

R_HW1

Zahra Soukhtedel-98105138

2023-04-28

Problem 1

In order to see the intensity and weakness of Corona in different months, we have drawn the heat map of the additional deaths. To calculate the excess death rate, we first predict the death rate in the absence of Corona and compare it with the actual death rate and declare it as an excess death.

To predict the amount of additional deaths, we use the data of 5 years before Corona and use the linear regression model, and if the p-value is high, we use the average mortality. (In case of excess death, we declare that the actual death rate has a significant difference of more than 1.96 sigma from the predicted death rate).

libraries to be added:

```
library(knitr)
library(RColorBrewer)
library(data.table)
library(ggplot2)
```

function to calculate p-value:

```
lmp <- function (modelobject) {
  if (class(modelobject) != "lm") stop("Not an object of class 'lm' ")
  f <- summary(modelobject)$fstatistic
  p <- pf(f[1],f[2],f[3],lower.tail=F)
  attributes(p) <- NULL
  return(p)
}
```

load data and prepare it for analyses(covid-data df to save comparable datas, and moral-num to save actual death values):

```
d = fread('C:\\Users\\acer\\Documents\\R_hw1\\iranprovs_mortality_monthly.csv', encoding = 'UTF-8')
d$ym_num = d$y + d$m / 12 - 1/24
ds = d[, .(n = sum(n)), .(y, m, ym_num, prov)]
ym_num_covid = 1398 + 10/12 - 1/24
covid_data = data.frame(matrix(nrow = 31, ncol = 34))
moral_num = data.frame(matrix(nrow = 31, ncol = 34))
rownames(covid_data) = unique(ds[ym_num >= ym_num_covid]$prov)
colnames(covid_data) = sort(unique(ds[ym_num >= ym_num_covid]$ym_num), decreasing = F)
rownames(moral_num) = unique(ds[ym_num >= ym_num_covid]$prov)
colnames(moral_num) = sort(unique(ds[ym_num >= ym_num_covid]$ym_num), decreasing = F)
ym_num_start = ym_num_covid - 5
```

fill the covid-data data frame with computing values of each year-month-province with the model discussed above:

```

for (M in c(1:12)) {
  for (PROV in unique(d$prov)) {
    dsm = ds[prov == PROV & m == M & ym_num > ym_num_start,]
    dsm2fit = dsm[ym_num < ym_num_covid]
    fit = lm(n ~ ym_num, dsm2fit)
    if (lmp(fit) > 0.1){
      dsm$n_predicted = mean(dsm2fit$n)
      dsm$sd = sd(dsm2fit$n)
    }else{
      dsm$n_predicted = predict(fit ,dsm)
      dsm$sd = summary(fit)$sigma
    }

    dsm$moral_limit = dsm$n_predicted + 1.96 * dsm$sd
    dsm$diff = (dsm$n - dsm$moral_limit)/dsm$n
    dsm$diff[dsm$diff < 0] = 0
    dsm$moral_value = (dsm$n - dsm$n_predicted)
    dsm$moral_value[dsm$moral_value < 0] = 0

    dsm$ym_num[dsm$ym_num[dsm$ym_num]]

    for (ym_i in unique(dsm$ym_num[dsm$ym_num >= ym_num_covid])) {
      covid_data[PROV, as.character(ym_i)] = dsm[ym_num == ym_i & prov == PROV]$diff[1]
      moral_num[PROV, as.character(ym_i)] = dsm[ym_num == ym_i & prov == PROV]$moral_value[1]
    }
  }
}

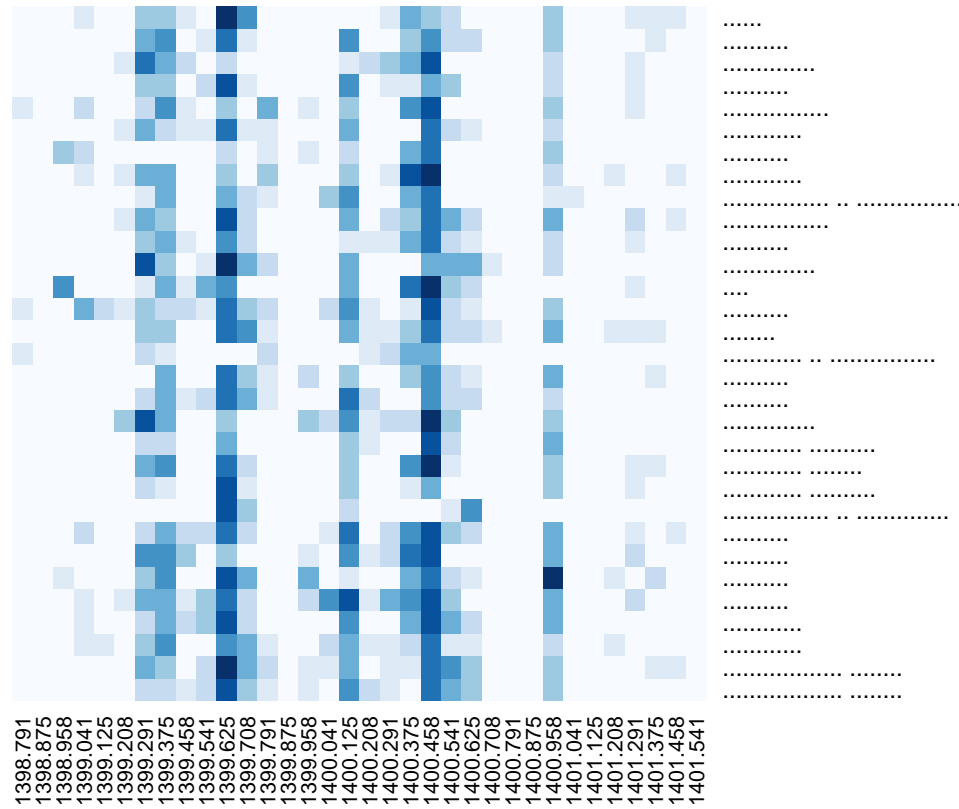
```

