Biomedical Data Science - Assignment 1

Damian Baeza

14 February 2018

Problem 1

Question (a)

```
# Data frame with the airquality data
df.a <- airquality

# Number of missing values in Ozone variable
Ozone.Miss <- sum(is.na(df.a$Ozone))
print(Ozone.Miss)
```

```
## [1] 37
```

```
# Mean imputation of Ozone variable
df.a$Ozone[is.na(df.a$Ozone)] <- mean(df.a$Ozone, na.rm = TRUE)

# Number of missing values in Ozone variable after mean imputation
Ozone.Miss <- sum(is.na(df.a$Ozone))
print(Ozone.Miss)
```

```
## [1] 0
```

Question (b)

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
impute.to.window.mean <- function(x, windowsize){
  ## Function that imputes the mean of a vector considering a windowsize.
  ## A missing value will be imputed with the mean of the values in the
  ## positions (i - windowsize) and (i + windowsize).
  ## Inputs: vector to be imputed and size of window
  ## Outputs: imputed vector
  if(windowsize > length(x)){
    warning("Window size is longer than vector, vector length will be used window size")
    impute.val <- mean(x, na.rm = TRUE)
    x[is.na(x)] <- impute.val
    return(x)
  }
  else if(windowsize < 0){
    stop("Window size is negative!")
  }
  else{
    # position of missing values
    Miss.pos <- which(is.na(x))
```

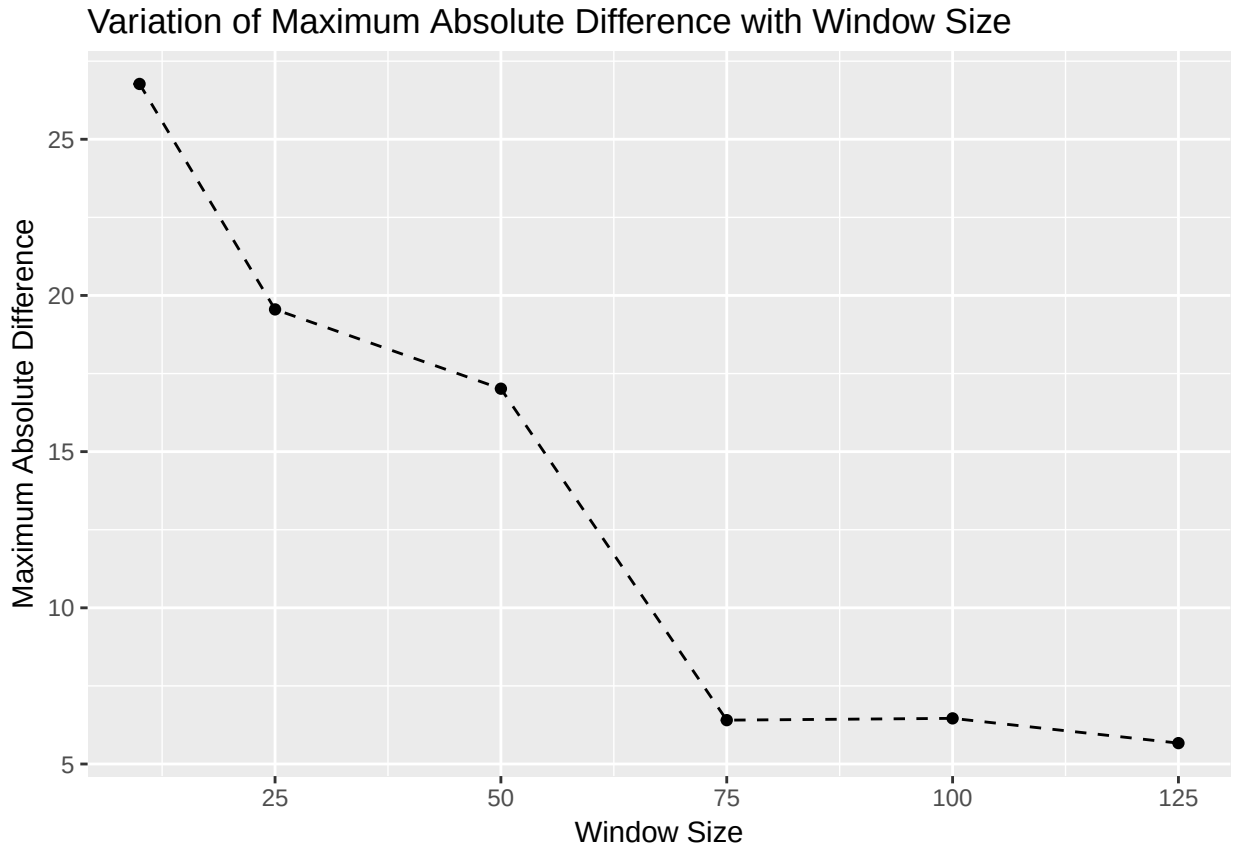
```
    new.x <- x
    for(i in Miss.pos){
      # calculation of begining of the window of values used for
      # imputation
      first <- max(i - windowsize, 1)
      # calculation of end of the window of values used for
      # imputation
      last <- min(i + windowsize, length(x))
      # mean calculation
      impute.val <- mean(x[first:last], na.rm = TRUE)
      # imputation of missing value
      new.x[i] <- impute.val
    }
    return(new.x)
  }
}
```

Question (c)

