Assignment 1 Solution

Biomedical Data Science (MATH11174), 22/23, Semester 2

Reproduced by Johnny MyungWon Lee

March 9, 2023

Due on Thursday, 9th of March 2023, 5:00pm

> ❗ Pay Attention
>
> The assignment is marked out of 100 points, and will contribute to ***20%*** of your final mark. The aim of this assignment is to produce a precise report in biomedical studies with the help of statistical and machine learning. Please complete this assignment using **Quarto/Rmarkdown file and render/knit this document only in PDF format** and submit using the **gradescope link on Learn**. You can simply click render on the top left of Rstudio (`Ctrl+Shift+K`). If you cannot render/knit to PDF directly, open **Terminal** in your RStudio (`Alt+Shift+R`) and type `quarto tools install tinytex`, otherwise please follow this link. If you have any code that does not run you will not be able to render nor knit the document so comment it as you might still get some grades for partial code.
> **Clear and reusable code will be rewarded**. Codes without proper indentation, choice of variable identifiers, **comments**, error checking, etc will be penalised. An initial code chunk is provided after each subquestion but **create as many chunks as you feel is necessary** to make a clear report. Add plain text explanations in between the chunks when required to make it easier to follow your code and reasoning. Ensure that all answers containing multiple values should be presented and formatted with `kable()` and `kable_styling()` or using Markdown syntax. All plots must be displayed with clear title, label and legend.

Problem 1 (25 points)

Files `longegfr1.csv` and `longegfr2.csv` (available on Assessment > Assignment 1) contain information regarding a longitudinal dataset containing records on 250 patients. For each subject, eGFR (**estimated glomerular filtration rate, a measure of kidney function**) was collected at irregularly spaced time points: variable `fu.years` contains the follow-up time (that is, the distance from baseline to the date when each eGFR measurement was taken, expressed in years).

Problem 1.a (4 points)

- Convert the files to data table format and merge in an appropriate way into a single data table.
- Order the observations according to subject identifier and follow-up time.
- Print first 10 values of the new dataset using `head()`.

```
longegfr1.dt <- fread("data_assignment1/longegfr1.csv")
longegfr2.dt <- fread("data_assignment1/longegfr2.csv")
#merging two dataset by id and follow-up years
longeGFR.dt <- merge(longegfr1.dt, longegfr2.dt, by.x = c('id', 'fu.years'),
                     by.y = c('ID','fu.years')) %>% .[order(id,fu.years)]
kable(head(longeGFR.dt, 10), caption = "Longitudinal eGFR of 250 patients") |>
  kable_styling(full_width = F, position = "center", latex_options = "hold_position")
```

Table 1: Longitudinal eGFR of 250 patients

id	fu.years	sex	baseline.age	egfr
1	0.0000	0	65.5	76.48
1	0.1533	0	65.5	47.36
1	0.6899	0	65.5	94.87
1	1.1882	0	65.5	52.12
1	1.8398	0	65.5	91.91
1	2.2806	0	65.5	76.52
1	3.3895	0	65.5	46.79
1	3.7563	0	65.5	35.56
1	4.5229	0	65.5	28.41
1	5.3607	0	65.5	20.85

Problem 1.b (6 points)

- Compute the average eGFR and length of follow-up for each patient.
- Print first 10 values of the new dataset using `head()`.
- Tabulate the number of patients with average eGFR in the following ranges: $(0, 15]$, $(15, 30]$, $(30, 60]$, $(60, 90]$, $(90, \texttt{max(eGFR)})$.
- Count and report the number of patients with missing average eGFR.

```
#Computing the average eGFR and length of follow-up for each patient
longeGFR.dt[, c('avg.egfr', 'max.fu.years') :=
            .(mean(egfr, na.rm = T), max(fu.years, na.rm = T)),
          by = id]
mean.length.eGFR <- data.table(id = unique(longeGFR.dt[,id]),
                               avg.eGFR = unique(longeGFR.dt[,avg.egfr]),
                      length.fu.years = unique(longeGFR.dt[,max.fu.years]))
```

```
Warning in as.data.table.list(x, keep.rownames = keep.rownames, check.names
= check.names, : Item 3 has 233 rows but longest item has 247; recycled with
remainder.
```

```
kable(head(mean.length.eGFR, 10),
      caption = "average eGFR and length of follow-up of 250 patients") |>
  kable_styling(full_width = F, position = "center", latex_options = "hold_position")
```

Table 2: average eGFR and length of follow-up of 250 patients

id	avg.eGFR	length.fu.years
1	43.04333	6.4586
2	38.93294	2.0698
3	85.72000	6.5161
4	76.59308	5.2786
5	13.90892	5.8262
6	85.66435	6.2313
7	64.21758	5.8453
8	66.28333	1.5606
9	86.35750	5.8700
10	107.00429	5.1964

```
#Tabulating the number of patients with average eGFR in the given ranges
rangeGFR <- table(cut(mean.length.eGFR$avg.eGFR,
```

```
                        c(0,15,30,60,90, max(longeGFR.dt$egfr , na.rm=TRUE))))
kable(t(rangeGFR), caption = "Number of patients with average eGFR") |>
  kable_styling(full_width = F, position = "center", latex_options = "hold_position")
```

Table 3: Number of patients with average eGFR

(0,15]	(15,30]	(30,60]	(60,90]	(90,175]
2	9	84	86	66

```
#Counting the number of patients with missing average eGFR
cat("number of patients with missing average eGFR:",
    sum(is.na(longeGFR.dt$avg.egfr)))
```

```
number of patients with missing average eGFR: 0
```

Note that we removed all the `NA` values when calculating the average eGFR and length of follow-up for each patient. Thus, the number of patients with missing average eGFR is 0.

Problem 1.c (6 points)

- For patients with average eGFR in the (90, `max(eGFR)`) range, collect their identifier, sex, age at baseline, average eGFR, time of last eGFR reading and number of eGFR measurements taken in a data table.
- Print the summary of the new dataset.

```
#Tabulating patients with average eGFR in the (90, max(eGFR)) range
hi.egfr.dt <- longeGFR.dt[!(is.na(egfr)), 'num.fu' := length(egfr), by = id] %>%
  #Computing the average eGFR & the time of last eGFR recording by id
                       .[(avg.egfr > 90) & (fu.years==max.fu.years),
  #Setting orders given by the question
                        .(id, sex, baseline.age, avg.egfr, max.fu.years, num.fu)]
summary(hi.egfr.dt)
```

```
       id              sex          baseline.age      avg.egfr
 Min.   : 10.00   Min.   :0.0000   Min.   :22.10   Min.   : 90.04
 1st Qu.: 86.25   1st Qu.:0.0000   1st Qu.:47.20   1st Qu.: 99.13
 Median :144.00   Median :0.0000   Median :55.20   Median :109.81
 Mean   :141.88   Mean   :0.3333   Mean   :55.27   Mean   :112.13
 3rd Qu.:197.50   3rd Qu.:1.0000   3rd Qu.:63.80   3rd Qu.:123.20
```

```
Max.   :250.00   Max.   :1.0000   Max.   :90.90   Max.   :147.69
 max.fu.years       num.fu
Min.   :0.000   Min.   : 1.00
1st Qu.:1.607   1st Qu.: 5.00
Median :4.093   Median : 8.00
Mean   :3.688   Mean   :11.91
3rd Qu.:5.513   3rd Qu.:13.75
Max.   :6.590   Max.   :57.00
```

Problem 1.d (9 points)

For patients 3, 37, 162 and 223:

- Plot the patient's eGFR measurements as a function of time.
- Fit a linear regression model and add the regression line to the plot.
- Report the 95% confidence interval for the regression coefficients of the fitted model.
- Using a different colour, plot a second regression line computed after removing the extreme eGFR values (one each of the highest and the lowest value).

(All plots should be displayed in the same figure. The plots should be appropriately labelled and the results should be accompanied by some explanation as you would communicate it to a colleague with a medical background with a very little statistical knowledge.)

```r
patients <- c(3, 37, 162, 223)

par(mfrow=c(2,2), mar = c(1.5,1.5,1.5,1.5), oma = c(4,4,2.5,2.5))
for (i in patients){
  data.i <- longeGFR.dt[id==i,]
  #fitting the time series of eGFR of each patient
  fit1 <-lm(egfr ~ fu.years, data = data.i)
  data.i.new <- data.i %>%
    #arranging by ascending order to find the minima and maxima
    arrange(egfr) %>% na.omit() %>%
    #removing the two extreme values
    slice(2:(n() - 1))
  #fitting the time series of eGFR after removal of extremas
  fit2 <- lm(egfr ~ fu.years, data = data.i.new)
  conf.interval <- data.frame(confint(fit1)["fu.years",],
                              confint(fit2)["fu.years",])
  cat("95% confidence interval of fit1 and fit2 when id =", i, "\n")
  print(conf.interval)
```