plot result heat-map:

```

covid_matrix <- as.matrix(covid_data)
for (i in c(1 : length(colnames(covid_matrix)))) {
  colnames(covid_matrix)[i] = substr(colnames(covid_matrix)[i], start=1, stop=8)
}
col <- brewer.pal(9, "Blues")
heatmap(covid_matrix, col=col, Colv = NA, Rowv = NA, scale="none")

```



As can be seen from the heat-map, it is easy to see the death waves of Corona.

Problem 2

What was the total number of additional deaths in the entire country?

the number of total number of additional deaths is shown as follows:

```
sum(moral_num)
```

```
## [1] 328925.8
```

Problem 3

What was the total number of additional deaths in each province?

the number of total number of additional deaths in each province is shown as follows:

```
rowSums(moral_num)
```

```
##
##          20589.70          14818.85          6015.30          20767.20
##
##          14450.75          2265.70          3649.50          55839.40
##
##          2628.70          2348.80          27672.20          3132.20
##
##          19403.40          4395.60          2814.70          6706.50
##
```

##	19740.20	6044.15	6097.80	7053.90
##				
##	10804.25	9329.80	2182.30	8232.50
##				
##	10246.20	6542.80	12611.60	5840.30
##				
##	4664.70	8077.20	3959.60	

Problem 4

Which provinces have been more successful in controlling the disease? Is there a significant difference between the provinces?

first we compare normalized death sum in provinces, we have:

```
sort(rowSums(covid_data))
```

##				
##	1.623538	1.661277	2.044131	2.435194
##				
##	2.471749	3.058584	3.153926	3.164760
##				
##	3.231746	3.233507	3.418042	3.521570
##				
##	3.615733	3.621725	3.636240	3.680428
##				
##	3.861881	3.965130	3.988357	3.993150
##				
##	4.029805	4.101250	4.149745	4.166678
##				
##	4.335598	4.354785	4.375494	4.653589
##				
##	4.721216	4.845674	5.738912	

from data above we can conclude that most successful provinces are {gilan, south khorasan, chaharmaha, sistanal, now we will compare it with actual death values :

```
sort(rowSums(moral_num))
```

##				
##	2182.30	2265.70	2348.80	2628.70
##				
##	2814.70	3132.20	3649.50	3959.60
##				
##	4395.60	4664.70	5840.30	6015.30
##				
##	6044.15	6097.80	6542.80	6706.50
##				
##	7053.90	8077.20	8232.50	9329.80
##				
##	10246.20	10804.25	12611.60	14450.75
##				
##	14818.85	19403.40	19740.20	20589.70
##				
##	20767.20	27672.20	55839.40	

