RAD-Median and Zt-Mean

Isaac Rankin

2024

Function to Compute the RAD-Median and Trimmed Mean

```r
TRIMMEDmean = function(X, RAD_Median = TRUE, TRIM_percentage = 80) {
  # Set RAD_Median = TRUE to compute the RAD-Median
  # To compute the Zt-Mean, set RAD_Median = FALSE and choose the percentage of points to trim using TR

  # X is a matrix of point coordinates
  n = nrow(X) # number of observations
  p = ncol(X) # dimension of data and must be greater or equal to 2
  rownames(X) = paste0(rep("x", n), 1:n) # xi is observation 1
  colnames(X) = paste0(rep("c", p), 1:p) # ci is coordinate i

  # If RAD_Median = TRUE, use TRIM_Remain = 1
  if (RAD_Median == TRUE) {
    TRIM_Remain = 1
  }

  # If RAD_Median = FALSE, use TRIM_proportion and n to find TRIM_Remain
  if (RAD_Median == FALSE) {
    TRIM_Remain = min(n, max(1, round(n * (1 - TRIM_percentage/100))))
  }

  # If all points are kept, return the regular mean
  if (TRIM_Remain == n) {
    print("The mean of all points is:")
    return(colMeans(X))
  }

  # Matrix of pairwise distances
  D = as.matrix(dist(X, method = "euclidean"))
  colnames(D) = paste0(rep("x", n), 1:n)
  rownames(D) = paste0(rep("x", n), 1:n)

  # Find points for trimmed mean or median
  points = n # initially have n points
  prev_points = n # number of points in previous iteration
  POINTS_to_use = 0

  # Iteratively remove points
  while (points > TRIM_Remain) {
    Prev = D
    prev_points = ncol(Prev)

    #### Handle the single point RAD-Median case separately
    if (points == 3 & TRIM_Remain == 1) {
      three_points = rownames(D)
      farthest = which(D == max(D), arr.ind = TRUE)
      last_point = three_points[!(three_points %in% rownames(farthest))]

      ## If all the last three points have farthest distance
      if (length(last_point) == 0) {
        POINTS_to_use = rownames(Prev)
        break
      }
      ##
      cat("The RAD-Median is:", last_point, "\n")
      return(X[last_point,])
    }
    #####

    farthest = which(D == max(D), arr.ind = TRUE)
    D = D[!(rownames(D) %in% rownames(farthest)), !(colnames(D) %in% rownames(farthest))]
    points = max(ncol(D), 0)

    # Ensure that the number of points kept is the fewest possible while at least TRIM_Remain
    if (points == TRIM_Remain) {
      POINTS_to_use = rownames(D)
      break
    } else if (points < TRIM_Remain) {
      POINTS_to_use = rownames(Prev)
      break
    }
  }

  n_use = length(POINTS_to_use)
  points_in_mean = rep(0, n_use)
  for (a in 1:n_use) {
    points_in_mean[a] = which(rownames(X) == POINTS_to_use[a])
  }


  mean_of_points = colMeans(X[points_in_mean,])
  print("The points used are:")
  print(X[points_in_mean,])

  if (RAD_Median == TRUE){
    print("The RAD-Median is:")
  } else{
    cat(paste("The Z", TRIM_percentage, "-Mean is:\n", sep = ""))
  }
  return(mean_of_points)
}
```

High-dimensional examples

Create an example dataset with 101 5-dimensional observations

```
set.seed(1)
n = 101 # n is the number of observations
p = 5 # p is the number of coordinates of each observation
Sample_data = matrix(data = rnorm(n*p, mean = 0, sd = 10), nrow = n, ncol = p)
```

Compute the RAD-Median

```
TRIMMEDmean(Sample_data)
```

```
## The RAD-Median is: x69
```

```
##        c1        c2        c3        c4        c5
##  1.532533  2.075383 -2.589326  2.779141 -1.096557
```

