

Ярылгасимов Султан
Домашнее задание №3

Задание 1. Выполнить задания из комментариев к занятию (см. приложенный R-файл).

1.1. Изменяя размер выборки, проверьте, как изменяется качество подгонки гистограммы

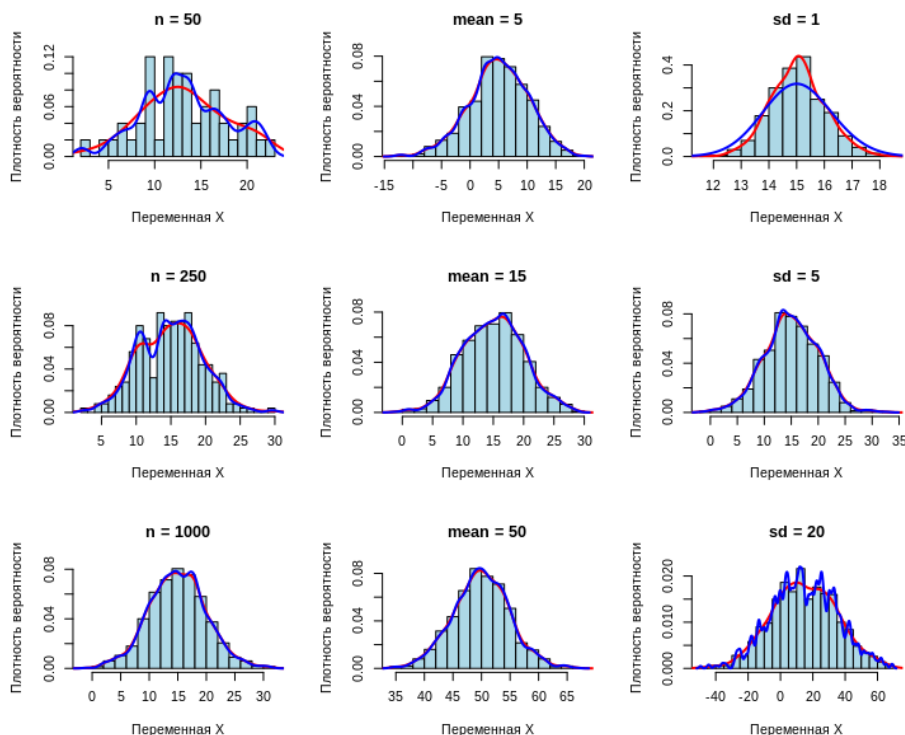
Код:

```
layout(matrix(1:9, ncol = 3, nrow=3))
for (i in c(50, 250, 1000)) {
  X <- rnorm(n = i, mean = 15, sd = 5)
  hist(X, breaks = 20, freq = FALSE, col = "lightblue",
       xlab = "Переменная X",
       ylab = "Плотность вероятности",
       main = paste("n =", i))
  lines(density(X), col = "red", lwd = 2)
  lines(density(X, bw = 0.8), col = "blue", lwd = 2)
}

for (i in c(5, 15, 50)) {
  X <- rnorm(n = 1000, mean = i, sd = 5)
  hist(X, breaks = 20, freq = FALSE, col = "lightblue",
       xlab = "Переменная X",
       ylab = "Плотность вероятности",
       main = paste("mean =", i))
  lines(density(X), col = "red", lwd = 2)
  lines(density(X, bw = 0.8), col = "blue", lwd = 2)
}

for (i in c(1, 5, 20)) {
  X <- rnorm(n = 1000, mean = 15, sd = i)
  hist(X, breaks = 20, freq = FALSE, col = "lightblue",
       xlab = "Переменная X",
       ylab = "Плотность вероятности",
       main = paste("sd =", i))
  lines(density(X), col = "red", lwd = 2)
  lines(density(X, bw = 0.8), col = "blue", lwd = 2)
}
```

Вывод:



