

advanced-plotting.R

julian

Tue Jun 2 16:07:31 2015

```
#!/usr/bin/env Rscript

# beyond base R graphics and other styles
# using previous data for examples

target.dir <- '~/GitHub/reproducible-research/Day-3/datasets'
target.file <- 'advanced-plotting-r-examples.txt'
sink(file = file.path(target.dir, target.file))

# import data -----

save.path <- '~/GitHub/reproducible-research/Day-3/datasets'
save.file <- '/basic-grouping-plotting.rda'

load(file.path(save.path, save.file))

cbbPalette <-
  c("#000000", "#E69F00", "#56B4E9",
    "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")

cbPalette <-
  c("#999999", "#E69F00", "#56B4E9",
    "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")

# base R with customization -----

# see help(par) on how to set base styles
# and the R reference card

# tutorial based on
# https://flowingdata.com/2014/10/23/moving-past-default-charts/
# which also has source code available for download

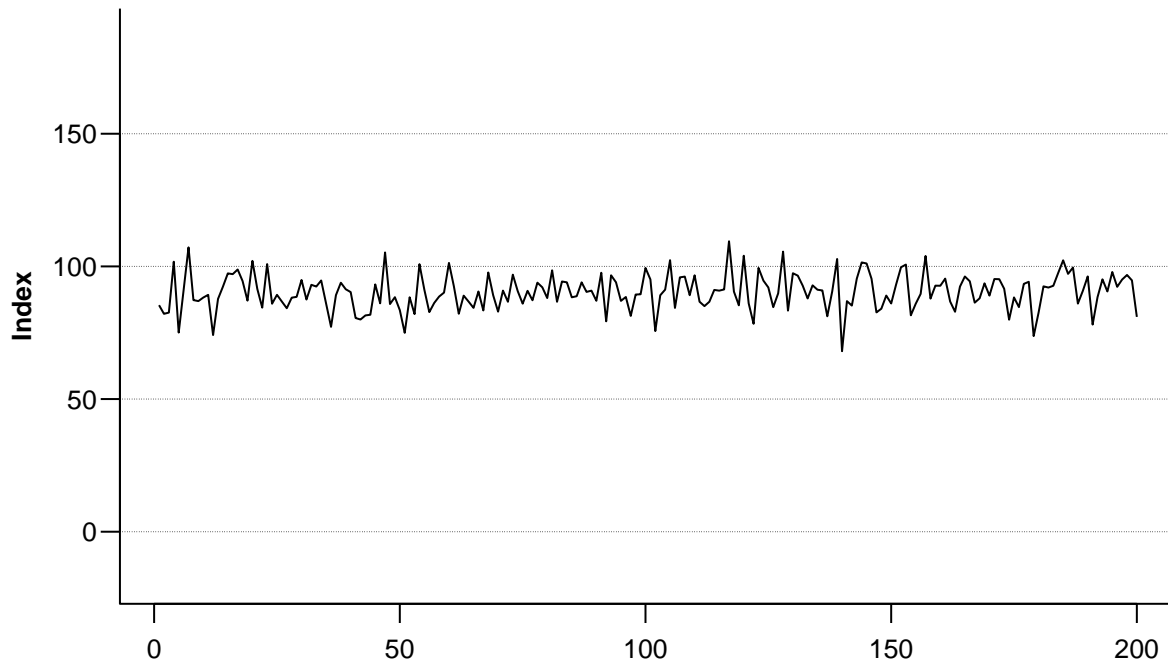
# quick rundown:
# mar: number of lines of margin on each side
# oma: size of outer margins in lines of text
# xpd: clipping parameter
# yaxs: style of y axis
# mgp: margin line for axis title, axis labels, axis line
# las: style of axis labels
# lend: line end style

# newspaper background
par(mar = c(4, 4, 3, 1), oma = c(0,0,0,0), xpd = FALSE, xaxs = "r",
    yaxs = "i", mgp = c(2.1,.6,0), las = 1, lend = 1)
```

```
plot(col1 ~ col5, data = plot.df, ylim = c(60, 110),
     type = "l", bty = "l", las = 1,
     main = "Column 1 vs. Column 5",
     xlab = "", ylab = expression(bold("Index")), family = "Helvetica",
     cex.axis = 0.8, cex.lab = 0.8, asp = 1/2)

grid(NA, NULL, col = "black", lty = "dotted", lwd = 0.3)
```

Column 1 vs. Column 5

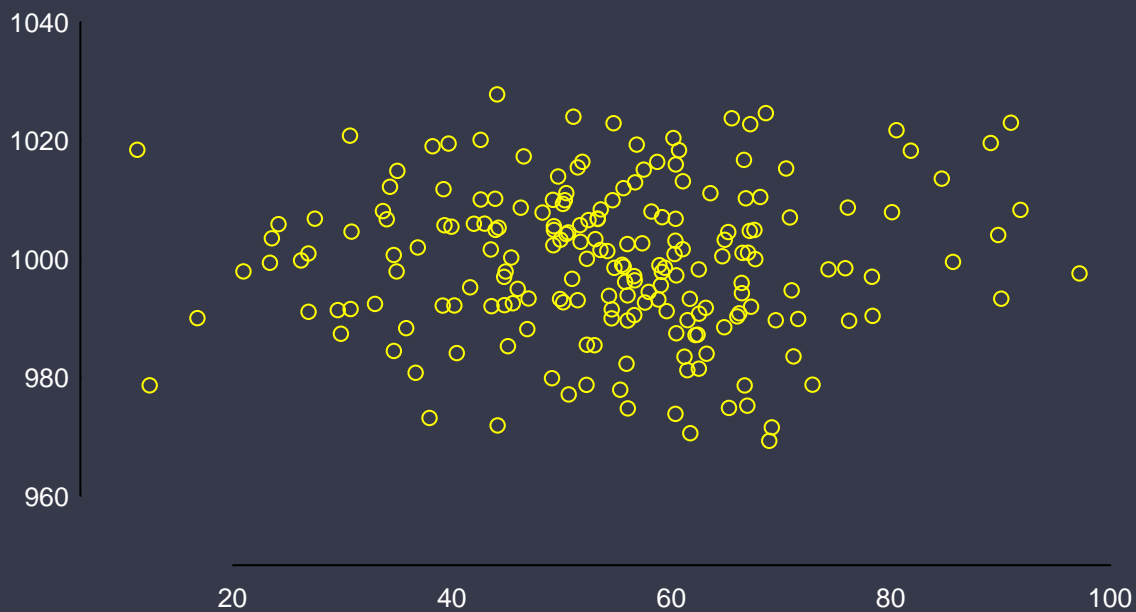


```
# Feltron style

par(bg = "#36394A", mar = c(5, 4, 3, 2), oma = c(0,0,0,0),
    xpd = FALSE, xaxs = "r", yaxs = "i", mgp = c(2.8,0.3,0.5),
    col.lab = "white", col.axis = "white", col.main = "white",
    font.main = 1, cex.main = 0.8, cex.axis = 0.8, cex.lab = 0.8,
    family = "Helvetica", lend = 1, tck = 0)

plot(x = rnorm(n = 200, mean = 55, sd = 14),
     y = rnorm(n = 200, mean = 1000, sd = 12),
     type = "p", bty = "n", las = 1, asp = 1/2, col = 'yellow',
     main = "THESE RANDOM NUMBERS ARE PRETTY, ARE THEY NOT?",
     xlab = "", ylab = "")
```

