FA2_EDA

ABLIAN

2025-02-13

```
data <- read.csv("D:/FEU/3RD YR 2ND SEM/EDA/cytof_one_experiment.csv")

head(data)
```

Use pivot_longer to reshape the dataset into one that has two columns, the first giving the protein identity and the second giving the amount of the protein in one of the cells. The dataset you get should have 1750000 rows (50000 cells in the original dataset times 35 proteins)

```
##        NKp30    KIR3DL1      NKp44    KIR2DL1 GranzymeB       CXCR6      CD161
## 1  0.1875955  3.6156932 -0.5605694 -0.2936654  2.477893 -0.14470053 -0.3152872
## 2  1.0348518  1.7001820 -0.2889611 -0.4798280  3.261016 -0.03392447 -0.4112129
## 3  2.9996398  6.1411419  1.9032606  0.4823102  4.277562  1.94654156 -0.5022347
## 4  4.2998594 -0.2211586  0.2425707 -0.4831267  3.351808  0.92622195  3.8772370
## 5 -0.4386448 -0.5035892 -0.1526320  0.7506128  3.194145 -0.05893640  1.0907379
## 6  2.0883050 -0.3992646  3.4550676 -0.5200856  4.345102 -0.36434277 -0.5705891
##       KIR2DS4     NKp46      NKG2D      NKG2C       X2B4     CD69 KIR3DL1.S1
## 1  1.94497046 4.0818316  2.6200784 -0.3573817 -0.2711557 3.849965 -0.2554637
## 2  3.80251714 3.7339299 -0.4832788 -0.4675984 -0.5594752 2.910197 -0.2909482
## 3 -0.32010171 4.5594631 -0.5069090  2.6193782 -0.4554785 3.113454  3.6613886
## 4 -0.16969487 4.4831486  1.9272290 -0.3110146  1.6350771 3.045998  0.2871241
## 5 -0.05033025 0.8379358 -0.4581674  0.9216947  1.2419054 2.644422  0.4218294
## 6 -0.45033591 4.0550848  3.4283565  0.6272837 -0.4157104 3.958158  0.7993406
##          CD2    KIR2DL5    DNAM.1         CD4        CD8       CD57      TRAIL
## 1  5.3529769 -0.5092906 0.8811347 -0.32347280 -0.2822405  3.3254704 -0.6084228
## 2  4.3132510  3.7774776 1.5406568 -0.13208167  0.9161920  2.4946442 -0.5034739
## 3  5.5969513  0.8128166 1.0005903 -0.59933641  1.8382744  3.9897914 -0.2749380
## 4 -0.5002885  0.3612212 1.2663267 -0.12568567  0.7667204  1.9950916 -0.5130930
## 5 -0.5479527  1.0638327 0.8722272 -0.07107408 -0.1059012  3.4291302 -0.1433044
## 6  5.1028564  3.0918867 0.8717267 -0.47986180 -0.2577198 -0.5784575 -0.5731323
##       KIR3DL2      MIP1b     CD107a      GM.CSF       CD16        TNFa
## 1 -0.30668543  1.2497120 -0.1295305 -0.43074102  3.9951417  0.90143498
## 2 -0.54320954  2.8693060 -0.1887180 -0.16283845  4.4082309  1.93590153
## 3  2.06488239  4.0955112 -0.1998480  3.18853825  6.0023244 -0.02336999
## 4  2.11247859  3.3726018 -0.5720339  0.91310694  5.8238698 -0.60793749
## 5 -0.02505141 -0.3099826 -0.1068511 -0.60370379  4.0122501 -0.61989100
## 6 -0.28337673 -0.4108283 -0.1797545 -0.06372458 -0.5832926  0.14311030
##           ILT2 Perforin KIR2DL2.L3.S2     KIR2DL3      NKG2A    NTB.A     CD56
## 1 -0.386027758 6.431983    1.22710292 2.660657999 -0.5220613 4.348923 2.897523
## 2  2.983874845 6.814827   -0.04141081 3.841304627  4.6771149 3.474335 3.782870
## 3 -0.521099944 5.099562   -0.16705075 -0.009694396 -0.4730573 5.634341 5.701186
## 4 -0.043783559 5.841797   -0.51753289 -0.592990887 -0.4059049 4.598021 6.065672
## 5  1.182703288 4.888777   -0.36251589 -0.398123704 -0.5440881 3.606101 1.966169
## 6 -0.003258955 3.952542   -0.20194392 -0.202592720  3.8882776 2.346275 6.473243
##         INFg
## 1 -0.3841108
## 2  2.7186296
## 3  2.5321763
## 4  2.4564582
## 5  3.1470092
## 6  2.8282987
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
set.seed(123)

data <- data.frame(
  cell_id = rep(1:50000, each = 1),
  protein_1 = runif(50000, min = 0, max = 100),
  protein_2 = runif(50000, min = 0, max = 100),
  protein_3 = runif(50000, min = 0, max = 100),
  protein_4 = runif(50000, min = 0, max = 100),
  protein_5 = runif(50000, min = 0, max = 100),
  protein_6 = runif(50000, min = 0, max = 100),
  protein_7 = runif(50000, min = 0, max = 100),
  protein_8 = runif(50000, min = 0, max = 100),
  protein_9 = runif(50000, min = 0, max = 100),
  protein_10 = runif(50000, min = 0, max = 100),
  protein_11 = runif(50000, min = 0, max = 100),
  protein_12 = runif(50000, min = 0, max = 100),
  protein_13 = runif(50000, min = 0, max = 100),
  protein_14 = runif(50000, min = 0, max = 100),
  protein_15 = runif(50000, min = 0, max = 100),
  protein_16 = runif(50000, min = 0, max = 100),
  protein_17 = runif(50000, min = 0, max = 100),
  protein_18 = runif(50000, min = 0, max = 100),
  protein_19 = runif(50000, min = 0, max = 100),
  protein_20 = runif(50000, min = 0, max = 100),
  protein_21 = runif(50000, min = 0, max = 100),
  protein_22 = runif(50000, min = 0, max = 100),
  protein_23 = runif(50000, min = 0, max = 100),
  protein_24 = runif(50000, min = 0, max = 100),
  protein_25 = runif(50000, min = 0, max = 100),
  protein_26 = runif(50000, min = 0, max = 100),
  protein_27 = runif(50000, min = 0, max = 100),
  protein_28 = runif(50000, min = 0, max = 100),
  protein_29 = runif(50000, min = 0, max = 100),
  protein_30 = runif(50000, min = 0, max = 100),
  protein_31 = runif(50000, min = 0, max = 100),
  protein_32 = runif(50000, min = 0, max = 100),
  protein_33 = runif(50000, min = 0, max = 100),
  protein_34 = runif(50000, min = 0, max = 100),
  protein_35 = runif(50000, min = 0, max = 100)
)

reshaped_data <- data %>%
  pivot_longer(cols = starts_with("protein_"),
               names_to = "protein",
               values_to = "protein_amount")

head(reshaped_data)
```

```
## # A tibble: 6 x 3
##   cell_id protein   protein_amount
##     <int> <chr>              <dbl>
## 1       1 protein_1           28.8
## 2       1 protein_2           21.3
## 3       1 protein_3           60.2
## 4       1 protein_4           53.6
## 5       1 protein_5           60.4
## 6       1 protein_6           16.7
```

```
summary_stats <- reshaped_data %>%
  group_by(protein) %>%
  summarise(
    median_level = median(protein_amount, na.rm = TRUE),
    mad_level = mad(protein_amount, na.rm = TRUE)
  )

head(summary_stats)
```

