Biomedical Data Science: Assignment 1

Agnieszka Słowik

2/25/2018

Problem 1

a) Overall mean imputation

```
airquality.imputed.mean <- airquality
mean.ozone <- mean(airquality$Ozone, na.rm=TRUE)

print(paste0("Overall mean of the Ozone variable: ", round(mean.ozone, 3)))
```

```
## [1] "Overall mean of the Ozone variable: 42.129"
```

```
na.idx <- is.na(airquality.imputed.mean$Ozone)
airquality.imputed.mean$Ozone[na.idx] <- mean.ozone
```

b) Window mean imputation function (NB: the original vector without imputed values is used to compute the local mean)

```
impute.to.window.mean <- function(x, windowsize){
  stopifnot(windowsize > 0)
  x.imputed <- x
  for (i in 1:length(x.imputed)){
    if (is.na(x.imputed[i])){
      window <- c(max(1, i-windowsize):min(i+windowsize, length(x.imputed)))
      local.mean <- mean(x[window], na.rm = TRUE)
      x.imputed[i] <- local.mean
    }
  }
  stopifnot(length(x.imputed)==length(x))
  return(x.imputed)
}
```

c) Ozone window mean imputation

```
tmp = list()
windows <- c(10, 25, 50, 75, 100, 125) # Windows sizes
for (w in windows){
  imputed <- impute.to.window.mean(airquality$Ozone, w)
  max.diff <- data.frame(max(abs(imputed-airquality.imputed.mean$Ozone)))
  max.diff$w <- w
  tmp[[w]] <- round(max.diff,3)
}

max.abs.diff = do.call(rbind, tmp)
```

```r
colnames(max.abs.diff) = c("Max.Abs.Diff", "Window.Size")
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
```

