Final assignment : Chronic Kidney Disease.

By:Setareh pishyar

Abstract:

Chronic kidney disease (CKD) refers to a condition where in the kidneys cannot perform its regular function of filtering blood. A family history of kidney diseases or failure, high blood pressure, type 2 diabetes may lead to CKD. This is a lasting damage to the kidney and chances of getting worse by time is high. The very common complications that results due to a kidney failure are heart diseases, anemia, bone diseases, high potassium and calcium. The worst case situation leads to complete kidney failure and necessitates kidney transplant to live.

Introduction:

Chronic Kidney Disease is one of the most critical illness nowadays and proper diagnosis is required as soon as possible. Signs and symptoms of chronic kidney disease develop over time if kidney damage progresses slowly. Loss of kidney function can cause a buildup of fluid or body waste or electrolyte problems. Depending on how severe it is, loss of kidney function can cause:

- Nausea
- Vomiting
- Loss of appetite
- Fatigue and weakness
- Sleep problems
- Urinating more or less
- Decreased mental sharpness
- Muscle cramps
- Swelling of feet and ankles
- Dry, itchy skin
- High blood pressure (hypertension) that's difficult to control
- Shortness of breath, if fluid builds up in the lungs
- Chest pain, if fluid builds up around the lining of the heart

Signs and symptoms of kidney disease are often nonspecific. This means they can also be caused by other illnesses.
Machine learning technique has become reliable for medical treatment. With the help of a machine learning classifier algorithms, the doctor can detect the disease on time. For this perspective, Chronic Kidney Disease prediction has been discussed in this article. Chronic Kidney Disease dataset has been taken from the UCI repository.

Chronical Kidney dataset:

This dataset contains 400 instances and 24 attributes with 1 target attribute. The target attribute has labelled in two-class to represent CKD or non-CKD. The dataset was collected from various hospitals in 2015. It contains also missing value.

No	Attribute name	Describe	variables
1	Age	Age in years	Numerical
2	Bp	Blood pressure	Numerical
3	Sg	Specific gravity	Nominal
4	Al	Albumin	Nominal
5	Su	Sugar	Nominal
6	Rbc	Red blood cells	Nominal
7	Pc	Pus cell	Nominal
8	Pcc	Pus cell clumps	Nominal
9	Ba	Bacteria	Nominal
10	Bgr	Blood glucose random	Numerical
11	Bu	Blood urea	Numerical
12	Sc	Serum creatinine	Numerical
13	Sod	Sodium	Numerical
14	Pot	Potassium	Numerical
15	Hemo	Hemoglobin	Numerical
16	Pcv	Packed cell volume	Numerical
17	Wc	White blood cell count	Numerical
18	Rc	Red blood cell count	Numerical
19	Htn	Hypertension	Nominal
20	Dm	Diabetes mellitus	Nominal
21	Cad	Coronary artery disease	Nominal
22	Appet	Appetite	Nominal
23	Pe	Pedal edema	Nominal
24	Ane	Anemia	Nominal

+

25	Class	Classification	Nominal

➢ Data Exploration:

Started to read the Chronical kidney dataset which has been downloaded from UCI website with **read.csv.**

```
# read dataset
dataset <- read.csv('C:/Users/STAR/OneDrive/Desktop/kidney_disease1.csv' ,
                    stringsAsFactors = FALSE)
```

The data has many missing values and needs to be clean, manipulated and pre-processed to the next step