THESE RANDOM NUMBERS ARE PRETTY, ARE THEY NOT?



```
# ggplot 2 with customization -----  
  
# create 538-style plot  
# adapted from http://minimaxir.com/2015/02/ggplot-tutorial/  
  
library(ggplot2)  
library(scales)  
library(grid)  
library(RColorBrewer)  
  
# changed font size so that it is readable (15 point)  
  
fte_theme <- function() {  
  
  # Generate the colors for the chart procedurally with RColorBrewer  
  palette <- brewer.pal("Greys", n=9)  
  color.background = palette[2]  
  color.grid.major = palette[3]  
  color.axis.text = palette[6]  
  color.axis.title = palette[7]  
  color.title = palette[9]  
  
  # Begin construction of chart  
  theme_bw(base_size=9) +  
  
  # Set the entire chart region to a light gray color
```

```

theme(panel.background=element_rect(fill=color.background, color=color.background)) +
theme(plot.background=element_rect(fill=color.background, color=color.background)) +
theme(panel.border=element_rect(color=color.background)) +

# Format the grid
theme(panel.grid.major=element_line(color=color.grid.major,size=0.25)) +
theme(panel.grid.minor=element_blank()) +
theme(axis.ticks=element_blank()) +

# Format the legend, but hide by default
theme(legend.position="none") +
theme(legend.background = element_rect(fill=color.background)) +
theme(legend.text = element_text(size=15,color=color.axis.title)) +

# Set title and axis labels, and format these and tick marks
theme(plot.title=element_text(color=color.title, size=20, vjust=1.25)) +
theme(axis.text.x=element_text(size=15,color=color.axis.text)) +
theme(axis.text.y=element_text(size=15,color=color.axis.text)) +
theme(axis.title.x=element_text(size=15,color=color.axis.title, vjust=0)) +
theme(axis.title.y=element_text(size=15,color=color.axis.title, vjust=1.25)) +

# Plot margins
theme(plot.margin = unit(c(0.35, 0.2, 0.3, 0.35), "cm"))
}

# theme that includes the legend
fte_theme2 <- function() {

# Generate the colors for the chart procedurally with RColorBrewer
palette <- brewer.pal("Greys", n=9)
color.background = palette[2]
color.grid.major = palette[3]
color.axis.text = palette[6]
color.axis.title = palette[7]
color.title = palette[9]

# Begin construction of chart
theme_bw(base_size=9) +

# Set the entire chart region to a light gray color
theme(panel.background=element_rect(fill=color.background, color=color.background)) +
theme(plot.background=element_rect(fill=color.background, color=color.background)) +
theme(panel.border=element_rect(color=color.background)) +

# Format the grid
theme(panel.grid.major=element_line(color=color.grid.major,size=0.25)) +
theme(panel.grid.minor=element_blank()) +
theme(axis.ticks=element_blank()) +

# Format the legend, but hide by default
theme(legend.position="bottom") +
theme(legend.background = element_rect(fill=color.background)) +
theme(legend.text = element_text(size=15,color=color.axis.title)) +

