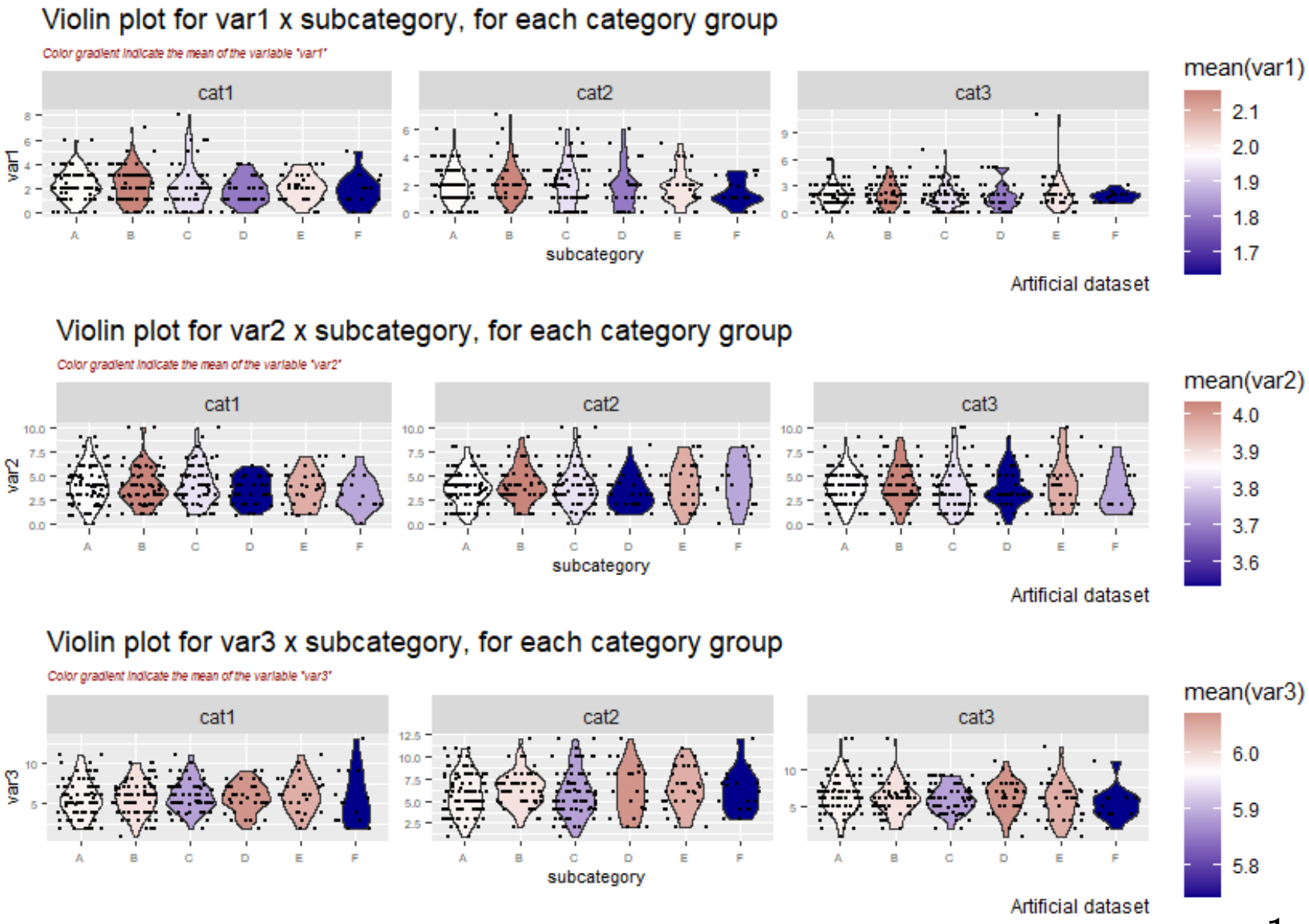
Objectif: multiple violin plots



Creating fake numerical and categorical data including missing values

```r
# load libraries and create artificial dataset
library(caret)
library(missForest)
library(tidyverse)
library(RANN)
library(gridExtra)

set.seed(2023) # for reproducibility
var1 <- rpois(900,2)
var2 <- rpois(900,4)
var3 <- rpois(900,6)
category <- c(rep('cat1', 300), rep('cat2', 300), rep('cat3', 300))
subcategory <- sample(LETTERS[1:6], size = 900, replace = TRUE,
                      prob = c(0.25, 0.3, 0.2, 0.1, 0.1, 0.05))
dataset <- data.frame(var1, var2, var3, subcategory, category)

# prodNA produce 5% missing data
dataset.mis = data.frame(prodNA(dataset[,1:3], noNA = 0.05), subcategory,
     category)

# save a copy of the dataset in .csv
write.csv(dataset.mis, "path/dataset.mis.csv", row.names = FALSE)
dataset.mis = read.csv("path/dataset.mis.csv", header = TRUE)

dataset.mis[298:301, ] # excerpt of the dataset contains NA values
#     var1 var2 var3 subcategory category
# 298    0   10    5           C     cat1
# 299   NA    1    5           E     cat1
# 300    2    3    5           A     cat1
# 301    2   NA    8           B     cat2
```

Missing value imputation using 'knn'

```r
# knn imputation using caret and 5 'neighbours'
set.seed(2023)