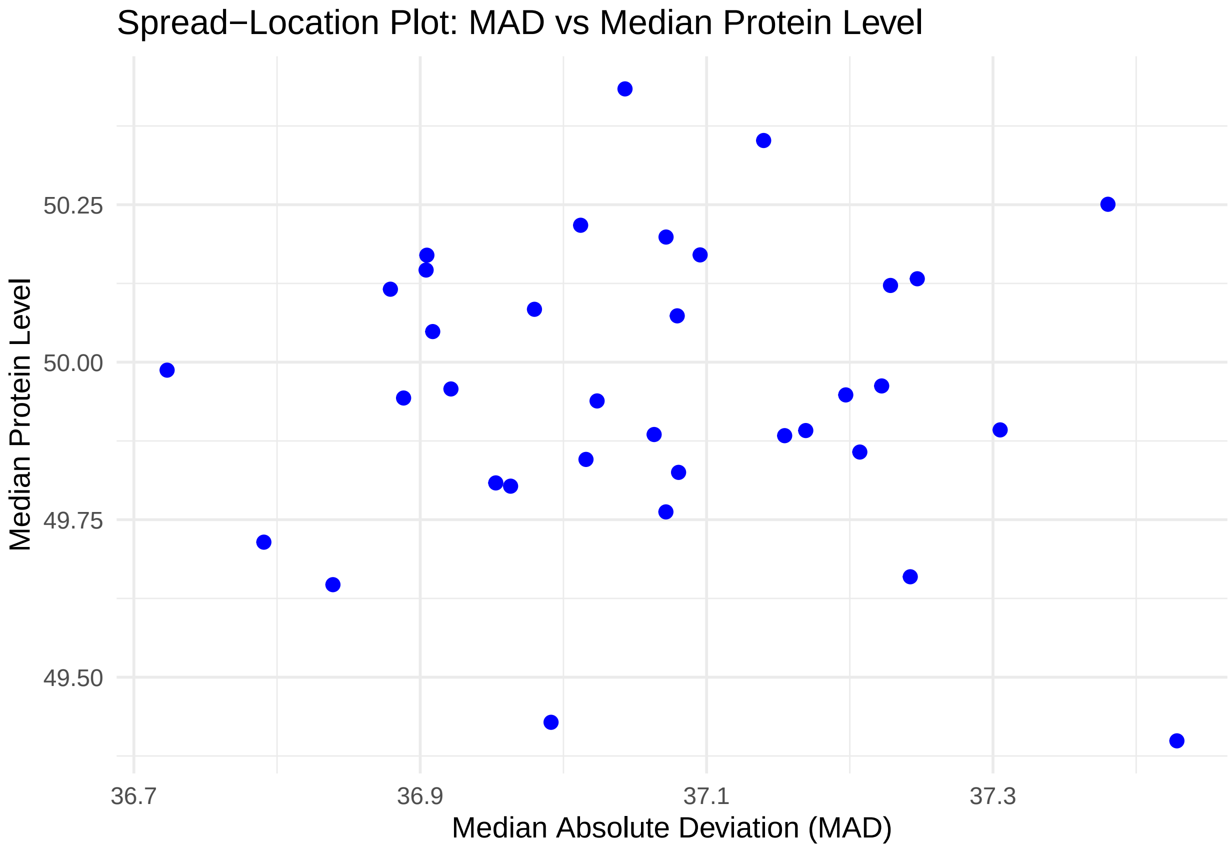
Use group_by and summarise to find the median protein level and the median absolute deviation of the protein level for each marker. (Use the R functions median and mad).

```
## # A tibble: 6 x 3
##   protein    median_level mad_level
##   <chr>             <dbl>     <dbl>
## 1 protein_1          49.4      37.0
## 2 protein_10         49.8      37.0
## 3 protein_11         49.6      36.8
## 4 protein_12         50.0      37.2
## 5 protein_13         49.9      37.2
## 6 protein_14         49.8      37.1
```

```r
library(ggplot2)

ggplot(summary_stats, aes(x = mad_level, y = median_level)) +
  geom_point(color = "blue", size = 2) +
  labs(
    x = "Median Absolute Deviation (MAD)",
    y = "Median Protein Level",
    title = "Spread-Location Plot: MAD vs Median Protein Level"
  ) +
  theme_minimal()
```