```

```

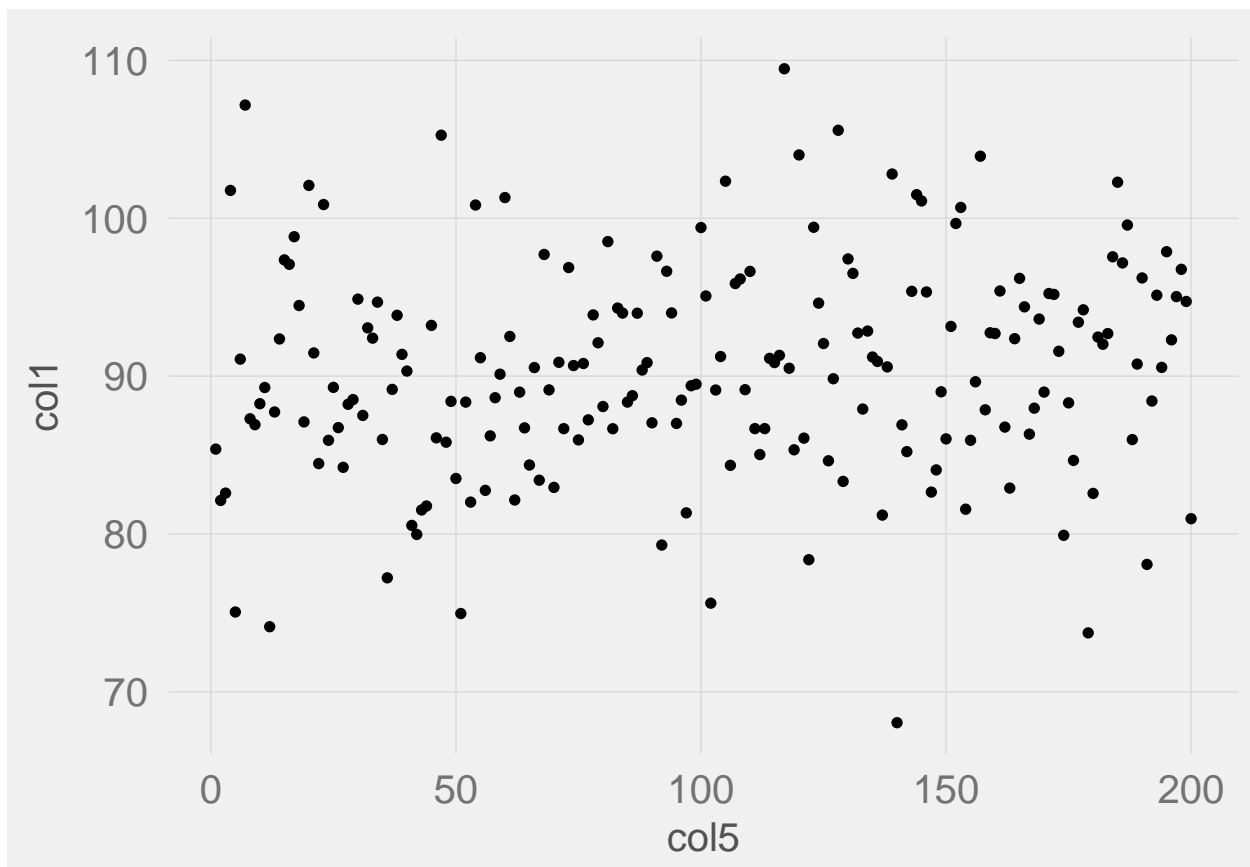
# Set title and axis labels, and format these and tick marks
theme(plot.title=element_text(color=color.title, size=20, vjust=1.25)) +
theme(axis.text.x=element_text(size=15,color=color.axis.text)) +
theme(axis.text.y=element_text(size=15,color=color.axis.text)) +
theme(axis.title.x=element_text(size=15,color=color.axis.title, vjust=0)) +
theme(axis.title.y=element_text(size=15,color=color.axis.title, vjust=1.25)) +

# Plot margins
theme(plot.margin = unit(c(0.35, 0.2, 0.3, 0.35), "cm"))
}

# plot data.frame

ggplot(plot.df, aes(x = col5, y = col1)) +
  geom_point() +
  fte_theme()

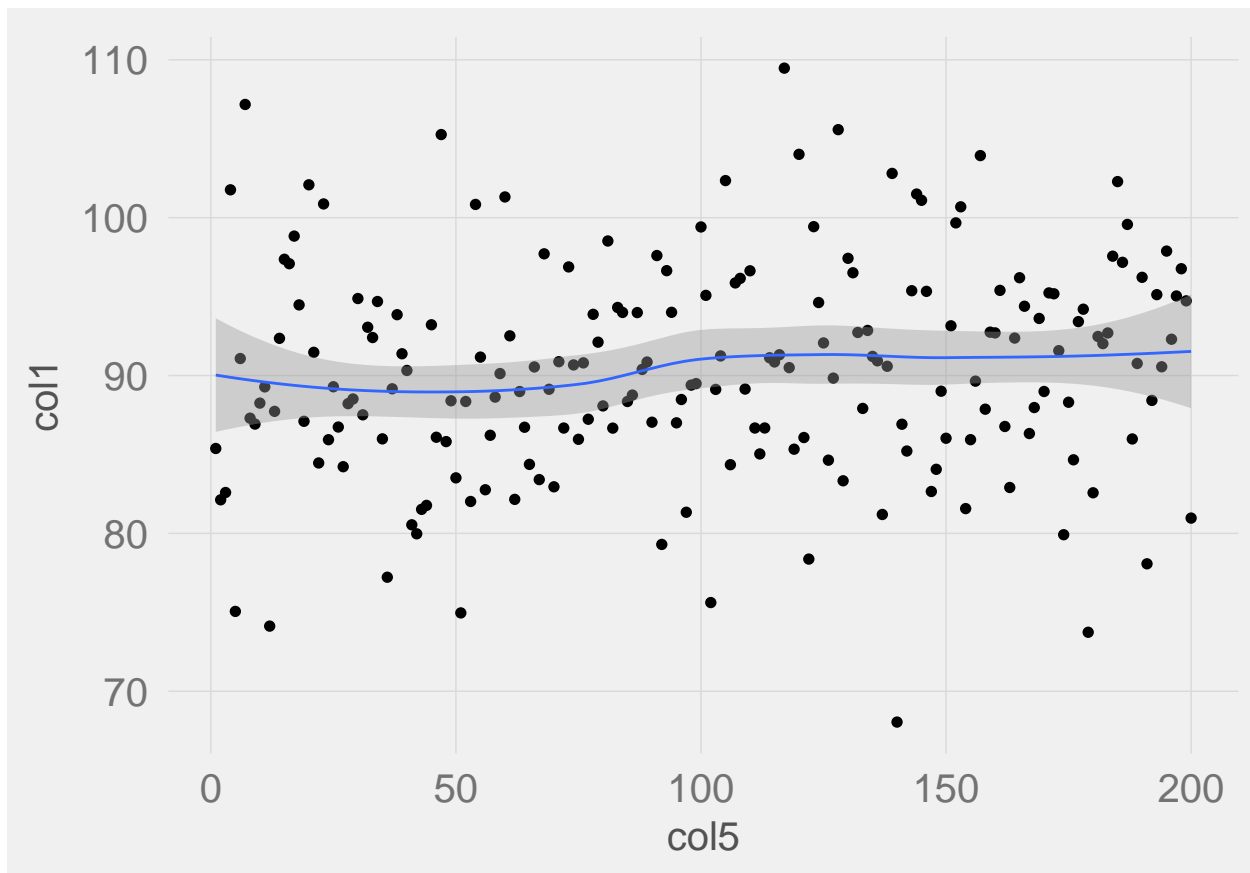
```



