Exploratory Data Analysis

Formative Assessment 2

February 14, 2025

By Romand Lansangan

In [33]:
```
library(tidyverse)
library(ggplot2)
library(tidyr)
options(scipen=0)
library(readr)
library(ggthemes)
library(lspline)
library(knitr)
library(splines)
library(formattable)
library(IRdisplay)
```

(I) CyTof data

In this chapter we followed the first set of instructions:

(a) 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).

In [32]:
```
df <- as_tibble(read.csv("cytof_one_experiment.csv"))
```

In [27]:
```
formatted_table <- df %>%
  head(7) %>%
  formattable()

display_html(paste0("<div style='display: flex; justify-content: center;'>", as.characte
```

.1.S1	CD2	KIR2DL5	DNAM.1	CD4	CD8	CD57	TRAIL	KIR3DL2
4637	5.3529769	-0.5092906	0.8811347	-0.32347280	-0.2822405	3.3254704	-0.6084228	-0.30668543
9482	4.3132510	3.7774776	1.5406568	-0.13208167	0.9161920	2.4946442	-0.5034739	-0.54320954
3886	5.5969513	0.8128166	1.0005903	-0.59933641	1.8382744	3.9897914	-0.2749380	2.06488239
1241	-0.5002885	0.3612212	1.2663267	-0.12568567	0.7667204	1.9950916	-0.5130930	2.11247859
8294	-0.5479527	1.0638327	0.8722272	-0.07107408	-0.1059012	3.4291302	-0.1433044	-0.02505141
3406	5.1028564	3.0918867	0.8717267	-0.47986180	-0.2577198	-0.5784575	-0.5731323	-0.28337673
2852	-0.5989730	-0.2517884	0.9207401	1.17093612	-0.6024213	2.5377810	1.7714566	-0.56939916

Table 1: Table representing original CyTof dataset (50,000 x 35)

In [30]:
```
df_longer <- pivot_longer(df, colnames(df), names_to = "type", values_to = "amount")
```

In [34]:
```
formatted_table <- df_longer %>%
  head(7) %>%
  formattable()

display_html(paste0("<div style='display: flex; justify-content: center;'>", as.characte
```

type	amount
NKp30	0.1875955
KIR3DL1	3.6156932
NKp44	-0.5605694
KIR2DL1	-0.2936654
GranzymeB	2.4778929
CXCR6	-0.1447005
CD161	-0.3152872

Table 2: Table representing CyTof dataset in pivot longer form (1,750,000 x 2)

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

In [20]:
```
med_mad <- df_longer %>%
    group_by(type) %>%
    summarise(median = median(amount), mad = mad(amount))
```

In [21]:
```
formatted_table <- med_mad %>%
  head(7) %>%
  formattable()

display_html(paste0("<div style='display: flex; justify-content: center;'>", as.characte
```

type	median	mad
CD107a	-0.1222997	0.6086976
CD16	5.1229802	0.8744054
CD161	0.7256933	1.6882296
CD2	3.9453789	1.6770427
CD4	-0.2036499	0.3953896
CD56	5.7107903	0.9981358
CD57	3.0709204	1.9868804

Table 3: Table of *median* and *mean absolute deviation* for each type of marker in CyTof dataset (35 x 3)

(c) 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?

In [48]:
```
options(repr.plot.width = 16, repr.plot.height =8)

med_mad %>%
ggplot(aes(x=median, y=mad)) +
geom_point(size=3) +
ggtitle("Spreadlocation (S-L) Plot of CyTOF") +
xlab("Median") +
ylab("Mean Absolute Deviation") +
theme_economist() +
theme(
    plot.title = element_text(hjust = 0.5, size=20),
    axis.title = element_text(size = 16),
    axis.text= element_text(size = 16),
    axis.title.x = element_text(margin = margin(t = 10)),
    axis.title.y = element_text(margin = margin(r = 10)))
```