Compute the Z90-Mean

```
TRIMMEDmean(Sample_data, RAD_Median = FALSE, TRIM_percentage = 90)
```

```
## [1] "The points used are:"
##             c1         c2         c3          c4         c5
## x10 -3.0538839 -6.3573645  4.2069464 -4.54136909 -7.7211080
## x25  6.1982575  7.1266631 -3.3113204  1.70489471 -6.4117693
## x32 -1.0278773  5.3149619  5.7050764 -3.34281365 -1.9819542
## x34 -0.5380504  3.0655786 -0.9817874  9.46585640 -0.2571507
## x57 -3.6722148 -6.2126669  3.7472441 -1.95258846 -4.5303708
## x60 -1.3505460  4.2510038 -3.8923718 -6.35543001  5.9549803
## x64  0.2800216 -6.1924305  2.7005490  6.78340177  6.2801715
## x69  1.5325334  2.0753834 -2.5893258  2.77914132 -1.0965570
## x73  6.1072635 -0.7715294  1.5601168 -0.08309014  3.6438459
## x76  2.9144624  7.8763961  7.4100116 -1.63910957 -0.2790997
## x90  2.6709879 -1.7710396  3.7637029 10.91668956  1.1110643
## The Z90-Mean is:
```

```
##         c1         c2         c3         c4         c5
##  0.9146322  0.7640869  1.6653492  1.2486893 -0.4807225
```

Example for 2-dimensional data, n odd

Compute the RAD-Median

```
# Create the dataset
set.seed(12345)
n = 35 # n is the number of observations
X <- matrix(data = runif(n = 2*n, min = 0, max = 10), nrow = n, ncol = 2, byrow = TRUE)
```

```r
# Set row names as x1, x2, ...
rownames(X) <- paste0("x", 1:nrow(X))

# Set column names as c1, c2, ...
colnames(X) <- paste0("c", 1:ncol(X))

# Prepare data for plotting
plot_data <- as.data.frame(X)
plot_data$Point <- rownames(plot_data)  # Add point names for labels

TRIMMEDmean(X)
```

```