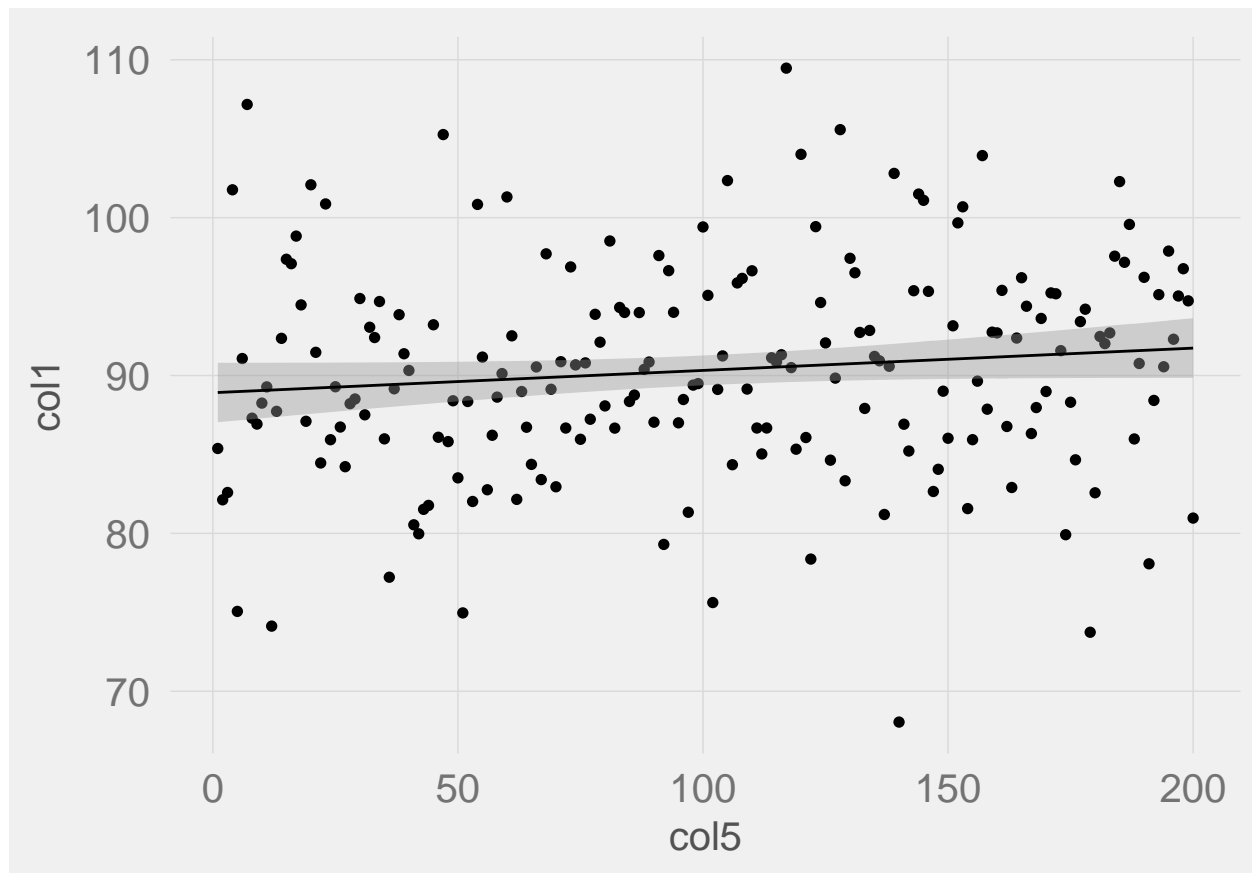
```

ggplot(plot.df, aes(x = col5, y = col1)) +
  geom_point() +
  geom_smooth() +
  fte_theme()

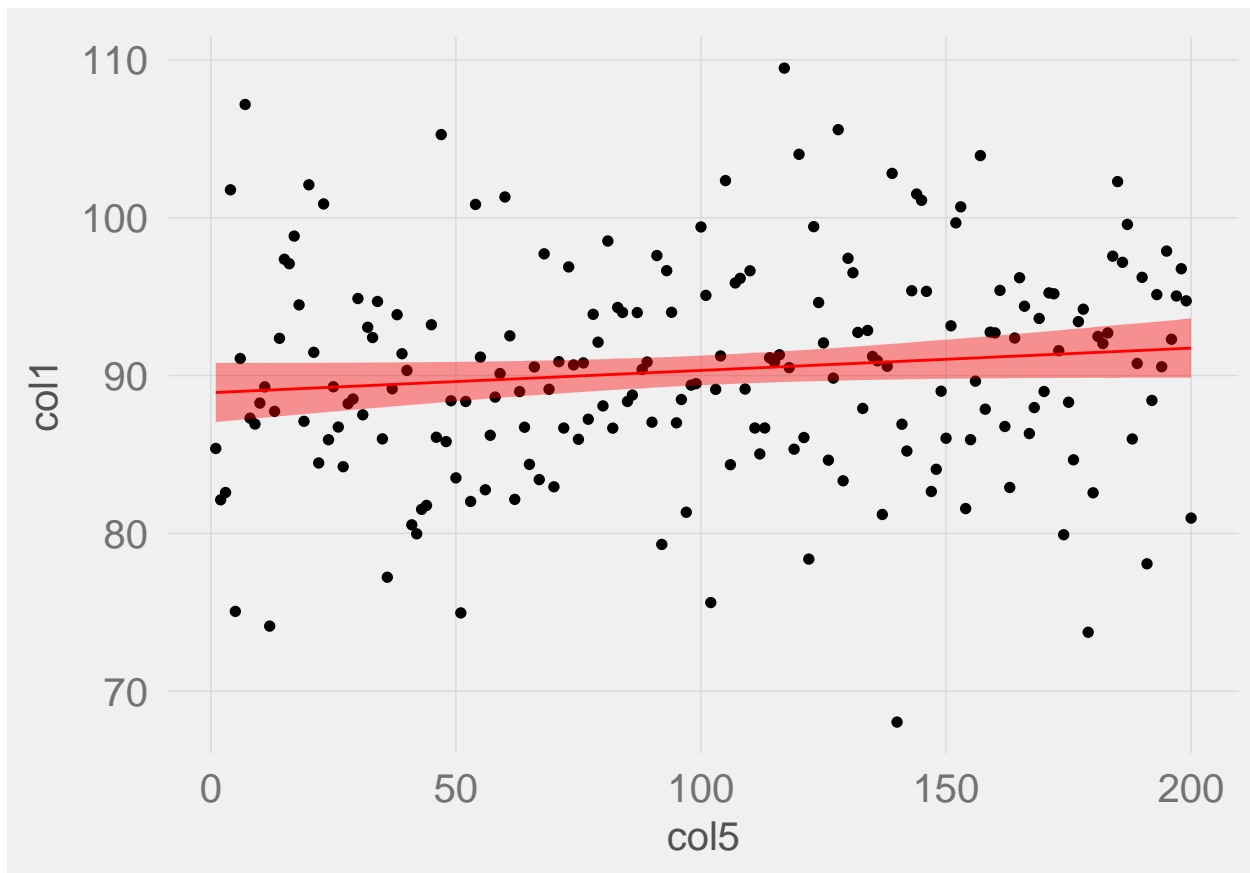
```



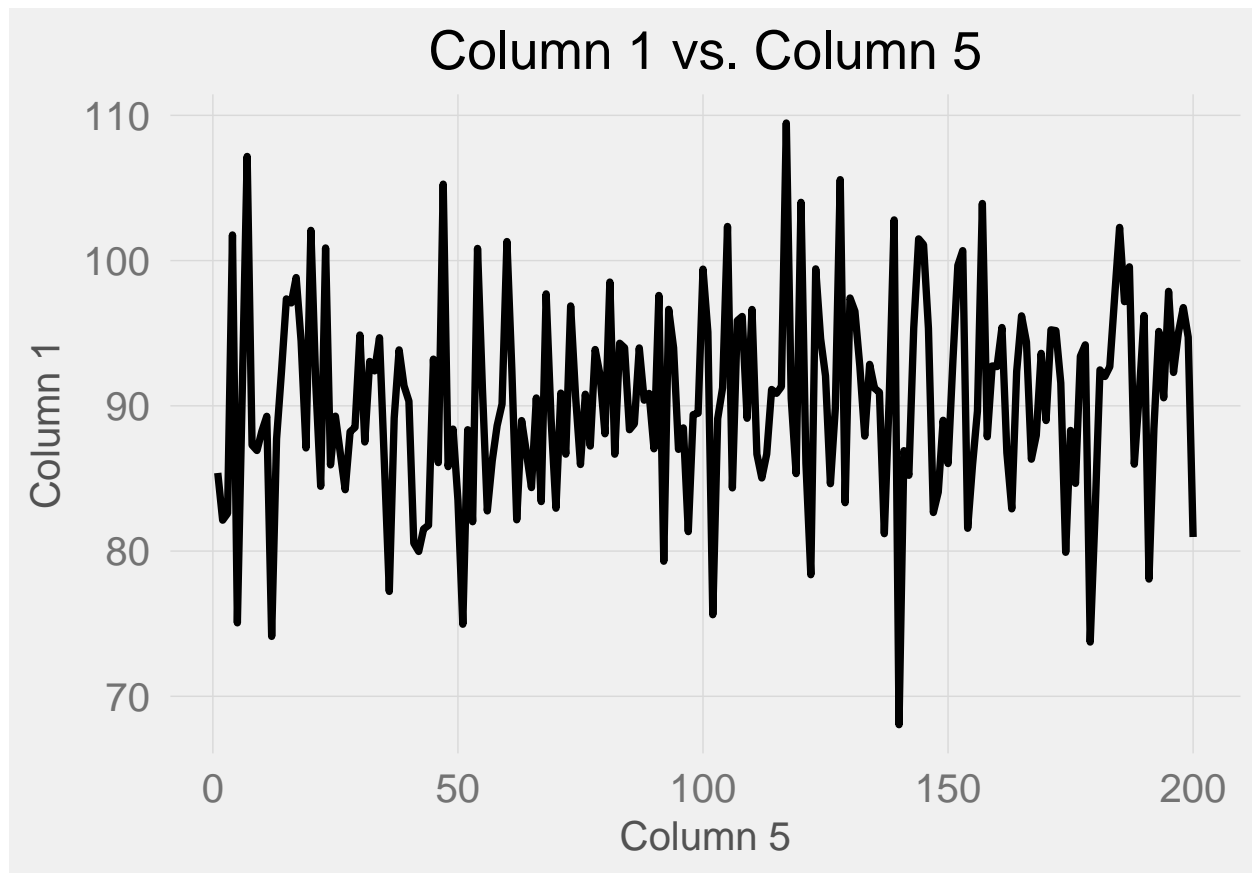
```
# change the smoothing method  
ggplot(plot.df, aes(x = col5, y = col1)) +  
  geom_point() +  
  geom_smooth(method = 'lm', colour = 'black') +  
  fte_theme()
```