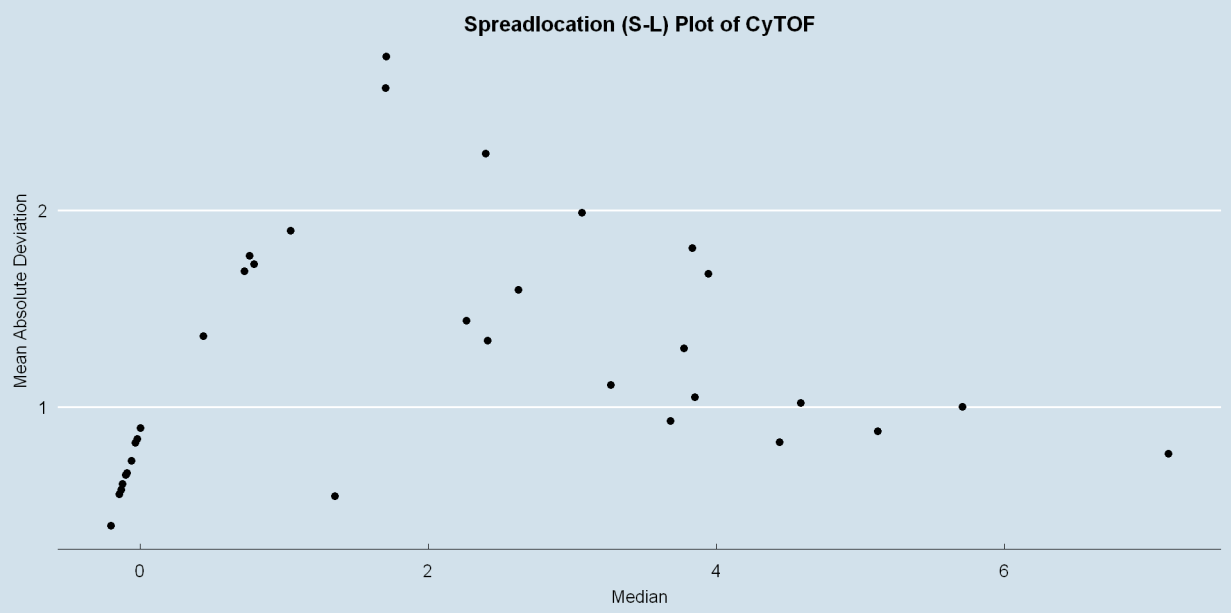

Figure 1: Scatter plot of *median* vs *mean absolute deviation* from Table 2

In [24]:
```
knot <- 1.7
model <- lm(mad ~ lspline(median, knot), data = med_mad)

coefs <- coef(model)
slope1 <- coefs[2]
slope2 <- coefs[3]
intercept <- coefs[1]

y_knot <- intercept + slope1 * knot

x_min <- min(med_mad$median)
```

```
x_max <- max(med_mad$median)

y_start <- intercept + slope1 * x_min
y_end <- y_knot + slope2 * (x_max - knot)

med_mad %>%
ggplot(aes(x=median, y=mad)) +
geom_point(size=3) +
geom_segment(aes(x = x_min, y = y_start, xend = knot, yend = y_knot),
               color = "red", linewidth = 1) +
geom_segment(aes(x = knot, y = y_knot, xend = x_max, yend = y_end),
               color = "red", linewidth = 1) +
ggtitle("Spreadlocation (S-L) Plot of CyTOF") +
xlab("Median") +
ylab("Mean Absolute Deviation") +
theme_economist() +
theme(
    plot.title = element_text(hjust = 0.5, size=20),
    axis.title = element_text(size = 16),
    axis.text= element_text(size = 16),
    axis.title.x = element_text(margin = margin(t = 10)),
    axis.title.y = element_text(margin = margin(r = 10)))
```

