

Cardiology Patient Records Analysis

Claire Jiaying Wu

22 Feb 2019

```
library(prettydoc)
library(data.table)
library(DT)
```

```
data.file <- "../homework 2/Data/Homework 2 Data.csv"
num.lines <- 1000
one.year <- 365.25
max.age <- 110

id.name <- "id"
t1.name <- "begin"
t2.name <- "end"
age.name <- "age"
diabetes.name <- "diabetes"
hypertension.name <- "hypertension"
kidney.disease.name <- "kidney_disease"
ace.name <- "ace"
beta.blocker.name <- "beta.blocker"
statin.name <- "statin"
hospital.name <- "hospital"
heart.attack.name <- "heart.attack"
death.name <- "death"
```

```
round.numerics <- function(x, digits){
  if(is.numeric(x)){
    x <- round(x = x, digits = digits)
  }
  return(x)
}

id.panel.overlaps.one.patient <- function(patient.dat, id.name, t1.name, t2.name, row.index.name){
  require(data.table)
  setDT(patient.dat)
  beginning.times <- patient.dat[, get(t1.name)]
  ending.times <- patient.dat[, get(t2.name)]
  overlapping.results <- patient.dat[, .(has_overlap = sum((get(t1.name) >
    beginning.times & get(t1.name) < ending.times) | (get(t2.name) < beginning.times &
    get(t2.name) > ending.times)) > 0), by = row.index.name]
  overlapping.true.false <- overlapping.results[, sum(has_overlap) > 0]
  return(overlapping.true.false)
}

id.panel.overlaps <- function(dat, id.name, t1.name, t2.name){
  require(data.table)
  setDT(dat)
  setorderv(x = dat, cols = c(id.name, t1.name), order = 1)
  dat[, record.index := 1:.N, by = id.name]
  ids.with.overlaps <- dat[, .(V1 = id.panel.overlaps.one.patient(patient.dat = .SD,
    id.name = id.name, t1.name = t1.name, t2.name = t2.name, row.index.name = "record.index")), by = get(id.name)]
  setnames(x = ids.with.overlaps, old = c("get", "V1"), new = c(id.name, "overlapping_panel_overlaps"))
  return(ids.with.overlaps)
}

identify.panel.gaps.one.patient <- function(patient.dat, t1.name, t2.name, first.value = 0, expected.gap.between = 1){
  require(data.table)
  setDT(patient.dat)
  gap.first.row <- (patient.dat[1, get(t1.name) > first.value])
  n <- patient.dat[, .N]
  if (n == 1) {
    res <- gap.first.row
  }
  if (n > 1) {
    t2.values <- patient.dat[1:(n - 1), get(t2.name)]
    gaps.other.rows <- patient.dat[2:n, get(t1.name) > t2.values + expected.gap.between]
    res <- c(gap.first.row, gaps.other.rows)
  }
}
```

```

    }
    return(res)
}

identify.panel.gaps <- function(dat, id.name, t1.name, t2.name, gap.name = "gap_before", fir
  require(data.table)
  setDT(dat)
  setorderv(x = dat, cols = c(id.name, t1.name), order = 1)
  dat[, `:=`(eval(gap.name), identify.panel.gaps.one.patient(patient.dat = .SD, t1.name = t1
  by = get(id.name))]
  return(dat[])
}

count.hospitalization <- function(x) {
  index <- which(x == 1 & c(0, x[1:(length(x)-1)]) != 1)
  return(length(index))
}

sum.usage.positive <- function(dat, colname){
  return(
    dat[get(colname)==1, .(`Medication Usage` = paste(colname, "- taking"),
      `Deaths Count` = sum(get(death.name), na.rm = TRUE),
      `Heart Attacks Count` = sum(get(heart.attack.name), na.rm = TRUE)
      `Hospitalization Count` = sum(get(hospital.name), na.rm = TRUE))]
  )
}

sum.usage.negative <- function(dat, colname){
  return(
    dat[get(colname)==0, .(`Medication Usage` = paste(colname, "- not taking"),
      `Deaths Count` = sum(get(death.name), na.rm = TRUE),
      `Heart Attacks Count` = sum(get(heart.attack.name), na.rm = TRUE)
      `Hospitalization Count` = sum(get(hospital.name), na.rm = TRUE))]
  )
}

utilization <- function(dat, colname){
  with_med <- dat[get(colname)==1, sum(get(t2.name)-get(t1.name))/one.year, by = id.name]
  without_med <- dat[get(colname)==0, sum(get(t2.name)-get(t1.name))/one.year, by = id.name]
  total_followup <- dat[, sum(get(t2.name)-get(t1.name))/one.year, by = id.name]
  overall <- dat[, .("Medicine" = eval(colname),
    "With-medicine Person-year" = sum(with_med$V1),
    "Without-medicine Person-year" = sum(without_med$V1),
    "Gap Person-year" = sum(total_followup$V1) - (sum(with_med$V1) + sum(wi
  return(overall[, lapply(X = .SD, FUN = "round.numerics", digits = 1)])
}

crude_event <- function(dat, colname){
  total_event <- dat[, sum(get(colname)), by = id.name]
  followup_time <- dat[, (sum(get(t2.name)-get(t1.name))/one.year)/100, by = id.name]
  dat2 <- dat[, .("Outcomes" = eval(colname),
    "Overall Follow-up Time (by units of 100 years)" = sum(followup_time$V1),
    "Nubmer of Events" = sum(total_event$V1),
    "Crude Rate" = sum(total_event$V1)/ sum(followup_time$V1))]
  return(dat2[, lapply(X = .SD, FUN = "round.numerics", digits = 1)])
}

crude_ace <- function(dat, colname){
  followup_time <- dat[get(ace.name) == 1,
    (sum(get(t2.name)-get(t1.name))/one.year)/100, by = id.name]
  total_event <- dat[get(ace.name) == 1, sum(get(colname)), by = id.name]
  dat2 <- dat[, .("Outcomes" = eval(colname),
    "Medication" = paste("With", ace.name),
    "Nubmer of Events" = sum(total_event$V1),
    "Overall Follow-up Time (by units of 100 years)" = sum(followup_time$V1),
    "Crude Rate" = sum(total_event$V1)/ sum(followup_time$V1))]
  return(dat2[, lapply(X = .SD, FUN = "round.numerics", digits = 1)])
}

crude_no_ace <- function(dat, colname){
  followup_time <- dat[get(ace.name) == 0,
    (sum(get(t2.name)-get(t1.name))/one.year)/100, by = id.name]
  total_event <- dat[get(ace.name) == 0, sum(get(colname)), by = id.name]
  dat2 <- dat[, .("Outcomes" = eval(colname),
    "Medication" = paste("Without", ace.name),
    "Nubmer of Events" = sum(total_event$V1),
    "Overall Follow-up Time (by units of 100 years)" = sum(followup_time$V1),
    "Crude Rate" = sum(total_event$V1)/ sum(followup_time$V1))]
  return(dat2[, lapply(X = .SD, FUN = "round.numerics", digits = 1)])
}

