

1\_2023900021

hrithik.rachakonda@research.iiit.ac.in

2026-02-02

## 1. Answer:

- The misleading visualization method is the bar plot of means. Although all four groups have similar mean values, their underlying distributions differ substantially in shape, spread, and modality, particularly the bimodal nature of x2.
- The bar plot collapses each group into a single summary statistic and hides these differences, creating a false impression of similarity across groups.
- Distribution-revealing visualizations such as histograms, density plots, jittered boxplots, and raincloud plots show the true structure of the data and therefore provide a more accurate representation.

```
# install.packages(c("readxl", "tidyr", "ggplot2", "dplyr", "ggridges", "ggdist", "waffle", "fmsb", "reshape2"))
```

```
library(readxl)
library(tidyr)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##   filter, lag
```

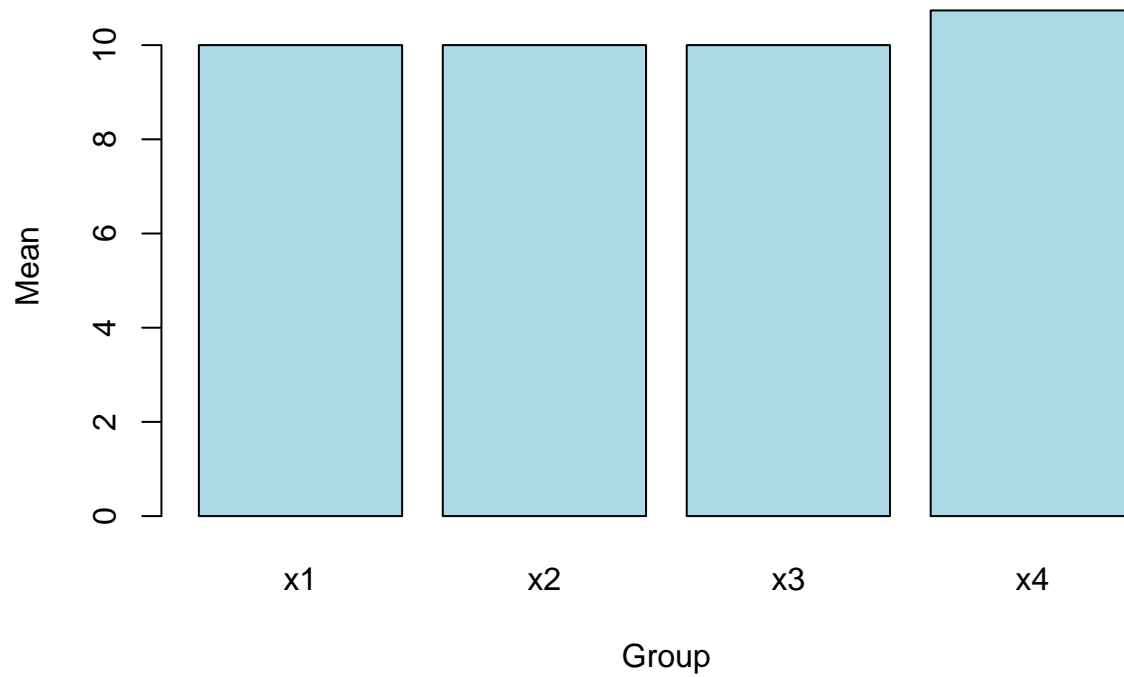
```
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
xlsx_path <- "BRSM_Visualisation_Assignment.xlsx"
data <- as.data.frame(read_excel(xlsx_path, sheet = "Statistical Deception"))
data <- data[, sapply(data, is.numeric), drop = FALSE]

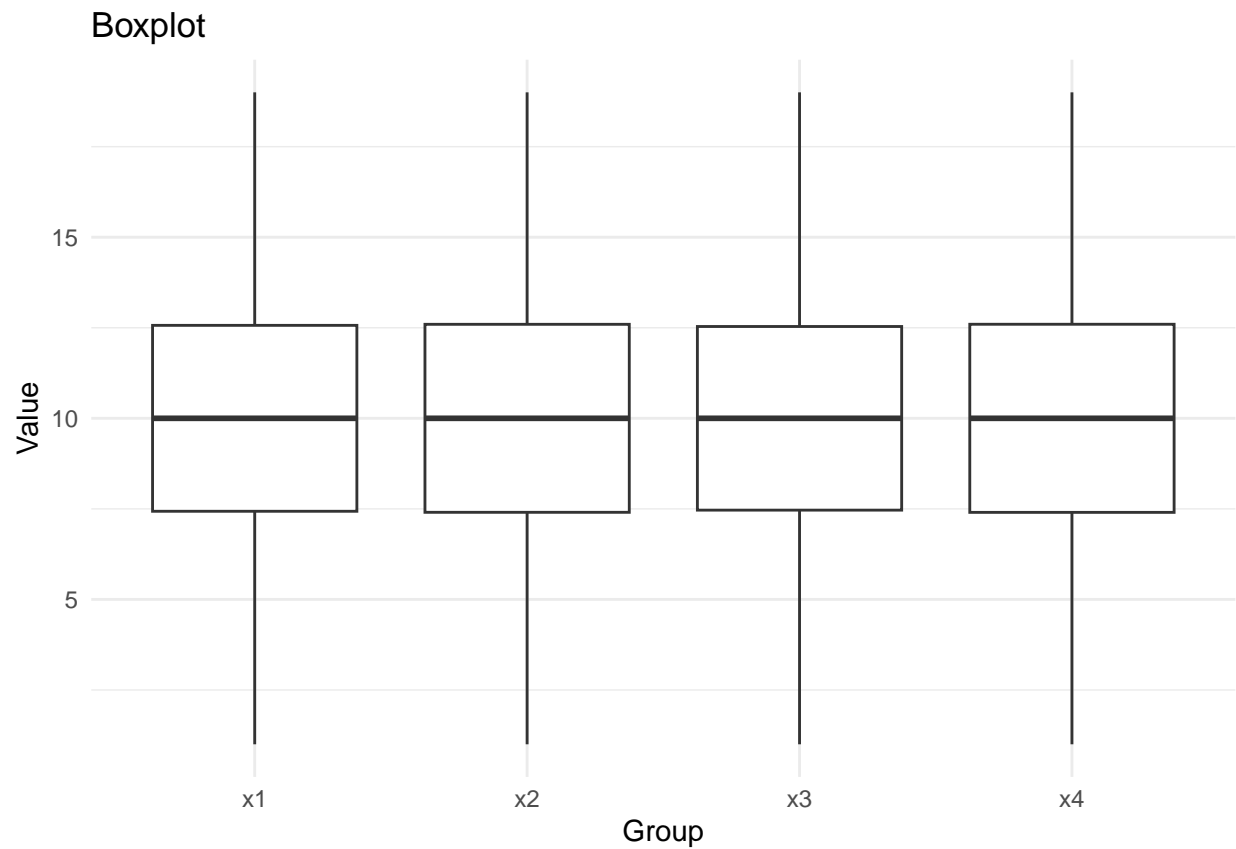
data_long <- pivot_longer(data, cols = everything(), names_to = "Group", values_to = "Value")

# 1) Bar plot of means
means <- colMeans(data, na.rm = TRUE)
barplot(means, col="lightblue", main="Bar Plot of Means (Misleading)", ylab="Mean", xlab="Group")
```