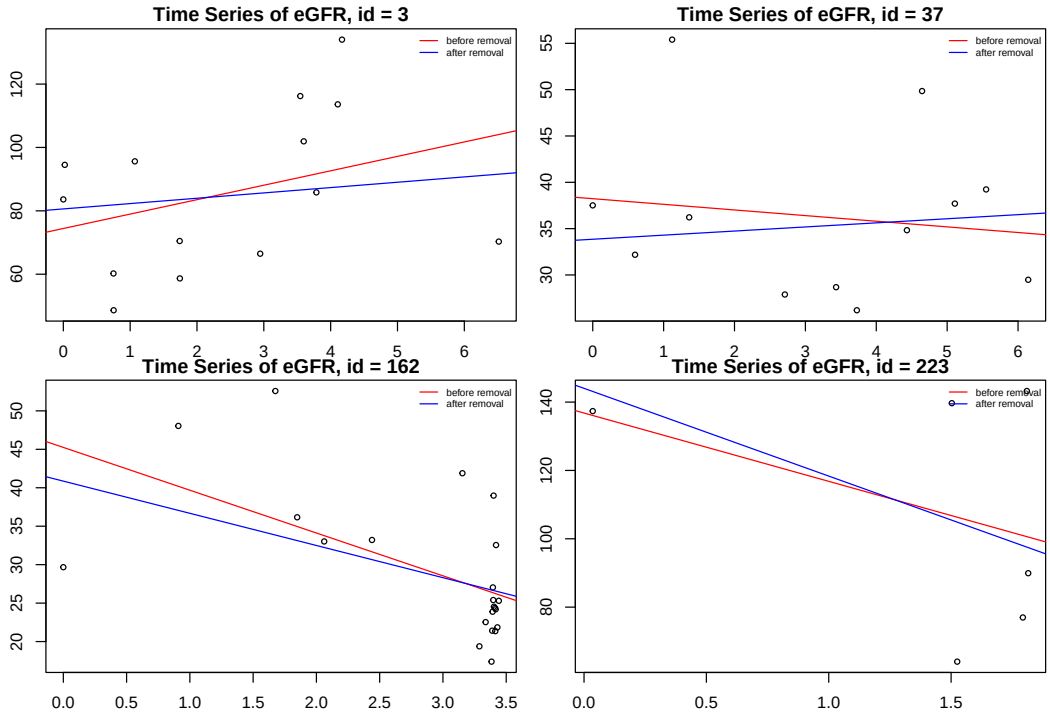
```r
    plot(egfr ~ fu.years, data = data.i,
         main = paste("Time Series of eGFR, id =", i), cex = 0.7)
    abline(fit1, col = "red")
    abline(fit2, col = "blue")
    legend("topright", legend = c("before removal", "after removal"),
           col = c("red", "blue"), lty = 1, cex = 0.6, bty = "n")
  }
```

```
95% confidence interval of fit1 and fit2 when id = 3
       confint.fit1...fu.years.... confint.fit2...fu.years....
2.5 %                   -3.151128                   -5.441923
97.5 %                  12.256121                    8.809287

95% confidence interval of fit1 and fit2 when id = 37
       confint.fit1...fu.years.... confint.fit2...fu.years....
2.5 %                   -3.595705                   -1.994624
97.5 %                   2.378590                    2.879692

95% confidence interval of fit1 and fit2 when id = 162
       confint.fit1...fu.years.... confint.fit2...fu.years....
2.5 %                   -9.257727                  -7.5621245
97.5 %                  -1.872262                  -0.8057698

95% confidence interval of fit1 and fit2 when id = 223
       confint.fit1...fu.years.... confint.fit2...fu.years....
2.5 %                   -85.93757                   -111.35297
97.5 %                   45.96590                    60.00585
```



eGFR stands for estimated Glomerular Filtration Rate which measures the functionality of patient's kidney and 60 or more is considered normal according to National Kidney Foundation. Also, the average measure of eGFR decreases with the decrease in age. Above, we fitted linear regression models to predict the eGFR measurements of four different patients as a function of time, i.e.

$$\texttt{eGFR} = \beta_0 + \beta_1 \times \texttt{time}$$

In the plot, patients have different number of measurements (data points) over different time range and this indicates that all patients are in different condition at the current measure. Thus, we will describe them one by one.

For patient 3 we obtained a 95% confidence interval of $(-3.15, 12.26)$ which is broad. The confidence interval without taking into account the extreme values is still reasonably broad, $(-5.44, 8.81)$. Thus, there seems to be a lot of variation in eGFR values for this patient. We can see from the plot that the eGFR value increases each time a measurement is taken. Removing the extreme values slows down the increment slightly and stabilises it more to give a smaller difference in filtration rate as time goes by. With the noticeable trend, we can conclude that this indicates good kidney health of the patient

Secondly, the patient seem to have a bad kidney functionality or either considered old as the values of the plot suggested. The 95 confidence interval for patient 37 for both linear models are relatively small, $(-3.60, 2.38)$ for the full data and $(-2.00, 2.88)$. As a result, it indicates that the eGFR value of the patient 37 is not varying largely during the time of measurement. The regression line disregarding the highest and lowest value does not change much too.

We elaborate for the patient 162, the general trend of the graph is decreasing through out the years and it suggests that the kidney functionality of the patient is becoming worse. From the confidence interval, $(-9.26, -1.87)$ we can see that the eGFR values tends to decrease as more measurements are taken. The confidence interval without the extreme values is of a similar width, $(-7.56, -0.81)$, however the values are slightly close to zero, indicating a less steep decline in eGFR. Moreover, the fitted line also indicates similar gradient and see more values were collected in the recent years. This indicates that the patient can possibly be in a serious state undergoing intensive care with multiple measurements before medication.

Lastly, we elaborate for the patient 233. Similar to patient 162, the patient shows a decreasing trend with steep gradient but the age or the condition of the kidney seem to be relatively young and better. Also, severity of the kidney conditions is not in a serious stage as all the measurements are above 60 and the follow-up years is shorter than the rest of the patients. Although the patient is having high eGFR values, the patient should be aware of its kidney condition and take medication to prevent further decrease in the kidney functionality. The change in the confidence interval is drastic as only a few eGFR measurements were taken which explains the wide confidence interval for the regression intervals compared to the other patients.

Problem 2 (25 points)

The MDRD4 and CKD-EPI equations are two different ways of estimating the glomerular filtration rate (eGFR) in adults:

$$\texttt{MDRD4} = 175 \times (\texttt{SCR})^{-1.154} \times \texttt{AGE}^{-0.203}[\times 0.742 \text{ if female}][\times 1.212 \text{ if black}]$$

, and

$$\texttt{CKD-EPI} = 141 \times \min(\texttt{SCR}/\kappa, 1)^{\alpha} \times \max(\texttt{SCR}/\kappa, 1)^{-1.209} \times 0.993^{\texttt{AGE}}[\times 1.018 \text{ if female}][\times 1.159 \text{ if black}]$$

, where:

- `SCR` is serum creatinine (in mg/dL)
- κ is 0.7 for females and 0.9 for males
- α is -0.329 for females and -0.411 for males

Problem 2.a (7 points)

For the `scr.csv` dataset,

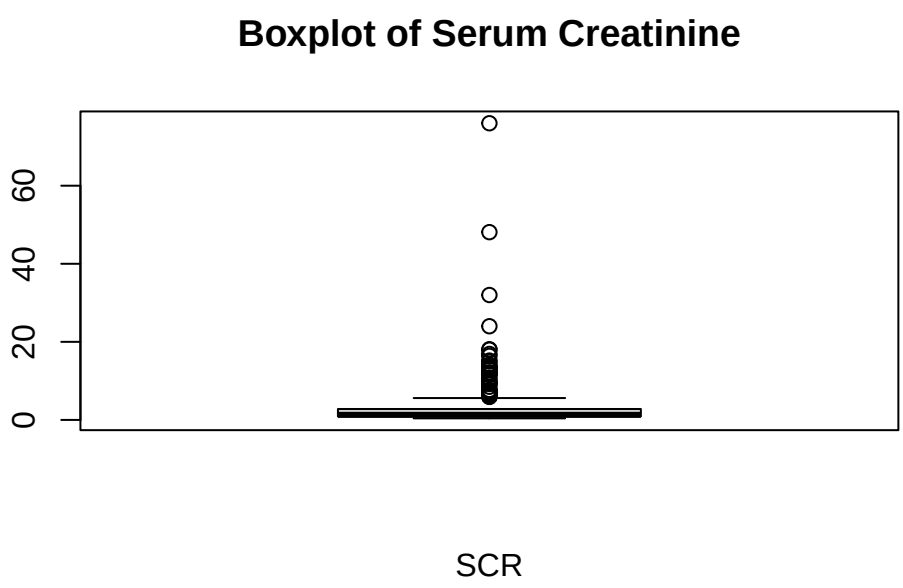
- Examine a summary of the distribution of serum creatinine and report the inter-quartile range.
- If you suspect that some serum creatinine values may have been reported in µmol/L convert them to mg/dL by dividing by 88.42.
- Justify your choice of values to convert and examine the distribution of serum creatinine following any changes you have made.

```
scr.dt <- fread('data_assignment1/scr.csv')
```

```
summary(scr.dt$scr)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  0.400   0.900   1.300   3.072   2.800  76.000      18
```

```
boxplot(scr.dt$scr, main = "Boxplot of Serum Creatinine", xlab = "SCR")
```




```r
scr.iqr <- IQR(scr.dt$scr, na.rm = T)
cat("The inter-quartile range is ", scr.iqr)
```

```
The inter-quartile range is  1.9
```

```r
scr.q3 <-  quantile(scr.dt$scr, 0.75, na.rm = T)
```