```
df.c <- airquality$Ozone

n <- length(df.c)

windows.size <- c(10, 25, 50, 75, 100, 125)

Results.Q1 <- data.frame(n10 = numeric(n),
                         n25 = numeric(n),
                         n50 = numeric(n),
                         n75 = numeric(n),
                         n100 = numeric(n),
                         n125 = numeric(n))

for (i in 1:length(windows.size)) {
  Results.Q1[,i] <- impute.to.window.mean(x = df.c,
                                          windowsize = windows.size[i])
}

Abs.Diff <- round(abs(Results.Q1 - df.a$Ozone), 3)

colnames(Abs.Diff) <- c()

Max.Abs.Diff <- data.frame(Max.Abs.Diff = sapply(Abs.Diff, max), Window.Size = windows.size)

print(Max.Abs.Diff)
```

```
##   Max.Abs.Diff Window.Size
## 1       26.771          10
## 2       19.553          25
## 3       17.015          50
## 4        6.404          75
## 5        6.461         100
## 6        5.669         125
```

```r
ggplot(data = Max.Abs.Diff, aes(x = Window.Size, y = Max.Abs.Diff)) +
  geom_point() +
  geom_line(linetype = "dashed") +
  labs(x = "Window Size", y = "Maximum Absolute Difference",
       title = "Variation of Maximum Absolute Difference with Window Size")
```



Question (d)

The smallest window size is 5.

```r
min.window <- numeric(2)

df.d <- airquality

df.d$Ozone <- airquality$Ozone
df.d$Solar.R <- airquality$Solar.R

for(i in 1:length(df.d$Ozone)){
  Ozone.chkr <- sum(is.nan(impute.to.window.mean(df.d$Ozone, i)))
  if(Ozone.chkr == 0){
    min.window[1] <- i
    break
  }
}
```

```
for(i in 1:length(df.d$Ozone)){
  Solar.R.chkr <- sum(is.nan(impute.to.window.mean(df.d$Solar.R, i)))
  if(Solar.R.chkr == 0){
    min.window[2] <- i
    break
  }
}

print("Smallest window that allows imputation of all missing values for Ozone: ")
```

```
## [1] "Smallest window that allows imputation of all missing values for Ozone: "
```

```
print(min.window[1])
```

```
## [1] 5
```

```
print("Smallest window that allows imputation of all missing values for Solar.R: ")
```

```
## [1] "Smallest window that allows imputation of all missing values for Solar.R: "
```

```
print(min.window[2])
```

```
## [1] 2
```

Problem 2

```
longe.egfr.1 <- read.csv(file = "longegfr1.csv", header = TRUE)

longe.egfr.2 <- read.csv(file = "longegfr2.csv", header = TRUE)

longe.egfr <- merge(x = longe.egfr.1, y = longe.egfr.2, by.x = c("id", "fu.years"),
                    by.y = c("ID", "fu.years"),
                    all.x = TRUE)

longe.egfr$id <- as.numeric(longe.egfr$id)

longe.egfr <- longe.egfr[order(longe.egfr$id, longe.egfr$fu.years),]

sort.chkr <- function(x = longe.egfr){
  if (is.unsorted(x$id)) {
    stop("ID is not sorted!")
  }
  for(i in unique(x$id)){
    if(is.unsorted(x$fu.years[x$ID == i])){
      stop("Follow up time is not sorted!")
    }
  }
  return("Data frame is ordered accordingly!")
}

print(sort.chkr())
```

```
## [1] "Data frame is ordered accordingly!"
```

Question (b)

