EDA

March 27, 2019

library packages

```r
library(caret)
theme1 <- trellis.par.get()
theme1$plot.symbol$col <- rgb(.2, .4, .2, .2)
theme1$plot.symbol$pch <- 16
theme1$plot.line$col <- rgb(.8, .1, .1, 1)
theme1$plot.line$lwd <- 2
theme1$strip.background$col <- rgb(.0, .2, .6, .2)
trellis.par.set(theme1)

library(tidyverse)
library(patchwork)
```

Data cleaning

```r
fifa = read_csv("..\\Data\\CompleteDataset.csv")
```

```r
calc_expression = function(fmula) {
    eval(parse(text = fmula))
}

data = fifa %>%
    janitor::clean_names() %>%
    select(-c(x1, name, photo, flag, club_logo, wage, overall, id)) %>%
    mutate(value = str_replace(value, "K", "/ 1000"),
           value = str_replace(value, "M", ""),
           value = str_replace(value, "€", "")) %>%
    mutate(value = map(value, calc_expression),
           value = as.numeric(value)) %>%
    mutate(value = readr::parse_number(value)) %>%
    filter(preferred_positions == "GK") %>%
    select(-(cam:st))

trans2int_cols = c(7:40)
trans2fct_cols = c(2, 4)
data[trans2int_cols] = map(data[trans2int_cols], as.integer)
data[trans2fct_cols] = map(data[trans2fct_cols], as.factor)

data = data %>%
    mutate(nationality = as.character(nationality))

#combining different nations on the same continent into a new variable
```

```r
#which has fewer categories

eu = c("Germany", "Spain", "Italy", "Belgium", "Slovenia", "France",
       "Czech Republic", "Croatia", "Switzerland", "Portugal",
       "Denmark", "Poland", "Greece", "Bosnia Herzegovina", "England",
       "Norway", "Netherlands", "Finland", "Russia", "Turkey", "Ukraine",
       "Romania", "Albania", "Hungary", "Lithuania",
       "Republic of Ireland", "Austria", "Sweden", "Wales", "Scotland",
       "Bulgaria", "Serbia", "Georgia", "Kosovo", "Slovakia", "Latvia",
       "Belarus", "FYR Macedonia", "Northern Ireland", "Iceland",
       "Luxembourg", "Montenegro", "Israel", "San Marino")
as = c("China PR", "Korea Republic", "Japan", "Oman", "Saudi Arabia",
       "Egypt", "Iran","Philippines", "India", "Lebanon", "Senegal",
       "Morocco",  "Comoros", "Nigeria", "Algeria", "Ivory Coast",
       "Ghana", "DR Congo", "Benin", "Kenya", "Equatorial Guinea",
       "Gabon", " Burkina Faso", "Congo", "Tunisia", "Cape Verde", "Angola")
af = c("Cameroon", "South Africa", "Mali")
na = c("United States", "Guatemala", "Canada", "Puerto Rico",
       "Haiti", "Bermuda")
sa = c("Costa Rica", "Argentina", "Brazil", "Uruguay", "Chile",
       "Colombia", "Mexico", "Venezuela", "Curacao", "Peru",
       "Paraguay", "Ecuador", "Bolivia")
oc = c("Australia", "New Zealand")

nation_eu = function(name){
  if (name %in% eu)
      name_new = "eu"
  else if (name %in% as)
      name_new = "as"
  else if (name %in% na)
      name_new = "na"
  else if (name %in% sa)
      name_new = "sa"
  else if (name %in% oc)
      name_new = "oc"
  else name_new = "af"
  name_new
}


data = data %>%
    mutate(nationality = map(.x = nationality, ~nation_eu(.x))) %>%
    mutate(nationality = as.factor(unlist(nationality)))
```

Split the data set into training and testing data

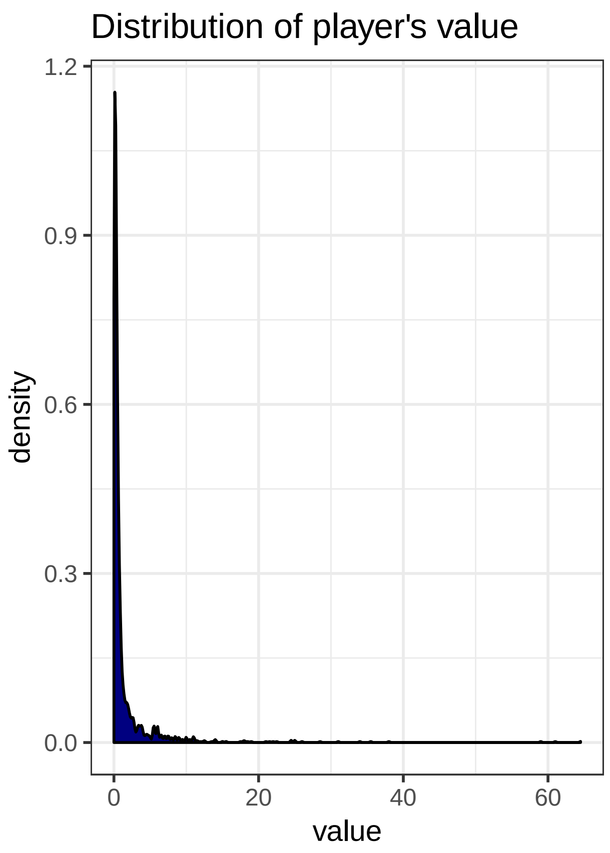
```r
set.seed(2)
trRows = createDataPartition(data$value, p = .75, list = FALSE)
data_split = data %>%
    select(-club)
```

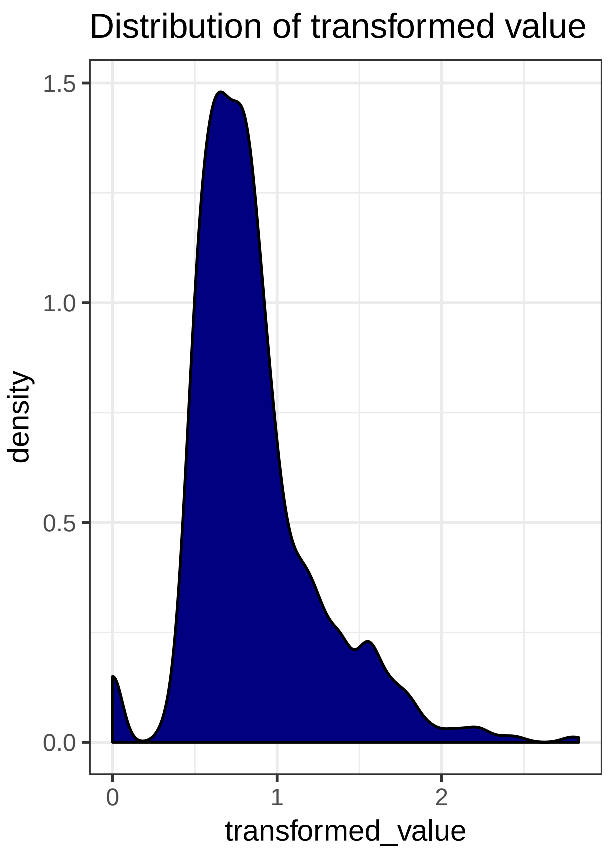
```
train = data_split[trRows,]
test = data_split[-trRows,]
```