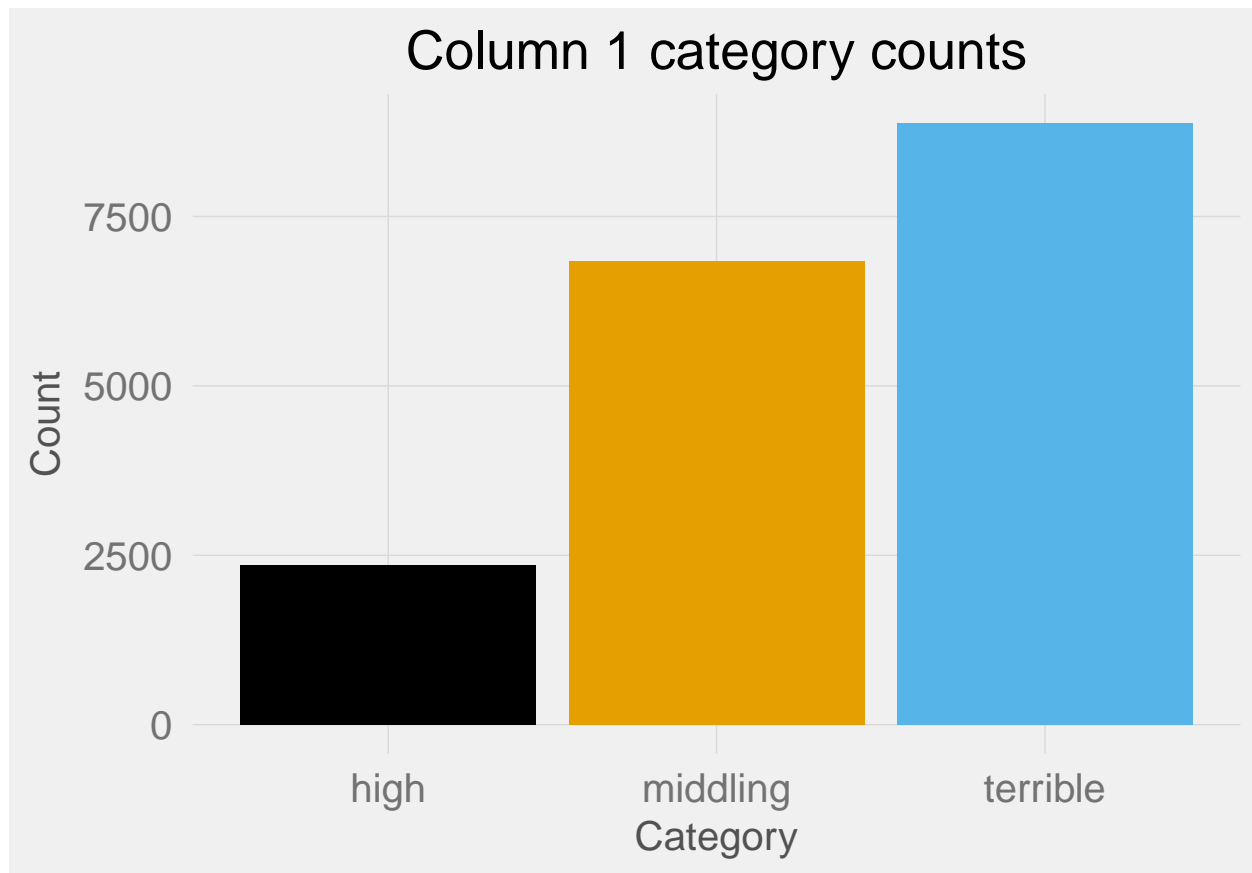
```
ggplot(plot.df, aes(x = col5, y = col1)) +  
  geom_point() +  
  geom_smooth(method = 'glm', colour = 'red', fill = 'red') +  
  fte_theme()
```



```
ggplot(plot.df, aes(x = col5, y = col1)) +  
  geom_line(linetype = 'solid', size = 1.3) +  
  labs(title = 'Column 1 vs. Column 5', x = 'Column 5', y = 'Column 1') +  
  fte_theme()
```

```
# bar plot
ggplot(plot.df, aes(x = Category, y = col1)) +
  geom_bar(stat = 'identity', aes(fill = Category)) +
  labs(title = 'Column 1 category counts', x = 'Category', y = 'Count') +
  fte_theme() +
  scale_colour_manual(values = cbbPalette) +
  scale_fill_manual(values = cbbPalette)
```