## The RAD-Median is: x11
```

```
##       c1       c2
## 4.537281 3.267524
```

Plot the dataset and RAD-Median

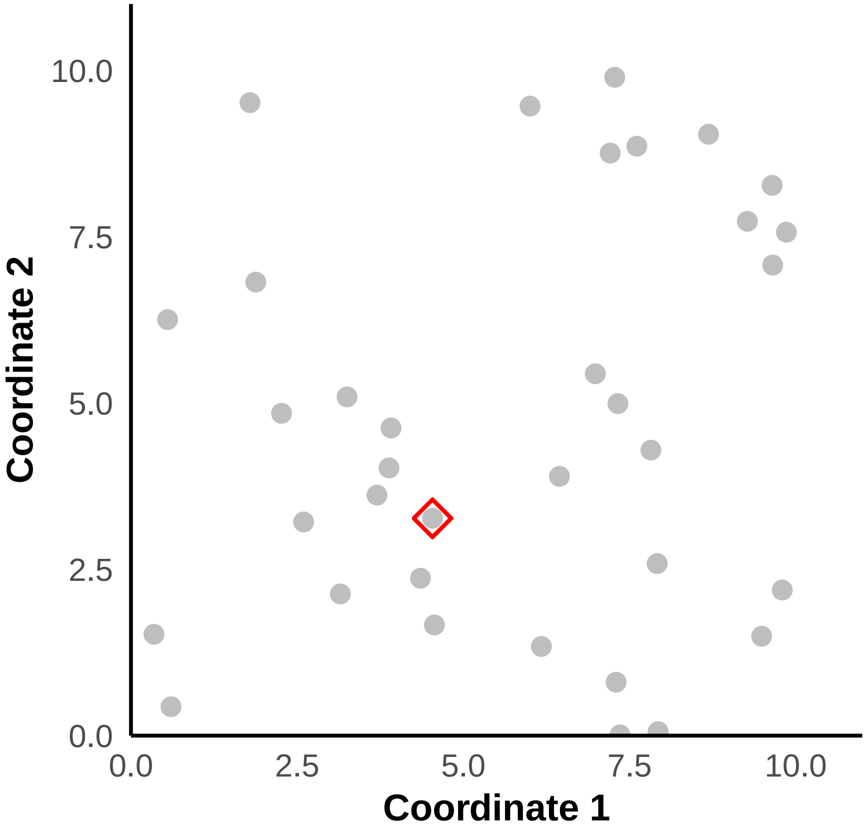
```r
# Compute the RAD-Median
trimmed_mean_result <- TRIMMEDmean(X)
```

```
## The RAD-Median is: x11
```

```r
print(trimmed_mean_result)
```

```
##       c1       c2
## 4.537281 3.267524
```

```r
# Plot the points and RAD-Median
suppressWarnings({
  p <- ggplot(plot_data, aes(x = c1, y = c2)) +
    geom_point(color = "grey", size = 3) +  # Original points
    geom_point(aes(x = trimmed_mean_result[1], y = trimmed_mean_result[2]),
               color = "red", shape = 5, size = 4, stroke = 1) +
    xlab("Coordinate 1") +
    ylab("Coordinate 2") +
    theme_minimal(base_size = 15) +
    theme(
      axis.title.x = element_text(size = 14, face = "bold"),  # Larger x-axis title
      axis.title.y = element_text(size = 14, face = "bold"),  # Larger y-axis title
      axis.text.x = element_text(size = 12),  # Larger x-axis text
      axis.text.y = element_text(size = 12),  # Larger y-axis text
      axis.line = element_line(colour = "black"),  # Add axis lines
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      panel.border = element_blank()
    ) +
    #geom_text(aes(label = Point), hjust = 1.5, vjust = 0, size = 4) +
    coord_fixed(ratio = 1) +  # Fixed aspect ratio
    scale_x_continuous(expand = c(0, 0),
                       limits = c(0, 11)) +  # Set x-axis to start from 0
    scale_y_continuous(expand = c(0, 0),
                       limits = c(0, 11))   # Set y-axis to start from 0
    print(p)
})
```



