

Ярылгасимов Султан  
Отчет по домашнему заданию №2

**Задание 1.** Написать функцию, которая выполняет загрузку данных и их рисование.  
Воспользоваться условными операторами и циклами.

**Код:**

```
# additional func to know month from date in dataset
get_month <- function(date) {
  new_date = as.Date(date, '%Y-%m-%dT%H:%M:%SZ')
  get_month = as.numeric(format(new_date, "%m"))
  # get_month = as.integer(substr(as.character(date), 6, 7))
}

plot_stats <- function(team_name, season) {
  match_results <- read.csv(file="/media/sultanyaril/989E6E129E6DE968/Users/
Султан/Desktop/Third year/Практикум на ЭВМ/dataset/results.csv")
  # str(match_results)
  points.ts <- ts(rep(0, times = 12), start = c(as.numeric(substr(season, 1,
4)), 8), frequency=12)
  points_accumulation=data.frame(
    month = seq(as.Date(paste(substr(season,1,4), "-8-1", sep='')),
by="month", length.out=12),
    points=0,
    games=0
  )
  # frame with Months from August to July and points, that we'll acumulate
  month_count = 1
  # this on we need for points.ts because season start in August and it's
indexed 1, not 8
  current_month = 0
  # initially its 0 but it's only needed to know when new month starts
  total_points = 0
  # accumulator
  total_games = 0
  for(i in seq(1, nrow(match_results), 1)) {
    row <- match_results[i, ]
    if (season != row$Season) {
      next;
    }
    if (row$HomeTeam == team_name) {
      # first we check if month ended
      # if true then update ts and frame
      if (current_month != get_month(row$DateTime)) {
        points.ts[month_count] = total_points

points_accumulation$points[get_month(points_accumulation$month)==current_mon
th] = total_points

points_accumulation$games[get_month(points_accumulation$month)==current_mont
h] = total_games
        # decrements by 1 month to correctly assign
        total_points = 0
        total_games = 0
        month_count = month_count + 1
        current_month = get_month(row$DateTime)
      }
      total_games = total_games + 1
      if (row$FTHG > row$FTAG)
        total_points = total_points + 3 # win
```

```

    else if (row$FTHG == row$FTAG)
      total_points = total_points + 1 # draw
  } else if (row$AwayTeam == team_name) {
    if (current_month != get_month(row$DateTime)) {
      points.ts[month_count] = total_points

points_accumulation$points[get_month(points_accumulation$month)==current_mon
th] = total_points

points_accumulation$games[get_month(points_accumulation$month)==current_mont
h] = total_games
      total_points = 0
      total_games = 0
      month_count = month_count + 1
      current_month = get_month(row$DateTime)
    }
    total_games = total_games + 1
    if (row$FTHG < row$FTAG)
      total_points = total_points + 3
    else if (row$FTHG == row$FTAG)
      total_points = total_points + 1
  }
}
# record last month
points.ts[month_count] = total_points

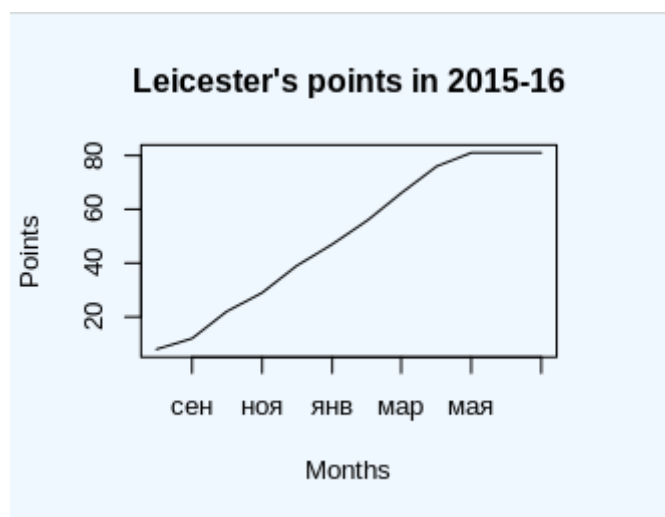
points_accumulation$points[get_month(points_accumulation$month)==current_mon
th] = total_points

points_accumulation$games[get_month(points_accumulation$month)==current_mont
h] = total_games
print(points.ts)
# multiplot(points_accumulation, team_name, season)
plot(
  cumsum(points_accumulation$points) ~ points_accumulation$month,
  type='l',
  main=paste(team_name, "'s points in ", season, sep=''),
  xlab="Months",
  ylab="Points"
)
}

plot_stats('Leicester', '2015-16')