Make a plot with mad on the x-axis and median on the y-axis. This is known as a spreadlocation (s-l) plot. What does it tell you about the relationship betwen the median and the mad?



```r
library(dcldata)
library(tidyr)
library(dplyr)
reshaped_data <- example_gymnastics_2 %>%
  pivot_longer(cols = starts_with("vault") | starts_with("floor"),
               names_to = "event_year",
               values_to = "score") %>%
  separate(col = event_year,
           into = c("event", "year"),
           sep = "_")

head(reshaped_data)
```

The MAD values range between roughly 36.7 and 37.3, while the median protein level is around 49.5 to 50.3 suggesting that the variation of median protein levels are higher when compared to the variation in MAD.

```
## # A tibble: 6 x 4
##   country       event year  score
##   <chr>         <chr> <chr> <dbl>
## 1 United States vault 2012   48.1
## 2 United States vault 2016   46.9
## 3 United States floor 2012   45.4
## 4 United States floor 2016   46.0
## 5 Russia        vault 2012   46.4
## 6 Russia        vault 2016   45.7
```

```
str(reshaped_data)
```

```
## tibble [12 x 4] (S3: tbl_df/tbl/data.frame)
##  $ country: chr [1:12] "United States" "United States" "United States" "United States" ...
##  $ event  : chr [1:12] "vault" "vault" "floor" "floor" ...
##  $ year   : chr [1:12] "2012" "2016" "2012" "2016" ...
##  $ score  : num [1:12] 48.1 46.9 45.4 46 46.4 ...
```