Filter

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classificatio
1	0	48	80	1.020	1	0		normal	notpresent	notpresent	121	36.0	1.20	NA	NA	15.4	44	7800	5.2	yes	yes	no	good	no	no	ckd
2	1	7	50	1.020	4	0		normal	notpresent	notpresent	NA	18.0	0.80	NA	NA	11.3	38	6000		no	no	no	good	no	no	ckd
3	2	62	80	1.010	2	3	normal	normal	notpresent	notpresent	423	53.0	1.80	NA	NA	9.6	31	7500		no	yes	no	poor	no	yes	ckd
4	3	48	70	1.005	4	0	normal	abnormal	present	notpresent	117	56.0	3.80	111.0	2.5	11.2	32	6700	3.9	yes	no	no	poor	yes	yes	ckd
5	4	51	80	1.010	2	0	normal	normal	notpresent	notpresent	106	26.0	1.40	NA	NA	11.6	35	7300	4.6	no	no	no	good	no	no	ckd
6	5	60	90	1.015	3	0			notpresent	notpresent	74	25.0	1.10	142.0	3.2	12.2	39	7800	4.4	yes	yes	no	good	yes	no	ckd
7	6	68	70	1.010	0	0		normal	notpresent	notpresent	100	54.0	24.00	104.0	4.0	12.4	36			no	no	no	good	no	no	ckd

➢ Clean, Prepare and Manipulate data:

As the data is raw it needs to be cleaned, manipulated and pre-processed. This involves:

1. Finding all the null values
2. Finding all the missing values
3. Finding all the wrong values
4. Imputing the correct values in the respective places

For cleaning ,prepare and manipulate the data first of all we should know about data structure. **str()** function can help us to show data summery and structure

```
str(dataset)
```

This is a structure of the Chronical kidney disease before started to be clean and prepare.

```
> str(dataset) # i get the structure
'data.frame':   400 obs. of  25 variables:
 $ age            : num  48 7 62 48 51 60 68 24 52 53 ...
 $ bp             : num  80 50 80 70 80 90 70 76 100 90 ...
 $ sg             : num  1.02 1.02 1.01 1 1.01 ...
 $ al             : num  1 4 2 4 2 3 0 2 3 2 ...
 $ su             : num  0 0 3 0 0 0 0 4 0 0 ...
 $ rbc            : chr  "" "" "normal" "normal" ...
 $ pc             : chr  "normal" "normal" "normal" "abnormal" ...
 $ pcc            : chr  "notpresent" "notpresent" "notpresent" "present" ...
 $ ba             : chr  "notpresent" "notpresent" "notpresent" "notpresent" ...
 $ bgr            : num  121 NA 423 117 106 74 100 410 138 70 ...
 $ bu             : num  36 18 53 56 26 25 54 31 60 107 ...
 $ sc             : num  1.2 0.8 1.8 3.8 1.4 1.1 24 1.1 1.9 7.2 ...
 $ sod            : num  NA NA NA 111 NA 142 104 NA NA 114 ...
 $ pot            : num  NA NA NA 2.5 NA 3.2 4 NA NA 3.7 ...
 $ hemo           : num  15.4 11.3 9.6 11.2 11.6 12.2 12.4 12.4 10.8 9.5 ...
 $ pcv            : num  44 38 31 32 35 39 36 44 33 29 ...
 $ wc             : num  7800 6000 7500 6700 7300 7800 NA 6900 9600 12100 ...
 $ rc             : num  5.2 NA NA 3.9 4.6 4.4 NA 5 4 3.7 ...
 $ htn            : chr  "yes" "no" "no" "yes" ...
 $ dm             : chr  "yes" "no" "yes" "no" ...
 $ cad            : chr  "no" "no" "no" "no" ...
 $ appet          : chr  "good" "good" "poor" "poor" ...
 $ pe             : chr  "no" "no" "no" "yes" ...
 $ ane            : chr  "no" "no" "yes" "yes" ...
 $ classification: chr  "ckd" "ckd" "ckd" "ckd" ...
>
```

Starting to find wrong objects such as character and changing that to numeric. On the other hand I removed id column because it was useless.

```
dataset$pcv <- as.numeric(dataset$pcv)
#change character to numeric because pcv column has number and we should define that as a numeric

dataset$wc <- as.numeric(dataset$wc)
#change character to numeric because wc column has number and we should define that as a numeric

dataset$rc <- as.numeric(dataset$rc)
#change character to numeric because rc column has number and we should define that as a numeric

dataset$id <- NULL  # remove the id column because i think it was useless
```