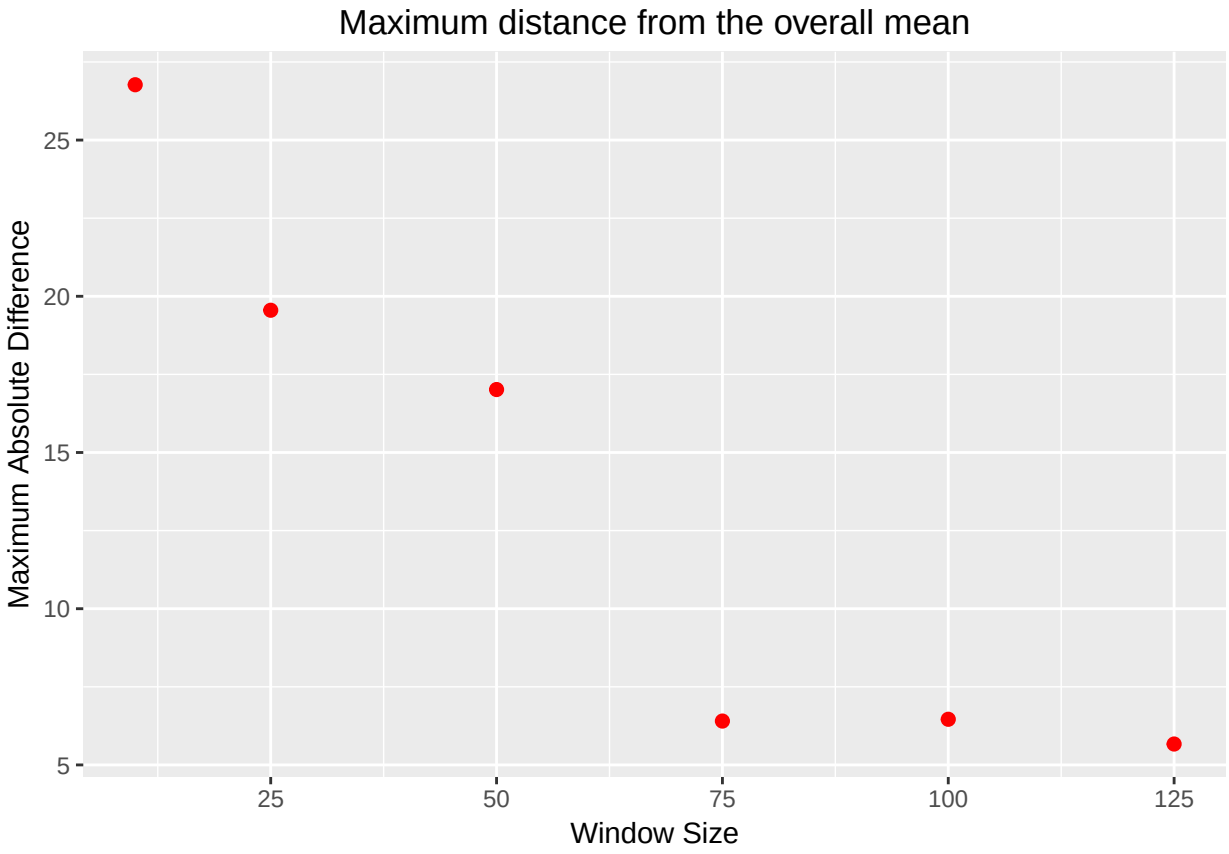


Table 1: Maximum distance from the overall mean

Window Size	Maximum absolute difference
10	26.771
25	19.553
50	17.015
75	6.404
100	6.461
125	5.669

Interpretation: Total number of examples is 153 so the imputation with window sizes of 75 and more gives similar results, and they are all close to the overall mean imputation since the majority of values used to compute the mean is the same (indices smaller than 1 and bigger than 153 are rounded so that we don't exceed the vector length, and NA values are omitted when computing the mean). For smaller window sizes (10, 25, 50) there is a bigger local variance since we use respectively max of 20, 50 and 100 of neighboring

examples.

d) Smallest window size that allows the imputation of all missing values for variables "Ozone" and "Solar.R"

```
smallest.window.size <- function(variable){
  windows = seq(1:length(variable))
  for (w in windows){
    imputed <- impute.to.window.mean(variable, w)
    if(sum(is.na(imputed))==0){
      cat(sprintf("Minimum window size for the variable %s is %d",
                  deparse(substitute(variable)), w))
      break
    }
  }
}
```

```
smallest.window.size(airquality$Ozone)
```

```
## Minimum window size for the variable airquality$Ozone is 5
```

```
smallest.window.size(airquality$Solar.R)
```

```
## Minimum window size for the variable airquality$Solar.R is 2
```

Problem 2

```
longegfr1 <-read.csv("data/longegfr1.csv")
longegfr2 <-read.csv("data/longegfr2.csv")
```

a)

```
colnames(longegfr1)[1] <- toupper(colnames(longegfr1)[1]) # we need the same column name
longegfr <- merge(longegfr1, longegfr2, by=c("ID", "fu.years"), all=TRUE)

longegfr <- longegfr[order( longegfr[,1], longegfr[,2] ),]
stopifnot(all.equal(longegfr, longegfr[order( longegfr[,1], longegfr[,2] ),]))
```

b)

```
average.egfrs <- aggregate(egfr~ID, longegfr, mean)
colnames(average.egfrs)[2] <- "average.egfr"
longegfr <- merge(longegfr, average.egfrs, all=TRUE)

followup.lengths <- aggregate(fu.years~ID, longegfr, max)
colnames(followup.lengths)[2] <- "followup.length"
longegfr <- merge(longegfr, followup.lengths, all=TRUE)
```

```
bins <- c(0, 15, 30, 60, 90, Inf)
print("Number of patients in the following average eGFR ranges: ")
```

```
## [1] "Number of patients in the following average eGFR ranges: "
```

```
table(cut(average.egfrs$average.egfr, bins, dig.lab=4))
```