Variable transformation

Transform the response(value), based on the distribution of the response in the training dataset

```
#The effect of log transformation is not ideal
#data = data %>%
#   mutate(value = log(value + 2)

p1 = train %>%
    ggplot(aes(x = value)) + geom_density(fill = "navy") + theme_bw() +
    labs(title = "Distribution of player's value")

train = train %>%
    mutate(value = value^(1/4)) %>%
    rename("transformed_value" = value)

test = test %>%
    mutate(value = value^(1/4)) %>%
    rename("transformed_value" = value)

train %>% write_csv("..\\exploratory analysis\\train.csv")
test %>% write_csv("..\\exploratory analysis\\test.csv")

data = train

p2 = data %>%
    ggplot(aes(x = transformed_value)) + geom_density(fill = "navy") +
    theme_bw() +
    labs(title = "Distribution of transformed value")

p1 + p2
```

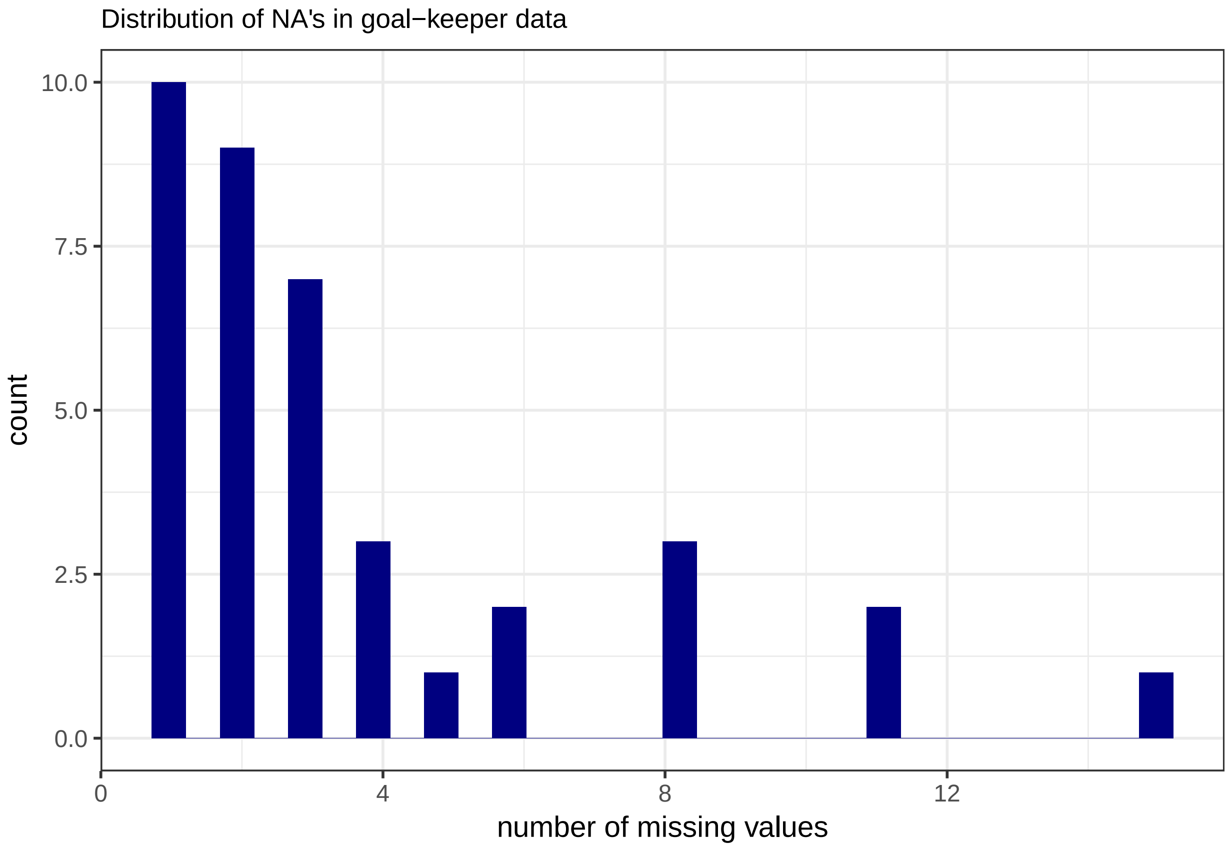




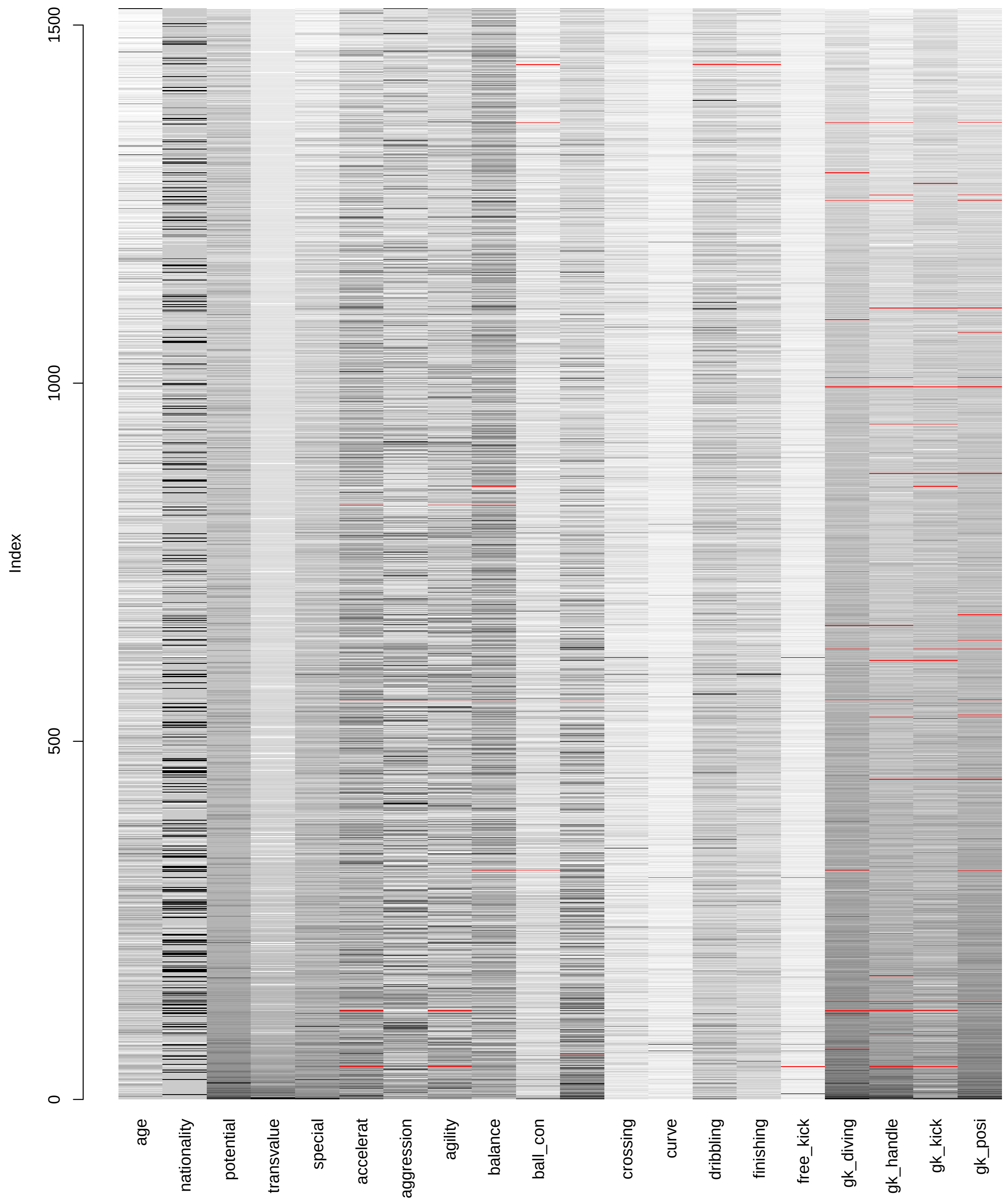
Checking NA's

NA's in each observation

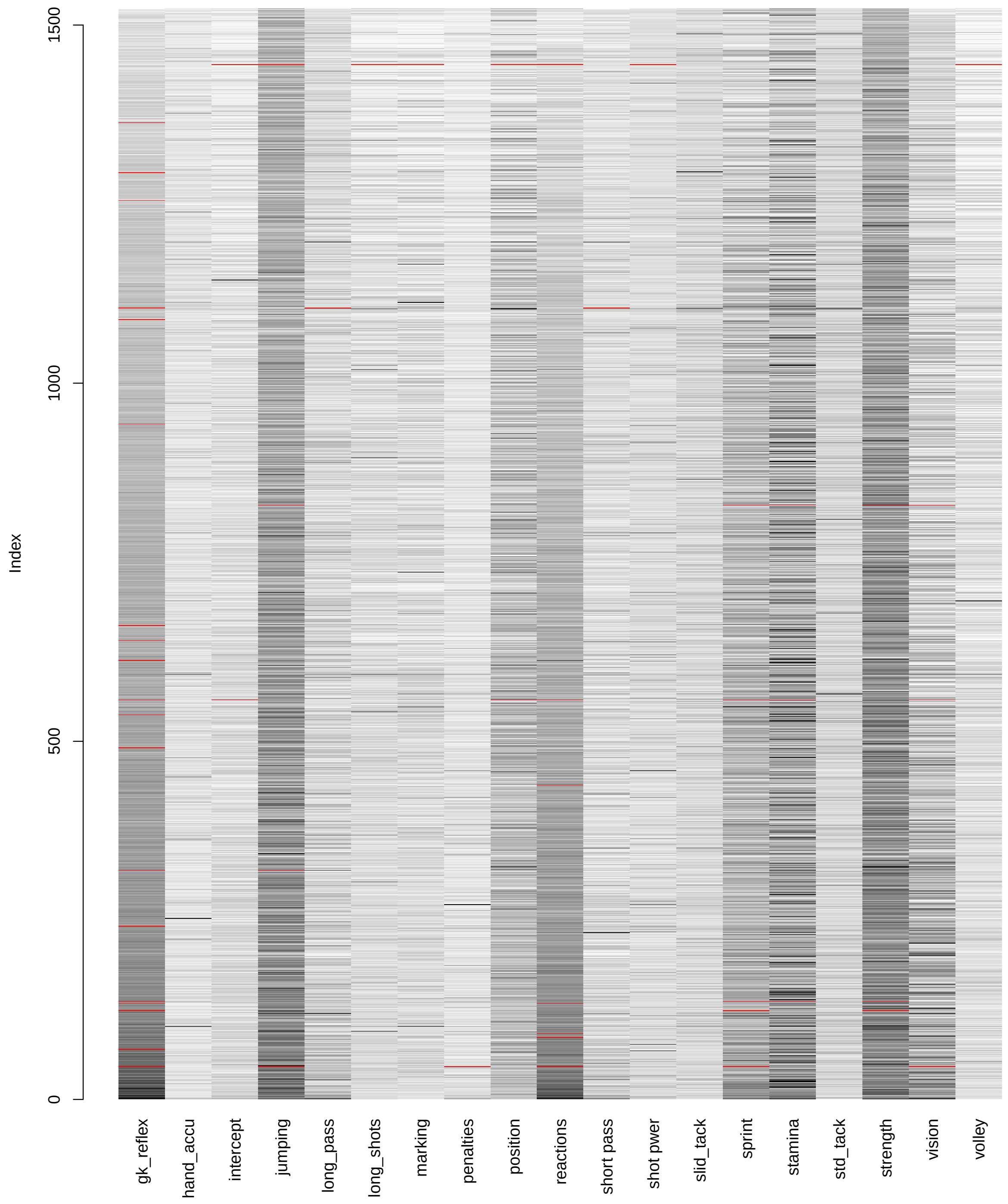
```
data = data %>%
    mutate(., na_count = apply(., 1, function(x) sum(is.na(x))))