These columns are numeric and it does not to change them to factor but by taking the average **(mean** function**)** of each column, the missing values can be replaced with the average number. Here I defined that as an integer to display numbers without decimal.

```
mean(dataset$age) # getting average of age column
#> NA
as.integer(mean(dataset$age, na.rm = T)) #this is a average age of age column
#> 51
mean_age <- as.integer(mean(dataset$age, na.rm = T))
dataset$age[is.na(dataset$age)]
#> NA NA NA NA NA NA NA NA NA
dataset$age[is.na(dataset$age)] <- mean_age #i define average age number instead of 'NA'


mean(dataset$bp)
#> NA
as.integer(mean(dataset$bp, na.rm = T))
#> 76
mean_bp <- as.integer(mean(dataset$bp, na.rm = T))
dataset$bp[is.na(dataset$bp)]
#> NA NA NA NA NA NA NA NA NA NA NA NA
dataset$bp[is.na(dataset$bp)] <- mean_bp

mean(dataset$bgr)
#> NA
as.integer(mean(dataset$bgr, na.rm = T)) # this is numeric and i change it as integer to get non decimal number
#> 148
mean_bgr <- as.integer(mean(dataset$bgr, na.rm = T))
dataset$bgr[is.na(dataset$bgr)] <- mean_bgr


mean(dataset$bu)
#> NA
as.integer(mean(dataset$bu, na.rm = T))
#> 57
mean_bu <- as.integer(mean(dataset$bu, na.rm = T))
dataset$bu[is.na(dataset$bu)] <- mean_bu

mean(dataset$pcv)
#> NA
as.integer(mean(dataset$pcv, na.rm = T))
#> 38
mean_pcv <- as.integer(mean(dataset$pcv, na.rm = T))
dataset$pcv[is.na(dataset$pcv)] <- mean_pcv


mean(dataset$wc)
#> NA
as.integer(mean(dataset$wc, na.rm = T))
#> 8406
mean_wc <- as.integer(mean(dataset$wc, na.rm = T))
dataset$wc[is.na(dataset$wc)] <- mean_wc

dataset$rc[is.na(dataset$rc)] <- 0
```

These are nominal and some of them does not have many NA because of that I just decided to change missing number (NA) to 0

```
dataset$sg[is.na(dataset$sg)] <- 0 # it is nominal because of that i just change the NA to 0
dataset$al[is.na(dataset$al)] <- 0 # this is also nominal
dataset$su[is.na(dataset$su)] <- 0 # this is also nominal

# those columns has not many NA because of that i decided to use this code just to change to the 0
dataset$sc[is.na(dataset$sc)] <- 0
dataset$sod[is.na(dataset$sod)] <- 0
dataset$pot[is.na(dataset$pot)] <- 0
dataset$hemo[is.na(dataset$hemo)] <- 0
```