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

```
paste0("The number of patients with missing average eGFR: ",
       with(longegfr[is.na(longegfr$average.egfr),], length(table(ID)) ))
```

```
## [1] "The number of patients with missing average eGFR: 3"
```

c)

```
count.egfr <- aggregate(egfr~ID, longegfr, length)
colnames(count.egfr)[2] <- "nr.of.egfrs"
longegfr <- merge(longegfr, count.egfr, all=TRUE)

longegfr.15 <- unique(subset(longegfr, average.egfr <= 15, select=c(ID,
          sex, baseline.age, average.egfr, followup.length, nr.of.egfrs)))

# Patient 5 has 11 measurements with missing values.
# I omitted those in the "number of eGFR measurements taken",
# and included in the "maximum follow-up time".
longegfr.15
```

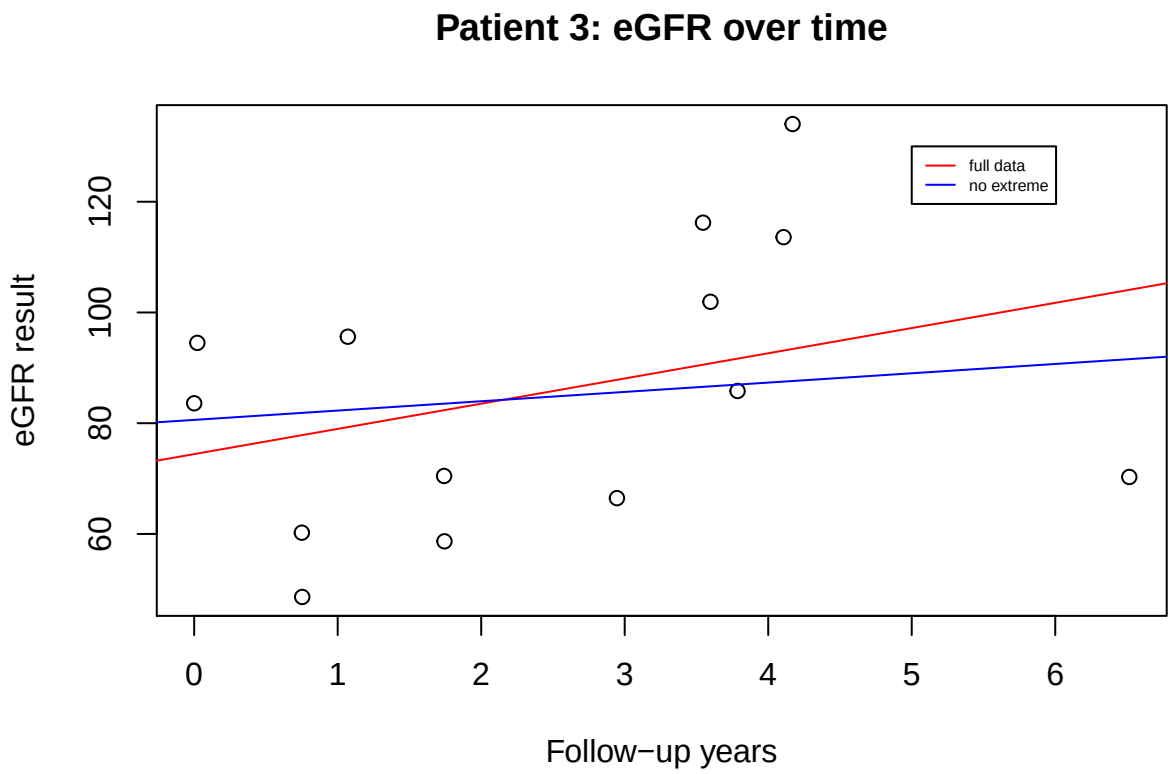
```
##       ID sex baseline.age average.egfr followup.length nr.of.egfrs
## 60     5   1         72.1     13.90892          6.3929          37
## 2298 138   1         75.7     14.87000          6.3628          88
```

d)