dataset.mis.model = preProcess(dataset.mis %>%
                               dplyr::select(names(dataset.mis)),
                               "knnImpute", k = 5, knnSummary = mean)
dataset.mis.model
dataset.mis.pred = predict(dataset.mis.model, dataset.mis) # variables are
    normalized
dataset.mis.pred[298:301, ]
#            var1       var2       var3 subcategory category
# 298 -1.37892580  3.0235674 -0.4137490           C     cat1
# 299  0.70767513 -1.4093204 -0.4137490           E     cat1
# 300  0.01214149 -0.4242342 -0.4137490           A     cat1
# 301  0.01214149 -0.1287083  0.8754844           B     cat2

# values in original scale
complete.dataset <- data.frame(col = names(dataset.mis[,1:3]),
                               mean = dataset.mis.model$mean,
                               sd = dataset.mis.model$std)
for(i in complete.dataset$col){
  dataset.mis.pred[i] <- dataset.mis.pred[i]*dataset.mis.model$std[i] +
                         dataset.mis.model$mean[i] }

# now the dataset is complete
complete.data..dataset <- dataset.mis.pred
complete.data.dataset[298:301, ]
#     var1 var2 var3 subcategory category
# 298    0 10.0    5           C     cat1
# 299    3  1.0    5           E     cat1
# 300    2  3.0    5           A     cat1
# 301    2  3.6    8           B     cat2
```