data %>%
    filter(na_count > 0) %>%
    ggplot(aes(x = na_count)) + geom_histogram(fill = "navy") + theme_bw() +
    labs(title = "Distribution of NA's in goal-keeper data",
         x = "number of missing values") +
    theme(plot.title = element_text(size = 10))
```



```
library(VIM)
#In the following plots, red color represents missing data

matrixplot(data[,1:20],
           labels = c("age","nationality","potential","transvalue","special",
                    "accelerat","aggression","agility","balance","ball_con",
                    "composure","crossing","curve","dribbling","finishing",
                    "free_kick","gk_diving","gk_handle","gk_kick","gk_posi"))
```

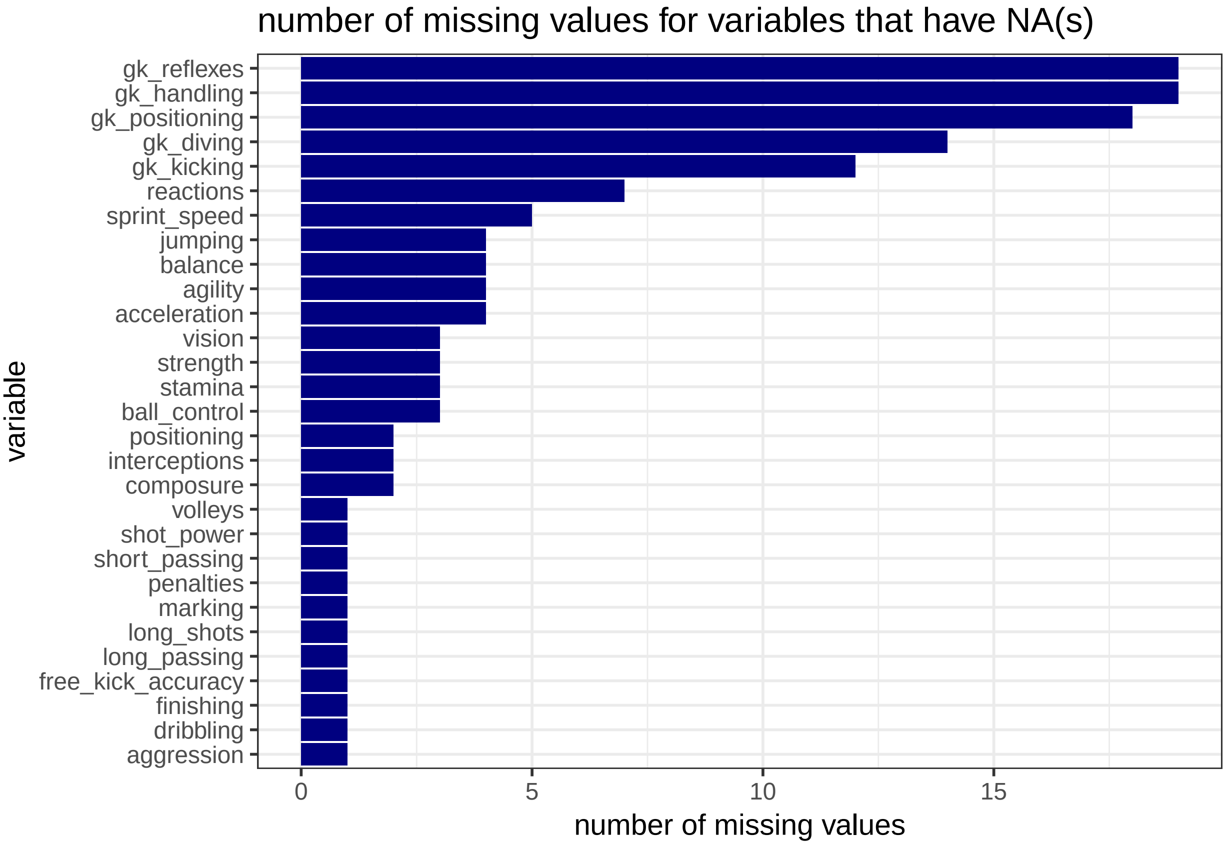


```
matrixplot(data[,21:39],
           labels = c("gk_reflex", "hand_accu", "intercept", "jumping",
                      "long_pass", "long_shots", "marking", "penalties",
                      "position", "reactions", "short pass", "shot pwer",
                      "slid_tack", "sprint", "stamina", "std_tack",
                      "strength", "vision", "volley"))
```