Example for 2-dimensional data, n even

Compute the RAD-Median

```
# Create the dataset
set.seed(2024)
n = 40 # n is the number of observations
X <- matrix(data = runif(n = 2*n, min = 0, max = 10), nrow = n, ncol = 2, byrow = TRUE)

# Set row names as x1, x2, ...
rownames(X) <- paste0("x", 1:nrow(X))

# Set column names as c1, c2, ...
colnames(X) <- paste0("c", 1:ncol(X))

# Prepare data for plotting
plot_data <- as.data.frame(X)
plot_data$Point <- rownames(plot_data)  # Add point names for labels
```

```r
TRIMMEDmean(X)
```

```
## [1] "The points used are:"
##           c1       c2
## x27 4.529547 4.894902
## x37 6.320666 4.440087
## [1] "The RAD-Median is:"

##       c1       c2
## 5.425107 4.667494
```

Plot the dataset and RAD-Median

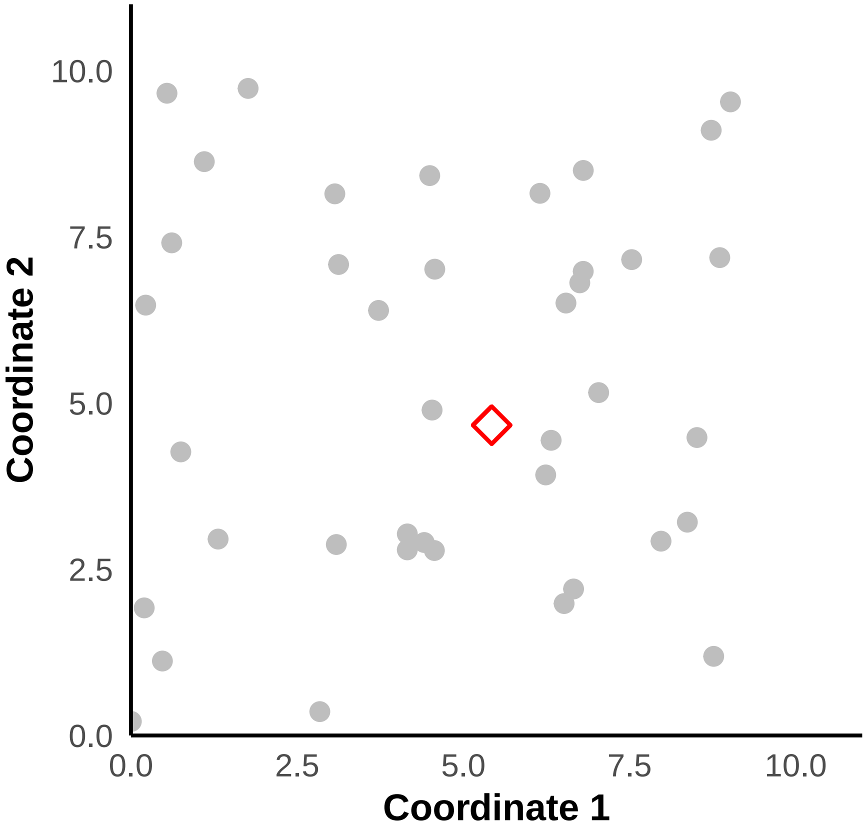
```r
# Compute the RAD-Median
trimmed_mean_result <- TRIMMEDmean(X)
```

```
## [1] "The points used are:"
##           c1       c2
## x27 4.529547 4.894902
## x37 6.320666 4.440087
## [1] "The RAD-Median is:"
```

```r
print(trimmed_mean_result)
```

```
##       c1       c2
## 5.425107 4.667494
```

```r
# Plot the points and RAD-Median
suppressWarnings({
  p <- ggplot(plot_data, aes(x = c1, y = c2)) +
    geom_point(color = "grey", size = 3) +  # Original points
    geom_point(aes(x = trimmed_mean_result[1], y = trimmed_mean_result[2]),
               color = "red", shape = 5, size = 4, stroke = 1) +
    xlab("Coordinate 1") +
    ylab("Coordinate 2") +
    theme_minimal(base_size = 15) +
    theme(
      axis.title.x = element_text(size = 14, face = "bold"),  # Larger x-axis title
      axis.title.y = element_text(size = 14, face = "bold"),  # Larger y-axis title
      axis.text.x = element_text(size = 12),  # Larger x-axis text
      axis.text.y = element_text(size = 12),  # Larger y-axis text
      axis.line = element_line(colour = "black"),  # Add axis lines
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      panel.border = element_blank()
    ) +
    #geom_text(aes(label = Point), hjust = 1.5, vjust = 0, size = 4) +
    coord_fixed(ratio = 1) +  # Fixed aspect ratio
    scale_x_continuous(expand = c(0, 0),
                       limits = c(0, 11)) +  # Set x-axis to start from 0
    scale_y_continuous(expand = c(0, 0),
                       limits = c(0, 11))   # Set y-axis to start from 0
    print(p)
})
```



Example for 2-dimensional data with outliers

Make a dataset with outliers

```
# Create the dataset
set.seed(2024)
X <- matrix(data = c(rnorm(n = 30, mean = 25, sd = 5), runif(n = 160, min = 100, max = 150)), ncol = 2,

# Set row names as x1, x2, ...
rownames(X) <- paste0("x", 1:nrow(X))

# Set column names as c1, c2, ...
colnames(X) <- paste0("c", 1:ncol(X))

# Prepare data for plotting
plot_data <- as.data.frame(X)
plot_data$Point <- rownames(plot_data)  # Add point names for labels
```

Compute the RAD-Median

```
TRIMMEDmean(X)
```

```
## The RAD-Median is: x89
```

```
##       c1       c2
## 127.5003 120.3308
```