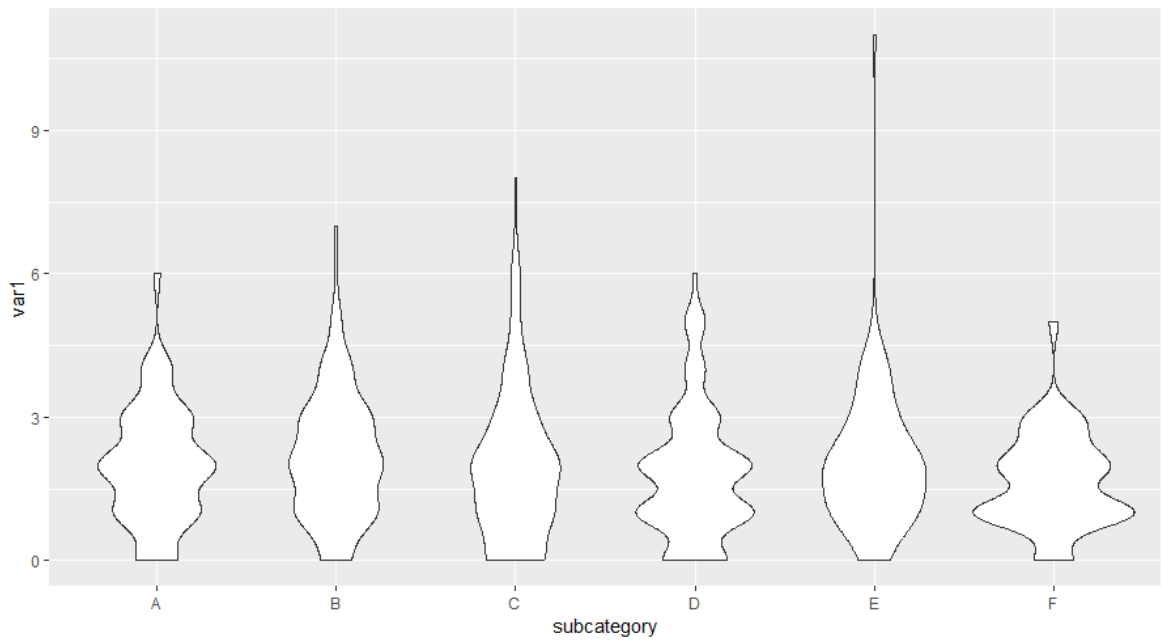
Rearranging and computing means using tydiverse functions

```r
dataset <- complete.data.dataset

# compute the mean of the variable 'var1' for each 'subcategory' group
dataset2 <- dataset %>%
  group_by(subcategory) %>%
  mutate(Mean_var1 = mean(var1))

# compute the mean of the variable 'var2' for each 'subcategory' group
dataset3 <- dataset %>%
  group_by(subcategory) %>%
  mutate(Mean_var2 = mean(var2))

# compute the mean of the variable 'var3' for each 'subcategory' group
dataset4 <- dataset %>%
  group_by(subcategory) %>%
  mutate(Mean_var3 = mean(var3))

dataset4[298:301, ]
# A tibble: 4    6
# Groups:   subcategory [2]
# var1  var2  var3 subcategory  category     Mean_var3
# <dbl> <dbl> <dbl> <chr>       <chr>            <dbl>
#   1     4     3     3 C           cat1          5.60
#   2     3     3     6 D           cat1          6.08
#   3     4     1     3 C           cat1          5.60
#   4     0     3     5 D           cat2          6.08
```

Creating a violin plot with minimal code

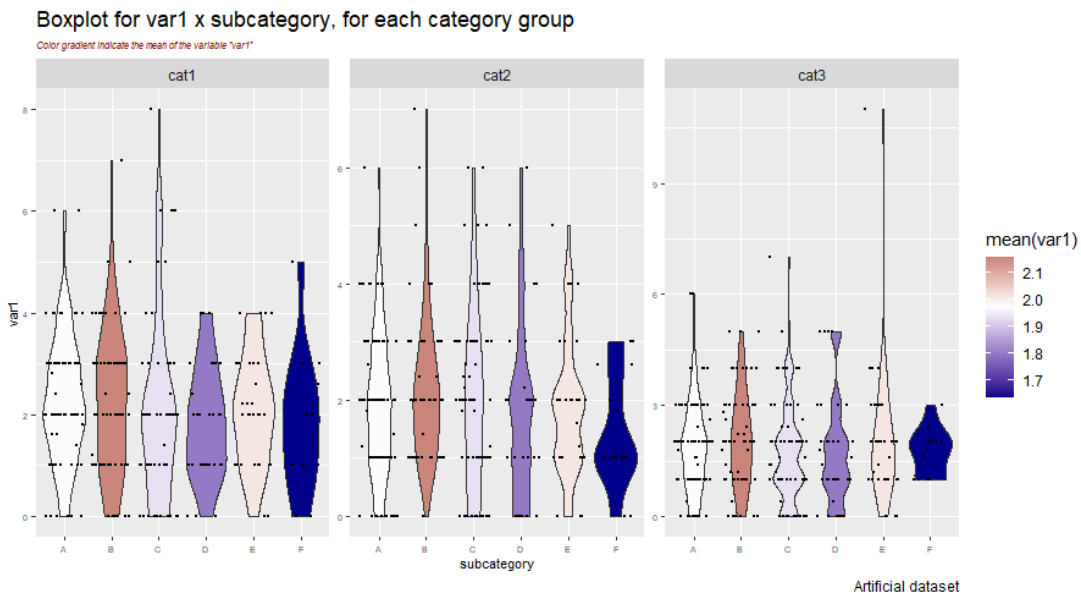
Two lines of code and with help of tidyverse including the powerful visualization library ggplot2 yield a 'quick and dirty' violin plot on which we will improve. We can have an idea of the distribution of the variable 1 for each subcategory.

```
1 ggplot(dataset, aes(x = subcategory, y = var1)) +
2   geom_violin()
```