NA's in each variable

```
na_col = colSums(is.na(data)) %>%
    as.list() %>%
    as.data.frame() %>%
    select(-na_count) %>%
    gather(age:volleys, key = "variable", value = "num_of_na")

na_col %>%
    filter(num_of_na > 0) %>%
    mutate(variable = fct_reorder(variable, num_of_na)) %>%
    ggplot(aes(x = variable, y = num_of_na)) +
  geom_col(fill = "navy") +
  theme(legend.position = "bottom") +
  labs(title = "number of missing values for variables that have NA(s)",
       y = "number of missing values") +
    coord_flip() + theme(axis.text.x = element_text(face = "plain",
                                                   color = "black",
                                                   size = 8)) +
    theme_bw()
```



Tables for descriptive statistics

```
descrip_list = data %>%
    skimr::skim_to_list()

descrip_list[[1]] %>%
    select(variable, n_unique, missing) %>%
    dplyr::rename("unique levels" = n_unique) %>%
    knitr::kable(caption = "Factor variables")
```

Table 1: Factor variables

variable	unique levels	missing
nationality	6	0

```
bind_rows(descrip_list[[2]] , descrip_list[[3]]) %>%
    dplyr::select(variable,
                Min = p0,
                `1st Q` = p25,
                Mean = mean,
                Median = p50,
                `3rd Q` = p75,
                Max = p100,
                `Std Dev` = sd,
                missing) %>%
  knitr::kable(digits = 3, caption = "Integer/numeric variables")
```

Table 2: Integer/numeric variables

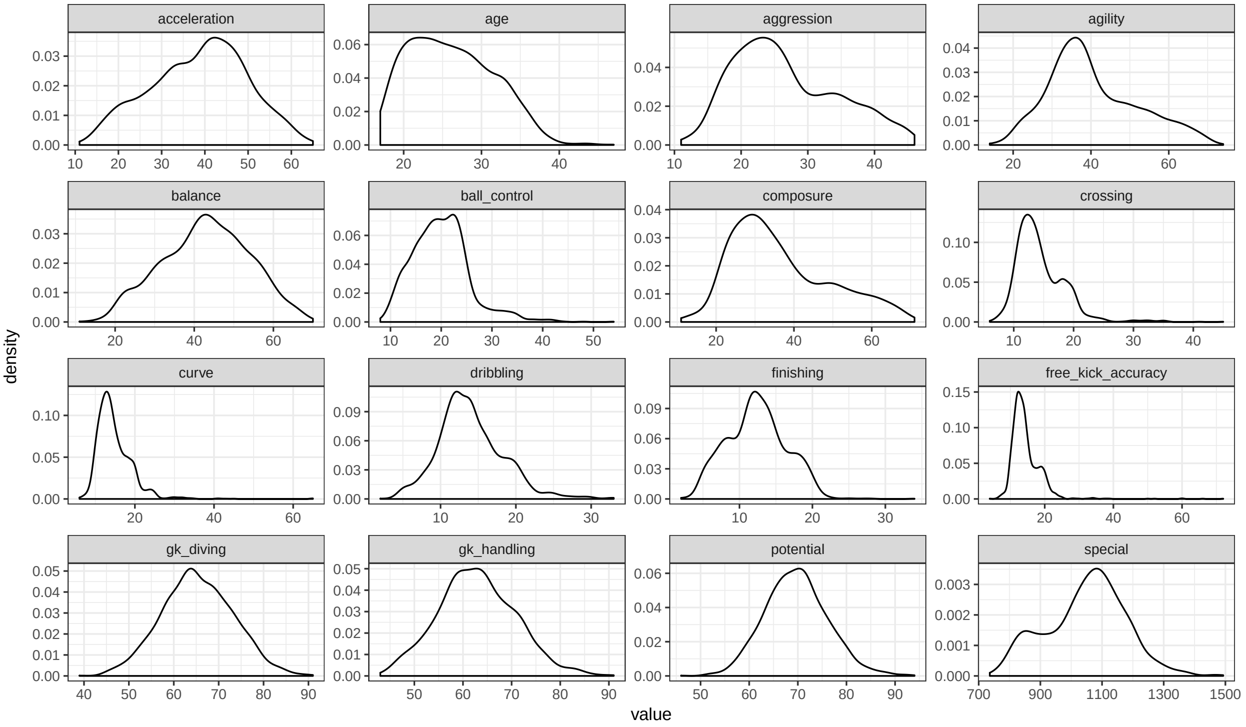
variable	Min	1st Q	Mean	Median	3rd Q	Max	Std Dev	missing
acceleration	11	31	38.65	40	47	65	11.07	4
age	17	22	26.12	26	30	47	5.41	0
aggression	11	21	26.75	25	33	46	7.86	1
agility	14	33	40.47	38	48	74	11.55	4
balance	11	35	43.03	43	51	70	10.94	4
ball_control	8	16	19.96	20	23	54	5.7	3
composure	11	27	36.59	33	45	71	12.72	2
crossing	6	12	14.43	13	17	45	4.1	0
curve	6	12	14.81	14	17	65	4.63	0
dribbling	2	11	14.01	14	16	33	4.38	1
finishing	2	9	12.37	12	15	34	4.06	1
free_kick_accuracy	4	12	14.51	14	16	72	4.73	1
gk_diving	39	60	65.32	65	71	91	7.95	14
gk_handling	43	58	62.88	63	68	91	7.96	19
gk_kicking	35	56	61.57	61	67	95	7.98	12
gk_positioning	38	57	63.08	63	69	91	8.78	18
gk_reflexes	37	60	66.17	66	72	90	8.43	19
heading_accuracy	4	12	14.58	14	17	47	4.18	0
interceptions	4	13	17.47	18	22	55	5.86	2
jumping	13	52	58.24	59	66	85	11.45	4
long_passing	7	20	25.47	24	30	62	7.72	1
long_shots	3	10	13.12	13	16	40	4.48	1
marking	4	10	12.67	13	15	35	4.31	1
na_count	0	0	0.091	0	0	15	0.77	0
penalties	5	15	20.4	20	24	73	6.83	1
positioning	2	8	11.61	12	15	24	4.16	2
potential	46	65	69.68	70	74	94	6.49	0