```

```

crude_beta <- function(dat, colname){
  followup_time <- dat[get(beta.blocker.name) == 1,
    (sum(get(t2.name)-get(t1.name))/one.year)/100, by = id.name]
  total_event <- dat[get(beta.blocker.name) == 1, sum(get(colname)), by = id.name]
  dat2 <- dat[, .("Outcomes" = eval(colname),
    "Medication" = paste("With", beta.blocker.name),
    "Nubmer of Events" = sum(total_event$V1),
    "Overall Follow-up Time (by units of 100 years)" = sum(followup_time$V1),
    "Crude Rate" = sum(total_event$V1)/ sum(followup_time$V1))]
  return(dat2[, lapply(X = .SD, FUN = "round.numerics", digits = 1)])
}

crude_no_beta <- function(dat, colname){
  followup_time <- dat[get(beta.blocker.name) == 0,
    (sum(get(t2.name)-get(t1.name))/one.year)/100, by = id.name]
  total_event <- dat[get(beta.blocker.name) == 0, sum(get(colname)), by = id.name]
  dat2 <- dat[, .("Outcomes" = eval(colname),
    "Medication" = paste("Without", beta.blocker.name),
    "Nubmer of Events" = sum(total_event$V1),
    "Overall Follow-up Time (by units of 100 years)" = sum(followup_time$V1),
    "Crude Rate" = sum(total_event$V1)/ sum(followup_time$V1))]
  return(dat2[, lapply(X = .SD, FUN = "round.numerics", digits = 1)])
}

crude_sta <- function(dat, colname){
  followup_time <- dat[get(statin.name) == 1,
    (sum(get(t2.name)-get(t1.name))/one.year)/100, by = id.name]
  total_event <- dat[get(statin.name) == 1, sum(get(colname)), by = id.name]
  dat2 <- dat[, .("Outcomes" = eval(colname),
    "Medication" = paste("With", statin.name),
    "Nubmer of Events" = sum(total_event$V1),
    "Overall Follow-up Time (by units of 100 years)" = sum(followup_time$V1),
    "Crude Rate" = sum(total_event$V1)/ sum(followup_time$V1))]
  return(dat2[, lapply(X = .SD, FUN = "round.numerics", digits = 1)])
}

crude_no_sta <- function(dat, colname){
  followup_time <- dat[get(statin.name) == 0,
    (sum(get(t2.name)-get(t1.name))/one.year)/100, by = id.name]
  total_event <- dat[get(statin.name) == 0, sum(get(colname)), by = id.name]
  dat2 <- dat[, .("Outcomes" = eval(colname),
    "Medication" = paste("Without", statin.name),
    "Nubmer of Events" = sum(total_event$V1),
    "Overall Follow-up Time (by units of 100 years)" = sum(followup_time$V1),
    "Crude Rate" = sum(total_event$V1)/ sum(followup_time$V1))]
  return(dat2[, lapply(X = .SD, FUN = "round.numerics", digits = 1)])
}

