ESS212 HW3

Yiting Yu

2024-02-24

Yiting's Github repository url

Problem 1

$$
\begin{aligned}
(x_1, y_1) =& (-4, 30) \\
(x_2, y_2) =& (0, 2) \\
(x_3, y_3) =& (4, 6) \\
y_i =& a(x_i)^2 + bx_i + c_i + e_i \qquad\qquad i = 1, 2, 3 \\
\Rightarrow & \begin{cases} 30 = a(-4)^2 + b(-4) + c \\ 2 = a(0)^2 + b(0) + c \\ 6 = a(4)^2 + 4b + c \end{cases} \\
\Rightarrow & \begin{cases} 30 = 16a - 4b + c \\ 2 = c \\ 6 = 16a + 4b + c \end{cases} \\
\Rightarrow & \begin{cases} a = 1 \\ b = -3 \\ c = 2 \end{cases}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{d} &= \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix}' \\
\mathbf{a} &= \begin{bmatrix} x_1^2 & x_2^2 & x_3^2 \end{bmatrix}' \\
\mathbf{b} &= \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}' \\
\mathbf{e} &= \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}' \\
\mathbf{A} &= \begin{bmatrix} x_1^2 & x_1 & c_1 \\ x_2^2 & x_2 & c_2 \\ x_3^2 & x_3 & c_3 \end{bmatrix} = \begin{bmatrix} 16 & -4 & 1 \\ 0 & 0 & 1 \\ 16 & 4 & 1 \end{bmatrix} \\
\mathbf{d} &= \begin{bmatrix} 30 \\ 2 \\ 6 \end{bmatrix} \\
\mathbf{A}\beta =& \mathbf{d} \\
\beta =& (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{d} \\
\mathbf{e} =& \mathbf{d} - \hat{\mathbf{d}}
\end{aligned}
$$

```r
A <- matrix(c(16, -4, 1, 0, 0, 1, 16, 4, 1), 3, byrow = TRUE)
d <- c(30, 2, 6)

beta <- solve(A, d)
```

```r
#beta <- solve(t(A) %*% A) %*% t(A) %*% d

d_hat <- A %*% beta
e <- d - d_hat

# result
beta
```

```
## [1]  1 -3  2
```

```r
e
```

```
##      [,1]
## [1,]    0
## [2,]    0
## [3,]    0
```

The above results show that:

$$\beta = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ -3 \\ 2 \end{bmatrix}$$

$$\mathbf{e} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$y_1 = A(x_1 - B)(x_1 - C)$$
$$y_2 = A(x_2 - B)(x_2 - C)$$
$$y_3 = A(x_3 - B)(x_3 - C)$$

```r
fn <- function(beta, x, y) {
  A <- beta[1]
  B <- beta[2]
  C <- beta[3]

  F <- numeric(length(y))

  for(i in 1:length(y)) {
    F[i] <- A*(x[i]-B)*(x[i]-C) - y[i]
  }
  return(F)
}

# A, B, C for initial iterate
beta <- c(A=1, B=2, C=3)

x <- c(-4, 0, 4)
y <- c(30, 2, 6)

# Newton's method
n <- 100
for (iter in 1:n) {
  F <- fn(beta, x, y)
  J <- jacobian(func = function(b) fn(b, x, y), x = beta)
  delta <- solve(J) %*% F
  beta <- beta - delta
```

```r
  if (max(abs(delta)) <  1e-6) {
    cat("Convergence achieved after", iter, "iterations.\n")
    break
  }
}
```

```
## Convergence achieved after 2 iterations.
```

```r
# result
cat("Estimates for A, B, C:\n")
```

```
## Estimates for A, B, C:
```

```r
print(beta)
```

```
##      [,1]
## [1,]    1
## [2,]    2
## [3,]    1
```

