First thing you need to do is register the Windows fonts in R. By default the fonts in R to choose from are limited.

# Make sure all Windows fonts are available within R library(extrafont)

# Auto detect all Windows TTFs. use fonts() or fonttable() for a complete overview afterwards

font\_import()

# Register fonts for Windows bitmap output loadfonts(device="win")

Next you need to create a color palette matching the corporate identity you want to be reflected in ggplot output.

# Cmotions default palette

palette\_cmotions <- c("#003D7C", #F5A507","#7C3F00","#000000"," #FFDC51","#ACACAC")

You have two options to go forward: first is to create a new ggplot template from scratch by specifying all elements. It will give you maximum control and also a lot of work. Second option is to use an existing theme and only edit those part as you see fit. I’m choosing the *theme\_bw* that comes with ggplot and alter only those parts I think are necessary to match my corporate identity. As you can see in the code below I’ve added many *element\_blank()* tags so my legend does not have a title, the panel does not have a border and by default the plot has no background.

# load ggplot library(ggplot2)

# Create Cmotions theme for ggplot theme\_cmotions <- function() {

theme\_bw(base\_size=8, base\_family="Verdana") %+replace% # use theme\_bw as default, replace font family

theme(

# adjust legend

legend.background = element\_blank(), legend.title = element\_blank(),

# adjust axis

axis.ticks = element\_blank(), axis.title.y = element\_blank(), axis.title.x = element\_blank(),

axis.text = element\_text(color = "black"), # adjust panel;

panel.background = element\_rect(colour = "white"), panel.border = element\_blank(),

panel.grid = element\_blank(), # adjust plot

plot.background = element\_blank(),

plot.title = element\_text(family="Arial Black", face="bold",

colour="black", size=14), complete = TRUE

)

}

Using the created template is easy, just specify which theme you want to use in the ggplot command. First let me set up some dummy data to use.

# create data for example plots

df <- data.frame(category = c('cat1', 'cat1', 'cat1', 'cat2', 'cat2',

'cat2', 'cat3', 'cat3', 'cat3', 'cat4', 'cat4', 'cat4', 'cat5', 'cat5', 'cat5', 'cat6', 'cat6', 'cat6'),

year = c('2018', '2019', '2020', '2018', '2019', '2020', '2018',

'2019', '2020', '2018', '2019', '2020', '2018', '2019', '2020', '2018',

'2019', '2020'),

value = as.numeric(c('2','5', '6', '4','7', '10', '5','8', '12',

'2','4', '6', '4','6', '8', '8','8', '12')))

Next I create 4 different plots and specify the new theme I would like to use and the color palette.

# Example for bar chart - no legend

bar <-ggplot(df, aes(category, value, fill = category)) + geom\_col(show.legend = FALSE) +

theme\_cmotions() + ggtitle("Category example") +

scale\_fill\_manual(values = palette\_cmotions)

# Example for stacked bar chart - only 3 categories

stacked<-ggplot(df[1:9,], aes(year, value, fill = category)) + geom\_col() +

theme\_cmotions() +

ggtitle("Category stacked example") + scale\_fill\_manual(values = palette\_cmotions)

# Example for line chart

line<-ggplot(df[1:9,], aes(year, value, group = category, color=category)) +

geom\_line(size=2) + theme\_cmotions() + ggtitle("Category trend") +

scale\_color\_manual(values = palette\_cmotions)

# Example for bar chart - facet

facet<-ggplot(df, aes(year, value, fill = category)) + geom\_bar(stat="identity") +

theme\_cmotions() +

ggtitle("Category example - facet") + facet\_wrap(~ category, nrow=2, ncol=3) +

scale\_fill\_manual(values = palette\_cmotions) + theme(strip.background = element\_blank(), strip.text.x = element\_blank()) # without the strip

# Display examples library(gridExtra)

grid.arrange(bar,stacked,line,facet, ncol=2)

If you want to use this new theme as a default theme just add the lines below to your .Rprofile and the theme will be set during R startup.

# code to add to .Rprofile for Cmotions theme as default setHook(packageEvent("ggplot2", "onLoad"),

function(...) ggplot2::theme\_set(ggplot2::theme\_cmotions()))

Of course more changes can be made, depending on your needs. I hope this basic example is useful to you when you want your corporate identity reflected in your R output.