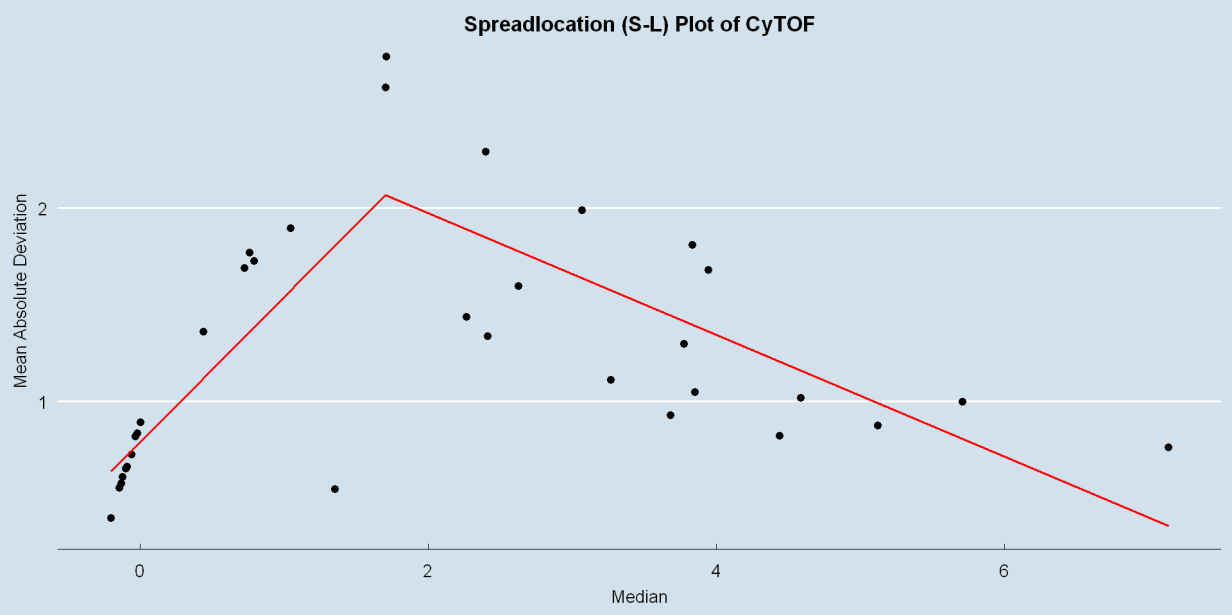

Figure 2: Scatter plot of *median* vs *mean absolute deviation* from Table 2, with a piecewise linear regression model fitted at a knot of *median = 1.7*

The scatter plot from `Figure 1` shows little to no correlation of the *median* and *mean absolute deviation* for each of marker in *CyTof* dataset. However a dividing the data into two, at *median = 1.7* and fitting a simple linear regression model showed a **positive relationship** between the two variables $\forall x \in \{x : x_{median} < 1.7\}$ and **negative relationship** for the rest of the data points.

Of course, we could calculate this numerically by calculating for correlation of (1) overall data, (2) first knot at median below 1.9, and (3) second knot a complement of the first knot.

In [36]:
```
cor_ovr <- cor(med_mad$median, med_mad$mad)
knot_one <- filter(med_mad,median < knot)
knot_two <- filter(med_mad,median >= knot)
cor_one <- cor(knot_one$median, knot_one$mad)
cor_two <- cor(knot_two$median, knot_two$mad)

res <- tibble(ovr_cor = cor_ovr, knot_1_cor = cor_one, knot_2_cor = cor_two)
```

In [37]:
```
formatted_table <- res %>%
  formattable()

display_html(paste0("<div style='display: flex; justify-content: center;'>", as.characte
```

ovr_cor	knot_1_cor	knot_2_cor
0.1542416	0.6697454	-0.7454767

Table 4: Correlational Table of *median* and *mean absolute deviation* for (i) overall, (2) $\forall x \in \{x : x_{median} < 1.7\}$, (3) $\forall x \notin \{x : x_{median} < 1.7\}$ (1 x 3)

(II) Example_Gymnastics_2 from dcldata

Using either pivot_longer on its own or pivot_longer in combination with separate, reshape the dataset so that it has columns for country, event, year, and score.