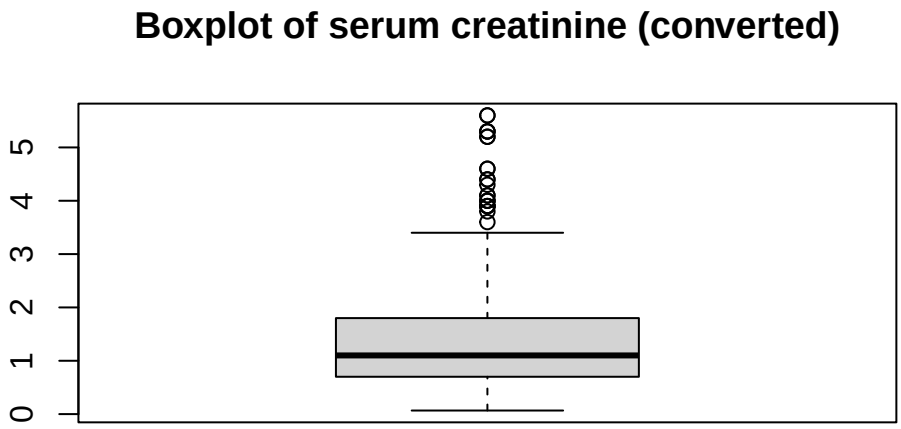
From the summary and boxplot above, most of the `SCR` values lie between 0 and 5. We also discovered that the inter-quartile range is 1.9 with 75 of the `SCR` measurements between $(0.9, 2.8)$. Normal SCR levels are known to lie between $(0.74, 1.35)$ mg/dL for adult males and $(0.59, 1.04)$ mg/dL for females source : Mayo Clinic. With that we follow the formal way of defining the outliers by setting the cutoff point that is greater than the 3^{rd} quantile with addition of 1.5 times of the interquartile range, i.e.

$$\text{outlier-cutoff} = Q3 + 1.5 \times IQR$$

```r
scr.dt[, 'scr2' := ifelse(scr > scr.q3 + 1.5*scr.iqr, scr/88.42, scr)]
summary(scr.dt$scr2)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
0.06673 0.70000 1.10000 1.39813 1.80000 5.60000      18
```

```r
boxplot(scr.dt$scr2, main = "Boxplot of serum creatinine (converted)")
```



Problem 2.b (11 points)

- Compute the eGFR according to the two equations using the newly converted `SCR` values.
- Report (rounded to the second decimal place) mean and standard deviation of the two eGFR vectors and their Pearson correlation coefficient.
- Report the same quantities according to strata of MDRD4 eGFR: $(0-60)$, $(60-90)$ and (>90).
- Print first 15 values for both datasets using `head()`.

```r
#computing MDRD4
#removing missing values
scr.mdrd <- scr.dt %>% copy() %>% #na.omit() %>%
  #equating into the equation
  .[, mdrd4:= 175 * scr2^(-1.154) * age^(-0.203)] %>%
  #special case for sex = Female
  .[, mdrd4:= ifelse(sex == "Female", mdrd4 * 0.742, mdrd4)] %>%
  #special case for ethnic = Black
  .[, mdrd4:= ifelse(ethnic == "Black", mdrd4 * 1.212, mdrd4)]
kable(head(scr.mdrd, 15), caption = "MDRD4 Calculation based on New SCR") |>
```

```
  kable_styling(full_width = F, position = "center", latex_options = "hold_position")
```

Table 4: MDRD4 Calculation based on New SCR

age	scr	sex	ethnic	scr2	mdrd4
48	1.2	Female	Other	1.2000000	47.94848
7	0.8	Male	Black	0.8000000	184.85020
62	1.8	Female	NA	1.8000000	NA
48	3.8	Female	Other	3.8000000	12.67885
51	1.4	Male	Other	1.4000000	53.42808
60	1.1	Male	Other	1.1000000	68.28199
68	24.0	Male	Other	0.2714318	334.65264
24	1.1	Male	Black	1.1000000	99.67601
52	1.9	Female	Other	1.9000000	27.75950
53	7.2	Male	Other	0.0814295	1412.43228
50	4.0	Female	Other	4.0000000	11.85151
63	2.7	Male	Other	2.7000000	23.98712
68	2.1	Female	Other	2.1000000	23.42079
68	4.6	Male	Other	4.6000000	12.77074
68	4.1	Male	Black	4.1000000	17.67620

```
#computing CKD_EPI
#removing missing values
scr.ckd <- scr.dt %>% copy() %>% #na.omit() %>%
  #computing the kappa for min and max
  .[, kappa := ifelse(sex=="Female", scr2/0.7, scr2/0.9)] %>%
  .[, minkappa := ifelse(kappa < 1, kappa, 1)] %>%
  .[, maxkappa := ifelse(kappa > 1, kappa, 1)] %>%
  #equating to the equation based on sex
  .[, ckd.epi := ifelse(sex=="Female",
    141 * (minkappa^(-0.329)) * (maxkappa^(-1.209)) *
      0.993^(age) * 1.018,
    141 * (minkappa^(-0.411)) * (maxkappa^(-1.209)) *
      0.993^(age))] %>%
  #special case for ethnic = Black
  .[, ckd.epi:= ifelse(ethnic == "Black", ckd.epi*1.159, ckd.epi)]
kable(head(scr.ckd, 15), caption = "CKD-EPI Calculation based on New SCR") |>
  kable_styling(full_width = F, position = "center", latex_options = "hold_position")
```

Table 5: CKD-EPI Calculation based on New SCR

age	scr	sex	ethnic	scr2	kappa	minkappa	maxkappa	ckd.epi
48	1.2	Female	Other	1.2000000	1.7142857	1.0000000	1.714286	53.39791
7	0.8	Male	Black	0.8000000	0.8888889	0.8888889	1.000000	163.29428
62	1.8	Female	NA	1.8000000	2.5714286	1.0000000	2.571429	NA
48	3.8	Female	Other	3.8000000	5.4285714	1.0000000	5.428571	13.25244
51	1.4	Male	Other	1.4000000	1.5555556	1.0000000	1.555556	57.76186
60	1.1	Male	Other	1.1000000	1.2222222	1.0000000	1.222222	72.57875
68	24.0	Male	Other	0.2714318	0.3015909	0.3015909	1.000000	143.12883
24	1.1	Male	Black	1.1000000	1.2222222	1.0000000	1.222222	108.32282
52	1.9	Female	Other	1.9000000	2.7142857	1.0000000	2.714286	29.78779
53	7.2	Male	Other	0.0814295	0.0904773	0.0904773	1.000000	260.85043
50	4.0	Female	Other	4.0000000	5.7142857	1.0000000	5.714286	12.28181
63	2.7	Male	Other	2.7000000	3.0000000	1.0000000	3.000000	23.99839
68	2.1	Female	Other	2.1000000	3.0000000	1.0000000	3.000000	23.58719
68	4.6	Male	Other	4.6000000	5.1111111	1.0000000	5.111111	12.16671
68	4.1	Male	Black	4.1000000	4.5555556	1.0000000	4.555556	16.20597

```r
#Addding MDRD4 and CKD-EPI values
scr.dt <- scr.dt %>% .[, mdrd4 := scr.mdrd$mdrd4] %>%
  .[, ckd.epi := scr.ckd$ckd.epi]
#defining a function that calculates the statistics between MDRD4 and CKD-EPI
two.egfr <- function(dataset, range=""){
  vals <- with(dataset, t(c(mdrd.mean = mean(mdrd4, na.rm = T),
                            mdrd.sd = sd(mdrd4, na.rm = T),
                            ckdepi.mean = mean(ckd.epi, na.rm = T),
                            ckdepi.sd = sd(ckd.epi, na.rm = T),
                            correlation = cor(mdrd4, ckd.epi,
                                              use = 'complete.obs'))))

  kable(data.frame(round(vals,2)),
        caption = paste("Statistics of MDRD4 and CKD-EPI", range)) |>
    kable_styling(full_width = F, position = "center", latex_options = "hold_position")
}

two.egfr(scr.dt)
```

Taking a look at the overall statistics between the computed values for `MDRD4` and `CKD-EPI` values, the values are not similar. However, by looking at the correlation values by different strata, we clearly see that there is strong positive relationship between the two computed

Table 6: Statistics of MDRD4 and CKD-EPI

mdrd.mean	mdrd.sd	ckdepi.mean	ckdepi.sd	correlation
188.98	359.03	85.84	64.27	0.86

values. It is reasonable to suspect as both are used to determine the eGFR values. This suggests that we want to further investigate the different strata of the values.

```
two.egfr(scr.dt[mdrd4 <= 60], "(0, 60)")
```

Table 7: Statistics of MDRD4 and CKD-EPI (0, 60)

mdrd.mean	mdrd.sd	ckdepi.mean	ckdepi.sd	correlation
31.9	15.14	33.15	16.72	0.99

```
two.egfr(scr.dt[mdrd4 > 60 & mdrd4 <= 90], "(60, 90)")
```

Table 8: Statistics of MDRD4 and CKD-EPI (60, 90)

mdrd.mean	mdrd.sd	ckdepi.mean	ckdepi.sd	correlation
73.41	8.4	80.18	10.42	0.93

```
two.egfr(scr.dt[mdrd4 >= 90], "(> 90)")
```

Table 9: Statistics of MDRD4 and CKD-EPI (> 90)