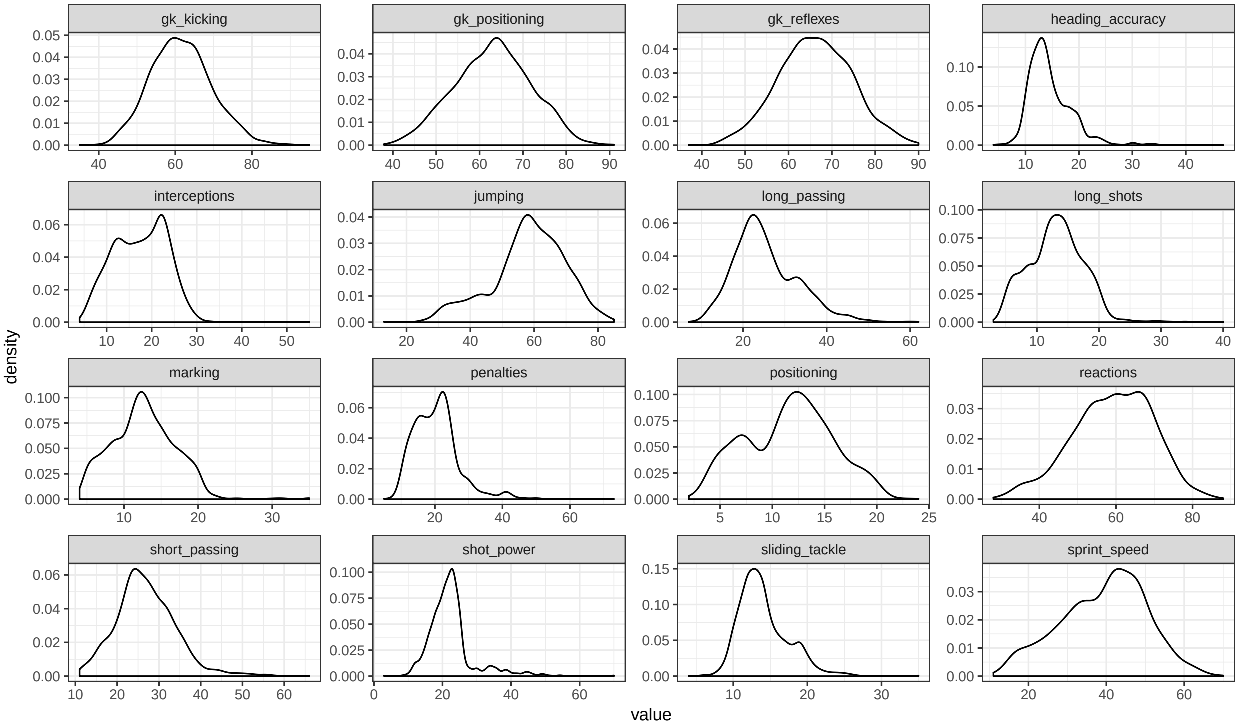
variable	Min	1st Q	Mean	Median	3rd Q	Max	Std Dev	missing
reactions	28	52	59.16	60	67	88	10.36	7
short_passing	11	22	26.93	26	31	66	7.22	1
shot_power	3	19	22.52	22	24	70	6.99	1
sliding_tackle	4	12	14.1	13	16	35	3.43	0
special	736	966	1048.4	1061	1133	1493	127.75	0
sprint_speed	11	32	39.25	41	47	70	10.98	5
stamina	12	25	30.51	30	36	45	7.51	3
standing_tackle	4	12	14.17	14	16	34	3.46	0
strength	12	54	61.09	62	69	85	11.26	3
vision	10	27	35.92	35	45	72	12.81	3
volleys	4	10	12.9	13	16	40	4.55	1
transformed_value	0	0.61	0.86	0.78	0.99	2.83	0.39	0

Figures for descriptive statistics

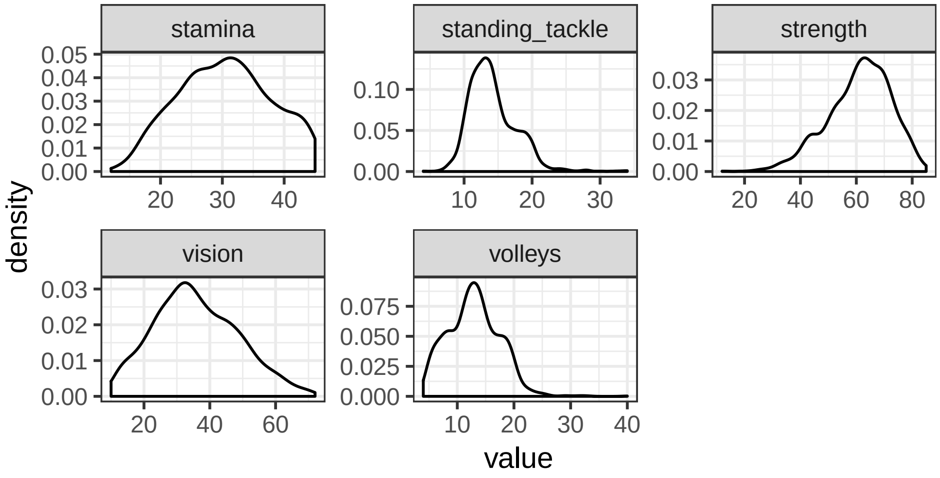
Check the distribution for each numeric/integer predictor

```
#library(gridExtra)

p1 = data[,1:18] %>%
    select(-transformed_value) %>%
  keep(is.numeric) %>%                     # Keep only numeric columns
  gather() %>%                             # Convert to key-value pairs
  ggplot(aes(value)) +                     # Plot the values
    facet_wrap(~ key, scales = "free") +   # In separate panels
    geom_density() +
    theme_bw()


p2 = data[,19:34] %>%
  keep(is.numeric) %>%                     # Keep only numeric columns
  gather() %>%                             # Convert to key-value pairs
  ggplot(aes(value)) +                     # Plot the values
    facet_wrap(~ key, scales = "free") +   # In separate panels
    geom_density() +
    theme_bw()

p1/p2
```


acceleration
age
aggression
agility
balance
ball_control
composure
crossing
curve
dribbling
finishing
free_kick_accuracy
gk_diving
gk_handling
potential
special
density
value
0.00 0.01 0.02 0.03
0.00 0.02 0.04 0.06
0.00 0.02 0.04
0.00 0.01 0.02 0.03 0.04
0.00 0.05 0.10
0.00 0.03 0.06 0.09
0.00 0.05 0.10 0.15
0.00 0.01 0.02 0.03 0.04 0.05
0.000 0.001 0.002 0.003
10 20 30 40 50 60
20 30 40
20 40 60
10 20 30 40 50
10 20 30 40
50 60 70 80 90
700 900 1100 1300 1500


gk_kicking
gk_positioning
gk_reflexes
heading_accuracy
interceptions
jumping
long_passing
long_shots
marking
penalties
positioning
reactions
short_passing
shot_power
sliding_tackle
sprint_speed
density
value
0.00 0.01 0.02 0.03 0.04 0.05
0.00 0.02 0.04 0.06
0.00 0.01 0.02 0.03 0.04
0.00 0.05 0.10 0.15
0.000 0.025 0.050 0.075 0.100
40 60 80
40 50 60 70 80 90
10 20 30 40 50
5 10 15 20 25
0 20 40 60
10 20 30 40 50 60

```
data[,35:39] %>%
  keep(is.numeric) %>%                     # Keep only numeric columns
  gather() %>%                             # Convert to key-value pairs
  ggplot(aes(value)) +                     # Plot the values
    facet_wrap(~ key, scales = "free") +   # In separate panels
    geom_density()  +
    theme_bw()
```