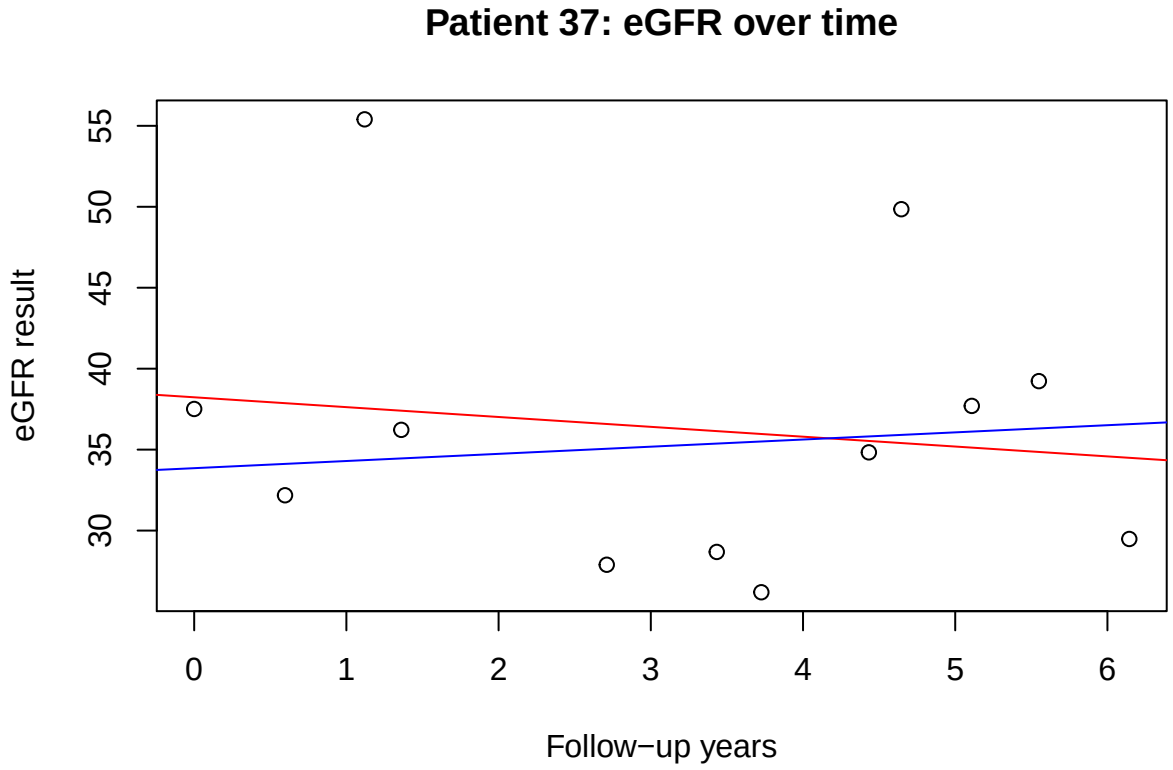
```
egfr.over.time <- function(id){
  subset <- subset(longegfr, ID == id)
  subset.ordered <- subset[order(subset$fu.years),]
  subset.ordered <- na.omit(subset.ordered) # Remove missing values: patient 223
  plot.title <- sprintf("Patient %d: eGFR over time" ,id)
  plot(subset.ordered$fu.years, subset.ordered$egfr, main=plot.title,
       xlab="Follow-up years", ylab="eGFR result")
  reg <- lm(egfr ~ fu.years , data=subset.ordered)
  abline(reg, col='red')
  print("Confidence interval: ")
  print(confint(reg))
  no.extreme <- subset(subset.ordered, egfr > min(egfr) & egfr < max(egfr))
  reg.no.extreme <- lm(egfr ~ fu.years, data = no.extreme)
  abline(reg.no.extreme, col='blue')
  legend(5, 130, legend=c("full data", "no extreme"), lty=c(1,1),
         lwd=c(1,1), col=c("red", "blue"), cex = 0.50)
}

ids = c(3, 37, 162, 223)
for(id in ids){
```