Changing character columns to factor because it should defined the number instead of those character on each columns. Machine can read the dataset just with numbers .

```
dataset$rbc <- factor(dataset$rbc,levels = c('normal' , 'abnormal') , labels = c(2,3))
dataset$rbc[is.na(dataset$rbc)] <- 2
# rbc was a character and i change that to factor because
# i wanted to change 'normal' , 'abnormal' to number 2 and 3
# also i change the NA to number 2

str(dataset) # i get the structure

dataset$pc <- factor(dataset$pc,levels = c("normal" , "abnormal") , labels = c(2,3))
dataset$pc[is.na(dataset$pc)] <- 2
# pc was a character and i change that to factor because
# i wanted to change 'normal' , 'abnormal' to number 2 and 3
# also i change the NA to number 2

dataset$pcc <- factor(dataset$pcc,levels = c("present" , "not present") , labels = c(2,3))
dataset$pcc[is.na(dataset$pcc)] <- 2
# pcc was a character and i change that to factor because
# i wanted to change 'present' , 'not present' to number 2 and 3
# also i change the NA to number 2


dataset$ba <- factor(dataset$ba,levels = c("present" , "notpresent") , labels = c(1,2))
dataset$ba[is.na(dataset$ba)] <- 2
# pcc was a character and i change that to factor because
# i wanted to change 'present' , 'not present' to number 1 and 2
# also i change the NA to number 2
```

```
dataset$htn <- factor(dataset$htn,levels = c("yes" , "no") , labels = c(1,2))# changing character to factor
dataset$htn[is.na(dataset$htn)] <- 2

dataset$dm <- factor(dataset$dm,levels = c("yes" , "no") , labels = c(1,2))# changing character to factor
dataset$dm[is.na(dataset$dm)] <- 2

dataset$cad <- factor(dataset$cad,levels = c("yes" , "no") , labels = c(1,2))# changing character to factor
dataset$cad[is.na(dataset$cad)] <- 2

dataset$appet<- factor(dataset$appet,levels = c("good" , "poor") , labels = c(1,2))# changing character to factor
dataset$appet[is.na(dataset$appet)] <- 2

dataset$pe <- factor(dataset$pe,levels = c("yes" , "no") , labels = c(1,2))# changing character to factor
dataset$pe[is.na(dataset$pe)] <- 2

dataset$ane <- factor(dataset$ane,levels = c("yes" , "no") , labels = c(1,2))# changing character to factor
dataset$ane[is.na(dataset$ane)] <- 2

dataset$classification <- factor(dataset$classification,levels = c("ckd" , "notckd") , labels = c(1,2))
dataset$classification[is.na(dataset$classification)] <- 2
```

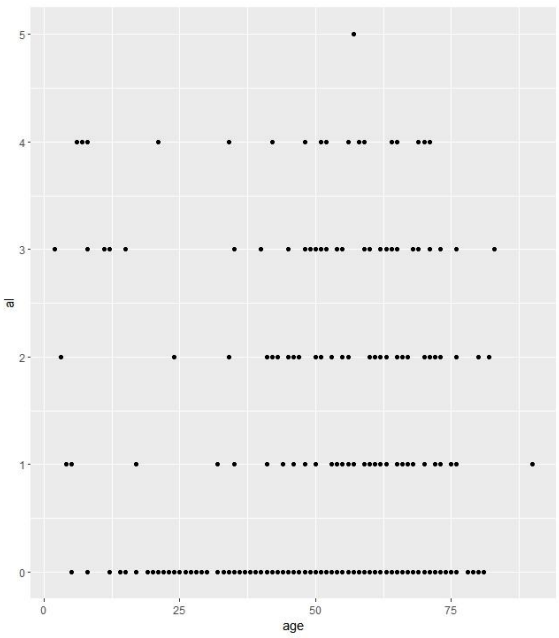
This is a result of dataset after cleaning and preparing values .

```
> str(dataset)
'data.frame':   400 obs. of  26 variables:
 $ id             : int  0 1 2 3 4 5 6 7 8 9 ...
 $ age            : num  48 7 62 48 51 60 68 24 52 53 ...
 $ bp             : num  80 50 80 70 80 90 70 NA 100 90 ...
 $ sg             : num  1.02 1.02 1.01 1 1.01 ...
 $ al             : num  1 4 2 4 2 3 0 2 3 2 ...
 $ su             : num  0 0 3 0 0 0 0 4 0 0 ...
 $ rbc            : chr  "" "" "normal" "normal" ...
 $ pc             : chr  "normal" "normal" "normal" "abnormal" ...
 $ pcc            : chr  "notpresent" "notpresent" "notpresent" "present" ...
 $ ba             : chr  "notpresent" "notpresent" "notpresent" "notpresent" ...
 $ bgr            : num  121 NA 423 117 106 74 100 410 138 70 ...
 $ bu             : num  36 18 53 56 26 25 54 31 60 107 ...
 $ sc             : num  1.2 0.8 1.8 3.8 1.4 1.1 24 1.1 1.9 7.2 ...
 $ sod            : num  NA NA NA 111 NA 142 104 NA NA 114 ...
 $ pot            : num  NA NA NA 2.5 NA 3.2 4 NA NA 3.7 ...
 $ hemo           : num  15.4 11.3 9.6 11.2 11.6 12.2 12.4 12.4 10.8 9.5 ...
 $ pcv            : chr  "44" "38" "31" "32" ...
 $ wc             : chr  "7800" "6000" "7500" "6700" ...
 $ rc             : chr  "5.2" "" "" "3.9" ...
 $ htn            : chr  "yes" "no" "no" "yes" ...
 $ dm             : chr  "yes" "no" "yes" "no" ...
 $ cad            : chr  "no" "no" "no" "no" ...
 $ appet          : chr  "good" "good" "poor" "poor" ...
 $ pe             : chr  "no" "no" "no" "yes" ...
 $ ane            : chr  "no" "no" "yes" "yes" ...
 $ classification: chr  "ckd" "ckd" "ckd" "ckd" ...
>
```