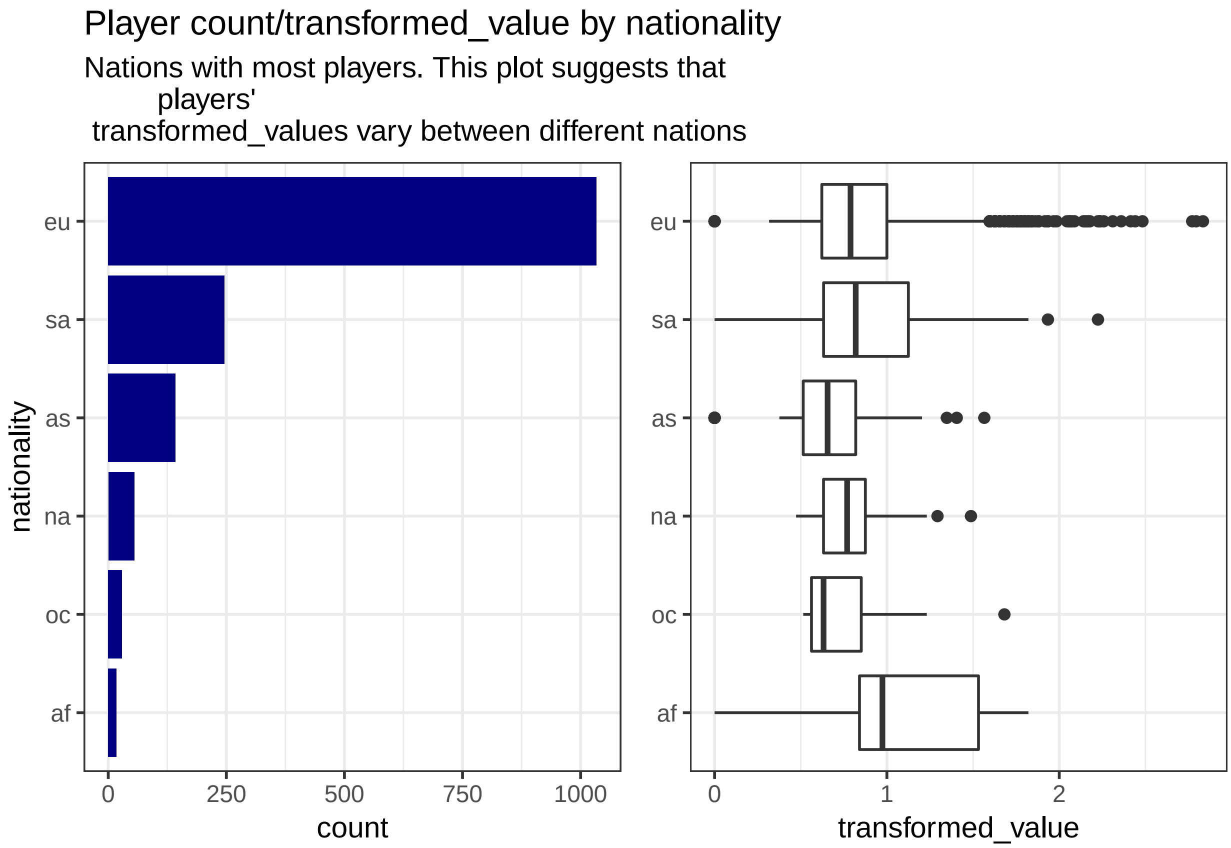
plot for variables

for factor variables

- nationality

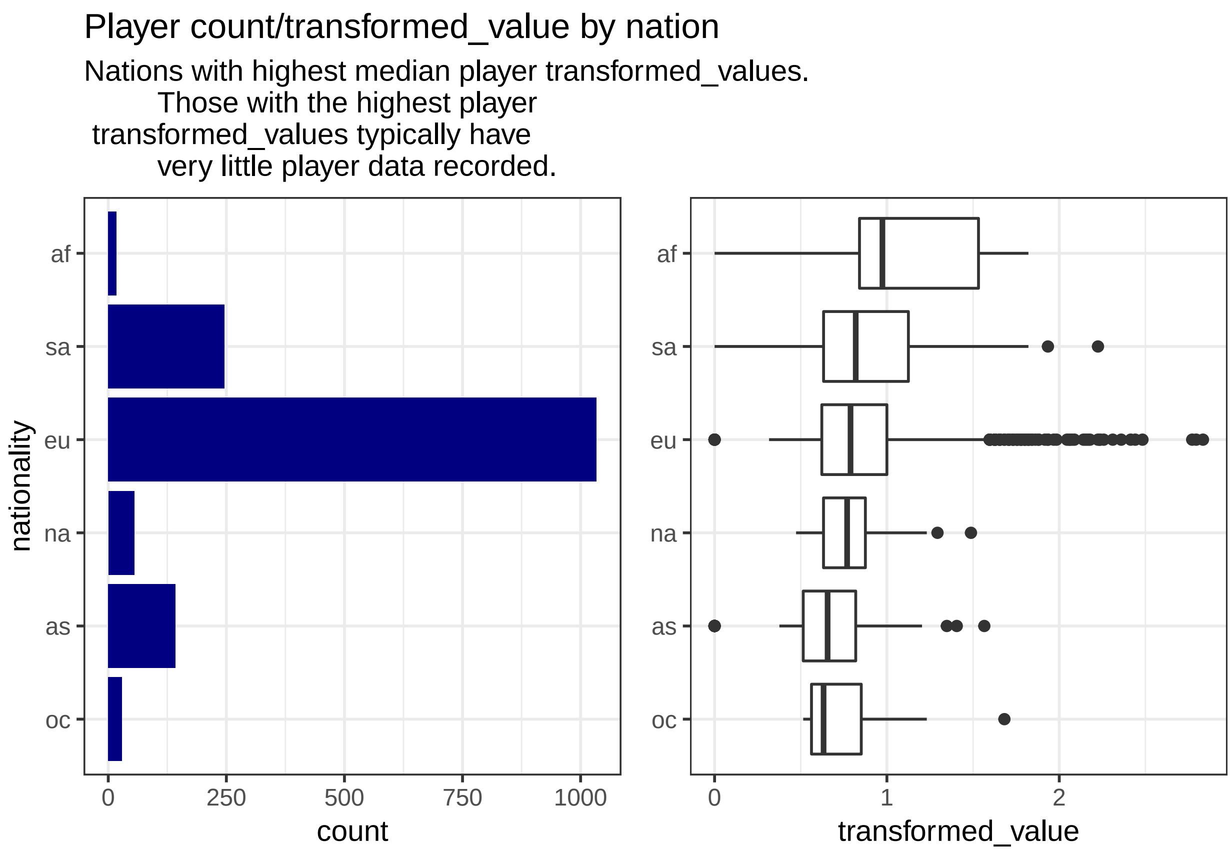
plot the situation for nations with the most number of players:

```
nation_box = data %>%
    mutate(nationality = fct_lump(nationality, 12)) %>%
    mutate(nationality = fct_infreq(nationality)) %>%
    mutate(nationality = fct_rev(nationality)) %>%
    #move "Other" level to the last:
    mutate(nationality = fct_relevel(nationality, "Other", after = 0)) %>%
    ggplot(aes(x = nationality, y = transformed_value)) +
geom_boxplot() +
theme(legend.position = "bottom") +
labs( x = NULL) +
coord_flip() +
    theme(axis.text.x = element_text(face = "plain",
                                     color = "black",
                                     size = 8)) +
    theme_bw()

nation_hist = data %>%
    #If focus on most common nations:
    mutate(nationality = fct_lump(nationality, 12)) %>%
    #nations that have fewer players will be denoted as "Other"
    mutate(nationality = fct_infreq(nationality)) %>%
    mutate(nationality = fct_rev(nationality)) %>%
    #move "Other" level to the last:
    mutate(nationality = fct_relevel(nationality, "Other", after = 0)) %>%
    ggplot(aes(x = nationality)) +
    geom_bar(fill = "navy") +
    theme(legend.position = "bottom") +
    labs(title = "Player count/transformed_value by nationality",
         subtitle = "Nations with most players. This plot suggests that
         players' \n transformed_values vary between different nations") +
    coord_flip() +
    theme(axis.text.x = element_text(face = "plain",
                                     color = "black",
                                     size = 8))  +
    theme_bw()

nation_hist + nation_box
```