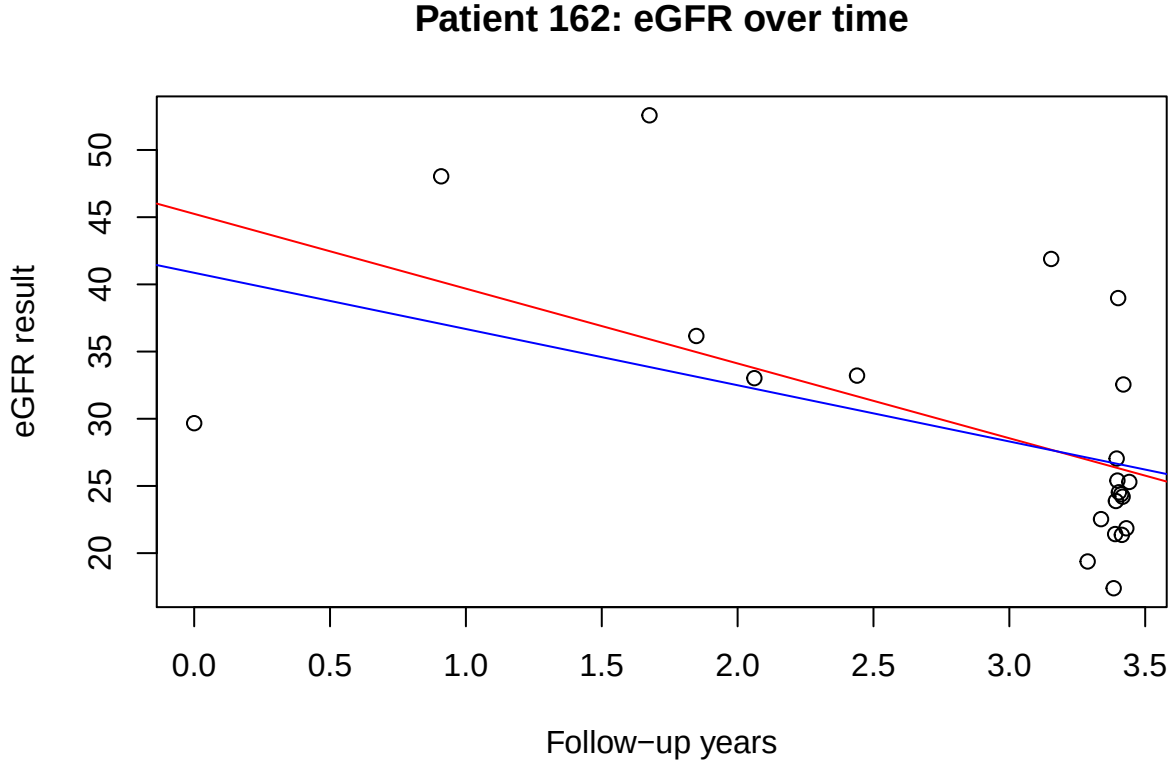
```
  egfr.over.time(id)
}
```



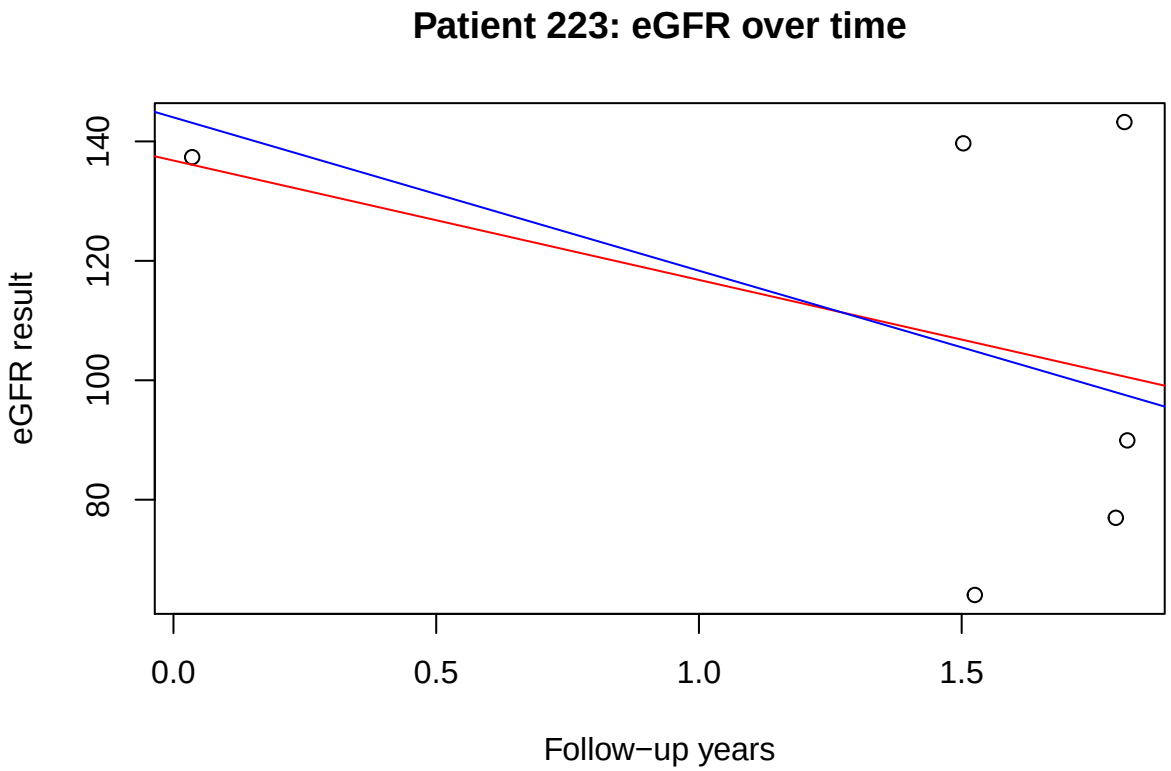

```
## [1] "Confidence interval: "
##                  2.5 %   97.5 %
## (Intercept) 50.623768 98.21718
## fu.years    -3.151128 12.25612
```




```
## [1] "Confidence interval: "
##                  2.5 %   97.5 %
## (Intercept) 26.911518 49.55334
## fu.years    -3.595705  2.37859
```



```
## [1] "Confidence interval: "
##                 2.5 %    97.5 %
## (Intercept) 34.109333 56.382006
## fu.years    -9.257727 -1.872262
```




```
## [1] "Confidence interval: "
##                 2.5 %   97.5 %
## (Intercept)  34.71838 238.8642
## fu.years    -85.93757  45.9659
```

Problem 3

a)

```
egfr.mdrd4 <- function(scr, age, sex, ethnic) {
  sex.coef <- (sex == 'Female')*0.742 + (sex == 'Male')*1
  ethnic.coef <- (ethnic == 'Black')*1.212 + (ethnic == 'Other')*1
  return(175 * scr^(-1.154) * age^(-0.203) * sex.coef * ethnic.coef)
}

egfr.ckdepi <- function(scr, age, sex, ethnic) {