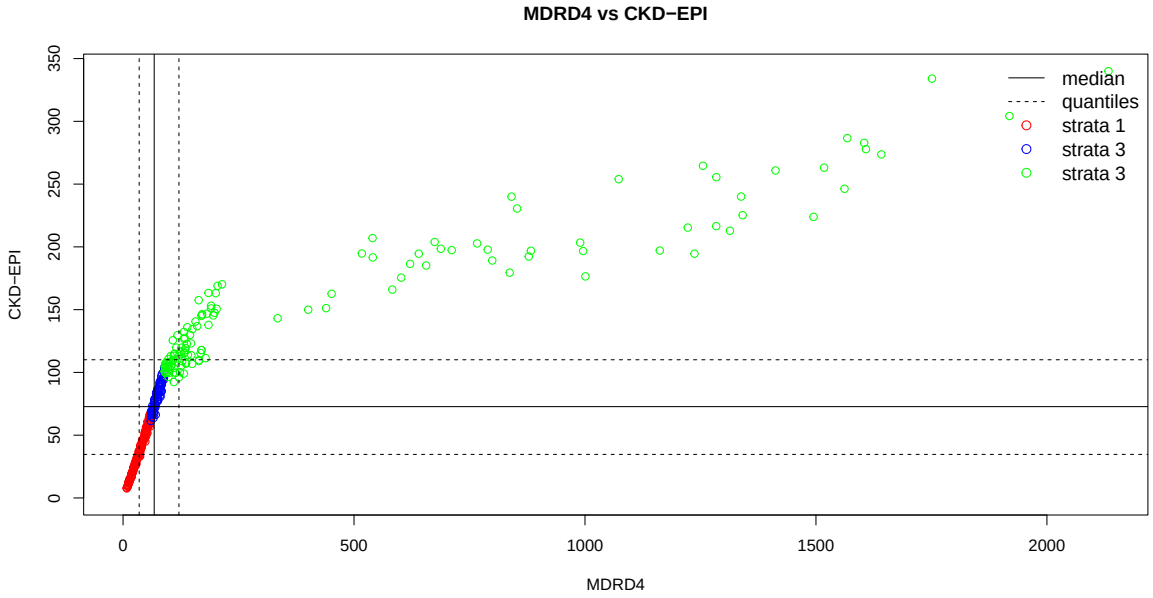
mdrd.mean	mdrd.sd	ckdepi.mean	ckdepi.sd	correlation
466.01	513.25	156.02	57.42	0.95

From the tables above, we can clearly see that there is a similarities in the mean and standard deviation values between $(0 - 90)$. We still observe strong positive correlation as creating strata increases linear dependency in all three cases. Looking at the values above 90, we can see that the values differ greatly here but the standard deviation in for `CKD-EPI` is much smaller compared to `MDRD4`. Therefore, we can conclude that the `CKD-EPI` values should be employed more than `MDRD4`.

Problem 2.c (7 points)

- Produce a scatter plot of the two eGFR vectors, and add vertical and horizontal lines (i.e.) corresponding to median, first and third quantiles.
- Is the relationship between the two eGFR equations linear? Justify your answer.

```r
#computing the quantiles for MDRD4
scr.dt <- scr.dt %>% na.omit()
firstmdrd <- quantile(scr.dt$mdrd4)[2]
secondmdrd <- quantile(scr.dt$mdrd4)[3]
thirdmdrd <- quantile(scr.dt$mdrd4)[4]
#computing the quantiles for CKD-EPI
firstckd <- quantile(scr.dt$ckd.epi)[2]
secondckd <- quantile(scr.dt$ckd.epi)[3]
thirdckd <- quantile(scr.dt$ckd.epi)[4]
#scatter plot of MDRD vs CKD-EPI by strata
plot(scr.dt[mdrd4 <= 60]$mdrd4, scr.dt[mdrd4 <= 60]$ckd.epi,
     main = "MDRD4 vs CKD-EPI", xlab="MDRD4", ylab = "CKD-EPI", col = "red",
     xlim = c(0, max(scr.dt$mdrd4)), ylim =c(0, max(scr.dt$ckd.epi)))
points(scr.dt[mdrd4 > 60 & mdrd4 <= 90]$mdrd4,
       scr.dt[mdrd4 > 60 & mdrd4 <= 90]$ckd.epi, col="blue")
points(scr.dt[mdrd4 > 90]$mdrd4, scr.dt[mdrd4 > 90]$ckd.epi, col="green")
#adding the quantiles of MDRD4
abline(v = c(firstmdrd, secondmdrd, thirdmdrd), lty=c(2,1,2))
#adding the quantiles of CKD-EPI
abline(h = c(firstckd, secondckd, thirdckd), lty=c(2,1,2))
legend("topright", legend = c("median", "quantiles", "strata 1",
                              "strata 3", "strata 3"),
         lty = c(1,2, NA, NA, NA), pch = c(NA, NA, 1,1,1), xpd = TRUE,
         col = c("black", "black", "red", "blue", "green"), cex = 1.2, bty = "n")
```



In the scatter plot, we can observe the linear relationship between `MDRD4` and `CKD-EPI`. The high variance is observable in the 3^{rd} strata but the variance in `MDRD4` is much greater than `CKD-EPI`. This result leads from the previous part and shows that our deduction was correct.

Problem 3 (31 points)

You have been provided with electronic health record data from a study cohort. Three CSV (Comma Separated Variable) files are provided on learn.

The first file is a cohort description file `cohort.csv` file with fields:

- `id` = study identifier
- `yob` = year of birth
- `age` = age at measurement
- `bp` = systolic blood pressure
- `albumin` = last known albuminuric status (categorical)
- `diabetes` = diabetes status

The second file `lab1.csv` is provided by a laboratory after measuring various biochemistry levels in the cohort blood samples. Notice that a separate lab identifier is used to anonymise results from the cohort. The year of birth is also provided as a check that the year of birth aligns between the two merged sets.

- `LABID` = lab identifier
- `yob` = year of birth
- `urea` = blood urea
- `creatinine` = serum creatinine
- `glucose` = random blood glucose

To link the two data files together, a third linker file `linker.csv` is provided. The linker file includes a `LABID` identifier and the corresponding cohort `id` for each person in the cohort.

Problem 3.a (6 points)

- Using all three files provided on learn, load and merge to create a single data table based dataset `cohort.dt`. This will be used in your analysis.
- Perform assertion checks to ensure that all identifiers in `cohort.csv` have been accounted for in the final table and that any validation fields are consistent between sets.
- After the checks are complete, drop the identifier that originated from `lab1.csv` dataset `LABID`.
- Ensure that a single `yob` field remains and rename it to `yob`.
- Ensure that the `albumin` field is converted to a factor and the ordering of the factor is `1="normo"`, `2="micro"`, `3="macro"`.
- Print first 10 values of the new dataset using `head()`.

```r
cohort <- fread('data_assignment1/cohort.csv', stringsAsFactors = F)
#setting albumin as factor
cohort$albumin <- factor(cohort$albumin, levels = c("normo", "micro","macro"))
link <- fread('data_assignment1/linker.csv', stringsAsFactors = F)
lab1 <- fread('data_assignment1/lab1.csv', stringsAsFactors = F)
#merging cohort.csv and link.csv first
cohort.dt <- merge(cohort, link)
#merging lab1 by LABID
diab.dt <- merge(cohort.dt, lab1, by = 'LABID')
#Performing assertive check
assertcheck <- c(all(diab.dt$id %in% link$id),
                 all(diab.dt$id %in% cohort$id),
                 #assertive check of the year of birth field
                 all(diab.dt$yob.x %in% cohort$yob),
                 all(diab.dt$yob.y %in% cohort$yob),
                 all(diab.dt$yob.x %in% lab1$yob),
                 all(diab.dt$yob.y %in% lab1$yob),
                 #assertive check of the LABID field
                 all(diab.dt$LABID %in% lab1$LABID),
                 all(diab.dt$LABID %in% link$LABID))
cat("Out of 8 assertive checks, we have", sum(assertcheck), "passed")
```

```
Out of 8 assertive checks, we have 8 passed
```

```r
#removing yob.y
diab.dt$yob.y <- NULL
setnames(diab.dt, 'yob.x', 'yob')
diab.dt <- diab.dt[,-1]
kable(head(diab.dt, 10), caption = "Complete Diabetes Dataset") |>
  kable_styling(full_width = F, position = "center", latex_options = "hold_position")
```

Table 10: Complete Diabetes Dataset

id	yob	age	bp	diabetes	albumin	urea	creatinine	glucose
PID_285	1986	33	80	0	normo	37.0	106.104	100
PID_153	1980	39	70	1	normo	20.0	70.736	121
PID_13	1951	68	70	1	micro	72.0	185.682	208
PID_110	1965	54	70	1	NA	50.1	167.998	233
PID_222	1953	66	70	1	micro	30.0	150.314	248
PID_103	2002	17	60	0	normo	32.0	185.682	92
PID_200	1954	65	80	0	normo	37.0	132.630	92
PID_378	1955	64	70	0	normo	27.0	61.894	97
PID_267	1964	55	80	0	normo	17.0	106.104	133
PID_271	1996	23	80	0	normo	34.0	97.262	111

Problem 3.b (10 points)