```
require(dplyr)
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
eGFR.averages <- longe.egfr %>% as_tibble() %>% group_by(id) %>%
  summarise(mean = mean(egfr, na.rm = TRUE),
            fu.years = max(fu.years, na.rm = TRUE))

eGFR.averages$interval <- cut(eGFR.averages$mean,
                              breaks = c(0, 15, 30, 60, 90, Inf))

print("Patients according to average eGFR: ")
```

```
## [1] "Patients according to average eGFR: "
```

```
print(table(eGFR.averages$interval))
```

```
##
##   (0,15]  (15,30]  (30,60]  (60,90] (90,Inf]
##        2        9       84       86       66
```

```
print("Number of patients with missing average: ")
```

```
## [1] "Number of patients with missing average: "
```

```
print(sum(is.nan(eGFR.averages$mean)))
```

```
## [1] 3
```

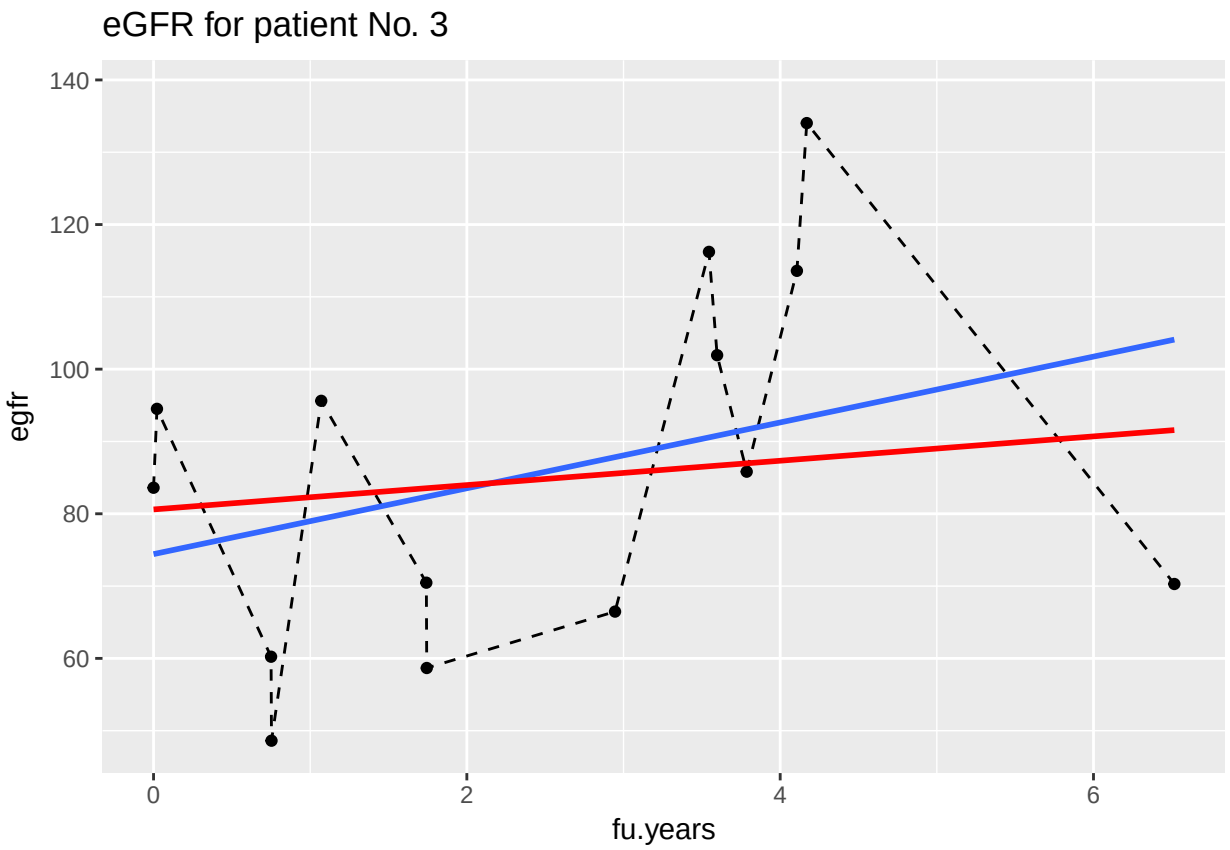
Question (c)