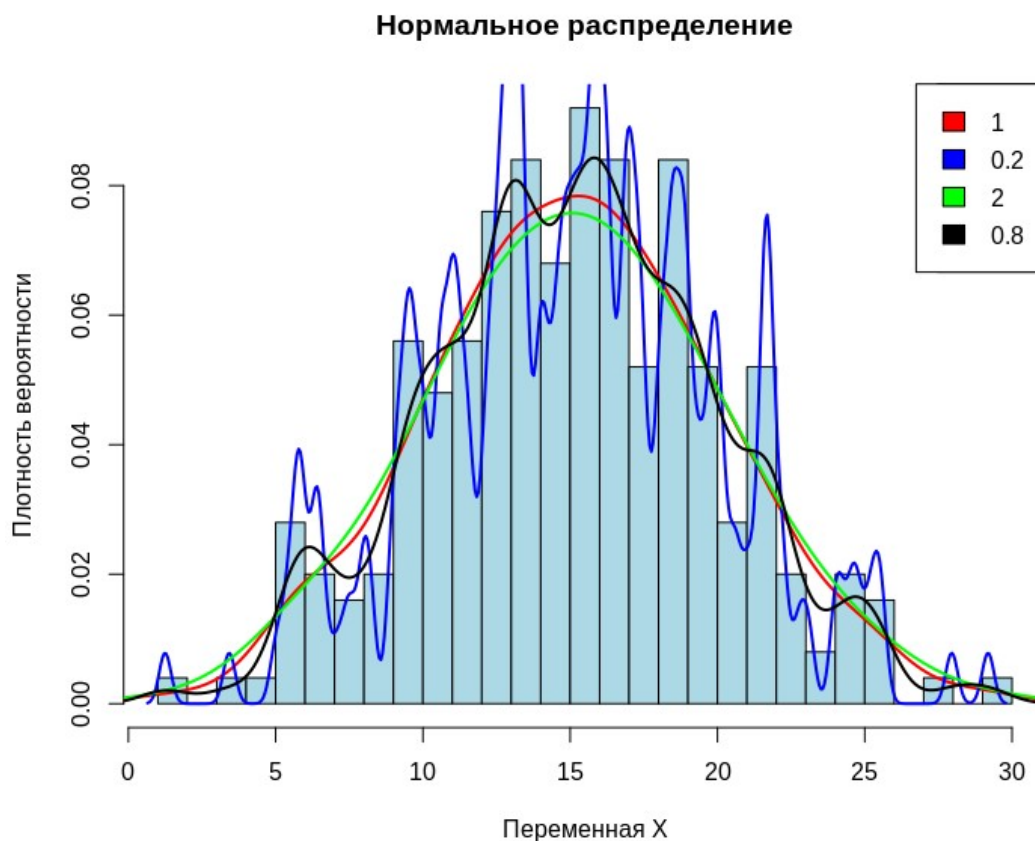
Комментарий: используя изученную функцию `layout`, выведены графики и гистограммы при разных параметрах. При увеличении `n` график все больше принимает форму колокола, при увеличении `mean`, график смещается по оси `x`, а при изменении `sd` график все меньше становится похож на колокол, и учащаются колебания кривой. Именно при изменении `sd`, можно заметить, что делает параметр `bw`.

1.2. Функция `density()` используется для отображения (ядерной) оценки плотности. Изучите ее параметры

Код:

```
X <- rnorm(n = 250, mean = 15, sd = 5)
hist(X, breaks = 20, freq = FALSE, col = "lightblue",
     xlab = "Переменная X",
     ylab = "Плотность вероятности",
     main = "Нормальное распределение")
lines(density(X), col = "red", lwd = 2)
lines(density(X, bw = 0.2), col = "blue", lwd = 2)
lines(density(X, bw = 2), col = "green", lwd = 2)
lines(density(X, bw = 0.8), col = "black", lwd = 2)
legend("topright", legend=c("1", "0.2", "2", "0.8"),
      fill=c("red", "blue", "green", "black"))
```

Вывод:



