world happiness - kaggle dataset

preeti swaminathan & patrick mcdevitt

20 may 2017

Happiness - à la Kaggle

```
setwd(home_dir)
setwd(data_dir)

hp_2015 <- read.csv("happiness_2015.csv", stringsAsFactors = FALSE)
setwd(home_dir)

names(hp_2015) <- tolower(names(hp_2015))

for (i in 2:(length(hp_2015)))
{
    if (class(hp_2015[,i]) == "character")
    {
        hp_2015[,i] <- factor (hp_2015[,i])
    }
}
```

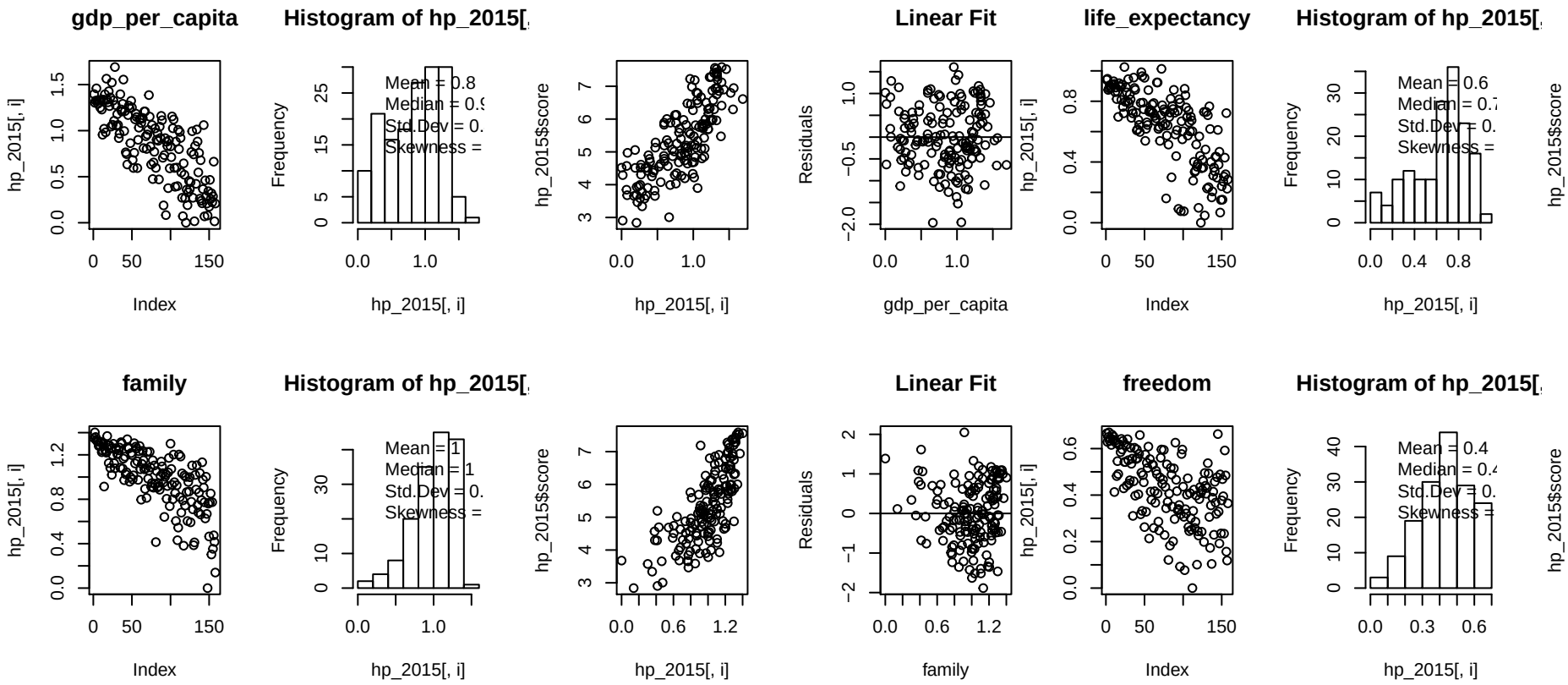
```
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ...   remove outliers ... more than 5 sigma from mean value
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    lst <- length(hp_2015) - 1  # sale price is (currently) last column

    for (i in 2 : lst)
    {
        if(class(hp_2015[,i]) == "integer" || class(hp_2015[,i]) == "numeric")
        {
            hp_2015[,i][which(scale(hp_2015[,i]) > 5)] <- NA
            hp_2015[,i][which(scale(hp_2015[,i]) < -5)] <- NA
        }
```