```
ids <- eGFR.averages$id[eGFR.averages$mean <= 15 & !is.na(eGFR.averages$mean)]
loc1 <- c(longe.egfr$id == ids[1])
loc2 <- c(longe.egfr$id == ids[2])

df.0.15 <- data.frame(id = ids,
                      sex = c(unique(longe.egfr$sex[loc1]),
                              unique(longe.egfr$sex[loc2])),
                      bl.age = c(unique(longe.egfr$baseline.age[loc1]),
                                 unique(longe.egfr$baseline.age[loc2])),
                      average.eGFR = c(eGFR.averages$mean[eGFR.averages$id == ids[1]],
                                       eGFR.averages$mean[eGFR.averages$id == ids[2]]),
                      Max.FU = c(eGFR.averages$fu.years[eGFR.averages$id == ids[1]],
                                 eGFR.averages$fu.years[eGFR.averages$id == ids[2]]),
```

```
                    eGFR.count = c(sum(!is.na(longe.egfr$egfr[longe.egfr$id == ids[1]])),
                                   sum(!is.na(longe.egfr$egfr[longe.egfr$id == ids[2]])))
```

Question (d)

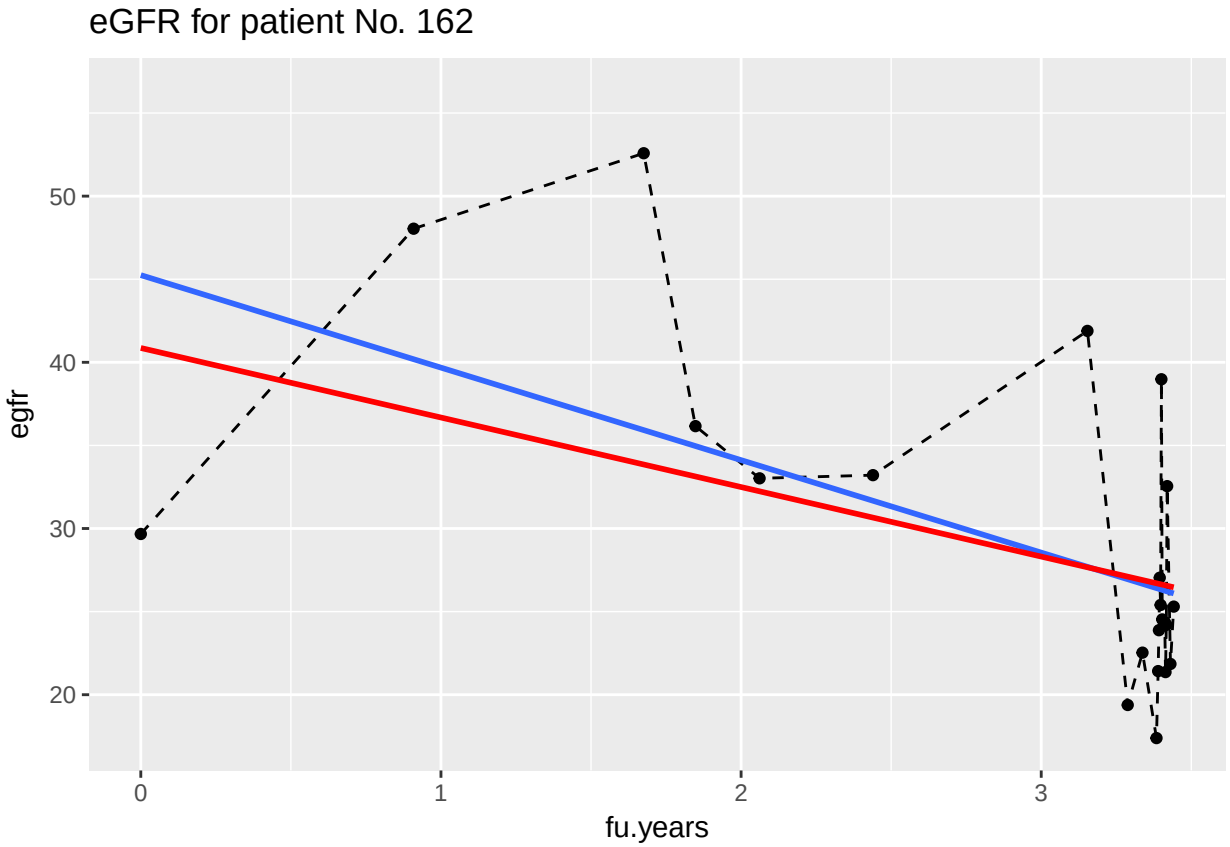
```
df.p3 <- longe.egfr[longe.egfr$id == 3,]
df.p37 <- longe.egfr[longe.egfr$id == 37,]
df.p162 <- longe.egfr[longe.egfr$id == 162,]
df.p223 <- longe.egfr[longe.egfr$id == 223,]