  sex.coef <- (sex == 'Female')*1.018 + (sex == 'Male')*1
  ethnic.coef <- (ethnic == 'Black')*1.159 + (ethnic == 'Other')*1
  kappa <- (sex == 'Female')*0.7 + (sex == 'Male')*0.9
  alpha <- (sex == 'Female')*(-0.329) + (sex == 'Male')*(-0.411)
  return(141 * pmin(scr/kappa, 1)^alpha * pmax(scr/kappa, 1)^(-1.209) *
           0.993^age * sex.coef * ethnic.coef)
}
```

b)

```
scr.data <-read.csv("data/scr.csv")
scr.data <- na.omit(scr.data)
```

```
mdrd4 <- egfr.mdrd4(scr.data$scr, scr.data$age, scr.data$sex, scr.data$ethnic)
ckdepi <- egfr.ckdepi(scr.data$scr, scr.data$age, scr.data$sex, scr.data$ethnic)

mdrd4.mean <- round(mean(mdrd4), 2)
mdrd4.std <- round(sd(mdrd4), 2)

sprintf("MDRD4: Mean: %1.2f Standard deviation: %1.2f", mdrd4.mean, mdrd4.std)
```

```
## [1] "MDRD4: Mean: 59.34 Standard deviation: 47.67"
```

```
ckdepi.mean <- round(mean(ckdepi), 2)
ckdepi.std <- round(sd(ckdepi), 2)

sprintf("CKD-EPI: Mean: %1.2f Standard deviation: %1.2f", ckdepi.mean, ckdepi.std)
```