- Create a copy of the dataset where you will impute all missing values.
- Update any missing age fields using the year of birth.
- Perform mean imputation for all other continuous variables by writing a single function called `impute.to.mean()` and impute to mean, impute any categorical variable to the mode.
- Print first 15 values of the new dataset using `head()`.
- Compare each distribution of the imputed and non-imputed variables and decide which ones to keep for further analysis. Justify your answer.

```r
impute.to.mean <- function(x) {
                        # only apply to numeric/integer columns
                        if (is.numeric(x) || is.integer(x)){
                          # find which values are missing
                          na.idx <- is.na(x)
                          # replace NAs with the median computed over the observed values
                          x[na.idx] <- mean(x, na.rm=TRUE)
                        }
                        else {
                          na.idx <- is.na(x)
                          uniqx <- unique(x)
                          # replace NAs with the mode computed over the observed values
                          x[na.idx] <- uniqx[which.max(tabulate(match(x, uniqx)))]
                        }
                        # return the vector with imputed values
                        return(x)
```

```r
}
numcols <- c('age', 'bp','urea','creatinine','glucose', 'albumin')
diab.dt.imputed <- diab.dt %>% copy() %>%
                          .[, age := ifelse(is.na(age), 2023 - yob, as.numeric(age))] %>%
                          .[,(numcols):=lapply(.SD, impute.to.mean),.SDcols = numcols]

kable(head(diab.dt.imputed, 15), caption = "Diabetes after Imputation") |>
  kable_styling(full_width = F, position = "center", latex_options = "hold_position")
```

Table 11: Diabetes after Imputation

id	yob	age	bp	diabetes	albumin	urea	creatinine	glucose
PID_285	1986	33	80	0	normo	37.00000	106.1040	100.0000
PID_153	1980	39	70	1	normo	20.00000	70.7360	121.0000
PID_13	1951	68	70	1	micro	72.00000	185.6820	208.0000
PID_110	1965	54	70	1	normo	50.10000	167.9980	233.0000
PID_222	1953	66	70	1	micro	30.00000	150.3140	248.0000
PID_103	2002	17	60	0	normo	32.00000	185.6820	92.0000
PID_200	1954	65	80	0	normo	37.00000	132.6300	92.0000
PID_378	1955	64	70	0	normo	27.00000	61.8940	97.0000
PID_267	1964	55	80	0	normo	17.00000	106.1040	133.0000
PID_271	1996	23	80	0	normo	34.00000	97.2620	111.0000
PID_105	1964	55	90	1	normo	88.00000	176.8400	143.0000
PID_375	1940	79	80	0	normo	44.00000	106.1040	111.0000
PID_89	1961	58	110	0	macro	52.00000	194.5240	251.0000
PID_24	1998	21	70	0	normo	57.42572	271.6664	148.0365
PID_349	1981	38	80	0	normo	19.00000	44.2100	99.0000

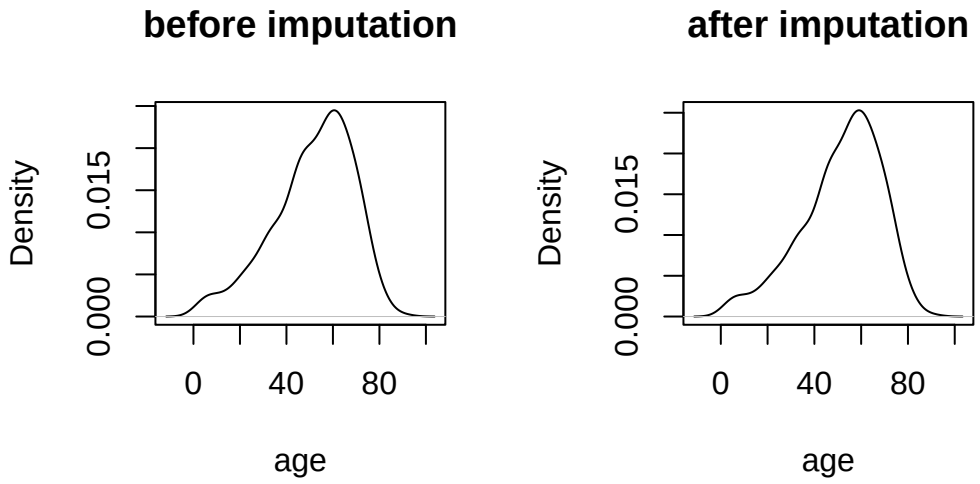
```r
#Before Imputation
summary(diab.dt[, .SD, .SDcols = numcols])
```

```
      age              bp              urea          creatinine     
 Min.   : 2.00   Min.   : 50.00   Min.   :  1.50   Min.   :  35.37  
 1st Qu.:42.00   1st Qu.: 70.00   1st Qu.: 27.00   1st Qu.:  79.58  
 Median :55.00   Median : 80.00   Median : 42.00   Median : 114.95  
 Mean   :51.48   Mean   : 76.47   Mean   : 57.43   Mean   : 271.67  
 3rd Qu.:64.50   3rd Qu.: 80.00   3rd Qu.: 66.00   3rd Qu.: 247.58  
 Max.   :90.00   Max.   :180.00   Max.   :391.00   Max.   :6719.92  
 NA's   :9       NA's   :12       NA's   :19       NA's   :17       
    glucose      albumin    
 Min.   : 22   normo:199  
 1st Qu.: 99   micro:130  
 Median :121   macro: 25  
 Mean   :148   NA's : 46  
 3rd Qu.:163              
 Max.   :490              
 NA's   :44               
```

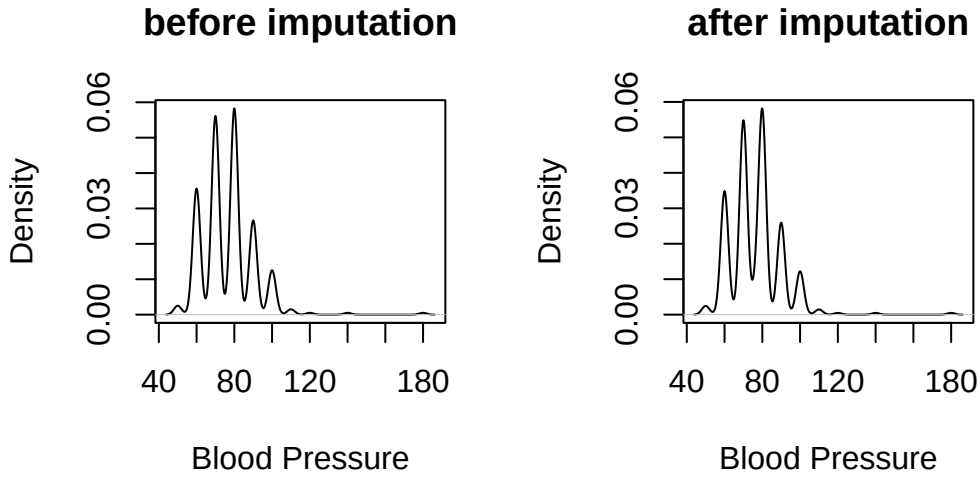
```r
#After Imputation
summary(diab.dt.imputed[, .SD, .SDcols = numcols])
```

```
      age              bp              urea          creatinine     
 Min.   : 2.00   Min.   : 50.00   Min.   :  1.50   Min.   :  35.37  
 1st Qu.:42.00   1st Qu.: 70.00   1st Qu.: 27.00   1st Qu.:  79.58  
 Median :55.00   Median : 78.23   Median : 44.00   Median : 123.79  
 Mean   :51.57   Mean   : 76.47   Mean   : 57.43   Mean   : 271.67  
 3rd Qu.:64.00   3rd Qu.: 80.00   3rd Qu.: 61.75   3rd Qu.: 271.67  
 Max.   :90.00   Max.   :180.00   Max.   :391.00   Max.   :6719.92  
    glucose      albumin    
 Min.   : 22   normo:245  
 1st Qu.:101   micro:130  
 Median :126   macro: 25  
 Mean   :148              
 3rd Qu.:150              
 Max.   :490              
```

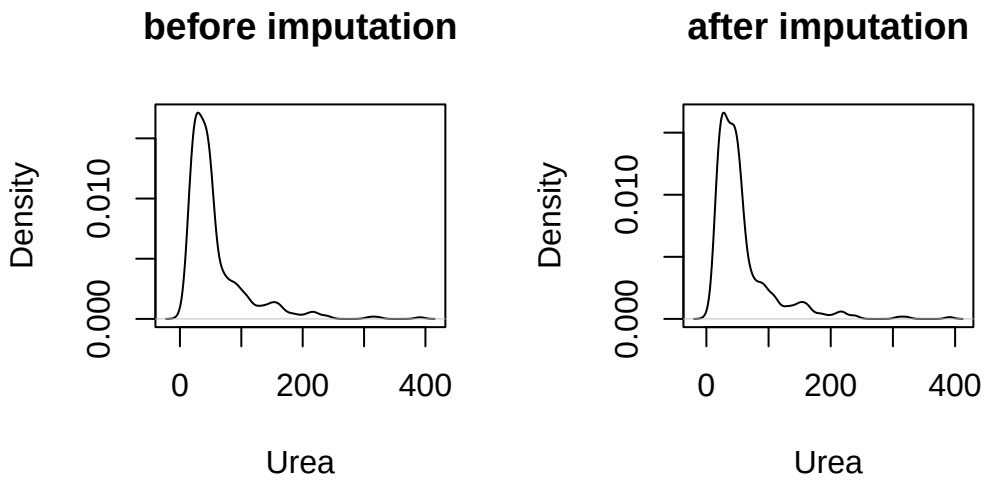
```r
par(mfrow=c(1, 2))
plot(density(diab.dt$age, na.rm = T), main = "before imputation", xlab = "age")
plot(density(diab.dt.imputed$age), main = "after imputation", xlab = "age")
```