**Bar Plot of Means (Misleading)**

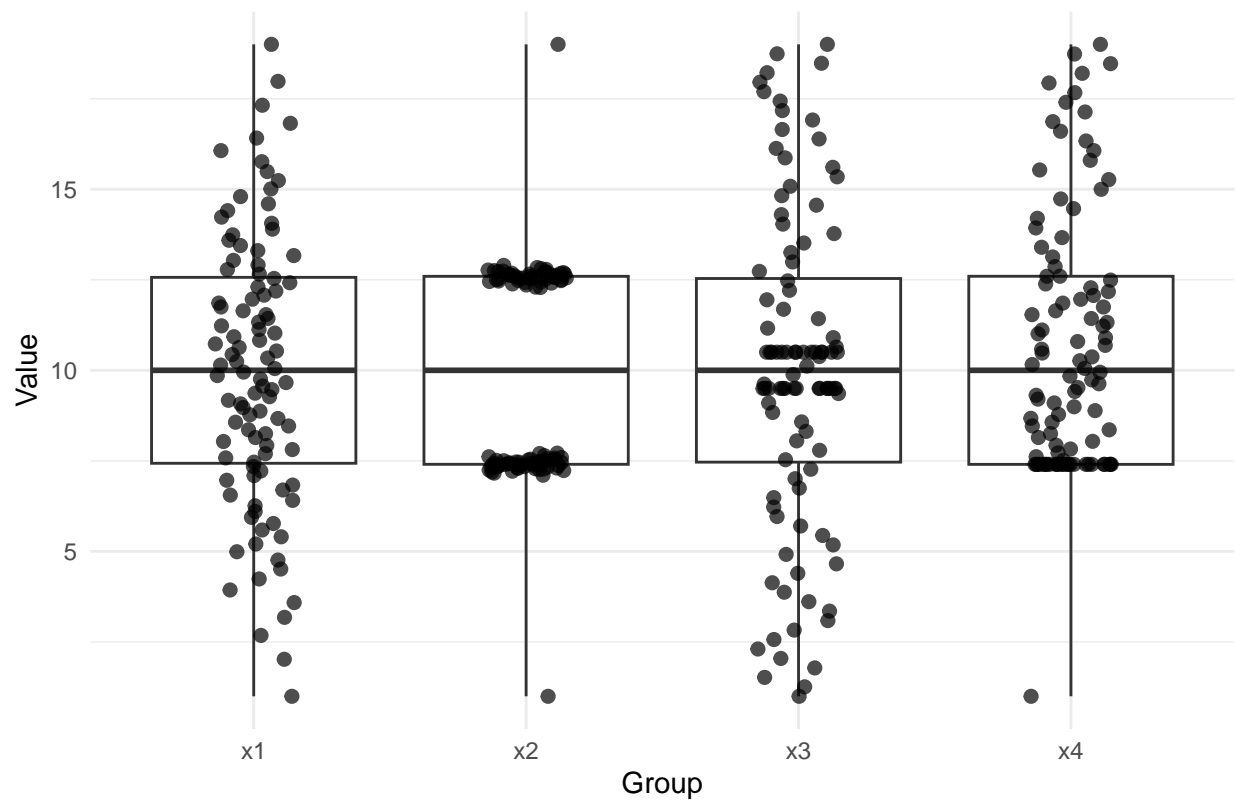


```
# 2) Boxplot  
ggplot(data_long, aes(Group, Value)) + geom_boxplot() +  
  labs(title="Boxplot", x="Group", y="Value") + theme_minimal()
```