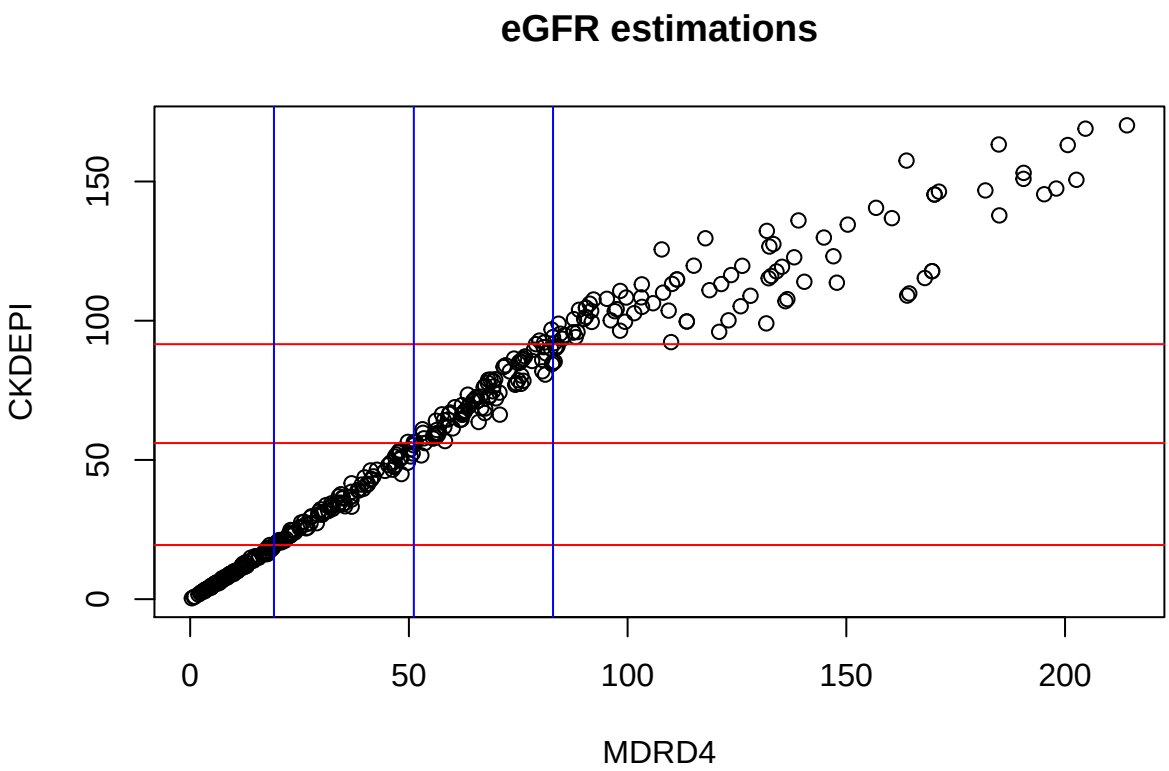
```
## [1] "CKD-EPI: Mean: 58.50 Standard deviation: 42.12"
```

```
pearson <- cor(ckdepi, mdrd4)
sprintf("Pearson correlation coefficient: %1.2f", pearson)
```

```
## [1] "Pearson correlation coefficient: 0.97"
```

c)

```
plot(mdrd4, ckdepi, main="eGFR estimations", xlab="MDRD4", ylab="CKDEPI")
abline(h=median(ckdepi), col='red')
abline(h=quantile(ckdepi, 0.25), col='red')
abline(h=quantile(ckdepi, 0.75), col='red')
abline(v=median(mdrd4), col='blue')
abline(v=quantile(mdrd4, 0.25), col='blue')
abline(v=quantile(mdrd4, 0.75), col='blue')
```



We can observe on the plot that the relationship is linear. It is confirmed by the Pearson correlation coefficient being close to 1 (0.97). However as the numbers grow beyond the 3rd quantile they start to deviate more from the linear line.