Scatterplot:

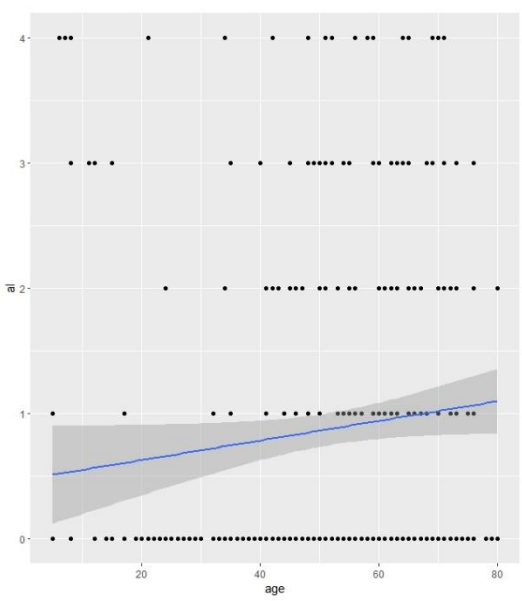
Starting to create plot with **ggplot** package.
First of all I insert ggplot on tool to active ggplot package on library .With library function could installed and use ggplot during coding.

```
library(ggplot2)# start to using ggplot
ggplot(dataset, aes(x = age , y = al)) # defind X and Y
ggplot(dataset, aes(x = age , y = al))+geom_point()
```



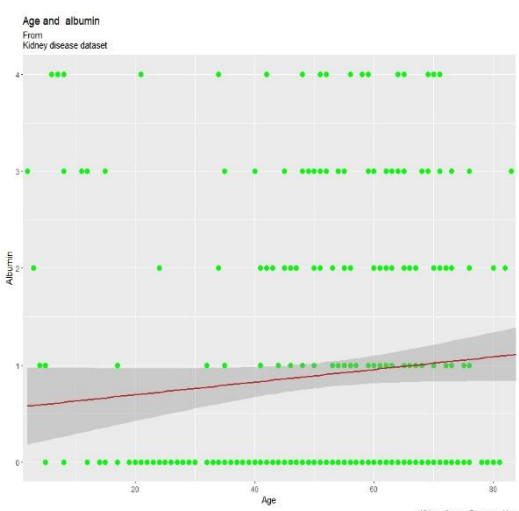
Add more layers as well as create line and deleting some rows and columns from plot which is defined by code .

```
a <- ggplot(dataset, aes(x = age , y = al)) +
  geom_point() +geom_smooth(method = "lm")
#  adding more layers

plot(a)
a + xlim(c( 5, 80)) + ylim(c(0 , 4))
# deleting 4 rows with xlim to age part and one row from ylim for Albumin par
```