Plot the RAD_Median

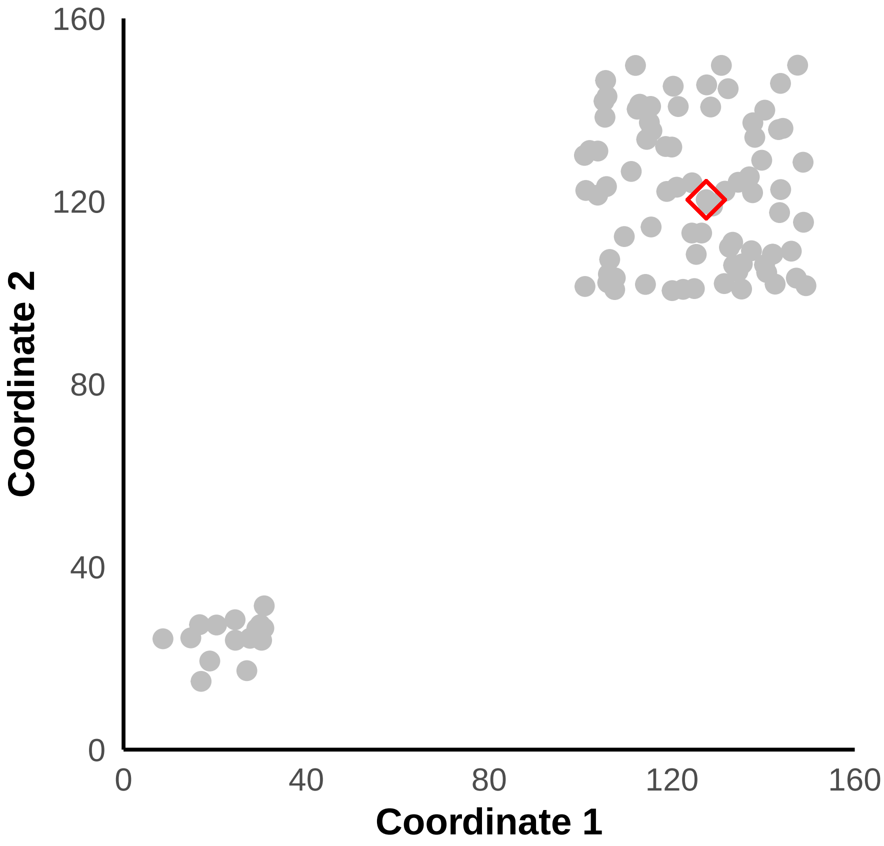
```
# Compute the RAD_Median
trimmed_mean_result <- TRIMMEDmean(X)
```

```
## The RAD-Median is: x89
```

```
print(trimmed_mean_result)
```

```
##       c1       c2
## 127.5003 120.3308
```

```
# Plot the points and RAD_Median
suppressWarnings({
  p <- ggplot(plot_data, aes(x = c1, y = c2)) +
    geom_point(color = "grey", size = 3) +  # Original points
    geom_point(aes(x = trimmed_mean_result[1], y = trimmed_mean_result[2]),
               color = "red", shape = 5, size = 4, stroke = 1) +
    xlab("Coordinate 1") +
    ylab("Coordinate 2") +
    theme_minimal(base_size = 15) +
    theme(
      axis.title.x = element_text(size = 14, face = "bold"),  # Larger x-axis title
      axis.title.y = element_text(size = 14, face = "bold"),  # Larger y-axis title
      axis.text.x = element_text(size = 12),  # Larger x-axis text
      axis.text.y = element_text(size = 12),  # Larger y-axis text
      axis.line = element_line(colour = "black"),  # Add axis lines
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      panel.border = element_blank()
    ) +
    #geom_text(aes(label = Point), hjust = 1.5, vjust = 0, size = 4) +
    coord_fixed(ratio = 1) +  # Fixed aspect ratio
    scale_x_continuous(expand = c(0, 0),
                       limits = c(0, 160)) +  # Set x-axis to start from 0
    scale_y_continuous(expand = c(0, 0),
                       limits = c(0, 160))   # Set y-axis to start from 0
    print(p)
})
```



Compute the Z80-Mean

```
TRIMMEDmean(X, RAD_Median = FALSE, TRIM_percentage = 80)
```

```
## [1] "The points used are:"
##            c1       c2
## x17 118.6232 131.9705
## x20 115.4449 114.3620
## x21 132.5791 109.9220
## x22 131.6033 122.2004
## x24 114.2039 101.7947
## x25 133.2903 111.0194
## x26 109.5739 112.2678
## x37 111.0962 126.5194
## x45 124.3500 113.0356
## x59 126.5000 113.0381
## x60 128.8879 118.9524
## x61 118.8616 122.1387
## x65 125.3128 108.3656
## x70 127.9574 119.4055
## x73 119.9645 131.8515
## x84 124.4065 124.0291
## x89 127.5003 120.3308
## x92 121.0447 123.0688
## x95 106.3806 107.2539
## The Z80-Mean is:
```

```
##       c1       c2
## 121.9779 117.4488
```