One of the factors that can bias our conclusions is the elderly people in the society, we eliminate their

influence, that is, we remove the data related to elderly people from our data and act as before and the normalized amount of extra deaths in the province We compare different.

```
ds = d[, .(n = sum(n)), .(y, m, ym_num, prov, age_group)]
for (M in c(1:12)) {
  for (PROV in unique(d$prov)) {
    dsm = ds[prov == PROV & m == M & ym_num > ym_num_start & age_group <= 4, ]
    dsm2fit = dsm[ym_num < ym_num_covid]
    fit = lm(n ~ ym_num, dsm2fit)
    if (lmp(fit) > 0.1){
      dsm$n_predicted = mean(dsm2fit$n)
      dsm$sd = sd(dsm2fit$n)
    }else{
      dsm$n_predicted = predict(fit ,dsm)
      dsm$sd = summary(fit)$sigma
    }

    dsm$moral_limit = dsm$n_predicted + 1.96 * dsm$sd
    dsm$diff = (dsm$n - dsm$moral_limit)/dsm$n
    dsm$diff[dsm$diff < 0] = 0
    dsm$moral_value = (dsm$n - dsm$n_predicted)
    dsm$moral_value[dsm$moral_value < 0] = 0

    dsm$ym_num[dsm$ym_num[dsm$ym_num]]

    for (ym_i in unique(dsm$ym_num[dsm$ym_num >= ym_num_covid])) {
      covid_data[PROV, as.character(ym_i)] = dsm[ym_num == ym_i & prov == PROV]$diff[1]
      moral_num[PROV, as.character(ym_i)] = dsm[ym_num == ym_i & prov == PROV]$moral_value[1]
    }
  }
}
```

now in new model we will check the normalized rate of death.

```
sort(rowSums(covid_data))
```

##				
##	0.00000000	0.00000000	0.00000000	0.00000000
##				
##	0.00000000	0.00000000	0.00000000	0.00000000
##				
##	0.00000000	0.00000000	0.00000000	0.00000000
##				
##	0.01940139	0.05280547	0.11145755	0.18494586
##				
##	0.22924794	0.25731168	0.35932612	0.38154041
##				
##	0.70411758	0.74719531	0.86791841	0.92334576
##				
##	0.94171353	0.94832906	1.07323523	1.11798230
##				
##	1.26325599	1.52956349	2.63384185	

here we can see death rate of young people (younger than 40) in different provinces are like above, so setting percent -of young people died over Corona over number of all young dead people- as our base.

if we took elder people as our base we have:

```

ds = d[, .(n = sum(n)), .(y, m, ym_num, prov, age_group)]
for (M in c(1:12)) {
  for (PROV in unique(d$prov)) {
    dsm = ds[prov == PROV & m == M & ym_num > ym_num_start & age_group > 4, ]
    dsm2fit = dsm[ym_num < ym_num_covid]
    fit = lm(n ~ ym_num, dsm2fit)
    if (lmp(fit) > 0.1){
      dsm$n_predicted = mean(dsm2fit$n)
      dsm$sd = sd(dsm2fit$n)
    }else{
      dsm$n_predicted = predict(fit ,dsm)
      dsm$sd = summary(fit)$sigma
    }

    dsm$moral_limit = dsm$n_predicted + 1.96 * dsm$sd
    dsm$diff = (dsm$n - dsm$moral_limit)/dsm$n
    dsm$diff[dsm$diff < 0] = 0
    dsm$moral_value = (dsm$n - dsm$n_predicted)
    dsm$moral_value[dsm$moral_value < 0] = 0

    dsm$ym_num[dsm$ym_num[dsm$ym_num]]

    for (ym_i in unique(dsm$ym_num[dsm$ym_num >= ym_num_covid])) {
      covid_data[PROV, as.character(ym_i)] = dsm[ym_num == ym_i & prov == PROV]$diff[1]
      moral_num[PROV, as.character(ym_i)] = dsm[ym_num == ym_i & prov == PROV]$moral_value[1]
    }
  }
}
sort(rowSums(covid_data))

```

```

##
##      0.00000000      0.00000000      0.00000000      0.00000000
##
##      0.00000000      0.00000000      0.00000000      0.00000000
##
##      0.00000000      0.00000000      0.00000000      0.00000000
##
##      0.00000000      0.00000000      0.00000000      0.00000000
##
##      0.00000000      0.00000000      0.00000000      0.00000000
##
##      0.00000000      0.00000000      0.00000000      0.00000000
##
##      0.00000000      0.00000000      0.00000000      0.00000000
##
##      0.00000000      0.00000000      0.00000000      0.07509412
##
##      0.12288434      0.13671454      0.45670094

```