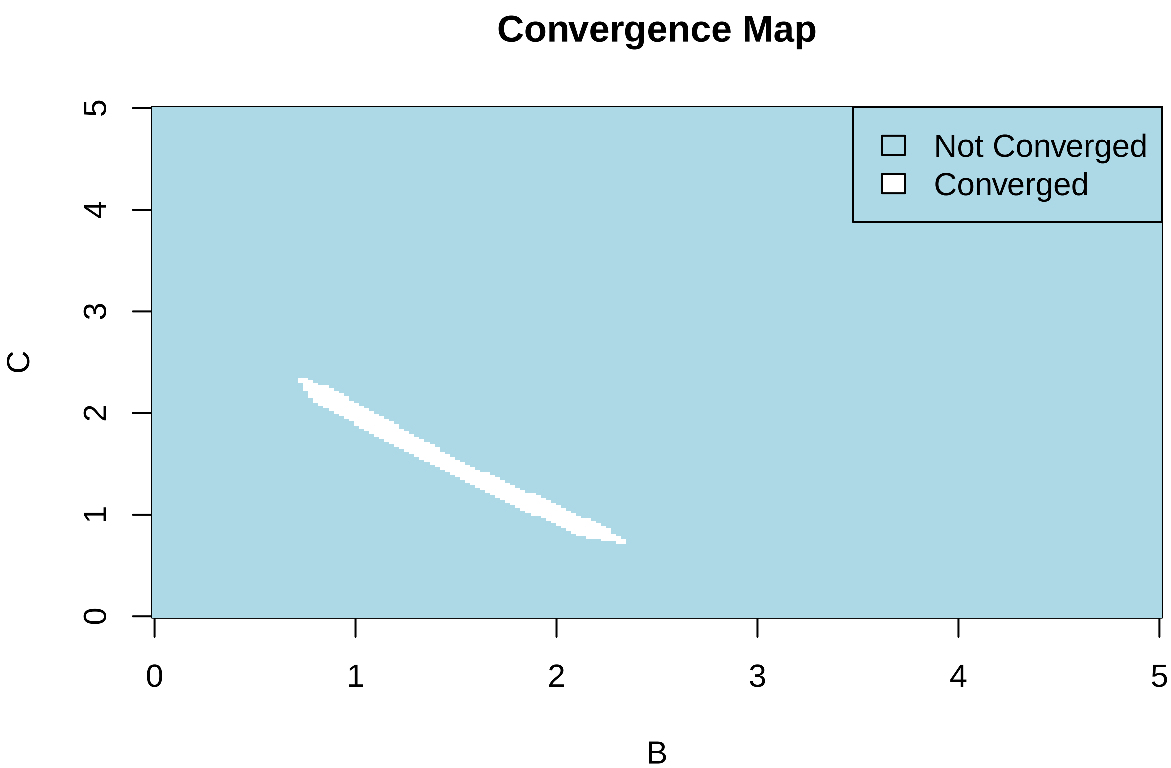
The above results show that:

$$\Rightarrow \begin{cases} A = 1 \\ B = 2 \\ C = 1 \end{cases}$$

If we choose $A = 0$ as the initial iteration, the model will not work because $A = 0$ will result in any value assigned to x_i, y_i always equal to 0.That's why I rule out the value of $A = 0$ as a possible value.

```r
B_range <- seq(0, 5, length.out = 200)
C_range <- seq(0, 5, length.out = 200)

x <- c(-4, 0, 4)
y <- c(30, 2, 6)

fn_F <- function(B, C, x, y) {
  A <- 1
  F <- numeric(length(x))
  for (i in 1:length(x)) {
    F[i] <- A*(x[i]-B)*(x[i]-C) - y[i]
  }
  return(F)
}

# initialize matrix to store convergence results
con_grid <- matrix(0, length(B_range), length(C_range))

# iterate over the ranges of B and C
for (i in 1:length(B_range)) {
  for (j in 1:length(C_range)) {
    F <- fn_F(B_range[i], C_range[j], x, y)
    con_criteria <- sum(abs(F))
    con_grid[i, j] <- ifelse(con_criteria < 1, 1, 0)
  }
}

colors <- colorRampPalette(c("lightblue", "aliceblue", "white"))(n = 2)
```

```r
image(x = B_range, y = C_range, z = con_grid, col = colors, xlab = "B", ylab = "C",
      main = "Convergence Map")
legend("topright", legend = c("Not Converged", "Converged"), fill = c(colors[1], colors[2]))
```