Multiple plots with mean color gradient

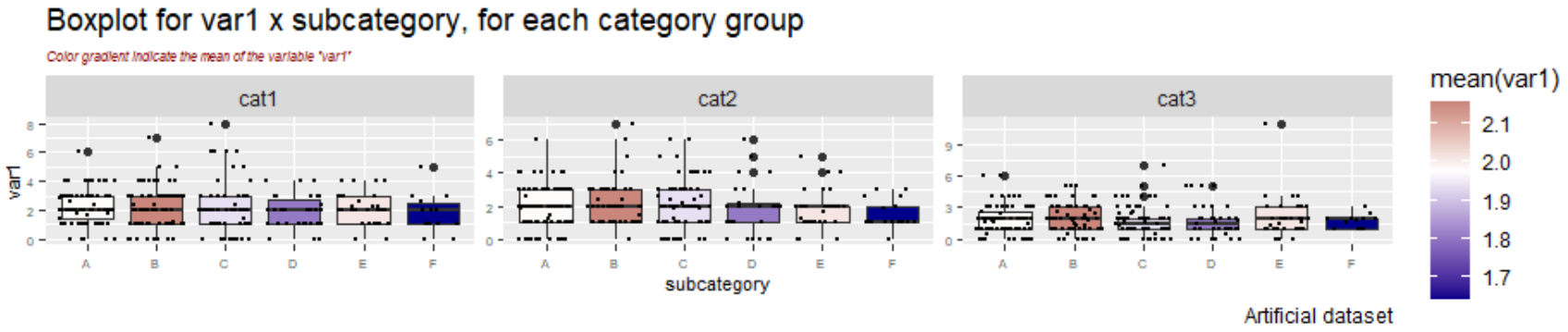
```r
ggplot(dataset2, aes(x = subcategory, y = var1)) +
  geom_violin(aes(fill=Mean_var1)) +
  geom_point(aes(x = subcategory, y = var1), position = 'jitter', size = 0.4) +
  scale_fill_gradient2('mean(var1)', low = "blue4",
                       mid = "white", high = "firebrick4",
                       midpoint = mean(dataset2$Mean_var1)) +
  facet_wrap(~category, scales="free") +
  labs(title = 'Boxplot for var1 x subcategory, for each category group',
       subtitle = "Color gradient indicate the mean of the variable 'var1'",
       caption = "Artificial dataset") +
  theme(axis.text=element_text(size=5),
        axis.title=element_text(size=8),
        plot.subtitle=element_text(size=6, face="italic", color="darkred"))
```