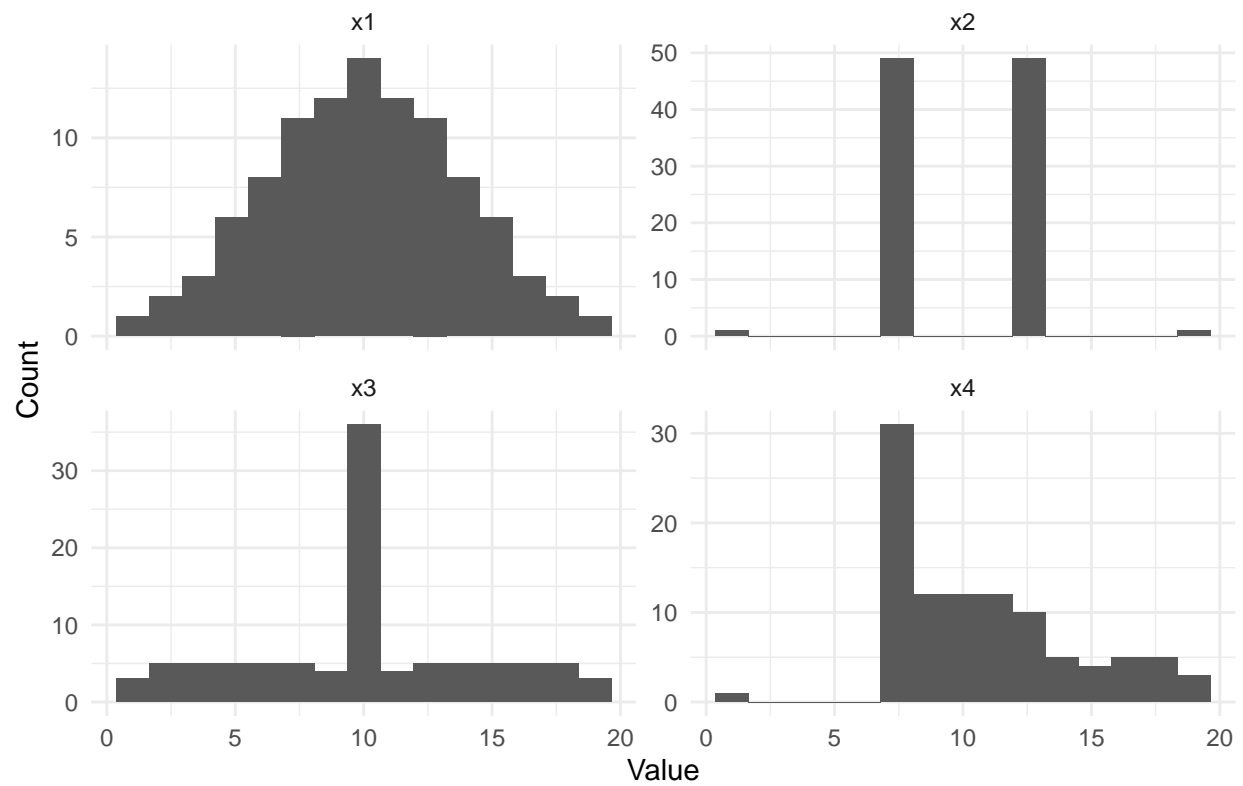
```
# 3) Jitter + boxplot
ggplot(data_long, aes(Group, Value)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(width=0.15, alpha=0.7, size=2) +
  labs(title="Jitter + Boxplot", x="Group", y="Value") + theme_minimal()
```