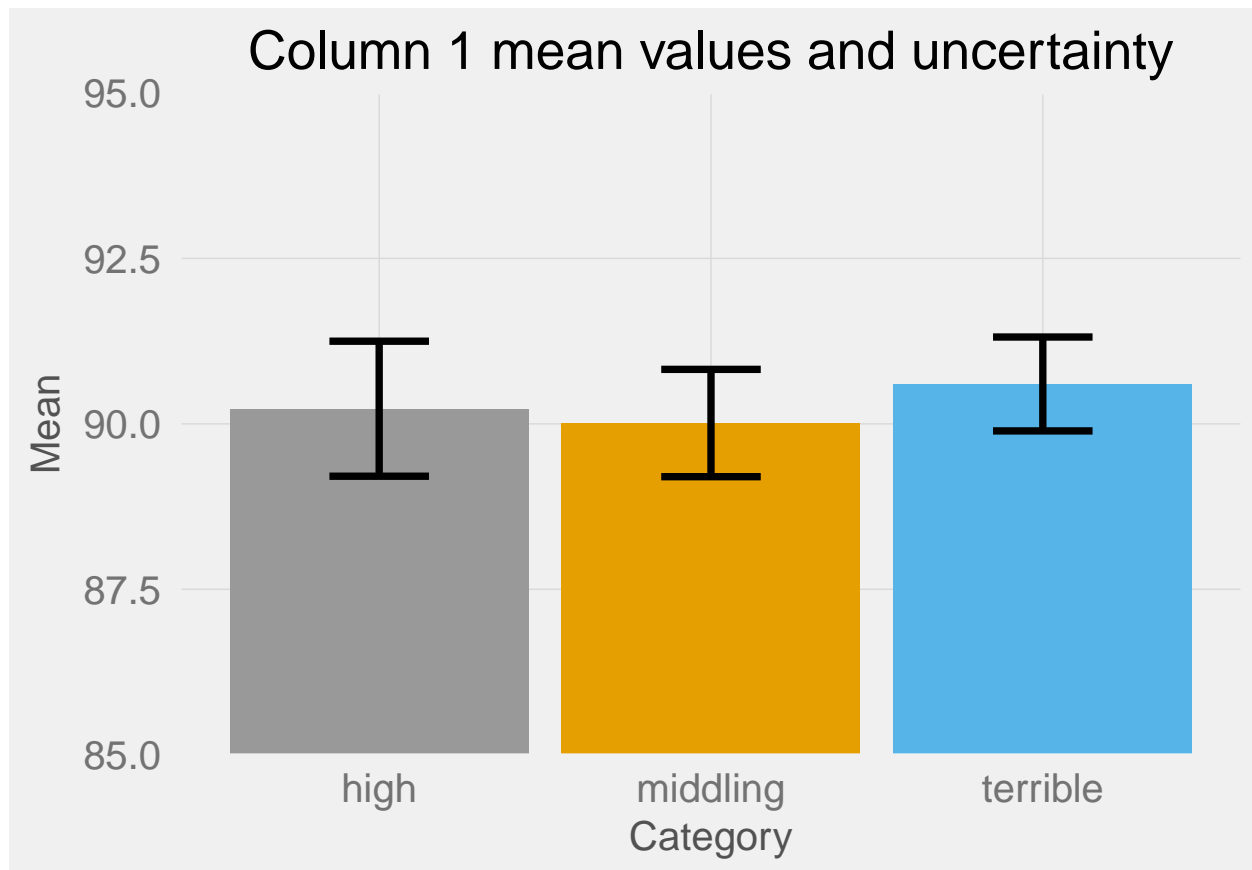
```
# bar plot with error bars

library(dplyr)

col1.err <-
  plot.df %>%
    group_by(Category) %>%
    summarize(meanCol1 = mean(col1), n = sum(complete.cases(col1)),
              errCol1 = sqrt(var(col1) / n))

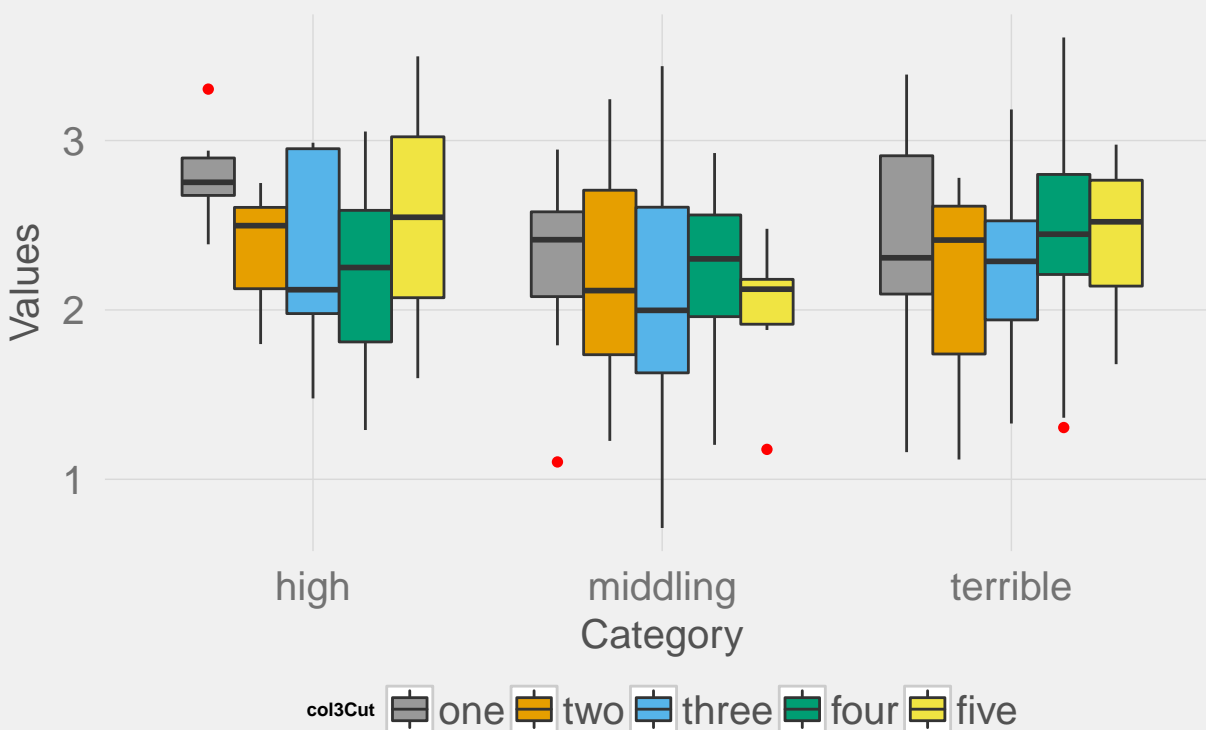
dodge <- position_dodge(width = 0.9)

ggplot(col1.err, aes(x = Category, y = meanCol1)) +
  geom_bar(stat = 'identity', aes(fill = Category)) +
  geom_errorbar(aes(ymin = meanCol1 - errCol1, ymax = meanCol1 + errCol1),
               position = 'dodge', width = 0.3, size = 1.3) +
  labs(title = 'Column 1 mean values and uncertainty',
       x = 'Category', y = 'Mean') +
  fte_theme() +
  coord_cartesian(ylim = c(85, 95)) +
  scale_colour_manual(values = cbPalette) +
  scale_fill_manual(values = cbPalette)
```



```
# boxplots
ggplot(plot.df, aes(x = Category, y = col2)) +
  geom_boxplot(aes(fill = col3Cut), outlier.colour = 'red') +
  labs(title = 'Column 3 distribution - normal approximation',
       x = 'Category', y = 'Values') +
  fte_theme2() +
  scale_colour_manual(values = cbPalette) +
  scale_fill_manual(values = cbPalette)
```