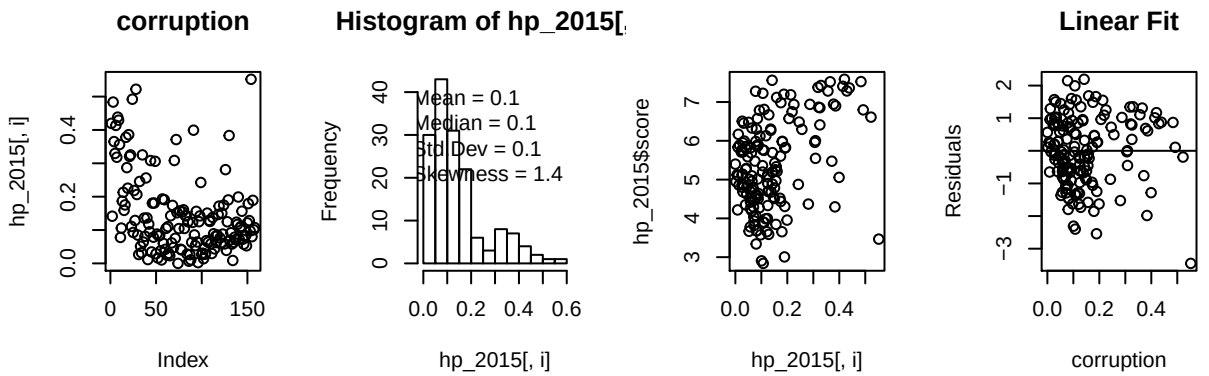
```
    }
```

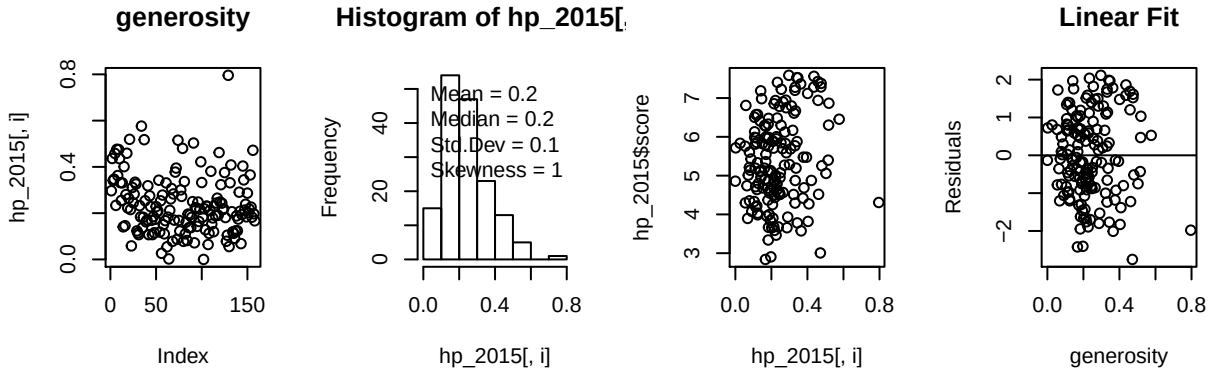
```
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ...   create a few new columns
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-


# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ...   scale each column independently
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

#   for (i in 2 : length(hp_2015))
#   {
#       if(class(hp_2015[,i]) == "integer" || class(hp_2015[,i]) == "numeric")
#       {
#           hp_2015[,i] <- scale(hp_2015[,i])
#       }
#   }
```

```
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ...   make some plots for numberic variables... linear, log_x, log_y, log_xy ...
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

#   pdf ("hp_2015_train_plots.pdf", width = 10, height = 7)

    par (mfrow = c (2, 4))
    for (i in 7 : (length(hp_2015)))
    {
        if(class(hp_2015[,i]) == "integer" || class(hp_2015[,i]) == "numeric" || class(hp_2015[,i]) == "matrix")
        {
            plot (hp_2015[,i], main = (names(hp_2015[i])))
            hist(hp_2015[,i])
            if (skewness(hp_2015[,i], na.rm = TRUE) < 0)
            {
                txt_pos <- "topleft"
            }
            else {
                txt_pos <- "topright"
            }
```

```r
            legend(txt_pos, legend = c(paste("Mean =", round(mean(hp_2015[,i], na.rm = TRUE), 1)),
                                paste("Median =",round(median(hp_2015[,i], na.rm = TRUE), 1)),
                                paste("Std.Dev =", round(sd(hp_2015[,i], na.rm = TRUE), 1)),
                                paste("Skewness =", round(skewness(hp_2015[,i], na.rm = TRUE), 1))),
                  bty = "n")


            plot(hp_2015$score  ~ hp_2015[,i])

# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ...   look at residuals from one-variable linear fit
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

            fit <- lm(hp_2015$score ~ hp_2015[,i])
            res <- resid(fit, na.action = na.exclude)
            plot (hp_2015[,i], res,
                  ylab = "Residuals",
                  xlab = (names(hp_2015[i])),
                  main = "Linear Fit")
            abline (0, 0)

        }
    }
```


gdp_per_capita
Histogram of hp_2015[
Linear Fit
life_expectancy
Histogram of hp_2015[
family
Histogram of hp_2015[
Linear Fit
freedom
Histogram of hp_2015[
hp_2015[, i]
Index
Frequency
Mean = 0.8
Median = 0.9
Std.Dev = 0.
Skewness =
hp_2015$score
Residuals
gdp_per_capita
family
Mean = 0.6
Median = 0.7
Mean = 1
Median = 1
Mean = 0.4
Median = 0.4






```
    for (i in 2:(length(hp_2015)))
    {
        if(class(hp_2015[,i]) == "factor")
        {
            plot_title <- names(hp_2015[i])

            p1 <- ggplot(hp_2015, aes(x = hp_2015[,i], fill = hp_2015[,i])) + geom_bar() + labs(title = plot_title )

            p2 <- ggplot(hp_2015, aes(x = hp_2015[,i], y = score, fill = hp_2015[,i])) + geom_boxplot() + labs(title = plot_title)

            print(p1)
            print(p2)
#           grid.arrange(p1, p2, ncol = 2, heights = 100, widths = c(100, 100))
```

```
#           p <- plot_grid(p1, p2, align='v', labels=c('', ''))

#           print(p)
        }
    }

#   dev.off()
```