### Jitter + Boxplot

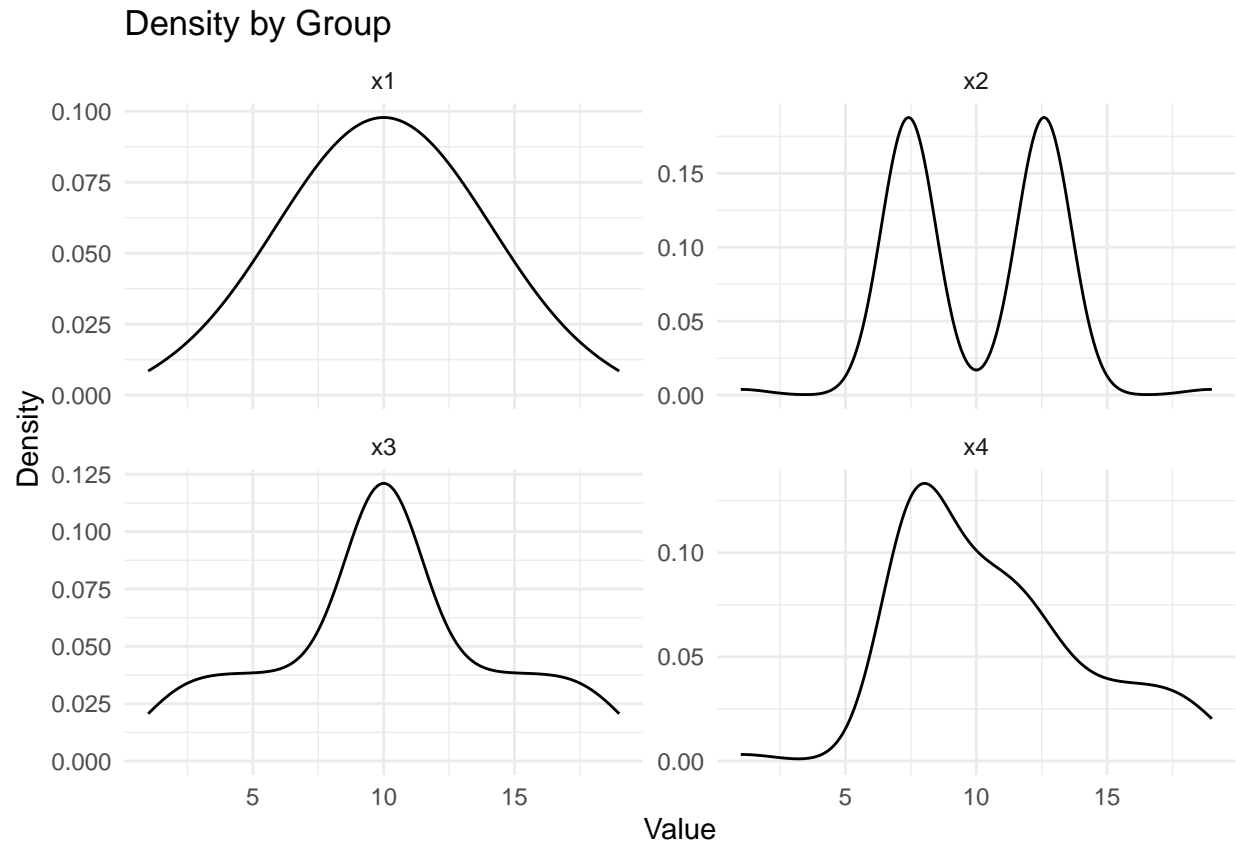


```
# 4) Histogram (facets)
ggplot(data_long, aes(Value)) +
  geom_histogram(bins=15) +
  facet_wrap(~Group, scales="free_y") +
  labs(title="Histograms by Group", x="Value", y="Count") + theme_minimal()
```

## Histograms by Group

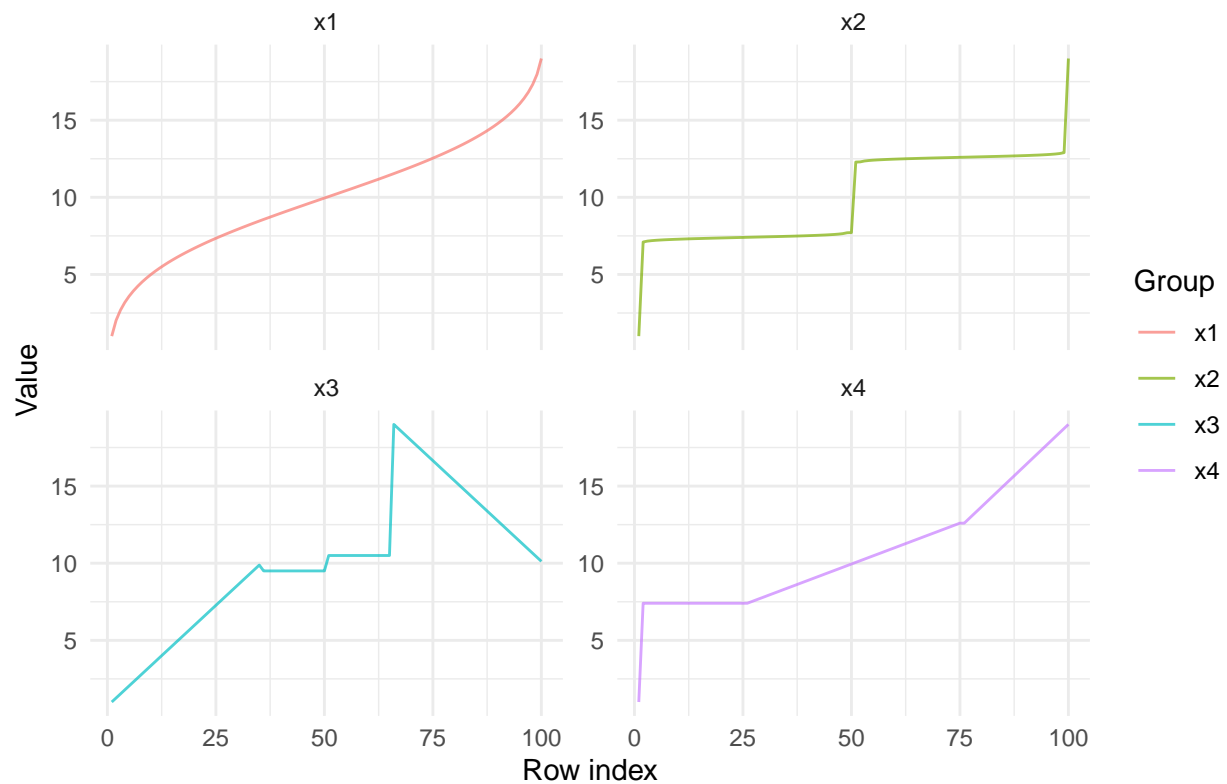


```
# 5) Density (facets)
ggplot(data_long, aes(Value)) +
  geom_density() +
  facet_wrap(~Group, scales="free_y") +
  labs(title="Density by Group", x="Value", y="Density") + theme_minimal()
```