Column 3 distribution – normal approximation



```
# create own theme -----
red_black_theme <- function() {

  # Generate the colors for the chart procedurally with RColorBrewer
  palette <- brewer.pal("RdGy", n = 11)
  color.background = palette[2]
  color.grid.major = palette[11]
  color.axis.text = palette[6]
  color.axis.title = palette[6]
  color.title = palette[11]

  # Begin construction of chart
  theme_bw(base_size = 15) +

  # Set the entire chart region to a light gray color
  theme(panel.background=element_rect(fill=color.background, color=color.background)) +
  theme(plot.background=element_rect(fill=color.background, color=color.background)) +
  theme(panel.border=element_rect(color=color.background)) +

  # Format the grid
  theme(panel.grid.major=element_line(color=color.grid.major,size = 0.5)) +
  theme(panel.grid.minor=element_blank()) +
  theme(axis.ticks=element_blank()) +

  # Format the legend, but hide by default
```

```

theme(legend.position="bottom") +
theme(legend.background = element_rect(fill=color.background)) +
theme(legend.text = element_text(size=15,color=color.axis.title)) +

# Set title and axis labels, and format these and tick marks
theme(plot.title=element_text(color=color.title, size=15, vjust=1.25)) +
theme(axis.text.x=element_text(size=15,color=color.axis.text)) +
theme(axis.text.y=element_text(size=15,color=color.axis.text)) +
theme(axis.title.x=element_text(size=15,color=color.axis.title, vjust=0)) +
theme(axis.title.y=element_text(size=15,color=color.axis.title, vjust=1.25))
}

dark2_vals <- c("#1B9E77", "#D95F02", "#7570B3",
               "#E7298A", "#66A61E", "#E6AB02", "#A6761D", "#666666")

norm.col1 <- rnorm(n = 2000, mean = 50, sd = 15)
norm.col2 <- rnorm(n = 2000, mean = 300, sd = 40)

norm.data <- data.frame(norm.col1, norm.col2)

norm.data$Category <-
  gl(n = 4, k = 20, labels = c('one', 'two', 'three', 'four'))

col2.vs.col5.2d.density <-
  ggplot(norm.data, aes(x = norm.col1, y = norm.col2)) +
  stat_density2d(aes(fill = Category, colour = Category),
                alpha = 0.5, geom = 'polygon') +
  red_black_theme() +
  labs(title = 'Col 1 vs. Col2 2D KDE', x = 'Column 5', y = 'Column 2') +
  scale_colour_manual(values = dark2_vals) +
  scale_fill_manual(values = dark2_vals) +
  facet_grid(. ~ Category)

straight_goth <- function() {

  color.background = '#000000'
  color.grid.major = 'white'
  color.axis.text = 'white'
  color.axis.title = 'white'
  color.title = 'red'

  # Begin construction of chart
  theme_bw(base_size = 15) +

  # Set the entire chart region to a light gray color
  theme(panel.background=element_rect(fill=color.background,
                                       color=color.background)) +
  theme(plot.background=element_rect(fill=color.background,
                                       color=color.background)) +
  theme(panel.border=element_rect(color=color.background)) +

  # Format the grid
  theme(panel.grid.major=element_line(color=color.grid.major,size = 0.5)) +

```

```

theme(panel.grid.minor=element_blank()) +
theme(axis.ticks=element_blank()) +

# Format the legend, but hide by default
theme(legend.position="bottom") +
theme(legend.background = element_rect(fill=color.background)) +
theme(legend.text = element_text(size=15,color=color.axis.title)) +

# Set title and axis labels, and format these and tick marks
theme(plot.title=element_text(color=color.title, size=15, vjust=1.25)) +
theme(axis.text.x=element_text(size=15,color=color.axis.text)) +
theme(axis.text.y=element_text(size=15,color=color.axis.text)) +
theme(axis.title.x=element_text(size=15,color=color.axis.title, vjust=0)) +
theme(axis.title.y=element_text(size=15,color=color.axis.title, vjust=1.25))
}

points <- seq(-20, 20, 1)
growth.saturation <- plogis(points, scale = 3)

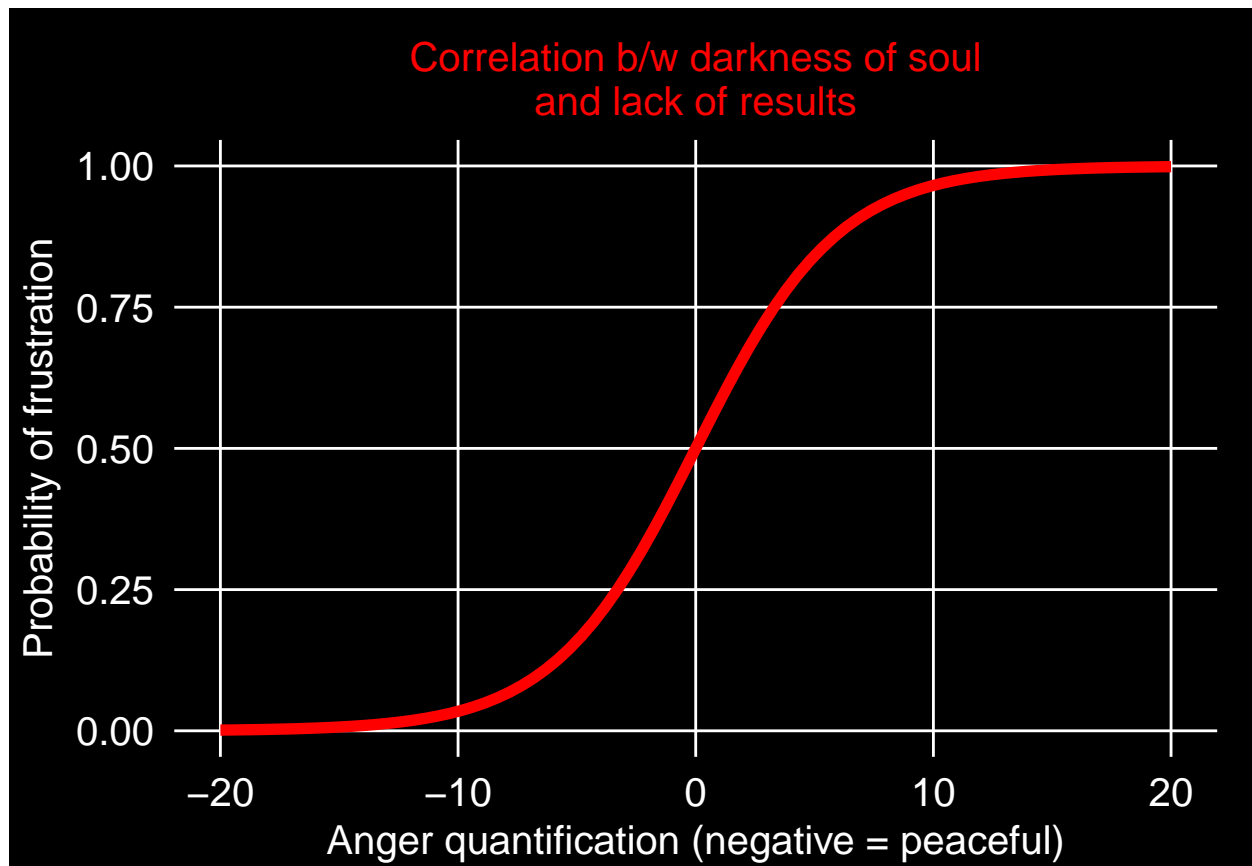
frustration.curve <- data.frame(points, growth.saturation)

goth.model <-
  ggplot(frustration.curve, aes(x = points, y = growth.saturation)) +
  geom_smooth(colour = 'red', size = 2,
             method = 'glm', family = 'binomial', se = FALSE) +
  red_black_theme() +
  labs(title = 'Correlation b/w darkness of soul\nand lack of results',
       x = 'Anger quantification (negative = peaceful)',
       y = 'Probability of frustration') +
  straight_goth()

goth.model

