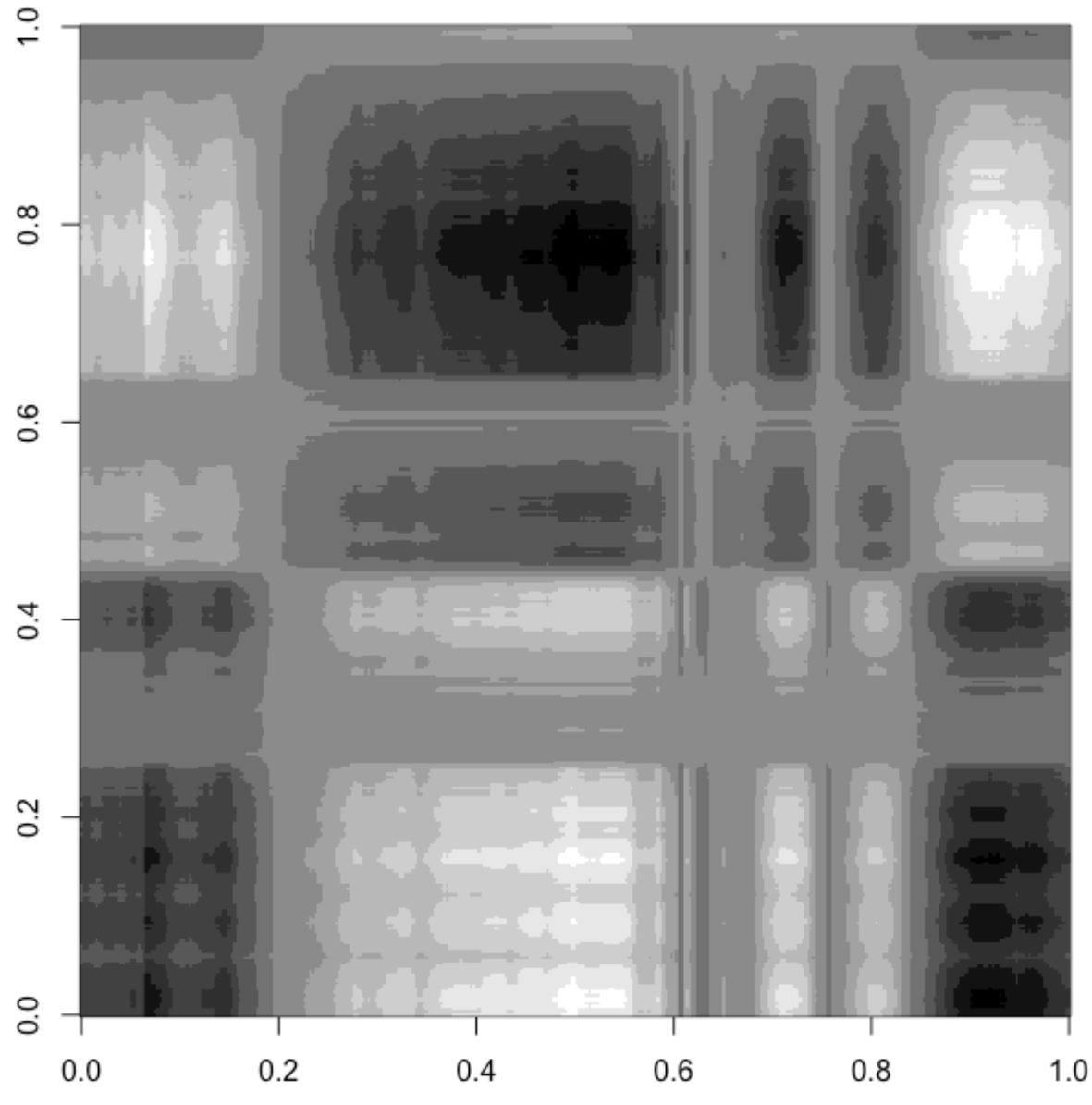


First Approximation

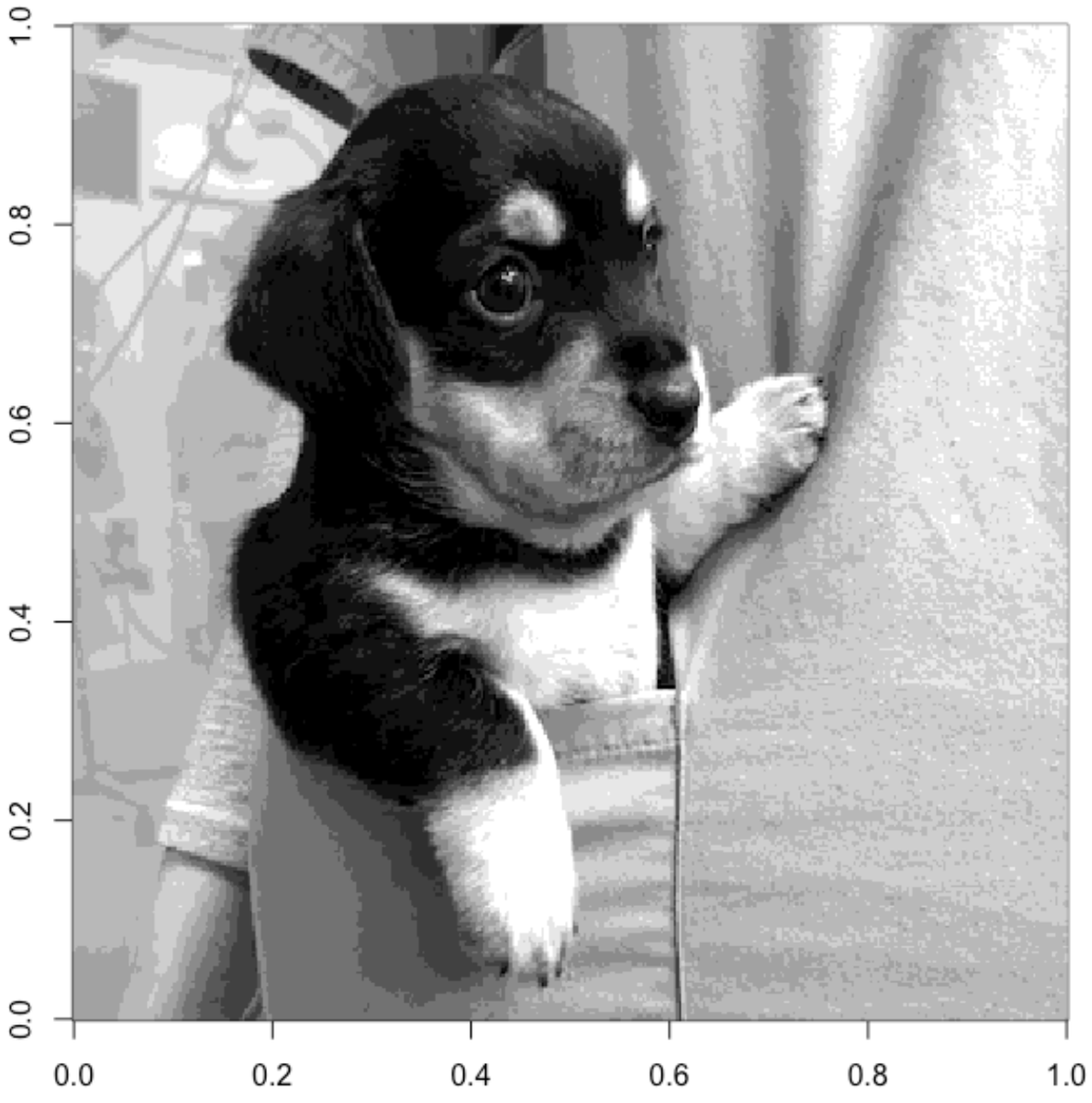


First Approximation

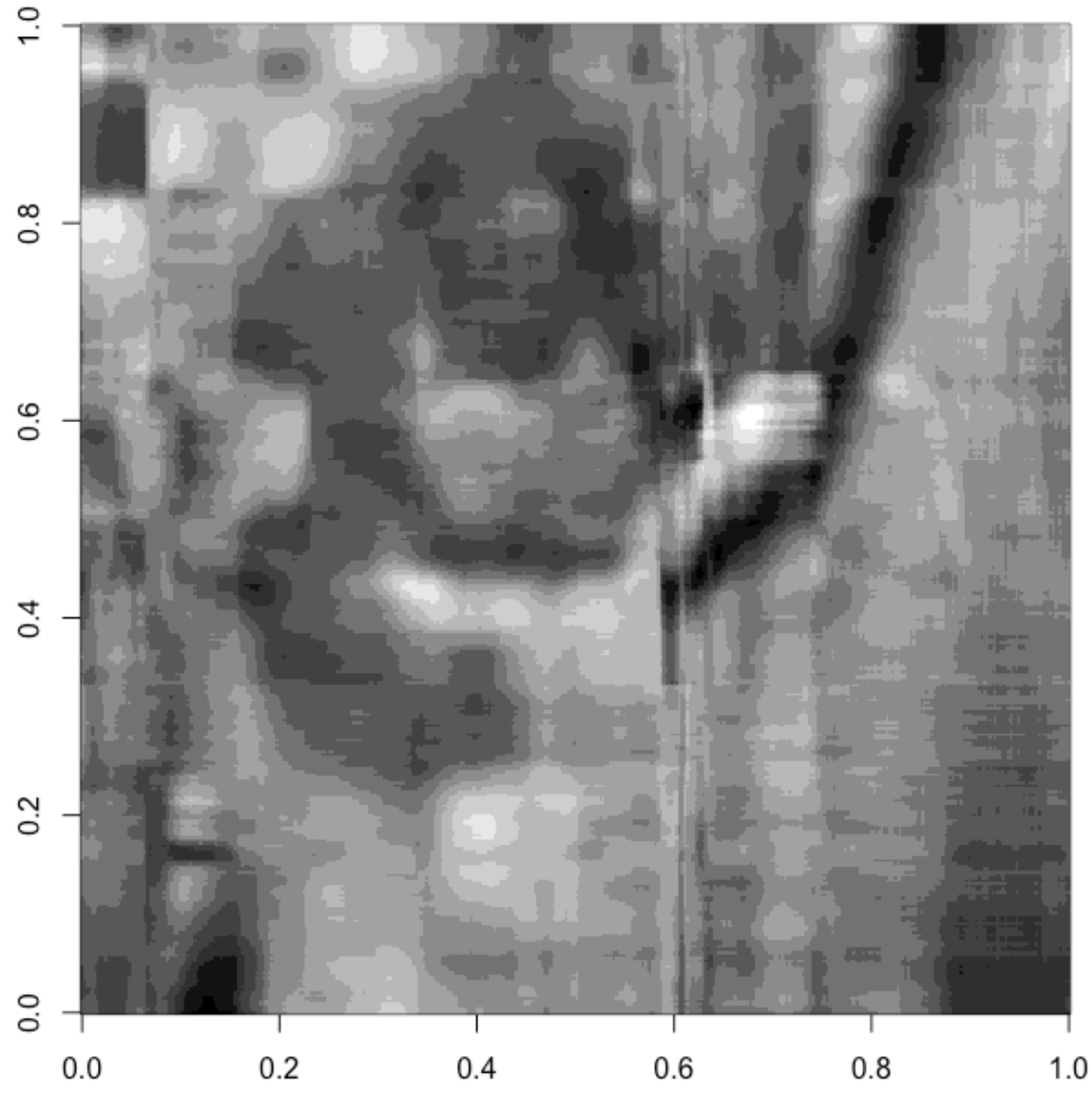




First Approximation



89 KB



59 KB

SVD Usage

- SVD can be applied to any matrix and is sometimes useful as a first cut
- Lack of functional form is good for exploration/discovery
- SVD is over-parametrized if there are strong prior hypotheses (i.e. just use a model)
- Can inspire specific models for next steps

Parsimony

- Most bases map N observations to N (or more!) basis functions/coefficients (by default, no data reduction)
- Penalization or truncation or thresholding is need to produce a parsimonious representation
- In practice, we often eyeball it (esp. for EDA purposes)
- We can also us things like cross-validation, i.e. minimize bias/variance

Summary

- Basis methods allow us to build on dictionaries of functions that summarize much previous experience

- A basis should be chosen based on the nature of the data, its characteristics, its context, and the goal of analysis

- Given a basis, we see if the data are correlated with basis elements