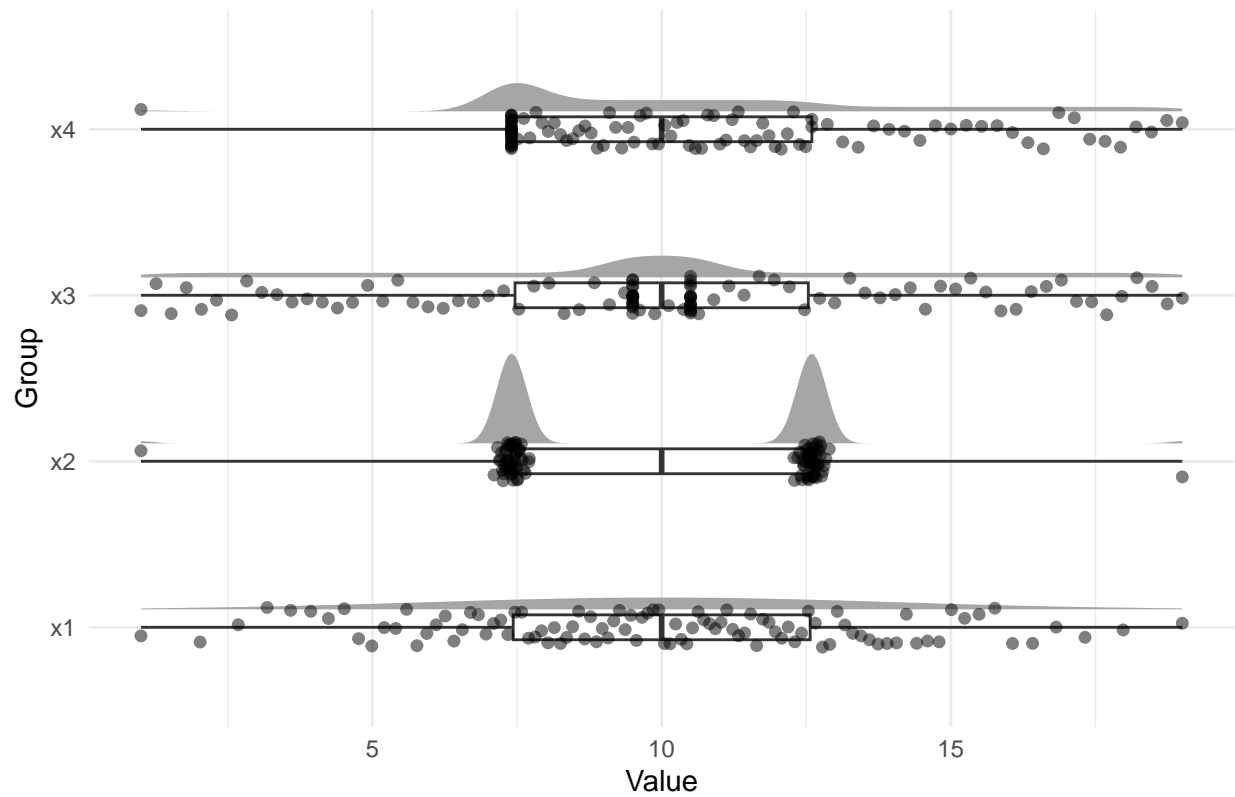
```
# 6) Line plot (index vs value)
data_long <- data_long %>% group_by(Group) %>% mutate(Index = row_number()) %>% ungroup()
ggplot(data_long, aes(Index, Value, color=Group)) +
  geom_line(alpha=0.7) +
  facet_wrap(~Group, scales="free_y") +
  labs(title="Line Plot (Index vs Value)", x="Row index", y="Value") + theme_minimal()
```