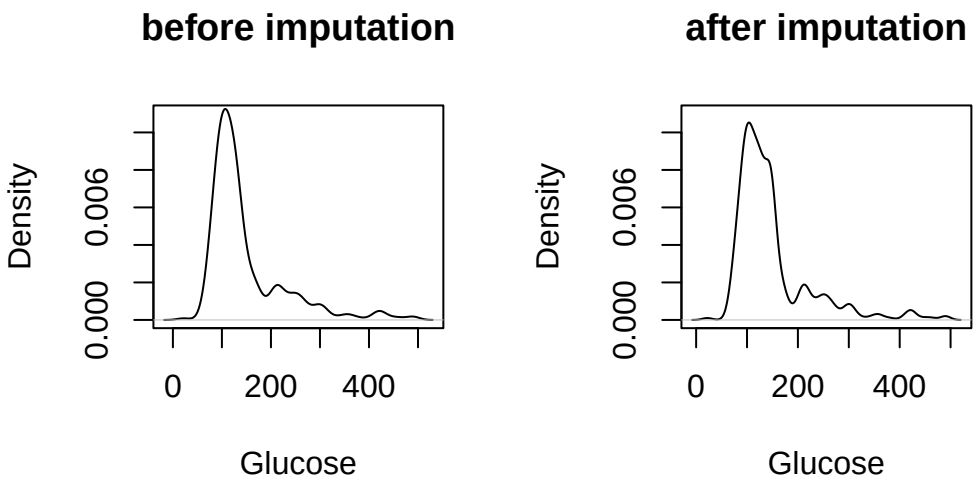

```r
par(mfrow=c(1, 2))
plot(density(diab.dt$bp, na.rm = T), main = "before imputation", xlab = "Blood Pressure")
plot(density(diab.dt.imputed$bp), main = "after imputation", xlab = "Blood Pressure")
```



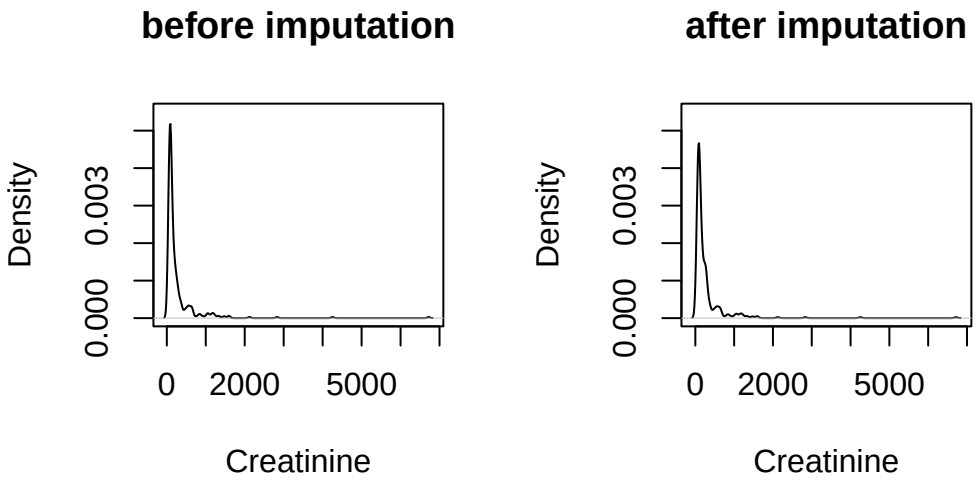
```r
par(mfrow=c(1, 2))
plot(density(diab.dt$urea, na.rm = T), main = "before imputation", xlab = "Urea")
plot(density(diab.dt.imputed$urea), main = "after imputation", xlab = "Urea")
```



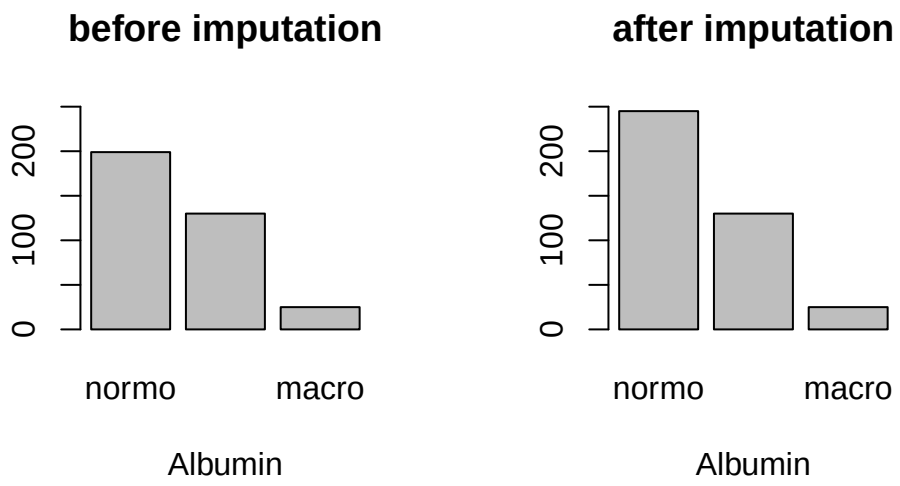

```r
par(mfrow=c(1, 2))
plot(density(diab.dt$glucose, na.rm = T), main = "before imputation",
     xlab = "Glucose", ylim=c(0,0.011))
plot(density(diab.dt.imputed$glucose), main = "after imputation",
     xlab = "Glucose", ylim=c(0,0.011))
```



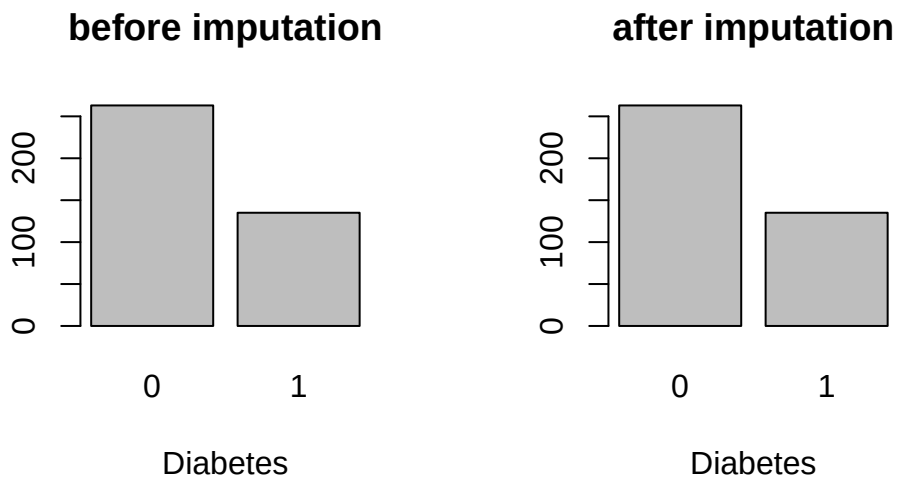

```
par(mfrow=c(1, 2))
plot(density(diab.dt$creatinine, na.rm = T), main = "before imputation",
     xlab = "Creatinine", ylim = c(0, 0.0055))
plot(density(diab.dt.imputed$creatinine), main = "after imputation",
     xlab = "Creatinine", ylim = c(0, 0.0055))
```



```r
par(mfrow=c(1, 2))
plot(na.omit(diab.dt$albumin), main = "before imputation",
     xlab = "Albumin", ylim = c(0, 250))
plot(diab.dt.imputed$albumin, main = "after imputation",
     xlab = "Albumin", ylim = c(0, 250))
```




```r
par(mfrow=c(1, 2))
plot(factor(na.omit(diab.dt$diabetes)), main = "before imputation", xlab = "Diabetes")
plot(factor(diab.dt.imputed$diabetes), main = "after imputation", xlab = "Diabetes")
```



Mean imputation can bias our understanding of `glucose`'s effect on `diabetes`. Since most missing `gluscose` values belong to non-diabetes, the imputed mean is higher than the true mean since diabetics are known to have higher `glucose` levels. As such, a model using `glucose` as the predictor would give less weight to `glucose` if it were imputed than if it were not imputed because the non-diabetic average would be closer to the diabetic average than it is in reality.