Problem 4

a)

```
fit.m1 <- glm(case ~ parity+age, data=infert, family="binomial")
pval <- signif(pchisq(fit.m1$null.deviance - fit.m1$deviance, df=2, lower.tail=FALSE), 3)

sprintf("P-value of the model using parity and age attributes: %1.3f", signif(pval, 3))
```

```
## [1] "P-value of the model using parity and age attributes: 0.989"
```

```
# P-value is very high - model doesn't predict the target variable very well
```

b)

```
fit.m2 <- glm(case ~ parity+age+spontaneous, data=infert, family="binomial")

or.spont <- exp(coef(fit.m2)[4])
ci.spont <- exp(confint(fit.m2))[4, ]
```

```
## Waiting for profiling to be done...
```

```
print("The odds ratio and the confidence interval:")
```

```
## [1] "The odds ratio and the confidence interval:"
```

```r
print(or.spont)
```

```
## spontaneous
##     3.44814
```

```r
# An odds ratio describes how the odds for an event change with a 1 unit increase
# in the "spontaneous variable"
print(ci.spont)
```

```
##    2.5 %   97.5 %
## 2.270900 5.389388
```

```r
pval2 <- pchisq(fit.m1$deviance - fit.m2$deviance, df=1, lower.tail=FALSE)
print("P-value of the model after adding the spontaneous attribute:")
```

```
## [1] "P-value of the model after adding the spontaneous attribute:"
```

```r
print(signif(pval2, 3))
```

```
## [1] 1.35e-09
```

```r
# The P-value has changed to a much smaller number after adding the
# "spontaneous" attribute to the model.
# This means that the "spontaneous" attribute is very significant when
# predicting the inability to get pregnant.
```

c)

```r
# Binomial log-likelihood function
loglik.binom <- function(y.obs, y.pred){
  return(sum(log(y.pred[y.obs==1])) + sum(log(1-y.pred[y.obs==0])))
}
```

```r
sprintf("Deviance of the model M2: %1.3f",
        -2*loglik.binom(infert$case, fit.m2$fitted.values))
```

```
## [1] "Deviance of the model M2: 279.406"
```

```r
null.model <- glm(case ~ 1, data=infert, family="binomial")
sprintf("Null deviance: %1.3f",
        -2 * loglik.binom(infert$case, null.model$fitted.values))
```

```
## [1] "Null deviance: 316.171"
```

```r
# For a comparison:

sprintf("Deviance of the model M1: %1.3f",
        -2*loglik.binom(infert$case, fit.m1$fitted.values))
```

```
## [1] "Deviance of the model M1: 316.150"
```

```r
# Deviance of the model M2 varies much from the null model deviance
# than in the case of model M1. It confirms that the model M2 is more
# powerful in terms of predicting the "case" target.
```

d)

```
# functions from the Lab 3

glm.cv <- function(formula, data, folds) {
  regr.cv <- NULL
  for (fold in 1:length(folds)) {
    regr.cv[[fold]] <- glm(formula, data=data[-folds[[fold]], ], family="binomial")
    }
  return(regr.cv)
}

predict.cv <- function(regr.cv, data, outcome, folds) {
  pred.cv <- NULL
  for (fold in 1:length(folds)) {
    test.idx <- folds[[fold]]
    pred.cv[[fold]] <- data.frame(obs=outcome[test.idx],
                                  pred=predict(regr.cv[[fold]], newdata=data,
                                               type="response")[test.idx])
    }
  return(pred.cv)
  }
```

```
set.seed(1)
folds <- createFolds(infert$case, k=10)

cv.m2 <- glm.cv(case ~ parity+age+spontaneous, data=infert, folds)
pred.cv.m2 <- predict.cv(cv.m2, infert, infert$case, folds)

loglik.sum <- 0 # sum variable
for (fold in 1:length(folds)) {
  loglik.sum <- loglik.sum + loglik.binom(pred.cv.m2[[fold]]$obs, pred.cv.m2[[fold]]$pred)
}
sprintf("Sum of the test log-likelihoods over 10 folds: %1.3f", loglik.sum)
```

```
## [1] "Sum of the test log-likelihoods over 10 folds: -144.513"
```