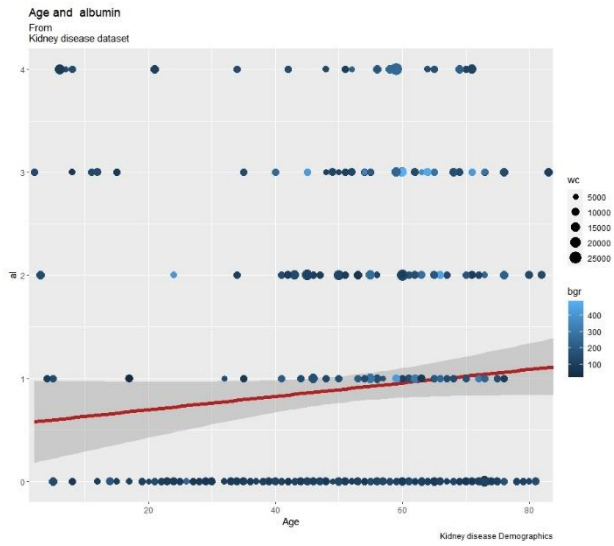
Add more layers for changing column color and size of points as well as change the line color. In addition add title , subtitle and caption

```
ggplot(dataset, aes(x=age, y=al)) +
  geom_point(col="green", size=3) +geom_smooth(method="lm", col="firebrick") +
  # changing color and size of points

  coord_cartesian(xlim=c(5, 80), ylim=c(0, 4)) +
  labs(title="Age and  albumin", subtitle="From
Kidney disease dataset", y="Albumin", x="Age",
       caption="Kidney disease Demographics")
```



Add more variables such as Wc and Bgr with geom_point and aes functions.

```
b <- ggplot(dataset, aes(x=age, y=al)) +
  geom_point(aes(col=bgr), size=3) +

  geom_smooth(method="lm", col="firebrick", size=2) +
  # changing color and size of the line

  coord_cartesian(xlim=c(5, 80), ylim=c(0, 4)) +
  labs(title="Age and  albumin", subtitle="From
Kidney disease dataset", y="al", x="Age",
       caption="Kidney disease Demographics")
b + geom_point(aes(col=bgr , size=wc))
```

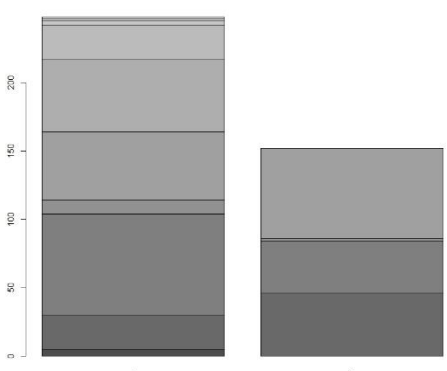


Barplot:

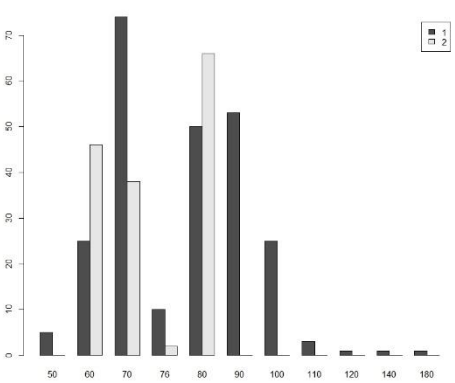
Defining barplot with table function and adding title. Defining color to title name and columns names.

```
# using barplot for making bar chart

barplot(table(dataset$bp,dataset$classification))
#defining barplot with table and bp and classification columns

classification=factor(dataset$classification)
barplot(table(dataset$classification,dataset$bp),beside=TRUE,legend.text=T)
barplot(table(dataset$classification,dataset$bp),beside=TRUE ,
        legend.text=c("less than 10","10_30","more than 30"))
title(main = "Kidney disease", xlab = "Blood pressure",
      ylab = "classification", col.lab= "orange", font.main= 4 , col.main="red")
# writing title , xlab name ,ylab name , color of the lab text ,
#size of the font and color for main title
```