ggplot(data = df.p3, aes(x = fu.years, y = egfr)) +
  geom_point() +
  geom_line(linetype = "dashed") +
  geom_smooth(method = "lm", fill = NA) +
  geom_smooth(data = subset(df.p3,
                            egfr > min(egfr, na.rm = TRUE) & egfr < max(egfr, na.rm = TRUE)),
              method = "lm", col = "red", fill = NA) +
  ggtitle(paste("eGFR for patient No.",unique(df.p3$id)))
```




```
ggplot(data = df.p37, aes(x = fu.years, y = egfr)) +
  geom_point() +
  geom_line(linetype = "dashed") +
  geom_smooth(method = "lm", fill = NA) +
  geom_smooth(data = subset(df.p37,
```

```
                          egfr > min(egfr, na.rm = TRUE) & egfr < max(egfr, na.rm = TRUE)),
              method = "lm", col = "red", fill = NA) +
  ggtitle(paste("eGFR for patient No.",unique(df.p37$id)))
```


```
ggplot(data = df.p162, aes(x = fu.years, y = egfr)) +
  geom_point() +
  geom_line(linetype = "dashed") +
  geom_smooth(method = "lm", fill = NA) +
  geom_smooth(data = subset(df.p162,
                          egfr > min(egfr, na.rm = TRUE) & egfr < max(egfr, na.rm = TRUE)),
              method = "lm", col = "red", fill = NA) +
  ggtitle(paste("eGFR for patient No.",unique(df.p162$id)))
```

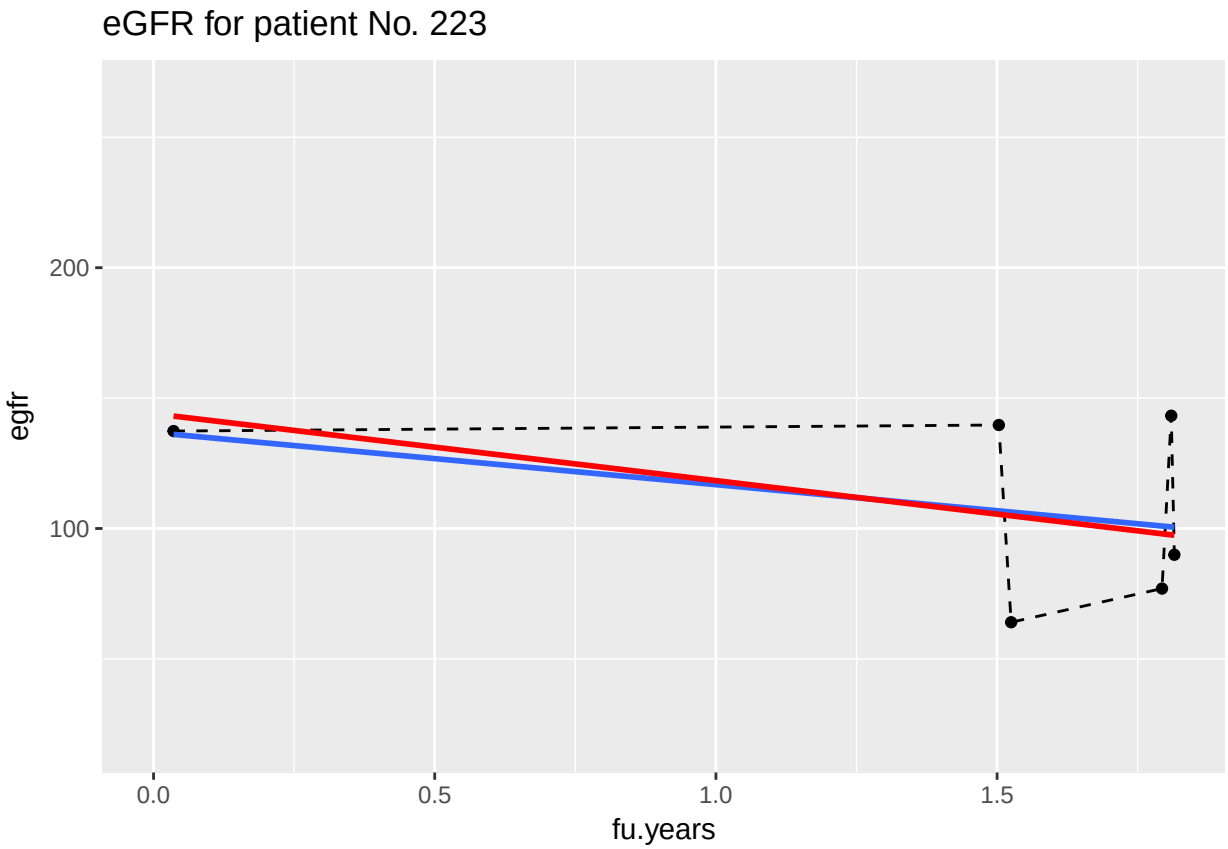