```

```
## Warning in loop_apply(n, do.ply): non-integer #successes in a binomial glm!
```



```
# saving ggplot objects
```

```
ggsave(filename = '~/GitHub/reproducible-research/Day-3/datasets/goth-eps.eps',  
        plot = goth.model, width = 10, height = 10, units = 'in', dpi = 600)
```

```
## Warning in loop_apply(n, do.ply): non-integer #successes in a binomial glm!
```

```
ggsave(filename = '~/GitHub/reproducible-research/Day-3/datasets/goth-pdf.pdf',  
        plot = goth.model, width = 10, height = 10, units = 'in', dpi = 1200)
```

```
## Warning in loop_apply(n, do.ply): non-integer #successes in a binomial glm!
```

```
ggsave(filename = '~/GitHub/reproducible-research/Day-3/datasets/goth-tiff.tiff',  
        plot = goth.model, width = 10, height = 10, units = 'in', dpi = 300)
```

```
## Warning in loop_apply(n, do.ply): non-integer #successes in a binomial glm!
```

```
ggsave(filename = '~/GitHub/reproducible-research/Day-3/datasets/goth-tex.tex',  
        plot = goth.model, width = 10, height = 10, units = 'in', dpi = 600)
```

```
## Warning in loop_apply(n, do.ply): non-integer #successes in a binomial glm!
```

```

# feltron style theme with large facets and bold headings

feltron_theme <- function() {

  color.background = '#36394A'
  color.grid.major = '#000000'
  color.axis.text = 'white'
  color.axis.title = 'white'
  color.title = 'white'

  # create base for plots
  theme_bw(base_size = 15) +

    # make background blue
    theme(panel.background = element_rect(fill = color.background,
                                           color = color.background)) +
    theme(plot.background = element_rect(fill = color.background,
                                           color = color.background)) +
    theme(panel.border = element_rect(color = color.background)) +

    # format grid
    theme(panel.grid.major = element_blank()) +
    theme(panel.grid.minor = element_blank()) +
    theme(axis.ticks = element_blank()) +

    # Format the legend
    theme(legend.position = "bottom") +
    theme(legend.background = element_rect(fill=color.background)) +
    theme(legend.text = element_text(size = 15, color=color.axis.title,
                                     face = 'bold')) +
    theme(legend.title = element_blank()) +

    # Set title and axis labels, and format these and tick marks
    theme(plot.title = element_text(color = color.title,
                                     size = 15, vjust = 1.25)) +
    theme(axis.text.x = element_text(size = 15,color = color.axis.text)) +
    theme(axis.text.y = element_text(size = 15,color = color.axis.text)) +
    theme(axis.title.x = element_text(size = 15,color = color.axis.title, vjust=0)) +
    theme(axis.title.y = element_text(size = 15,color = color.axis.title, vjust=1.25)) +

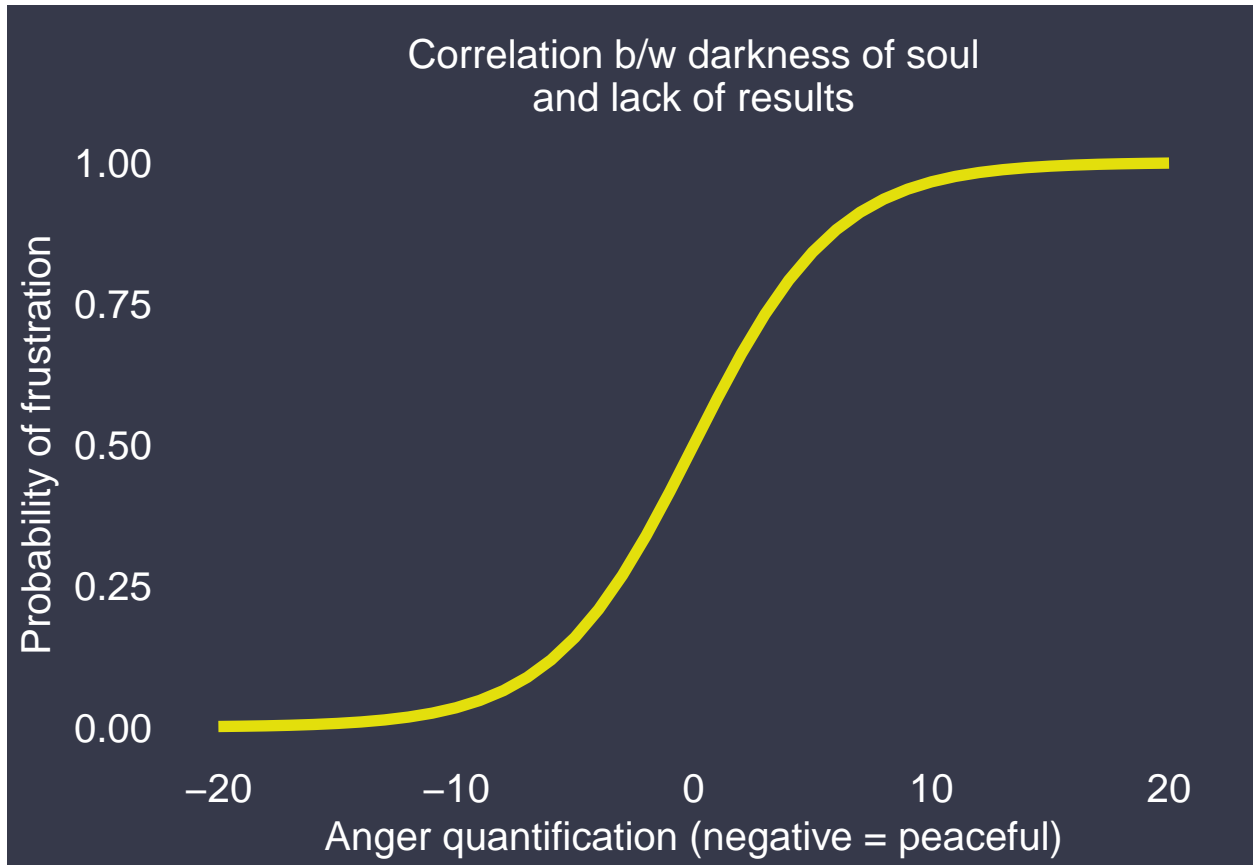
    # format the facet headings
    theme(strip.text = element_text(face = 'bold', size = rel(1.5),
                                     color = color.axis.text),
          strip.background = element_rect(fill = color.background,
                                           colour = color.background, size = 1))
}

feltron.anger <-
  ggplot(frustration.curve, aes(x = points, y = growth.saturation)) +
  geom_line(colour = '#E3DF0C', size = 2) +
  red_black_theme() +
  labs(title = 'Correlation b/w darkness of soul\and lack of results',
       x = 'Anger quantification (negative = peaceful)',

```



```
y = 'Probability of frustration') +  
feltron_theme()  
  
feltron.anger
```



```
ggsave(filename = '~/GitHub/reproducible-research/Day-3/datasets/feltron-anger-pdf.pdf',  
        plot = feltron.anger, width = 10, height = 10, units = 'in', dpi = 1200)  
  
sink()
```