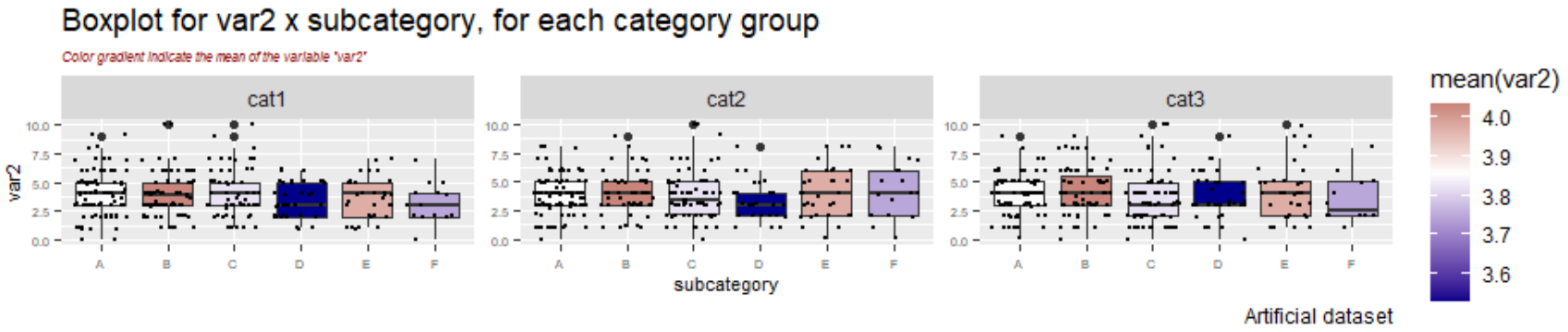


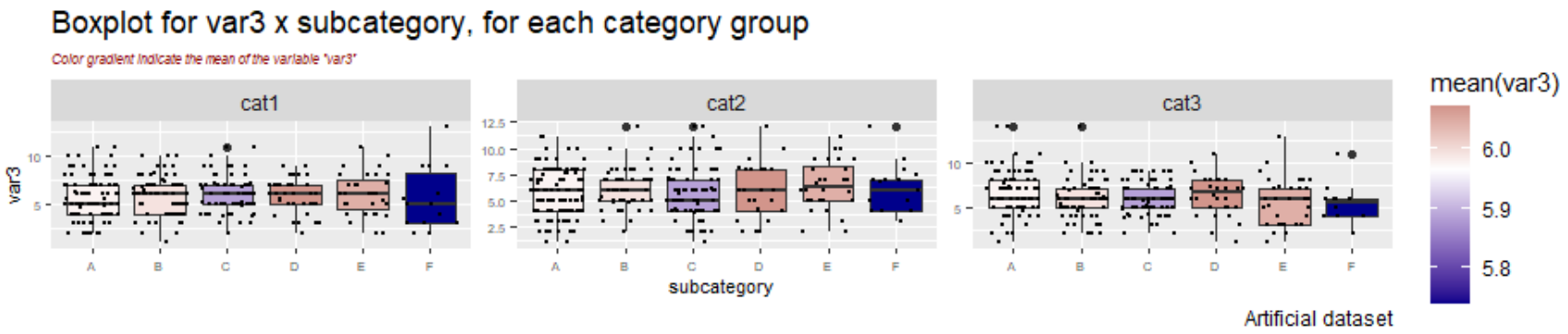
R code to reproduce the figure of slide 1

```r
# Violin plot with facet_wrap
p1 <- ggplot(dataset2, aes(x = subcategory, y = var1)) +
  geom_violin(aes(fill=Mean_var1)) +
  geom_point(aes(x = subcategory, y = var1), position = 'jitter', size = 0.4) +
  scale_fill_gradient2('mean(var1)', low = "blue4",
                       mid = "white", high = "firebrick4",
                       midpoint = mean(dataset2$Mean_var1)) +
  facet_wrap(~category, scales="free") +
  labs(title = 'Violin plot for var1 x subcategory, for each category group',
       subtitle = "Color gradient indicate the mean of the variable 'var1'",
       caption = "Artificial dataset") +
  theme(axis.text=element_text(size=5),
        axis.title=element_text(size=8),
        plot.subtitle=element_text(size=6, face="italic", color="darkred"))

# Violin plot with facet_wrap
p2 <- ggplot(dataset3, aes(x = subcategory, y = var2)) +
  geom_violin(aes(fill= Mean_var2)) +
...

# Violin plot with facet_wrap
p3 <- ggplot(dataset4, aes(x = subcategory, y = var3)) +
  geom_violin(aes(fill= Mean_var3)) +
...

final.plot <- grid.arrange(p1, p2, p3)
```

Objectif: multiple boxplots







What is a boxplot?