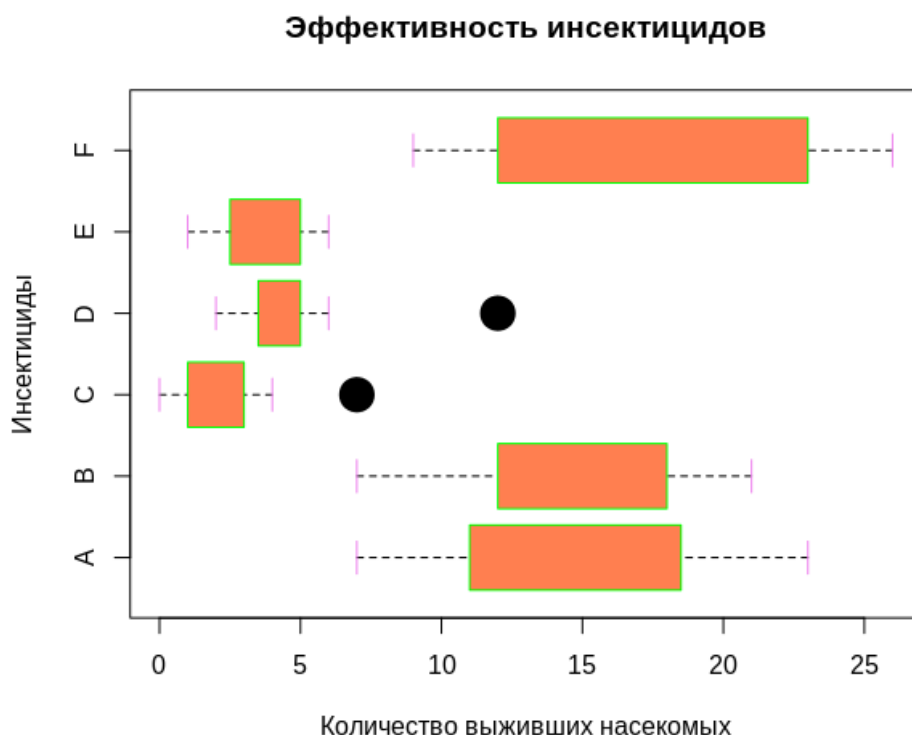
Комментарий: Как видно из рисунка, параметр `bw` определяет то, насколько кривые будут сглаживаться относительно `sd`. Были использованы функции нанесения кривых на график и вывода легенды.

1.3. Изучить параметры boxplot()

Код:

```
boxplot(count ~ spray,  
        ylab = "Инсектициды",  
        xlab = "Количество выживших насекомых",  
        main = "Эффективность инсектицидов",  
        col = "coral", horizontal = TRUE,  
        data = InsectSprays,  
        whisklty=2, # тип усов (3=dashed)  
        medlty=0, # тип прямой на медиане (0 чтобы убрать)  
        boxcol="green", # цвет границы коробки  
        staplecol = "violet", # цвет максимумов и минимумов  
        outlwd = 15 # ширина выбросов  
)
```

Вывод:



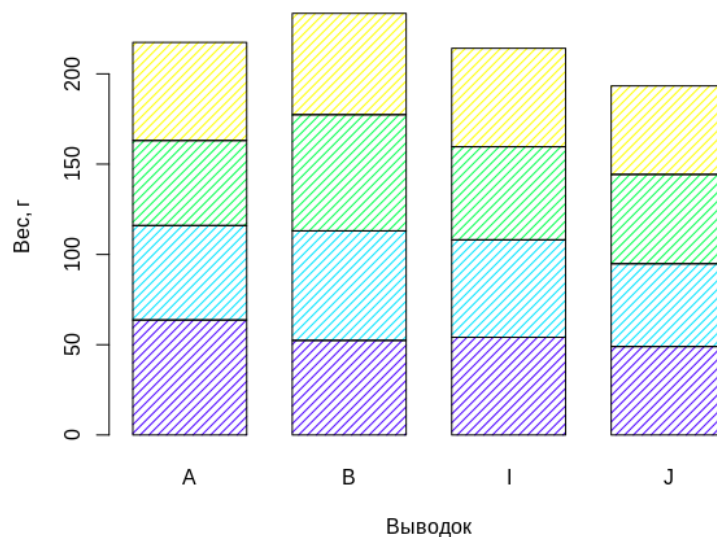
Комментарий: достаточно гибко можно изменить внешний вид всех частей диаграммы, используя сокращения box, med, whisk, staple, out и добавляя сокращения для изменения параметров (wd, col, lty и другие).