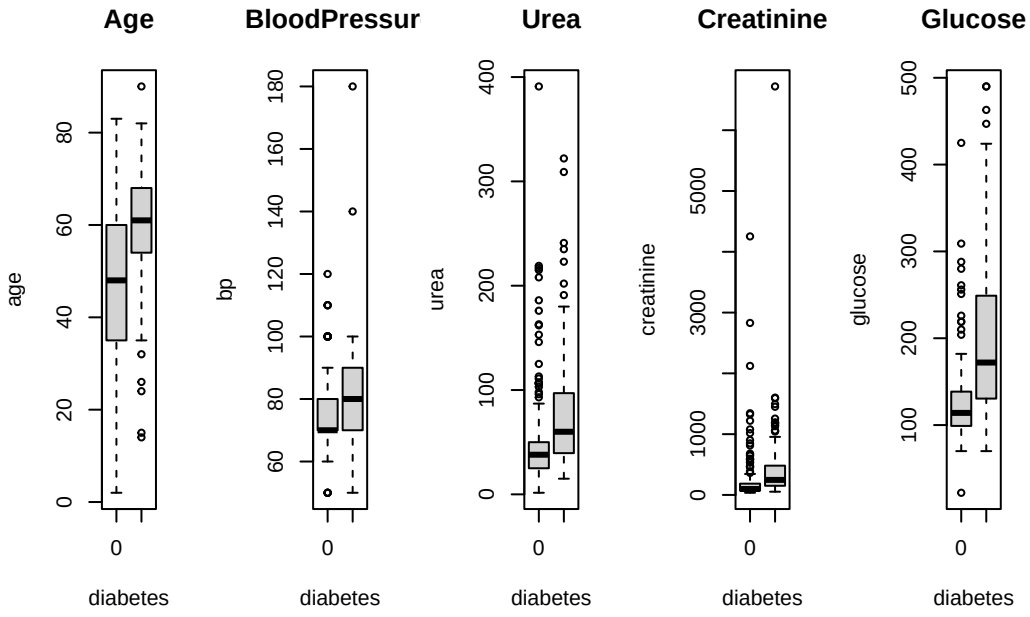
Mode imputation could bias the `albumin` results. If someone with `diabetes` is more likely to be missing `albumin` results, we do not know if those results are expected to be either high or low, so the proportion we detect in the observed data might not match the true data. Simply filling in with the mode, might also change the distribution because most `albumin` values for `diabetes` are `micro` not `normo`. If most missing values are for `diabetes` and they are replaced with `normo` instead of micro‘ it would bias the results.

Looking at the distributions of `before imputation` and `after imputation` we do not detect a significant change in the distribution. Since, the difference in distribution is not major, we will continue to employ the imputed dataset.

Problem 3.c (6 points)

- Plot a single figure containing boxplots of potential predictors for `diabetes` grouped by cases and controls. (Hint : `par(mfrow=c(1,5))`)
- Use these to decide which predictors to keep for future analysis.
- For any categorical variables create a table instead. Justify your answers.

```r
#computing the boxplot of diabetes against each continuous variables
par(mfrow=c(1, 5))
boxplot(age ~ diabetes, data = diab.dt.imputed, main = "Age")
boxplot(bp ~ diabetes, data = diab.dt.imputed, main = "BloodPressure")
boxplot(urea ~ diabetes, data = diab.dt.imputed, main = "Urea")
boxplot(creatinine ~ diabetes, data = diab.dt.imputed, main = "Creatinine")
boxplot(glucose ~ diabetes, data = diab.dt.imputed, main = "Glucose")
```



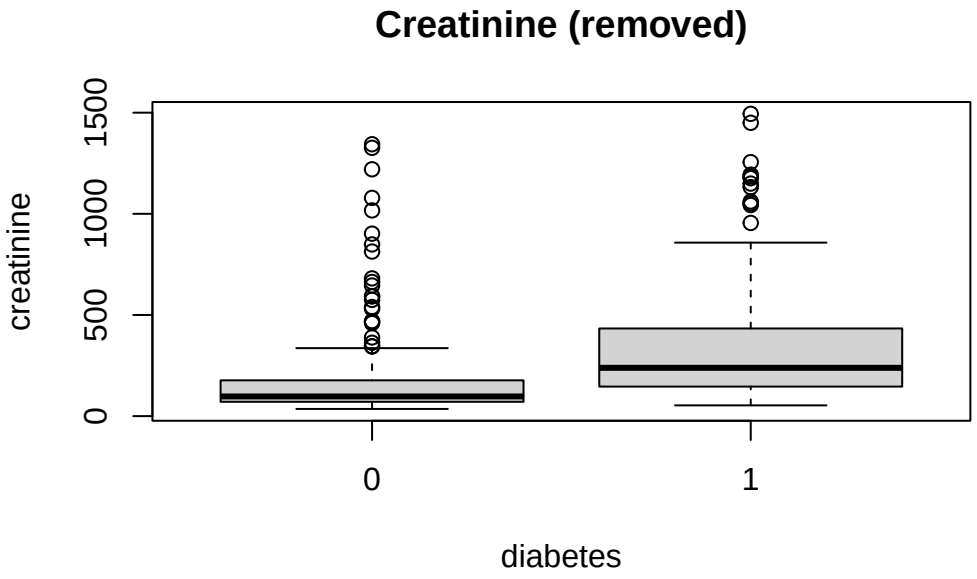

```r
kable(table(diab.dt[,albumin,diabetes]), caption = "Albumin stratified by Diabetes") |>
  kable_styling(full_width = F, position = "center", latex_options = "hold_position")
```

Table 12: Albumin stratified by Diabetes

	normo	micro	macro
0	174	61	12
1	23	69	13

```r
#removing the outlier in creatinine
par(mfrow=c(1, 1))
boxplot(creatinine ~ diabetes,
```

```
        data = diab.dt.imputed[diab.dt.imputed$creatinine<1500],
        main = "Creatinine (removed)")
```



The above boxplots suggest that `age`, `urea`, creatinine,blood pressure`and glucose` measurements. We see that the `age`, `urea` and `glucose` measurements are higher in people with `diabetes` than those without. `blood pressure` does not appear to be different from the boxplots.

Table (12), suggests that people with `diabetes` have a higher proportion of `micro albumin` levels than `normo` or `macro`. As such `albumin` could potentially be a predictor; however, given that `albumin` data is more likely to be missing for those with `diabetes` and the possibility of mode imputation leading to higher biasness, `albumin` may not be a good predictor.

Therefore, we conclude that `glucose`, `urea` and `age` are suitable predictors since the boxplots shows a noticeable difference for cases and controls.

Problem 3.d (9 points)

- Use your findings from the previous exercise and fit an appropriate model of `diabetes` with two predictors.
- Print a summary and explain the results as you would communicate it to a colleague with a medical background with a very little statistical knowledge.

```r
#removing NA values
diab.dt.imputed <- diab.dt.imputed[!is.na(diabetes),]
#fitting logistic regression with 2variables
#response variable : diabetes
#explanatory variables : glucose and urea
diab.regr.1 <- glm(diabetes ~ glucose + urea,
                   data = diab.dt.imputed, family='binomial')
summary(diab.regr.1)
```

```
Call:
glm(formula = diabetes ~ glucose + urea, family = "binomial",
    data = diab.dt.imputed)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0343  -0.6786  -0.5065   0.5941   2.2893

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.266076   0.422115 -10.106  < 2e-16 ***
glucose      0.018516   0.002473   7.487 7.02e-14 ***
urea         0.014040   0.002964   4.738 2.16e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 509.84  on 397  degrees of freedom
Residual deviance: 373.82  on 395  degrees of freedom
AIC: 379.82

Number of Fisher Scoring iterations: 5
```

```r
oddsratio.1 <- suppressMessages(exp(confint(diab.regr.1)))
kable(oddsratio.1, caption = "Confidence Interval of Odd Ratios") |>
  kable_styling(full_width = F, position = "center", latex_options = "hold_position")
```

We want to fit a model using the two most important variables. According to the plots above, we deduced that `glucose` and `urea` appear to be the most different between people with and

Table 13: Confidence Interval of Odd Ratios

	2.5 %	97.5 %
(Intercept)	0.0058806	0.0308746
glucose	1.0140510	1.0239506
urea	1.0084952	1.0202942

without `diabetes`. As a result, we will fit a model with `diabetes` as the response variable and `glucose` and `urea` as the explanatory variables.

Since the outcome is a binary variable, we will fit a logistic regression with a logit link. For brevity, we denote the odds of diabetes as odd(diabetes).

The form of the logistic regression is:

$$\log(\frac{\mathbb{P}(diabetes)}{1-\mathbb{P}(diabetes)}) = \beta_0 + \beta_1 \times \texttt{glucose} + \beta_2 \times \texttt{urea}$$

We can interpret the model as follows:

1. The coefficient of `glucose` tells how the log($odds$(diabetes)) changes with a unit increase in `glucose`. The interpretation is that a unit increase in `glucose` raises the odds of `diabetes` by 1.85%. The confidence interval of the odd ratio does not overlap with 1 where the odds ratio corresponding with no effect, we can conclude that the `glucose` has an effect on `diabetes` status.
2. The interpretation is that a unit increase in `urea` raises the odds of `diabetes` by 1.40%. The confidence interval of the odd ratio does not overlap with 1 again, we conclude that the `urea` has an effect on `diabetes` status.

We can also learn that the p-values for each estimate are less than 0.05, meaning that the effect of `glucose` and `urea` is likely to be significant.