## Line Plot (Index vs Value)



```
# 7) Raincloud / raindrop (needs ggdist)
if (!requireNamespace("ggdist", quietly = TRUE)) {
  message("Skipping Raincloud: install.packages('ggdist')")
} else {
  library(ggdist)
  ggplot(data_long, aes(x=Group, y=Value)) +
    ggdist::stat_halfeye(adjust=0.6, width=0.6, justification=-0.2, .width=0) +
    geom_boxplot(width=0.15, outlier.shape=NA) +
    geom_jitter(width=0.12, alpha=0.5, size=1.6) +
    coord_flip() +
    labs(title="Raincloud (Half-eye + Box + Points)", x="Group", y="Value") +
    theme_minimal()
}
```

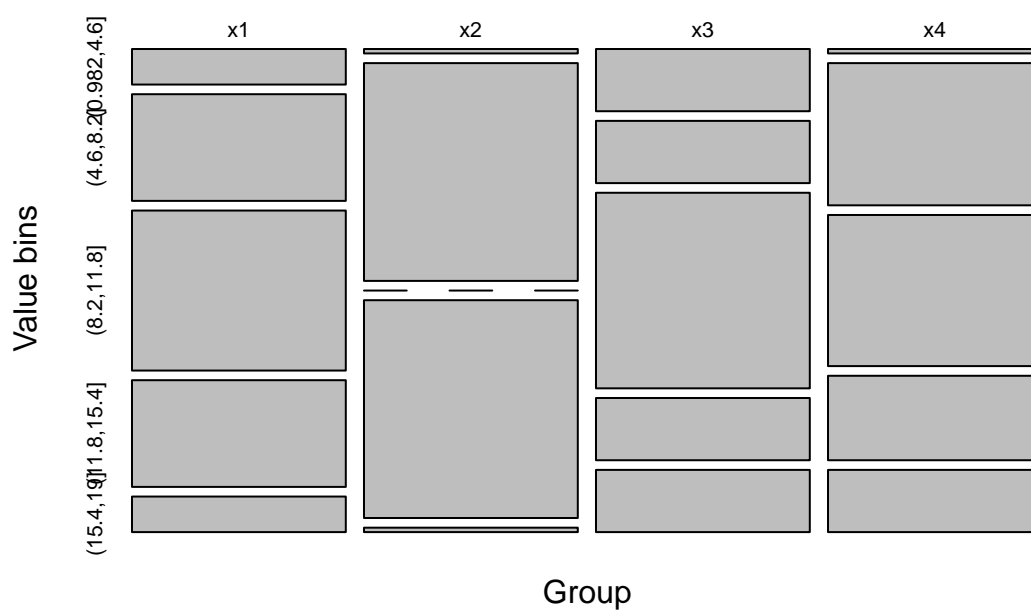
### Raincloud (Half-eye + Box + Points)



```
# 8) Mosaic plot (needs categorical data; bin values)
data_long$Bin <- cut(data_long$Value, breaks=5)
tab <- table(data_long$Group, data_long$Bin)
mosaicplot(tab, main="Mosaic Plot (Group x Binned Value)", xlab="Group", ylab="Value bins")
```