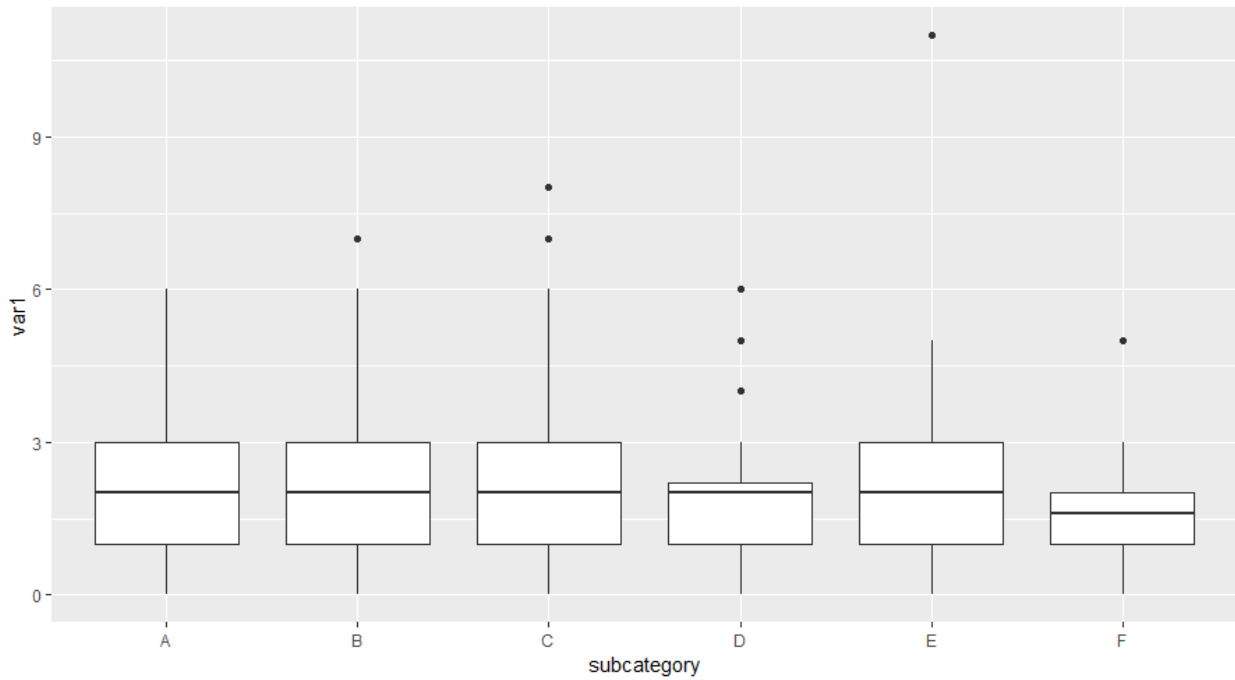
A Boxplot is a non-parametric descriptive statistical way to summarize and visualize the distribution of grouped data. It is used when we have a quantitative variable and a qualitative variable (nominal or ordinal). It gives an indication of the Median (measure of central tendency), the Interquartile Range (measure of dispersion), the 95% range for the observations, as well as outlying observations (observations outside the 95% range). We give the definition of those statistics, F^{-1} being the inverse CDF or Quantile function.

$$\text{First quartile (Q1)} = F^{-1}(0.25)$$
$$\text{Median (Q2)} = F^{-1}(0.5)$$
$$\text{Third quartile (Q3)} = F^{-1}(0.75)$$
$$\text{Interquartile Range (IQR)} = Q3 - Q1$$
$$\text{Outlying observations : } x_i > F^{-1}(0.95) \text{ or } x_i < F^{-1}(0.05)$$