1.4. Изучить параметры barplot()

Код:

```
boxplot(count ~ spray,  
        ylab = "Инсектициды",  
        xlab = "Количество выживших насекомых",  
        main = "Эффективность инсектицидов",  
        col = "coral", horizontal = TRUE,  
        data = InsectSprays,  
        whisklty=2, # тип усов (3=dashed)  
        medlty=0, # тип прямой на медиане (0 чтобы убрать)  
        boxcol="green", # цвет границы коробки  
        staplecol = "violet", # цвет максимумов и минимумов  
        outlwd = 15 # ширина выбросов  
)
```

Вывод:



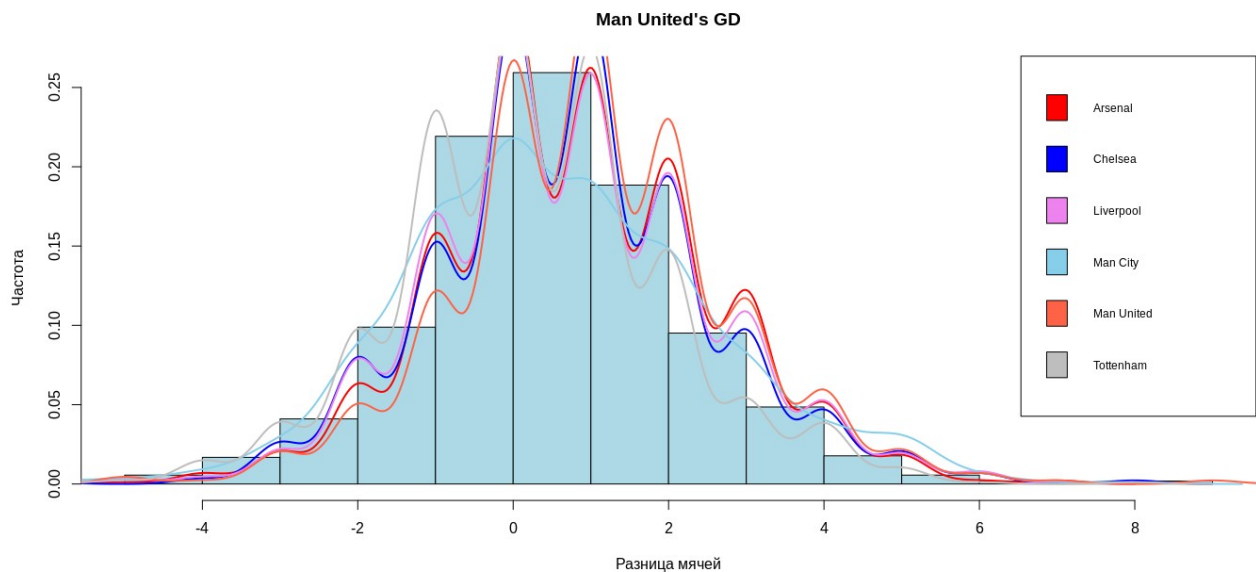
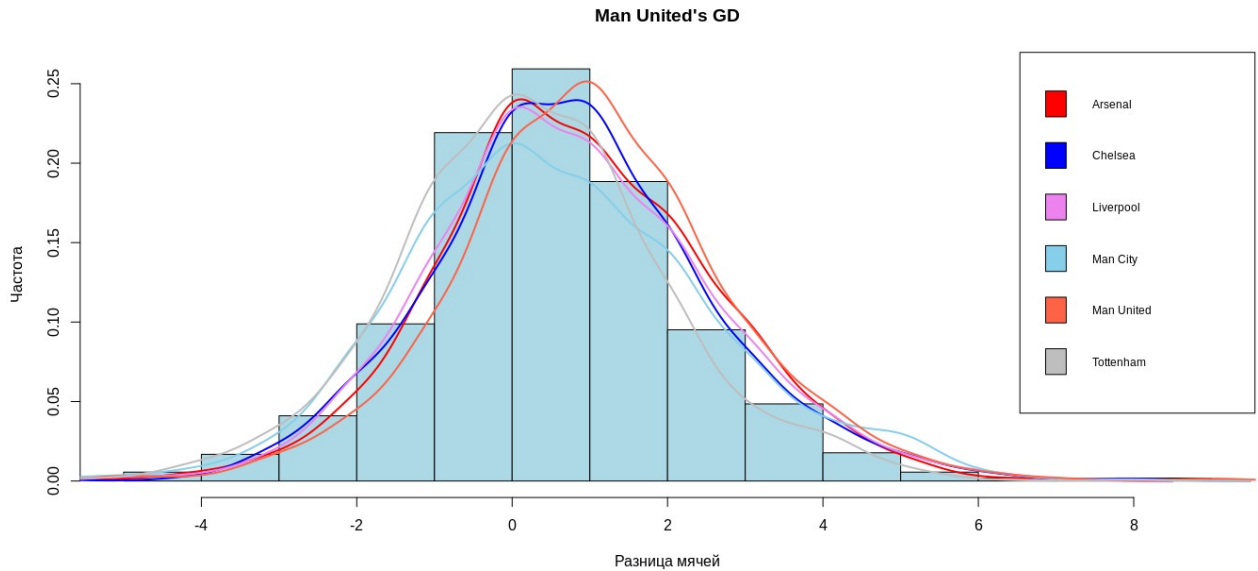
Задание 2. Реализовать аппроксимацию с помощью ядерных оценок для своих данных, использовать различные параметры при выводе. В этом и последующем пунктах снабдить графики легендами.