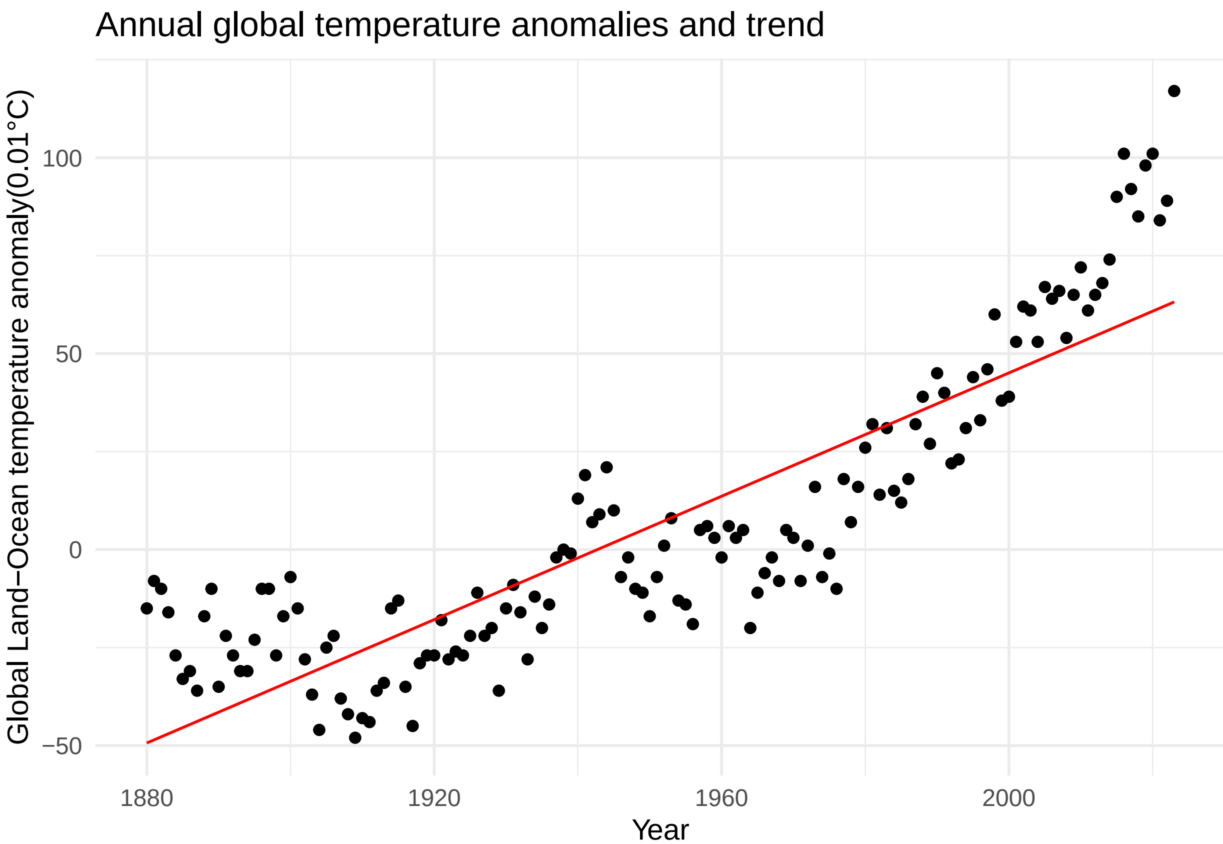
Problem 2

```r
# download the GISTEMP dataset
url <- "https://data.giss.nasa.gov/gistemp/tabledata_v4/GLB.Ts+dSST.txt"
save_f <- tempfile()
GET(url, write_disk(save_f))
```

```
## Response [https://data.giss.nasa.gov/gistemp/tabledata_v4/GLB.Ts+dSST.txt]
##   Date: 2024-03-01 07:55
##   Status: 200
##   Content-Type: text/plain
##   Size: 16.8 kB
## <ON DISK>  /var/folders/wb/8ngyfb090nz04193w4760y6c0000gn/T//RtmpkyMRgq/filea6db5cde60e1
```

```r
# clean the dataset
lines <- read_lines(save_f)

# filter lines to include only those with numeric data at the beginning (years)
filter_l <- grep("^\\s*\\d{4}", lines, value = TRUE)

# determine the maximum number of columns
max_cols <- max(sapply(filter_l, function(line) length(strsplit(line, "\\s+")[[1]])))

col_names <- c("Year", "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep",
               "Oct", "Nov", "Dec", "J-D", "D-N", "DJF", "MAM", "JJA", "SON")
```

```r
extra_cols <- max_cols - length(col_names)
if(extra_cols > 0) {
  col_names <- c(col_names, paste0("Extra", 1:extra_cols))
}

# combine the filtered lines into a single character vector
data_text <- paste(filter_l, collapse = "\n")

# read the combined text as a table
data <- read.table(text = data_text, header = FALSE, fill = TRUE, col.names = col_names)
clean_d <- data[data$`J.D` != "****", ]

#-----------------------------------------

# fit the model
yrs <- as.numeric(clean_d$Year)
mid_point <- mean(yrs)
annual_means <- as.numeric(clean_d$`J.D`) # `annual_means` is the mean of the annual temperature

model <- lm(annual_means ~ I(yrs - mid_point), na.action = na.exclude)

mu <- coef(model)[1] # intercept
m <- coef(model)[2]

# plotting
trend <- data.frame(Year = yrs, Trend = mu + m * (yrs - mid_point))
ggplot(data = clean_d, aes(x = Year, y = annual_means)) +
  geom_point() +
  geom_line(data = trend, aes(x = Year, y = Trend), color = "red") +
  labs(title = "Annual global temperature anomalies and trend",
       x = "Year", y = "Global Land-Ocean temperature anomaly(0.01°C)") +
  theme_minimal()
```




```
# the estimated values of mu and m
cat("Estimated mu:", mu, "\n")
```

```
## Estimated mu: 6.923611
```

```
cat("Estimated m:", m, "\n")
```

```
## Estimated m: 0.7868961
```

```
# the standard deviation of the residuals
sd_residual <- sd(resid(model))
cat("Standard deviation of the residuals:", sd_residual, "\n")
```

```
## Standard deviation of the residuals: 18.42912
```