```

**Вывод:**



	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2015								0	8	4	10	7
2016	10	8	9	10	10	5	0					

> |

**Комментарий:** Функция принимает на вход название команды и сезон, и выводит кумулятивный график набор очков. Для этого создается data.frame с информацией о выступлениях команды в сезоне, которая заполняется использованием циклов и условных операторов. Дополнительно, создана функция для вывода номера месяца из полной даты.

**Задание 2.** Продемонстрировать визуализацию данных на одном листе для различных комбинаций настроек функции plot (использовать максимальный набор настраиваемых параметров).

**Код:**

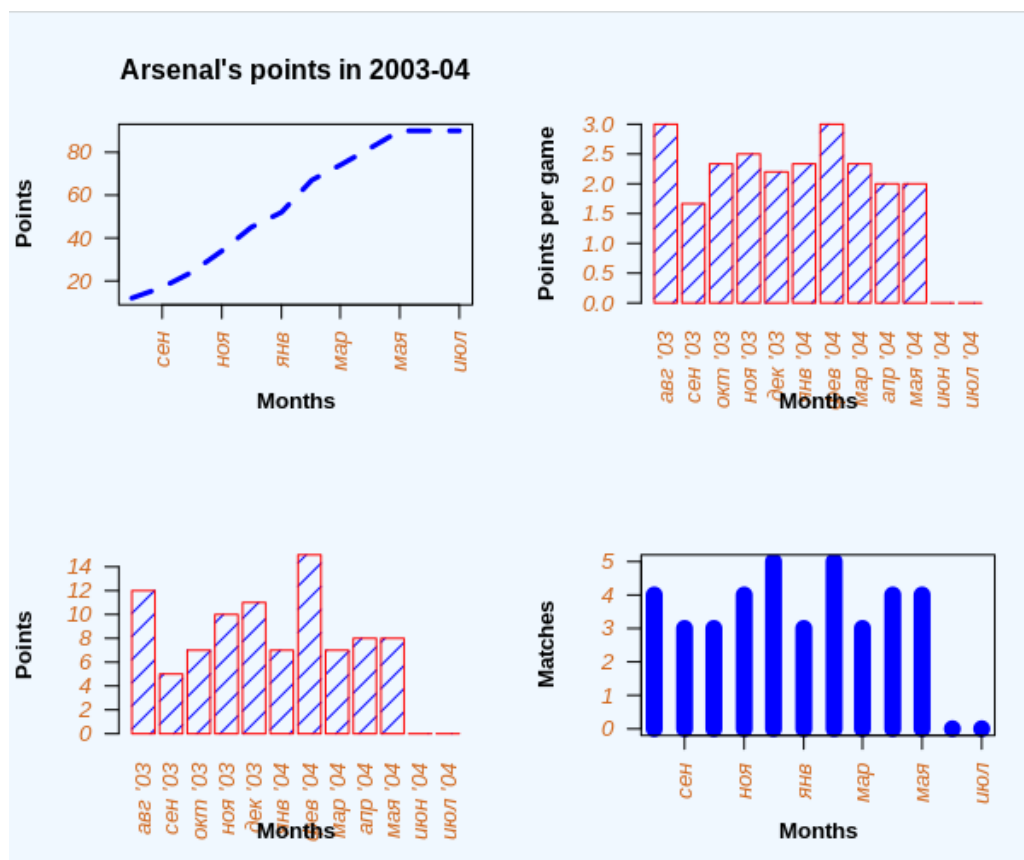
```
multiplot <- function(data, team_name, season) {
  par(mfcol=c(2,2))
  # plot of cumulative points
  plot(
    cumsum(data$points) ~ data$month,
    type='l',
    main=paste(team_name, "'s points in ", season, sep=''),
    xlab="Months",
    ylab="Points",
    lty = 2,
    lwd = 3,
    font = 3,
    font.lab = 2,
    col = "blue",
    col.axis = 'chocolate',
    las = 2
  )
  # barplot of points per month
  barplot(
    t(as.matrix(data$points)),
    names.arg=format(data$month, "%b '%y"),
    xlab="Months",
    ylab="Points",
    col = "blue",
    font = 3,
    font.lab = 2,
    col.axis = 'chocolate',
    las = 2,
    density = 10,
    border = 'red'
  )
  #barplot of average points in the month
  barplot (
    t(as.matrix(data$points/data$games)),
    names.arg=format(data$month, "%b '%y"),
    xlab="Months",
    ylab="Points per game",
    col = "blue",
    font = 3,
```

```

font.lab = 2,
col.axis = 'chocolate',
las = 2,
density=10,
border='red'
)
# barplot made by plot
plot (
  data$games ~ data$month,
  type='h',
  lwd=10,
  xlab="Months",
  ylab="Matches",
  col = "blue",
  font = 3,
  font.lab = 2,
  col.axis = 'chocolate',
  las = 2,
)
}