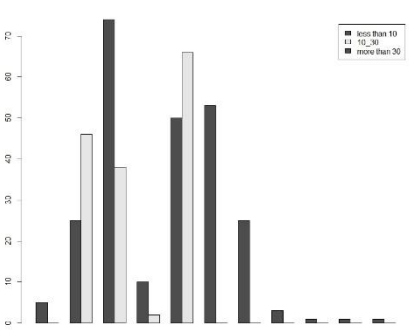
When you running every code you can get these results



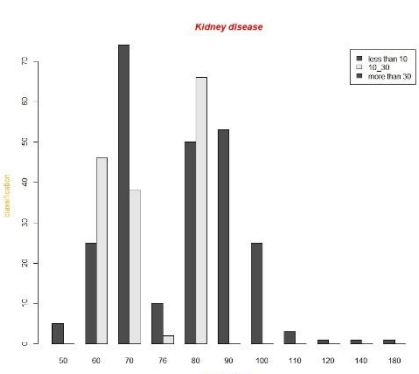
1



2



3



4

The final step , I used write.csv function for exporting final csv file after cleaning process to excel .
And read that again for see the final result.

```
write.csv(dataset,"C:\\Users\\STAR\\OneDrive\\Desktop\\city capture\\QUESTION2.csv"
          ,row.names = FALSE)
#exporting csv file to excel

QUSTION2 <- read.csv("C:\\Users\\STAR\\OneDrive\\Desktop\\city capture\\QUESTION2.csv" , sep = ',')
# reading a file again
```

	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
1	48	80	1.020	1	0	2	2	2	2	121	36.0	1.20	0.0	0.0	15.4	44	7800	5.2	1	1	2	1	2	2	1
2	7	50	1.020	4	0	2	2	2	2	148	18.0	0.80	0.0	0.0	11.3	38	6000	0.0	2	2	2	1	2	2	1
3	62	80	1.010	2	3	2	2	2	2	423	53.0	1.80	0.0	0.0	9.6	31	7500	0.0	2	1	2	2	2	1	1
4	48	70	1.005	4	0	2	3	2	2	117	56.0	3.80	111.0	2.5	11.2	32	6700	3.9	1	2	2	2	1	1	1
5	51	80	1.010	2	0	2	2	2	2	106	26.0	1.40	0.0	0.0	11.6	35	7300	4.6	2	2	2	1	2	2	1
6	60	90	1.015	3	0	2	2	2	2	74	25.0	1.10	142.0	3.2	12.2	39	7800	4.4	1	1	2	1	1	2	1
7	68	70	1.010	0	0	2	2	2	2	100	54.0	24.00	104.0	4.0	12.4	36	8406	0.0	2	2	2	1	2	2	1
8	24	76	1.015	2	4	2	3	2	2	410	31.0	1.10	0.0	0.0	12.4	44	6900	5.0	2	1	2	1	1	2	1
9	52	100	1.015	3	0	2	3	2	2	138	60.0	1.90	0.0	0.0	10.8	33	9600	4.0	1	1	2	1	2	1	1
10	53	90	1.020	2	0	3	3	2	2	70	107.0	7.20	114.0	3.7	9.5	29	12100	3.7	1	1	2	2	2	1	1
11	50	60	1.010	2	4	2	3	2	2	490	55.0	4.00	0.0	0.0	9.4	28	8406	0.0	1	1	2	1	2	1	1
12	63	70	1.010	3	0	3	3	2	2	380	60.0	2.70	131.0	4.2	10.8	32	4500	3.8	1	1	2	2	1	2	1
13	68	70	1.015	3	1	2	2	2	2	208	72.0	2.10	138.0	5.8	9.7	28	12200	3.4	1	1	1	2	1	2	1
14	68	70	0.000	0	0	2	2	2	2	98	86.0	4.60	135.0	3.4	9.8	38	8406	0.0	1	1	1	2	1	2	1
15	68	80	1.010	3	2	2	3	2	1	157	90.0	4.10	130.0	6.4	5.6	16	11000	2.6	1	1	1	2	1	2	1
16	40	80	1.015	3	0	2	2	2	2	76	162.0	9.60	141.0	4.9	7.6	24	3800	2.8	1	2	2	1	2	1	1