The above results show that:

$$\mu = 6.923611$$
$$m = 0.7868961$$

The standard deviation of the components of the $\mathbf{e}$ vector is 18.42912.

Problem 3

There are 16 seedlings to be planted in 4 different conditions house. Since four seedlings are planted in each greenhouse, I assume seedlings $1, 5, 9, 13$ were planted in Ba greenhouse, $2, 6, 10, 14$ were planted in Bb greenhouse, $3, 7, 11, 15$ were planted in Aa greenhouse, and seedlings $4, 8, 12, 16$ were planted in Ab greenhouse.

$$
\begin{aligned}
d_i =& \mu + \beta_1 \Delta_{CO_2,i} + \beta_2 \Delta_{H_2O,i} + \beta_3 \Delta_{CO_2,i} \Delta_{H_2O,i} + e_i \\
\Rightarrow & \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} = \begin{bmatrix} d_5 \\ d_6 \\ d_7 \\ d_8 \end{bmatrix} = \begin{bmatrix} d_9 \\ d_{10} \\ d_{11} \\ d_{12} \end{bmatrix} = \begin{bmatrix} d_{13} \\ d_{14} \\ d_{15} \\ d_{16} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \\
\Rightarrow \mathbf{d} =& \mathbf{A}\beta + \mathbf{e}
\end{aligned}
$$

$$
\Rightarrow \mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \\ d_9 \\ d_{10} \\ d_{11} \\ d_{12} \\ d_{13} \\ d_{14} \\ d_{15} \\ d_{16} \end{bmatrix} \mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \beta = \begin{bmatrix} \mu \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}
$$

```
A <- matrix(c(
  1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0,
  1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0,
  1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0,
  1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0
), nrow = 16, byrow = TRUE)

(Aprime_A <- t(A) %*% A)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   16    8    8    4
## [2,]    8    8    4    4
## [3,]    8    4    8    4
## [4,]    4    4    4    4
```

$$
\mathbf{A}'\mathbf{A} = \begin{bmatrix} 16 & 8 & 8 & 4 \\ 8 & 8 & 4 & 4 \\ 8 & 4 & 8 & 4 \\ 4 & 4 & 4 & 4 \end{bmatrix}
$$

$$
\mathbf{A}'\mathbf{d} = \begin{bmatrix} \Sigma_{i=1}^{16} d_i \\ d_1 + d_2 + d_5 + d_6 + d_9 + d_{10} + d_{13} + d_{14} \\ d_2 + d_4 + d_6 + d_8 + d_{10} + d_{12} + d_{14} + d_{16} \\ d_2 + d_6 + d_{10} + d_{14} \end{bmatrix}
$$