Problem 4 (19 points)

Problem 4.a. (9 points)

- Add a third predictor to the final model from **problem 3**, perform a likelihood ratio test to compare both models and report the p-value for the test.
- Is there any support for the additional term?
- Plot a ROC curve for both models and report the AUC, explain the results as you would communicate it to a colleague with a medical background with a very little statistical knowledge.
- Print a summary and explain the results as you would communicate it to a colleague with a medical background with a very little statistical knowledge.

```r
#fitting logistic regression with 3 variables
#response variable : diabetes
#explanatory variables : glucose, urea and age
diab.regr.2 <- glm(diabetes ~ glucose + urea + age,
                   data = diab.dt.imputed, family='binomial')
summary(diab.regr.2)
```

```
Call:
glm(formula = diabetes ~ glucose + urea + age, family = "binomial",
    data = diab.dt.imputed)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0445  -0.6510  -0.3837   0.6112   2.9118

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.587697   0.710134  -9.277  < 2e-16 ***
glucose      0.016944   0.002467   6.869 6.45e-12 ***
urea         0.012602   0.002952   4.269 1.96e-05 ***
age          0.047780   0.009955   4.799 1.59e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 509.84  on 397  degrees of freedom
```

```
Residual deviance: 345.90  on 394  degrees of freedom
AIC: 353.9

Number of Fisher Scoring iterations: 5
```

```r
oddsratio.2 <- suppressMessages(exp(confint(diab.regr.2)))
kable(oddsratio.2, caption = "Confidence Interval of Odd Ratios") |>
  kable_styling(full_width = F, position = "center", latex_options = "hold_position")
```

Table 14: Confidence Interval of Odd Ratios

	2.5 %	97.5 %
(Intercept)	0.0003131	0.0051056
glucose	1.0124918	1.0223544
urea	1.0070250	1.0187659
age	1.0294423	1.0705002

Similarly, we fitted a model using additional variable known as `age`. The form of the logistic regression is:

$$\log(\frac{\mathbb{P}(diabetes)}{1-\mathbb{P}(diabetes)}) = \beta_0 + \beta_1 \times \texttt{glucose} + \beta_2 \times \texttt{urea} + \beta_3 \times \texttt{age}$$

We can interpret the model as follows:

1. The coefficient of `glucose` tells how the $\log(odds(\text{diabetes}))$ changes with a unit increase in `glucose`. The interpretation is that a unit increase in `glucose` raises the odds of `diabetes` by 1.01%. The confidence interval of the odd ratio does not overlap with 1 where the odds ratio corresponding with no effect, we can conclude that the `glucose` has an effect on `diabetes` status.
2. The interpretation is that a unit increase in `urea` raises the odds of `diabetes` by 1.01%. The confidence interval of the odd ratio does not overlap with 1 again, we conclude that the `urea` has an effect on `diabetes` status.
3. The interpretation is that a unit increase in `age` raises the odds of `diabetes` by 1.03%. The confidence interval of the odd ratio does not overlap with 1 again, we conclude that the `age` has an effect on `diabetes` status.

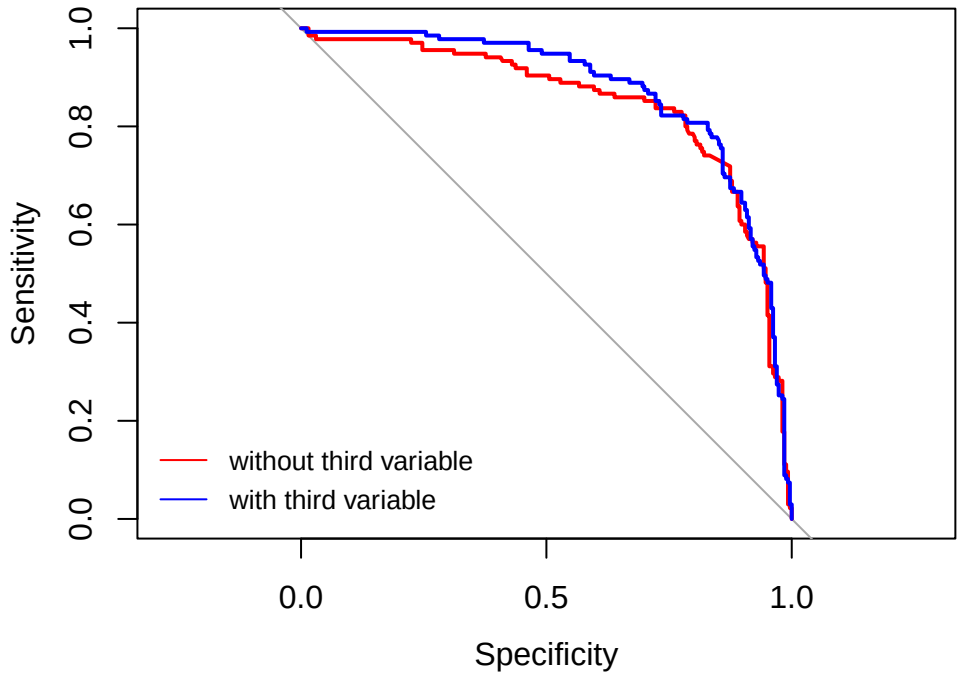
We can also learn that the p-values for each estimate are less than 0.05, meaning that the effect of `glucose` and `urea` is likely to be significant.

```r
#testing the goodness of fit by deriving p-value
gof.1 <- pchisq(diab.regr.1$null.deviance - diab.regr.1$deviance,
                df = 2, lower.tail = FALSE)
#testing the goodness of fit by deriving p-value
gof.2 <- pchisq(diab.regr.2$null.deviance - diab.regr.2$deviance,
                df = 3, lower.tail = FALSE)
#Performing likelihood ratio test
lrt <- pchisq(diab.regr.1$deviance - diab.regr.2$deviance,
                 df = 1, lower.tail=FALSE)
df.test <- data.frame(t(c(gof.1, gof.2, lrt)))
colnames(df.test) <- c("gof.1", "gof.2", "lrt")
kable(df.test, caption = "Likelihood Ratio Test of Models", digits = c(32, 37, 9)) |>
  kable_styling(full_width = F, position = "center", latex_options = "hold_position")
```

Table 15: Likelihood Ratio Test of Models

gof.1	gof.2	lrt
2.91e-30	2.59e-35	1.27e-07

```r
#computing the predicted values for both fits
diabetes.pred1 <- predict(diab.regr.1)
diabetes.pred2 <- predict(diab.regr.2)
#Computing the ROC Curve for the 2 models
suppressMessages(invisible({
  roc(diab.dt.imputed$diabetes, diabetes.pred1, plot = TRUE,
    xlim = c(0,1), col = "red")
  roc(diab.dt.imputed$diabetes, diabetes.pred2, plot = TRUE,
      add = TRUE, col = "blue")
  legend("bottomleft", legend = c("without third variable", "with third variable"),
         col = c("red", "blue"), lty = 1, cex = 0.8, bty = "n")
}))
```




```r
#Extracting AUC values for both models
suppressMessages(invisible(
  df.roc <- data.frame(t(c(roc(diab.dt.imputed$diabetes, diabetes.pred1)$auc,
              roc(diab.dt.imputed$diabetes, diabetes.pred2)$auc)))
))
colnames(df.roc) <- c("Model 1", "Model 2")
kable(df.roc, caption = "AUC values for Model 1 and Model 2") |>
  kable_styling(full_width = F, position = "center", latex_options = "hold_position")
```

Table 16: AUC values for Model 1 and Model 2

Model 1	Model 2
0.8481763	0.8736094

By comparing the deviance from both models, `diab.regr.1` had $2.91e-30$ where as `diab.regr.2` has $2.91e-35$ from the Table above. Since `diab.regr.2` has a smaller deviance, we can claim that `diab.regr.2` is a better model. This is can be further solidified by performing the likelihood ratio test where the p-value yields, $1.43e-7 < 0.05$. Thus, there is a sufficient evidence to reject the null hypothesis and conclude that adding an additional variable in the model will lead to better fit.

Now let us find more evidence by comparing the ROC plot and their AUC values. ROC curve let us visualise sensitivity vs specificity for all possible classification thresholds. According to the result above, both models can predict the outcome better than the random chance. The AUC values of `diab.regr.2` shows higher value than `diab.regr.1` according to Table (16). As a result, we say that the model including `age` variable has a better fit to the data and better predictive accuracy.

Problem 4.b (10 points)

- Perform 10-folds cross validation for your chosen model based on the above answers.
- Report the mean cross-validated AUCs in 3 significant figures.

```
#defining function to perform cross validation
glm.cv <- function(formula, data, folds) {
  #initialising list of list to store regression of each fold
  regr.cv <- NULL
  for (f in 1:length(folds)) {
    #computing logistic regression on the training set
    regr.cv[[f]] <- glm(formula, data = data[-folds[[f]], ],
                        family = "binomial")
  }
  #returning the regression outputs
  return(regr.cv)
}
```

```
#setting seed
set.seed(3)
#initialising number of folds
num.folds <- 10
folds <- createFolds(diab.dt.imputed$diabetes, k = num.folds)
```

```
suppressMessages({invisible({
  #storing the output of cross validation
  cv.m <- glm.cv(diabetes ~ glucose + urea + age, diab.dt.imputed, folds)
  #initialsing list of list to store prediced values of each fold
  pred.cv <- NULL
  #initalising list to store auc valude of each fold
  auc.cv <- numeric(num.folds)
  for(f in 1:num.folds) {
    test.idx <- folds[[f]]
```

```
    #computing the predicted values
    pred.cv[[f]] <- data.frame(obs = diab.dt.imputed$diabetes[test.idx],
                               pred = predict(cv.m[[f]],
                                              newdata = diab.dt.imputed,
                                              type = "response")[test.idx])
    #computing the auc value of fold
    auc.cv[f] <- roc(obs ~ pred, data = pred.cv[[f]])$auc
  }
})})
#computing the mean of AUC of the 10-folds cross validation
round(mean(auc.cv), 3)
```

```
[1] 0.869
```

The mean cross-validated AUCs for 10-fold cross-validation is 0.869. This is slightly lower than the AUC from the full dataset due to the reductions in sample size for each fold. Since these results came from cross-validation, we are much confident in the values than the single partitioned value.