Create a boxplot with minimal code

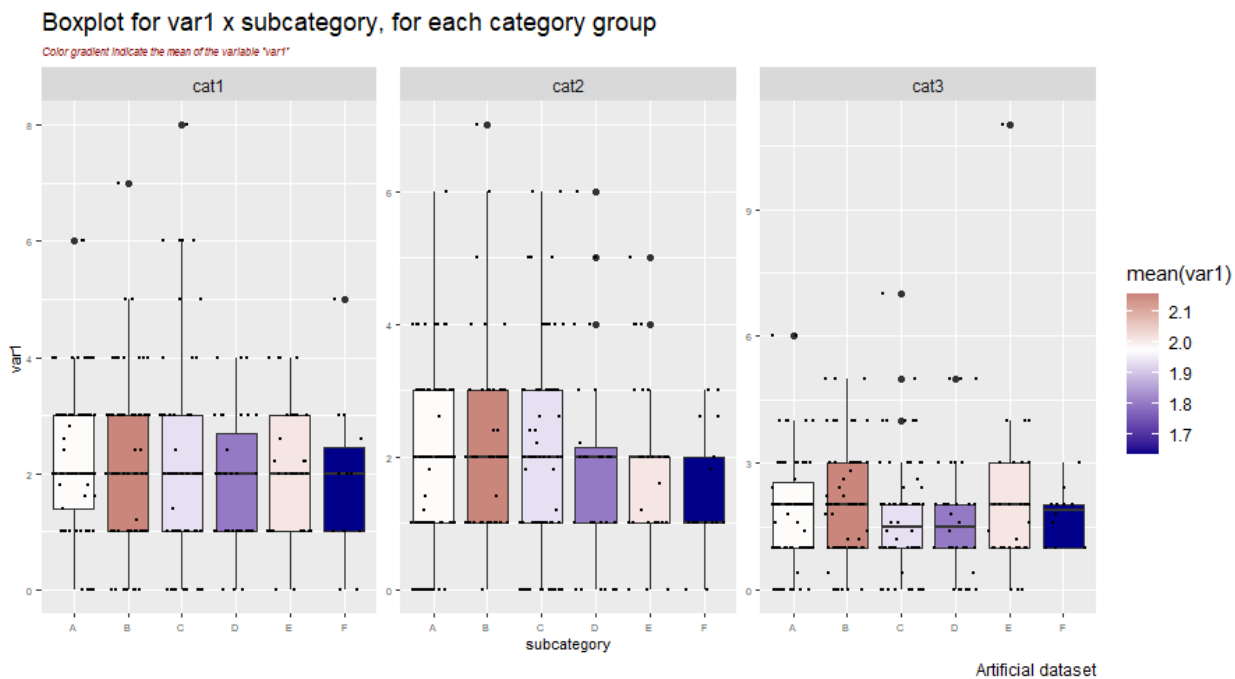
Two lines of code and with help of tidyverse including the powerful visualization library ggplot2 yield a 'quick and dirty' boxplot on which we will improve.

```
1 ggplot(dataset, aes(x = subcategory, y = var1)) +
2   geom_boxplot()
```



Multiple plots with mean color gradient

```r
ggplot(dataset2, aes(x = subcategory, y = var1)) +
  geom_boxplot(aes(fill=Mean_var1)) +
  geom_point(aes(x = subcategory, y = var1), position = 'jitter', size = 0.4) +
  scale_fill_gradient2('mean(var1)', low = "blue4",
                       mid = "white", high = "firebrick4",
                       midpoint = mean(dataset2$Mean_var1)) +
  facet_wrap(~category, scales="free") +
  labs(title = 'Boxplot for var1 x subcategory, for each category group',
       subtitle = "Color gradient indicate the mean of the variable 'var1'",
       caption = "Artificial dataset") +
  theme(axis.text=element_text(size=5),
        axis.title=element_text(size=8),
        plot.subtitle=element_text(size=6, face="italic", color="darkred"))
```



R code to reproduce the figure of slide 8

```r
# to access the function grid.arrange() for multiple plotting
library(gridExtra)

# Boxplot with facet_wrap
p1 <- ggplot(dataset2, aes(x = subcategory, y = var1)) +
  geom_boxplot(aes(fill=Mean_var1)) +
  geom_point(aes(x = subcategory, y = var1), position = 'jitter', size = 0.4) +
  scale_fill_gradient2('mean(var1)', low = "blue4",
                       mid = "white", high = "firebrick4",
                       midpoint = mean(dataset2$Mean_var1)) +
  facet_wrap(~category, scales="free") +
  labs(title = 'Boxplot for var1 x subcategory, for each category group',
       subtitle = "Color gradient indicate the mean of the variable 'var1'",
       caption = "Artificial dataset") +
  theme(axis.text=element_text(size=5),
        axis.title=element_text(size=8),
        plot.subtitle=element_text(size=6, face="italic", color="darkred"))

# Boxplot with facet_wrap
p2 <- ggplot(dataset3, aes(x = subcategory, y = var2)) +
  geom_boxplot(aes(fill= Mean_var2)) +
...

# Boxplot with facet_wrap
p3 <- ggplot(dataset4, aes(x = subcategory, y = var3)) +
  geom_boxplot(aes(fill= Mean_var3)) +
...

final.plot <- grid.arrange(p1, p2, p3)
```