```
ggplot(data = df.p223, aes(x = fu.years, y = egfr)) +
  geom_point() +
  geom_line(linetype = "dashed") +
  geom_smooth(method = "lm", fill = NA) +
  geom_smooth(data = subset(df.p223,
                            egfr > min(egfr, na.rm = TRUE) & egfr < max(egfr, na.rm = TRUE)),
              method = "lm", col = "red", fill = NA) +
  ggtitle(paste("eGFR for patient No.",unique(df.p223$id)))
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Warning: Removed 1 rows containing missing values (geom_path).
```



Problem 3

Question (a)

```
egfr.mdrd4 <- function(scr, age, sex, ethnic){
  x <- numeric(length(scr))
  for(i in 1:length(scr)){
    if(is.na(scr[i]) | is.na(age[i]) | is.na(sex[i]) | is.na(ethnic[i])){
      mdrd4 <- NA
    } else{
      mdrd4 <- 175*scr[i]^(-1.154)*age[i]^(-0.203)
      if(sex[i] == "Female"){
        mdrd4 <- mdrd4*0.742
      }
      if(ethnic[i] == "Black"){
        mdrd4 <- mdrd4*1.212
      }
    }
    x[i] <- mdrd4
  }
  return(x)
}
```

```r
egfr.ckdepi <- function(scr, age, sex, ethnic){
  x <- numeric(length(scr))
  for(i in 1:length(scr)){
    if(is.na(scr[i]) | is.na(age[i]) | is.na(sex[i]) | is.na(ethnic[i])){
      ckdepi <- NA
    } else{
      if(sex[i] == "Female"){
        Kappa <- 0.7
        alpha <- -0.329
        adic.factor <- 1.018
      } else{
        Kappa <- 0.9
        alpha <- -0.411
        adic.factor <- 1
      }
      ckdepi <- 141*min(scr[i]/Kappa,1)^alpha*max(scr[i]/Kappa,1)^(-1.209)
      ckdepi <- ckdepi*0.993^(age[i])*adic.factor
      if(ethnic[i] == "Black"){
        ckdepi <- ckdepi*1.159
      }
    }
    x[i] <- ckdepi
  }
  return(x)
}
```

Question (b)

```r
scr.data <- read.csv(file = "scr.csv", header = TRUE)

data.mdrd4 <- round(egfr.mdrd4(scr = scr.data$scr,
                               age = scr.data$age,
                               sex = scr.data$sex,
                               ethnic = scr.data$ethnic), 2)

data.ckdepi <- round(egfr.ckdepi(scr = scr.data$scr,
                                 age = scr.data$age,
                                 sex = scr.data$sex,
                                 ethnic = scr.data$ethnic), 2)

mean.mdrd4 <- mean(data.mdrd4, na.rm = TRUE)
```