In [38]:
```
load("example_gymnastics_2.rda")
```

In [39]:
```
formatted_table <- example_gymnastics_2 %>%
  formattable()

display_html(paste0("<div style='display: flex; justify-content: center;'>", as.characte
```

country	vault_2012	floor_2012	vault_2016	floor_2016
United States	48.132	45.366	46.866	45.999
Russia	46.366	41.599	45.733	42.032
China	44.266	40.833	44.332	42.066

Table 5: Original *example_gymnastic_2* data from dcldata that shows the score of each country on vault & floor event during 2012 & 2016 Olympics (3 x 5)

In [40]:
```
gymnastics_longer <- example_gymnastics_2 %>%
pivot_longer(names(example_gymnastics_2[-1]), names_sep = "_", names_to = c("event", "ye
```

In [41]:
```
formatted_table <- gymnastics_longer %>%
  formattable()
```

```
display_html(paste0("<div style='display: flex; justify-content: center;'>", as.characte
```

country	event	year	score
United States	vault	2012	48.132
United States	floor	2012	45.366
United States	vault	2016	46.866
United States	floor	2016	45.999
Russia	vault	2012	46.366
Russia	floor	2012	41.599
Russia	vault	2016	45.733
Russia	floor	2016	42.032
China	vault	2012	44.266
China	floor	2012	40.833
China	vault	2016	44.332
China	floor	2016	42.066

Table 5: Pivot longer of *Table 5* with event and year separated (12 x 4)

In [54]:
```
ggplot(data = gymnastics_longer, aes(x = interaction(event, year), y = score, fill = cou
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    x = "Year and Event",
    y = "Score",
    title = "Score Distribution by Year, Event, and Country (2015 & 2016 Olympics)"
  ) +
  theme_economist() +
  theme(
    axis.text = element_text(size=14),
    axis.title = element_text(size=16),
    legend.text = element_text(size = 14),
    legend.title = element_text(size = 16))
```

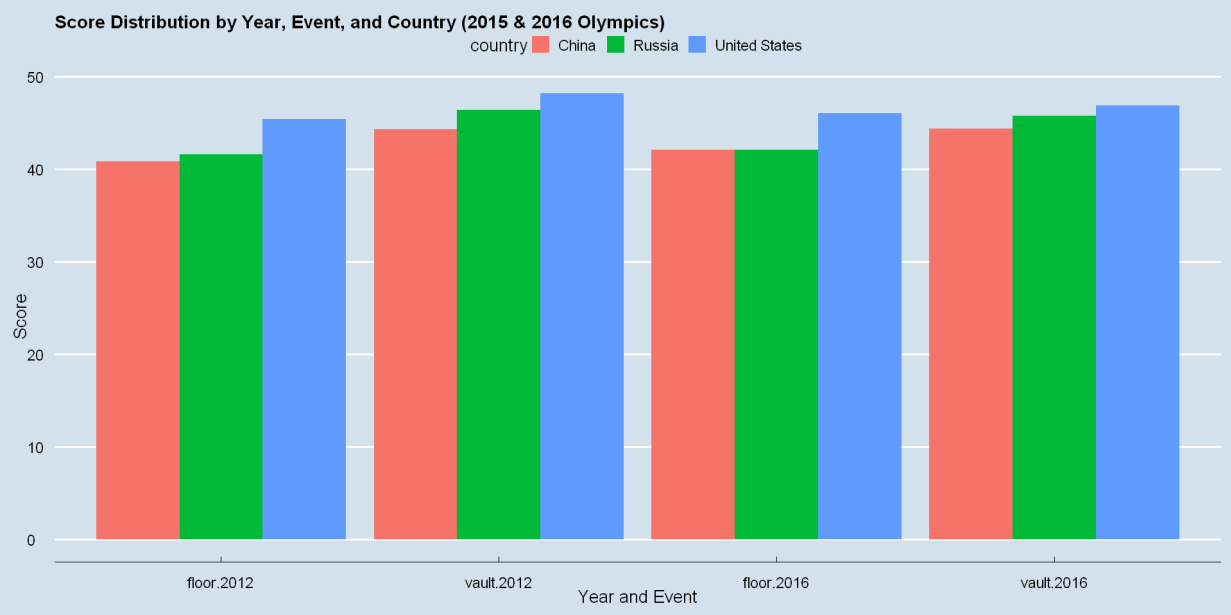

Figure 3: Bar chart of score distribution by year, event, and country in the 2015 & 2016 Olympics

`Figure 3` shows that *United States of America* has won both the *floor* and *vault* event during the 2015 and 2016 Olympics, followed by *Russia* and with *China* being the last place among the three through out. Although the ranking remains constant, the scores of three countries makes it clear that the two *vault* events are more competitive than *floor* events.