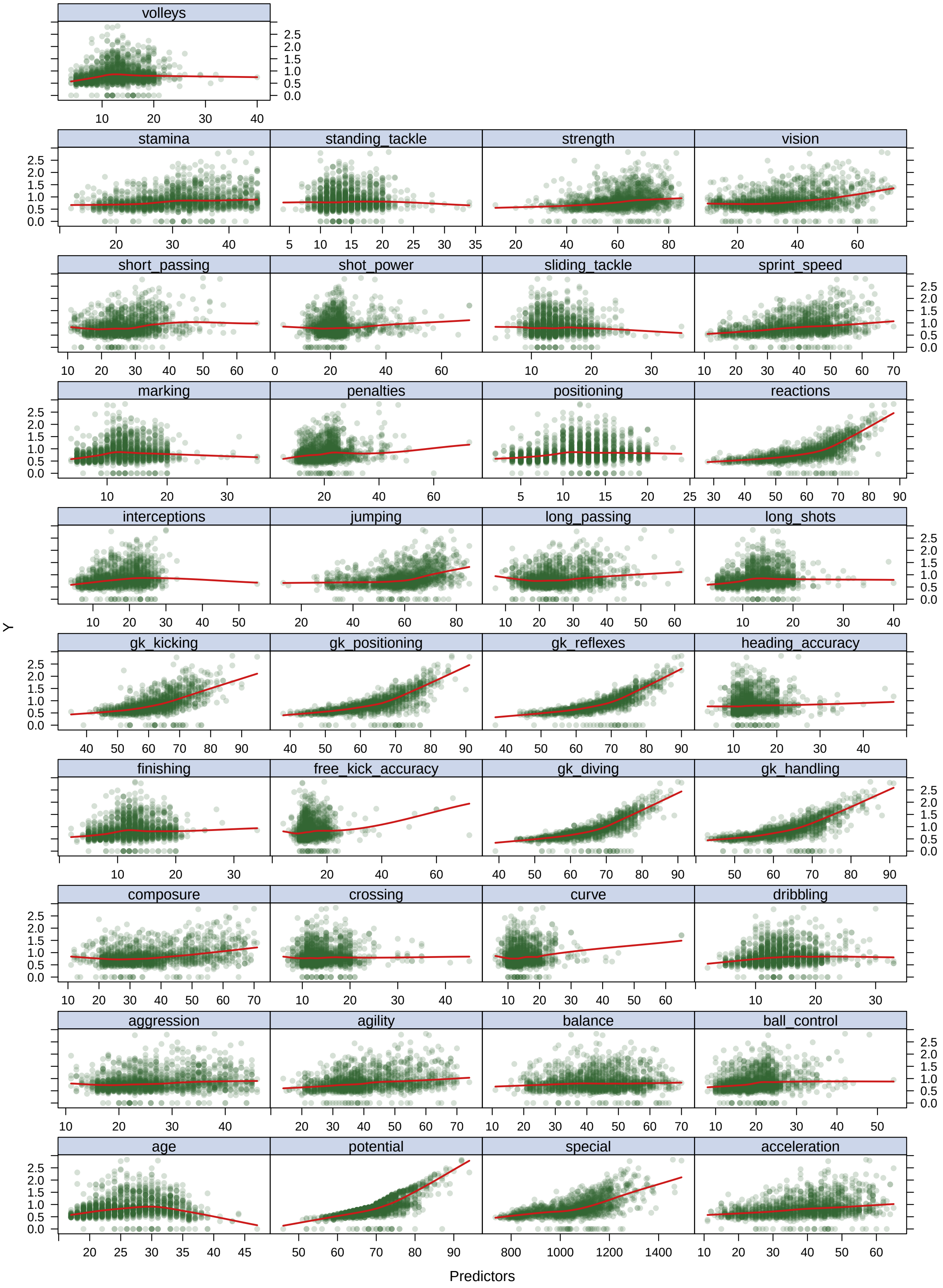
plot the nations with highest players' transformed_values:

```
nation_box = data %>%
    group_by(nationality) %>%
    mutate(med_by_nation = median(transformed_value)) %>%
    ungroup() %>%
    mutate(nationality = fct_reorder(nationality, med_by_nation)) %>%
    ggplot(aes(x = nationality, y = transformed_value)) +
    geom_boxplot() +
    theme(legend.position = "bottom") +
    labs( x = NULL) +
    coord_flip() +
    theme(axis.text.x = element_text(face = "plain",
                                     color = "black",
                                     size = 8)) +
    theme_bw()

nation_hist = data %>%
    group_by(nationality) %>%
    mutate(med_by_nation = median(transformed_value)) %>%
    ungroup() %>%
    mutate(nationality = fct_reorder(nationality, med_by_nation)) %>%
    ggplot(aes(x = nationality)) +
    geom_bar(fill = "navy") +
    theme(legend.position = "bottom") +
    labs(title = "Player count/transformed_value by nation",
         subtitle = "Nations with highest median player transformed_values.
         Those with the highest player\n transformed_values typically have
         very little player data recorded.") +
    coord_flip() +
    theme(axis.text.x = element_text(face = "plain",
                                     color = "black",
                                     size = 8))  +
    theme_bw()

nation_hist + nation_box
```



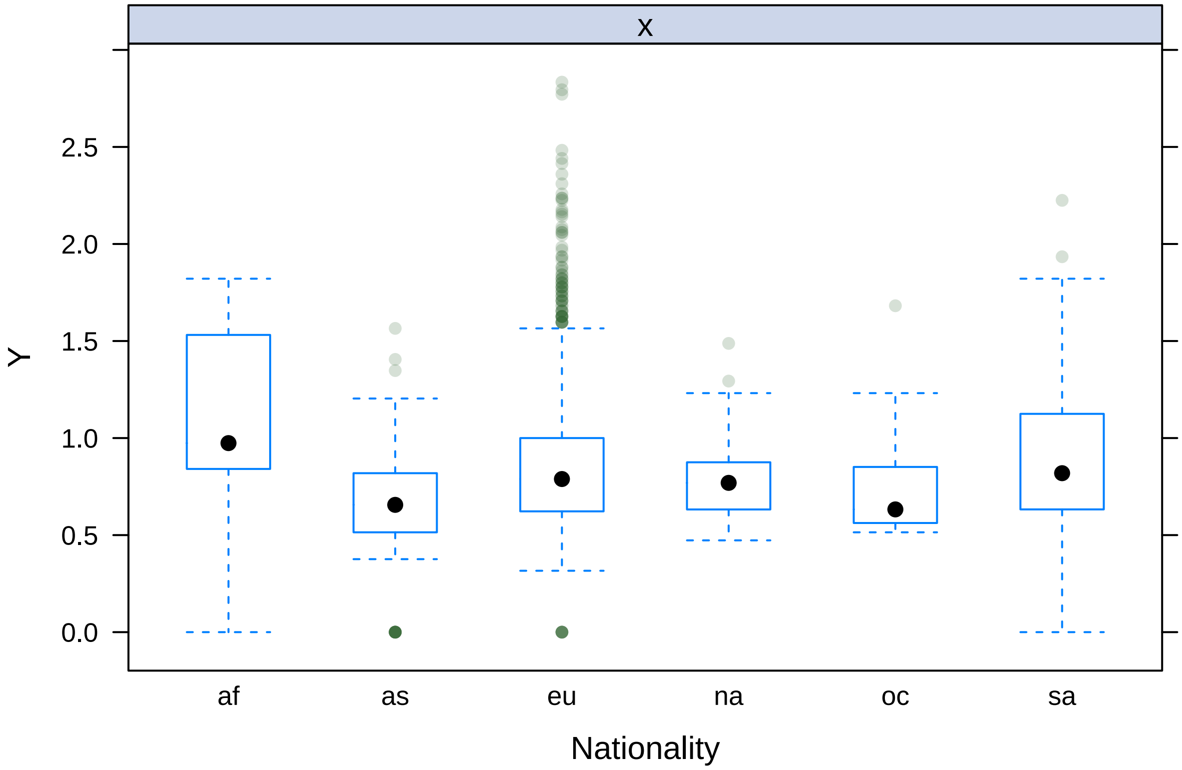
for int/num variables

```
# matrix of predictors

data_num  = data %>%
    keep(is.numeric) %>%
    select(-na_count, -transformed_value)

#for factor variables
data_fct  = data %>%
    select(-transformed_value) %>%
    select_if(~ is.factor(.))


# vector of response
y <- data$transformed_value

featurePlot(data_num,
y,
plot = "scatter",
span = .5,
labels = c("Predictors","Y"),
type = c("p", "smooth"),
layout = c(4, 10))
```