Plot the Z80-Mean

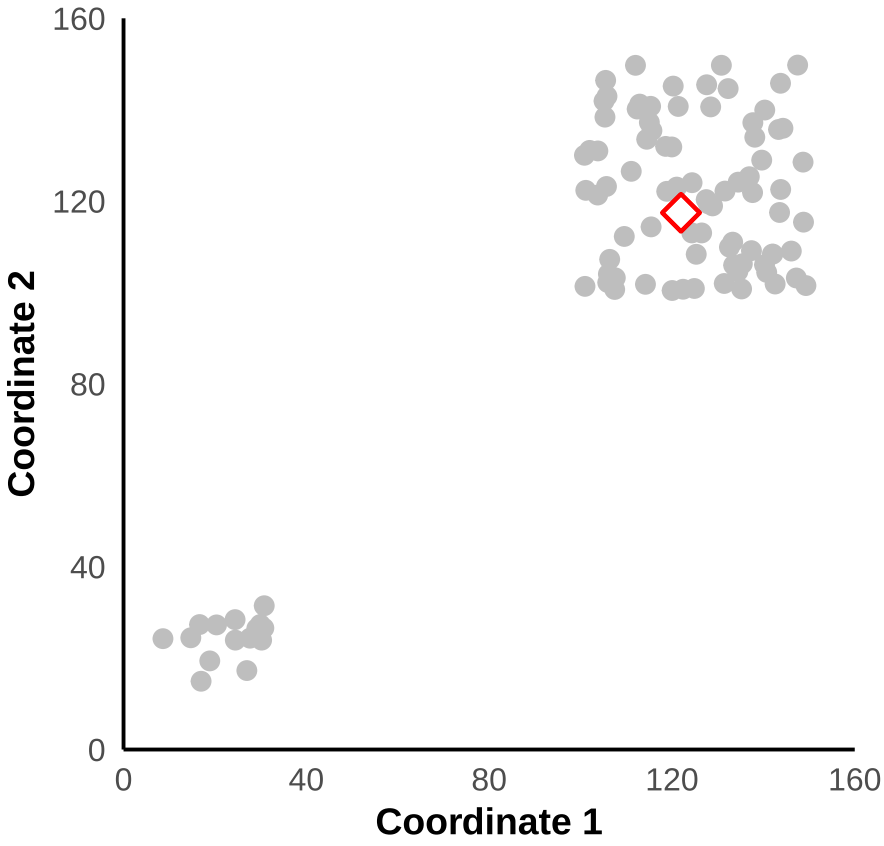
```
# Compute the Z80-Mean
trimmed_mean_result <- TRIMMEDmean(X, RAD_Median = FALSE, TRIM_percentage = 80)
```

```
## [1] "The points used are:"
##           c1       c2
## x17 118.6232 131.9705
## x20 115.4449 114.3620
## x21 132.5791 109.9220
## x22 131.6033 122.2004
## x24 114.2039 101.7947
## x25 133.2903 111.0194
## x26 109.5739 112.2678
## x37 111.0962 126.5194
## x45 124.3500 113.0356
## x59 126.5000 113.0381
## x60 128.8879 118.9524
## x61 118.8616 122.1387
## x65 125.3128 108.3656
## x70 127.9574 119.4055
## x73 119.9645 131.8515
## x84 124.4065 124.0291
## x89 127.5003 120.3308
## x92 121.0447 123.0688
## x95 106.3806 107.2539
## The Z80-Mean is:
```

```
print(trimmed_mean_result)
```

```
##       c1       c2
## 121.9779 117.4488
```

```
# Plot the points and Z80-Mean
suppressWarnings({
  p <- ggplot(plot_data, aes(x = c1, y = c2)) +
    geom_point(color = "grey", size = 3) +  # Original points
    geom_point(aes(x = trimmed_mean_result[1], y = trimmed_mean_result[2]),
               color = "red", shape = 5, size = 4, stroke = 1) +
    xlab("Coordinate 1") +
    ylab("Coordinate 2") +
    theme_minimal(base_size = 15) +
    theme(
      axis.title.x = element_text(size = 14, face = "bold"),  # Larger x-axis title
      axis.title.y = element_text(size = 14, face = "bold"),  # Larger y-axis title
      axis.text.x = element_text(size = 12),  # Larger x-axis text
      axis.text.y = element_text(size = 12),  # Larger y-axis text
      axis.line = element_line(colour = "black"),  # Add axis lines
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      panel.border = element_blank()
    ) +
    #geom_text(aes(label = Point), hjust = 1.5, vjust = 0, size = 4) +
    coord_fixed(ratio = 1) +  # Fixed aspect ratio
    scale_x_continuous(expand = c(0, 0),
                       limits = c(0, 160)) +  # Set x-axis to start from 0
    scale_y_continuous(expand = c(0, 0),
                       limits = c(0, 160))   # Set y-axis to start from 0
    print(p)
})
```


Coordinate 2
Coordinate 1