Код:

```
match_results = read.csv(file.choose())
attach(match_results)
big_six=c('Arsenal','Chelsea','Liverpool','Man City','Man
United','Tottenham')

layout(matrix(1:1, ncol=1, nrow=1))
home_gd=FTHG[HomeTeam=='Man United'] - FTAG[HomeTeam=='Man United']
away_gd=FTAG[AwayTeam=='Man United'] - FTHG[AwayTeam=='Man United']
total_gd <- c(home_gd, away_gd)
hist(total_gd,
      freq=F,
      xlab = "Разница мячей",
      ylab = "Частота",
      main = paste("Man United", "'s GD", sep=''),
      col="lightblue"
) # гистограмма разницы мячей МЮ, на нее будут ложиться все кривые
density(total_gd)
colors()
cols = c("red", "blue", "violet", "skyblue", "tomato", "grey")
# цвета выбранные для кривых
col_num = 1 # счетчик цветом для кривых
for (i in big_six) {
  home_gd=FTHG[HomeTeam==i] - FTAG[HomeTeam==i] # домашние результаты
  away_gd=FTAG[AwayTeam==i] - FTHG[AwayTeam==i] # результаты на выезде
  total_gd <- c(home_gd, away_gd) # общие результаты
  lines(density(total_gd, wd=0.45), col = cols[col_num], lwd = 2)
  col_num = col_num + 1
}
legend("topright", big_six, fill=cols, cex=0.75)
```

Вывод:



Комментарий: Верхний график показывает результат при параметре 0.45, а второй при стандартном параметре (nrd0).

Задание 3. На примере собственных данных продемонстрировать применение `cdplot` и `boxplot`.