volleys
10 20 30 40
0.0 0.5 1.0 1.5 2.0 2.5
stamina
standing_tackle
strength
vision
20 30 40
5 10 15 20 25 30 35
20 40 60 80
20 40 60
short_passing
shot_power
sliding_tackle
sprint_speed
10 20 30 40 50 60
0 20 40 60
10 20 30
10 20 30 40 50 60 70
marking
penalties
positioning
reactions
10 20 30
20 40 60
5 10 15 20 25
30 40 50 60 70 80 90
interceptions
jumping
long_passing
long_shots
10 20 30 40 50
20 40 60 80
10 20 30 40 50 60
10 20 30 40
Y
gk_kicking
gk_positioning
gk_reflexes
heading_accuracy
40 50 60 70 80 90
40 50 60 70 80 90
40 50 60 70 80 90
10 20 30 40
finishing
free_kick_accuracy
gk_diving
gk_handling
10 20 30
20 40 60
40 50 60 70 80 90
50 60 70 80 90
composure
crossing
curve
dribbling
10 20 30 40 50 60 70
10 20 30 40
10 20 30 40 50 60
10 20 30
aggression
agility
balance
ball_control
10 20 30 40
20 30 40 50 60 70
10 20 30 40 50 60 70
10 20 30 40 50
age
potential
special
acceleration
20 25 30 35 40 45
50 60 70 80 90
800 1000 1200 1400
10 20 30 40 50 60
Predictors

FeaturePlots for factor variables

```
featurePlot(data$transformed_value, data_fct$nationality, "box", labels = c("Nationality","Y"))
```

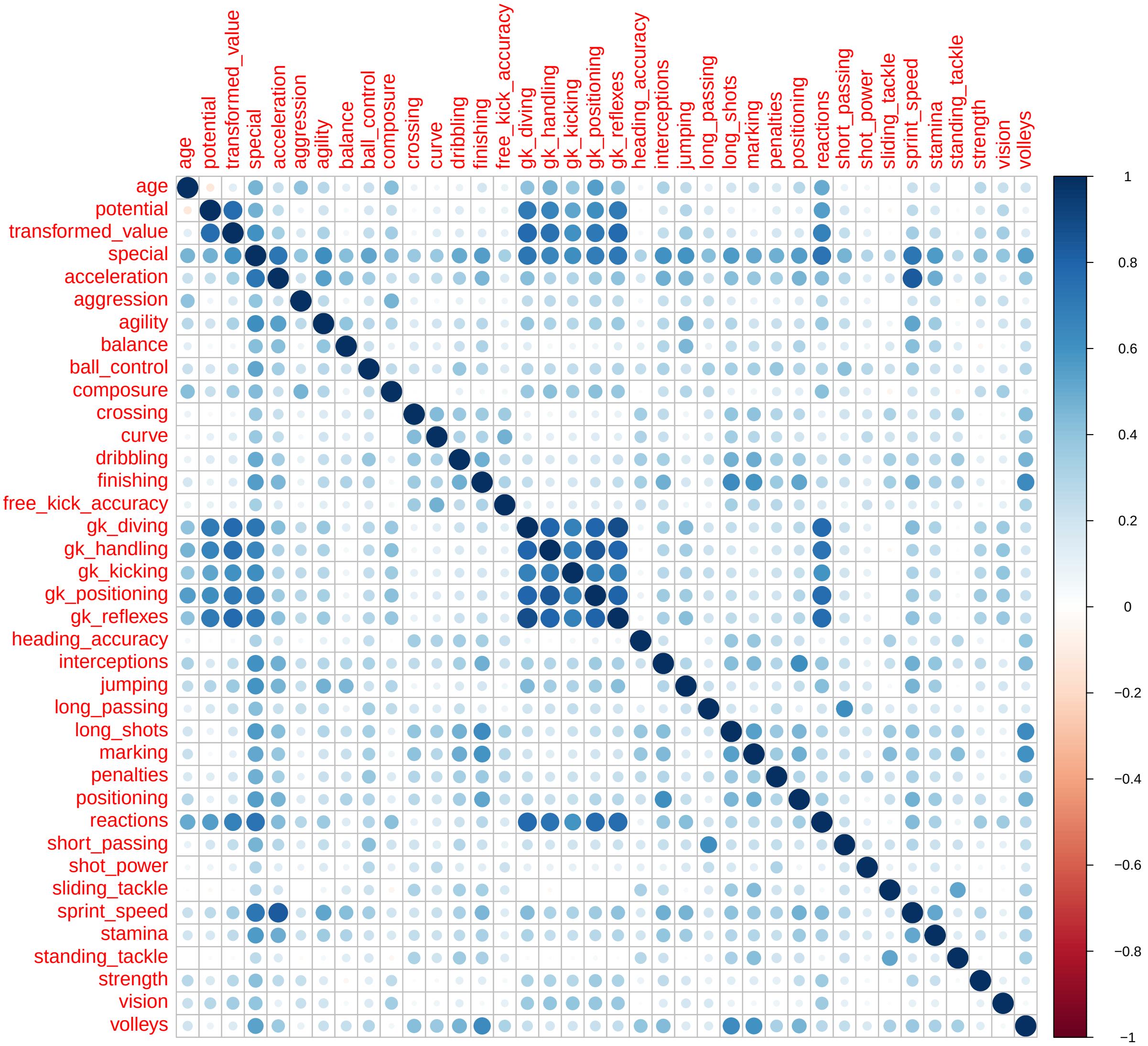


```
featurePlot(data$transformed_value, data_fct$club, "box", labels = c("Club","Y"))
```

```
## NULL
```

Correlation plot

```
library(corrplot)

cor_data = data %>%
    filter(na_count == 0) %>%
    select(-nationality, -na_count)

corrplot(cor(cor_data), tl.cex = 1.2)
```


age
potential
transformed_value
special
acceleration
aggression
agility
balance
ball_control
composure
crossing
curve
dribbling
finishing
free_kick_accuracy
gk_diving
gk_handling
gk_kicking
gk_positioning
gk_reflexes
heading_accuracy
interceptions
jumping
long_passing
long_shots
marking
penalties
positioning
reactions
short_passing
shot_power
sliding_tackle
sprint_speed
stamina
standing_tackle
strength
vision
volleys
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1