Categories

1. [Programming](https://datascienceplus.com/category/programming/)

Tags

1. [Data Visualisation](https://datascienceplus.com/tag/data-visualisation/)
2. [R Markdown](https://datascienceplus.com/tag/r-markdown/)
3. [R Programming](https://datascienceplus.com/tag/rstats/)

In this article, you learn how to make Automated Dashboard visualizations with distribution in R. First you need to install the `rmarkdown` package into your R library. Assuming that you installed the `rmarkdown`, next you create a new `rmarkdown` script in R.

After this you type the following code in order to create a dashboard with rmarkdown and flexdashboard:

---

title: "Dashboard visualizations in R: Distrubution"

author: "Kristian Larsen"

output:

flexdashboard::flex\_dashboard:

orientation: rows

vertical\_layout: scroll

---

```{r setup, include=FALSE}

library(ggplot2)

library(plotly)

theme\_set(theme\_classic())

```

Row

-----------------------------------------------------------------------

### Chart A: Histogram with Auto Binning

```{r}

g <- ggplot(mpg, aes(displ)) + scale\_fill\_brewer(palette = "Spectral")

g + geom\_histogram(aes(fill=class),

binwidth = .1,

col="black",

size=.1) + # change binwidth

labs(title="Histogram with Auto Binning",

subtitle="Engine Displacement across Vehicle Classes")

ggplotly(p = ggplot2::last\_plot())

```

### Chart B: Histogram with Fixed Bins

```{r}

g + geom\_histogram(aes(fill=class),

bins=5,

col="black",

size=.1) + # change number of bins

labs(title="Histogram with Fixed Bins",

subtitle="Engine Displacement across Vehicle Classes")

ggplotly(p = ggplot2::last\_plot())

ggplotly(p = ggplot2::last\_plot())

```

Row

-----------------------------------------------------------------------

### Cart C: Histogram on a Categorical variable

```{r}

library(ggplot2)

theme\_set(theme\_classic())

g <- ggplot(mpg, aes(manufacturer))

g + geom\_bar(aes(fill=class), width = 0.5) +

theme(axis.text.x = element\_text(angle=65, vjust=0.6)) +

labs(title="Histogram on Categorical Variable",

subtitle="Manufacturer across Vehicle Classes")

ggplotly(p = ggplot2::last\_plot())

```

### Cart D: Density plot

```{r}

library(ggplot2)

theme\_set(theme\_classic())

# Plot

g <- ggplot(mpg, aes(cty))

g + geom\_density(aes(fill=factor(cyl)), alpha=0.8) +

labs(title="Density plot",

subtitle="City Mileage Grouped by Number of cylinders",

caption="Source: mpg",

x="City Mileage",

fill="# Cylinders")

ggplotly(p = ggplot2::last\_plot())

```

Row

-----------------------------------------------------------------------

### Cart E: Box plot

```{r}

library(ggplot2)

theme\_set(theme\_classic())

# Plot

g <- ggplot(mpg, aes(class, cty))

g + geom\_boxplot(varwidth=T, fill="plum") +

labs(title="Box plot",

subtitle="City Mileage grouped by Class of vehicle",

caption="Source: mpg",

x="Class of Vehicle",

y="City Mileage")

ggplotly(p = ggplot2::last\_plot())

```

### Cart F: Box plot

```{r}

library(ggthemes)

g <- ggplot(mpg, aes(class, cty))

g + geom\_boxplot(aes(fill=factor(cyl))) +

theme(axis.text.x = element\_text(angle=65, vjust=0.6)) +

labs(title="Box plot",

subtitle="City Mileage grouped by Class of vehicle",

caption="Source: mpg",

x="Class of Vehicle",

y="City Mileage")

ggplotly(p = ggplot2::last\_plot())

```

Row

-----------------------------------------------------------------------

### Cart G: Box plot + Dot plot

```{r}

library(ggplot2)

theme\_set(theme\_bw())

# plot

g <- ggplot(mpg, aes(manufacturer, cty))

g + geom\_boxplot() +

geom\_dotplot(binaxis='y',

stackdir='center',

dotsize = .5,

fill="red") +

theme(axis.text.x = element\_text(angle=65, vjust=0.6)) +

labs(title="Box plot + Dot plot",

subtitle="City Mileage vs Class: Each dot represents 1 row in source data",

caption="Source: mpg",

x="Class of Vehicle",

y="City Mileage")

ggplotly(p = ggplot2::last\_plot())

```

### Cart H: Tufte Boxplot

```{r}

library(ggthemes)

library(ggplot2)

theme\_set(theme\_tufte()) # from ggthemes

# plot

g <- ggplot(mpg, aes(manufacturer, cty))

g + geom\_tufteboxplot() +

theme(axis.text.x = element\_text(angle=65, vjust=0.6)) +

labs(title="Tufte Styled Boxplot",

subtitle="City Mileage grouped by Class of vehicle",

caption="Source: mpg",

x="Class of Vehicle",

y="City Mileage")

```

Row

-----------------------------------------------------------------------

### Cart I: Violin plot

```{r}

library(ggplot2)

theme\_set(theme\_bw())

# plot

g <- ggplot(mpg, aes(class, cty))

g + geom\_violin() +

labs(title="Violin plot",

subtitle="City Mileage vs Class of vehicle",

caption="Source: mpg",

x="Class of Vehicle",

y="City Mileage")

ggplotly(p = ggplot2::last\_plot())

```

### Cart J: Population Pyramid

```{r}

library(ggplot2)

library(ggthemes)

options(scipen = 999) # turns of scientific notations like 1e+40

# Read data

email\_campaign\_funnel <- read.csv("https://raw.githubusercontent.com/selva86/datasets/master/email\_campaign\_funnel.csv")

# X Axis Breaks and Labels

brks <- seq(-15000000, 15000000, 5000000)

lbls = paste0(as.character(c(seq(15, 0, -5), seq(5, 15, 5))), "m")

# Plot

ggplot(email\_campaign\_funnel, aes(x = Stage, y = Users, fill = Gender)) + # Fill column

geom\_bar(stat = "identity", width = .6) + # draw the bars

scale\_y\_continuous(breaks = brks, # Breaks

labels = lbls) + # Labels

coord\_flip() + # Flip axes

labs(title="Email Campaign Funnel") +

theme\_tufte() + # Tufte theme from ggfortify

theme(plot.title = element\_text(hjust = .5),

axis.ticks = element\_blank()) + # Centre plot title

scale\_fill\_brewer(palette = "Dark2") # Color palette

```

Screenshot:  
[](https://i1.wp.com/datascienceplus.com/wp-content/uploads/2018/12/V4-Distribution.jpg?ssl=1)

The result of the above coding are published with RPubs [here](https://rpubs.com/knl84/445854).

**References**

1. [Using flexdashboard in R](https://rmarkdown.rstudio.com/flexdashboard/)