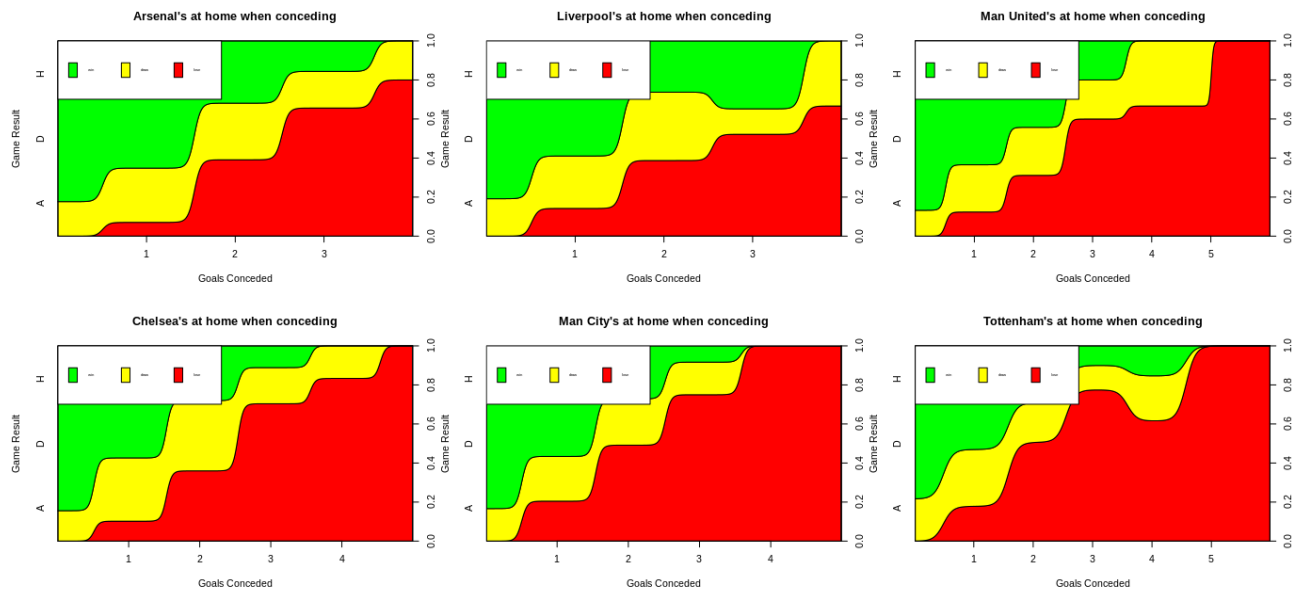
3.1. `cdplot`.

Код:

```
layout(matrix(1:6, ncol=3, nrow=2))
for (i in big_six) {
  home_gd=FTHG[HomeTeam==i] - FTAG[HomeTeam==i] # домашние результаты
  away_gd=FTAG[AwayTeam==i] - FTHG[AwayTeam==i] # результаты на выезде
  game_results = FTR[HomeTeam==i]
  total_gd <- c(home_gd) # общие результаты
  cdplot(FTR[HomeTeam==i]~FTAG[HomeTeam==i], col=c('red','yellow','green'),
        xlab="Goals Conceded",
        ylab="Game Result",
        main=paste(i,"'s at home when conceding", sep=''))
```

```
legend("topleft",legend=c("win","draw","lose"),fill=c("green","yellow","red"),ncol=3,cex=0.4)
}
```

Вывод:



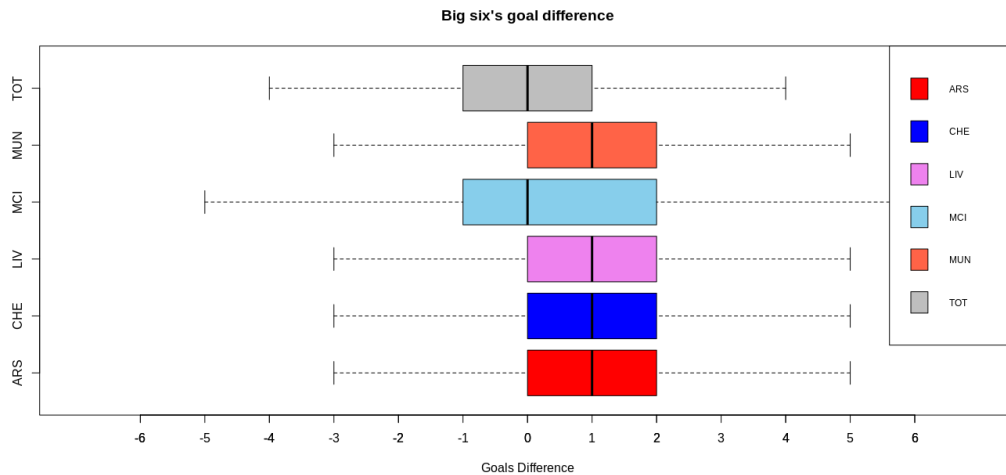
Комментарий: выводится `cpdplot` для 6 лучших команд лиги: зависимость результата игры от пропущенных голов на своем стадионе. Легенда выводится маленькой, так как при стандартном размере она занимает весь график, а размер текста отдельно от размера легенды не меняется.