```

**Вывод:**



**Комментарий:** Вызов plot() в функции из прошлого задания был заменен на вызов новой пользовательской функции multiplot(). Функция выводит на экран 4 графика, изображающих разную статистику за сезон команды. Построены: прямая, как в прошлом задании, но измененная параметрами функции; две гистограммы (функция hist()) с датами по осям x; гистограмма построенная функцией plot() (на самом деле не самая красивая, но добавлена, чтобы разнообразить картинку).

**Задание 3.** Реализовать векторную обработку данных с помощью функций `apply` / `lapply` / `sapply` / `vapply` / `mapply` / `rapply` / `tapply`.

**Код:**

```
teams = levels(dataset$HomeTeam)
results = sapply(teams, season='2003-04', plot_stats)
tournament_table = sort(results[results != 0], T)
tournament_table

seasons = levels(dataset$Season)
chelsea_results = sapply(seasons, team_name='Chelsea', plot_stats)
chelsea_results
```

**Вывод:**

```
> tournament_table
  Arsenal   Chelsea  Man United  Liverpool  Aston Villa  Newcastle  Bolton
    90         79         75         60         56         56         53
  Charlton   Fulham  Birmingham  Middlesbrough  Southampton  Portsmouth  Tottenham
    53         52         50         48         47         45         45
  Blackburn  Man City   Everton    Leeds    Leicester    Wolves
    44         41         39         33         33         33

> |

> chelsea_results
1993-94 1994-95 1995-96 1996-97 1997-98 1998-99 1999-00 2000-01 2001-02 2002-03 2003-04 2004-05
    51     54     50     59     63     75     65     61     64     67     79     95
2005-06 2006-07 2007-08 2008-09 2009-10 2010-11 2011-12 2012-13 2013-14 2014-15 2015-16 2016-17
    91     83     85     83     86     71     64     75     82     87     50     93
2017-18 2018-19 2019-20 2020-21
    70     72     66     67

> |
```

**Комментарий:** Первая часть кода возвращает лист с результатами команд в определенный сезон и после сортирую его и убирая нулевые значения, выводим турнирную таблицу. Вторая часть же выводит количество набранных очков определенной команды за все время существования турнира.