## Mosaic Plot (Group x Binned Value)

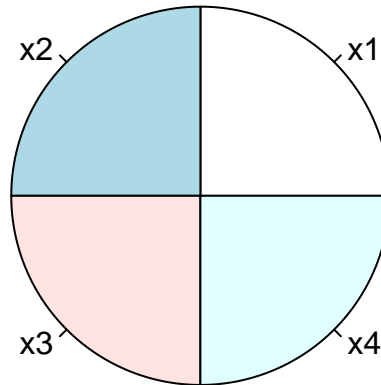


```
# 9) Waffle plot (counts per group; needs waffle)
counts <- table(data_long$Group)
if (!requireNamespace("waffle", quietly = TRUE)) {
  message("Skipping Waffle: install.packages('waffle')")
} else {
  waffle::waffle(as.list(counts), rows=5, title="Waffle Plot (Counts per Group)")
}
```

```
## Skipping Waffle: install.packages('waffle')
```

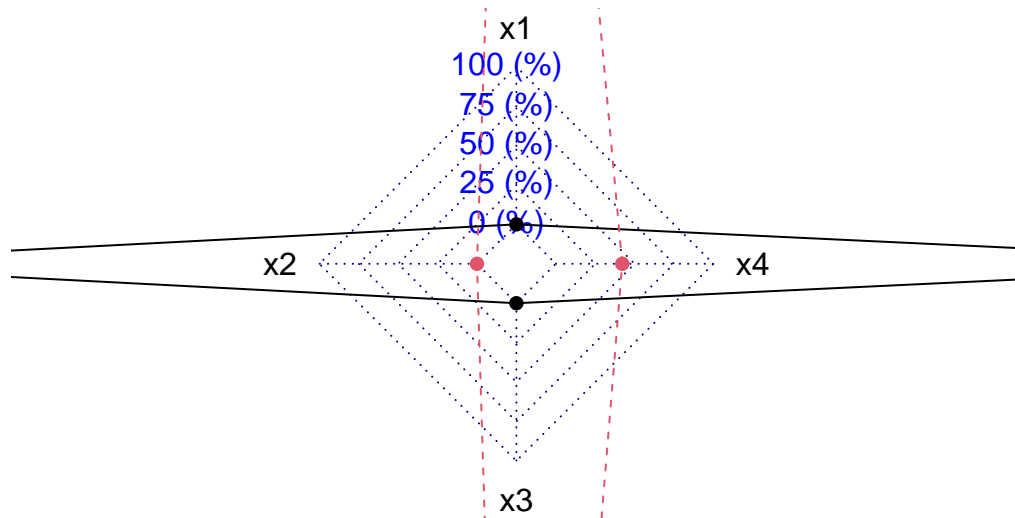
```
# 10) Pie chart (counts per group)
pie(counts, main="Pie Chart (Counts per Group)")
```

## Pie Chart (Counts per Group)



```
# 11) Radar chart (mean + sd per group; needs fmsb)
stats <- data_long %>% group_by(Group) %>% summarise(mean=mean(Value), sd=sd(Value), .groups="drop")
rad <- as.data.frame(t(stats[,c("mean","sd")]))
colnames(rad) <- stats$Group
rad <- rbind(apply(rad,1,max), apply(rad,1,min), rad)
if (!requireNamespace("fmsb", quietly = TRUE)) {
  message("Skipping Radar: install.packages('fmsb')")
} else {
  library(fmsb)
  fmsb::radarchart(rad, axistype=1, title="Radar Chart (Mean & SD)")
}
```

## Radar Chart (Mean & SD)



```
# 12) Heatmap (correlation across x1-x4)
cor_mat <- cor(data, use="complete.obs")
if (!requireNamespace("reshape2", quietly = TRUE)) {
  message("Skipping ggplot heatmap: install.packages('reshape2')")
  heatmap(cor_mat, main="Heatmap: Correlation (base R)")
} else {
  library(reshape2)
  cor_long <- reshape2::melt(cor_mat)
  ggplot(cor_long, aes(Var1, Var2, fill=value)) +
    geom_tile() +
    labs(title="Heatmap: Correlation", x="", y="", fill="corr") +
    theme_minimal()
}
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
## smiths
```

Heatmap: Correlation