3.2. boxplot.

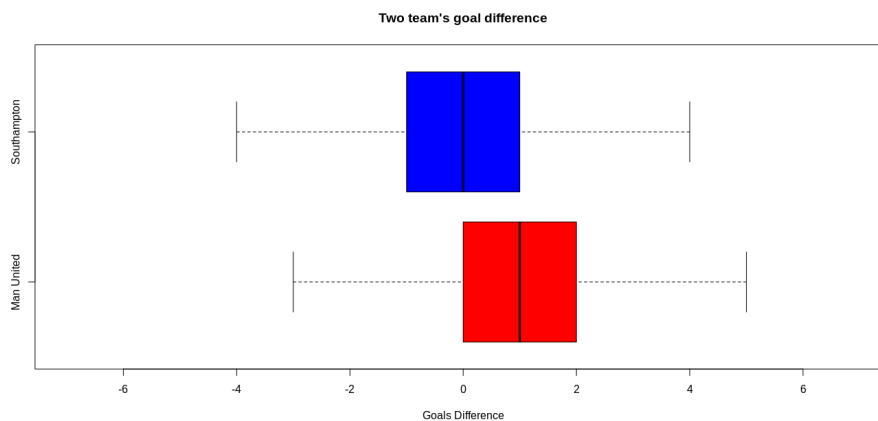
Код:

```
cols = c("red","blue","violet","skyblue","tomato","grey")
big_six_abbreviations=c("ARS", "CHE", "LIV","MCI","MUN","TOT")
boxplot(totals,
        ylim=c(-7,7),
        horizontal=T,
        names=big_six_abbreviations,
        xlab="Goals Conceded",
        main=paste("Big six's goal difference",sep=''),
        col=cols,
        outline=F
)
axis(side=1,at=seq(-6,6))
legend("topright",legend=big_six_abbreviations,fill=cols,cex=0.75)
```

Вывод:



Комментарий: Вывод коробки с усами для лучших шести команд не совсем впечатляет, но на примере хорошей и средней команды результаты выглядят немного более впечатляюще.



Задание 4. Используя функцию `pie`, нарисовать пример круговой диаграммы для собственных данных.

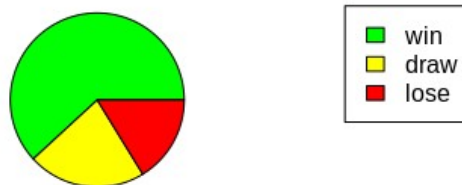
Код:

```
draw_pie_for <- function(team_name) {
  home_results=table(FTR[HomeTeam==team_name])
  away_results=table(FTR[AwayTeam==team_name])
  wins = home_results[names(home_results)=='H'] +
  away_results[names(away_results)=='A']
  draws = home_results[names(home_results)=='D'] +
  away_results[names(away_results)=='D']
  loses = home_results[names(home_results)=='A'] +
  away_results[names(away_results)=='H']
  pie(c(wins,draws,loses),
      labels=c("", "", ""),
      col=c("green", "yellow", "red"),
      main=paste(team_name, "'s all-time results", sep=""),
      radius=1)

  legend("topright", fill=c("green", "yellow", "red"), legend=c("win", "draw", "lose"))
}
draw_pie_for("Man United")
```

Вывод:

Man United's all-time results



Комментарий: создана функция, которая будет выводить круговую диаграмму выступлений команды за все время существования турнира.