```

Reading the Data

```
dat <- fread(input = data.file, verbose = FALSE)
```

Correct the time of death for the patients:

```
dat <- dat[death == 1, begin := as.integer(end - 1)]
```

- How many rows are there?

```
dat[, .N]
```

```
[1] 2426922
```

- How many columns?

```
ncol(dat)
```

```
[1] 13
```

- How many unique patients are there?

```
dat[, .('Unique Patients' = length(unique(get(id.name))), `Number of Rows` = .N)]
```

```
Unique Patients Number of Rows
1:          10000          2426922
```

- What are the names of the columns? Do they match up with our description of the data set?

```
names(dat)
```

```
[1] "id"          "begin"       "end"         "age"
[5] "diabetes"    "hypertension" "kidney_disease" "ace"
[9] "beta.blocker" "statin"      "hospital"    "heart.attack"
[13] "death"
```

The column names above match up with the dataset description.

Inspection and Cleaning

Checking begin

```
# checking the missing values
dat[sum(is.na(t1.name)), .N]
```

```
[1] 0
```

```
# checking if the beginning times have negatives or are equal to / greater than the ending times
dat[get(t1.name) < 0 | get(t1.name) >= get(t2.name), .N]
```

```
[1] 0
```

Checking end

```
# checking the missing values
dat[sum(is.na(t2.name)), .N]
```

```
[1] 0
```

```
# checking if the ending times have negatives
dat[get(t2.name) < 0, .N]
```

```
[1] 0
```

```
# investigating if there are overlapping time frames
overlaps <- id.panel.overlaps(dat, id.name, t1.name, t2.name)
overlaps[overlapping_panels == TRUE, .N]
```

```
[1] 0
```

```
# investigating if there are gaps within time frames
gaps <- identify.panel.gaps(dat, id.name, t1.name, t2.name)
gaps[gap_before == TRUE & get(t1.name) != get(t2.name), .N]
```

```
[1] 0
```

Checking age

```
# checking the missing values
dat[sum(is.na(age.name)), .N]
```

```
[1] 0
```

```
# checking basic statistical summary
dat[, summary(get(age.name))]
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
29.00	59.00	66.00	65.51	72.00	100.00

Checking diabetes

```
# checking the missing values
dat[sum(is.na(diabetes.name)), .N]
```

```
[1] 0
```

Checking hypertension

```
# checking the missing values
dat[sum(is.na(hypertension.name)), .N]
```

```
[1] 0
```

Checking kidney_disease

```
# checking the missing values
dat[sum(is.na(kidney.disease.name)), .N]
```

```
[1] 0
```

Checking ace

```
# checking the missing values
dat[sum(is.na(ace.name)), .N]
```

```
[1] 0
```

Checking beta.blocker

```
# checking the missing values
dat[sum(is.na(beta.blocker.name)), .N]
```

```
[1] 0
```

Checking statin

```
# checking the missing values
dat[sum(is.na(statin.name)), .N]
```

```
[1] 0
```

Checking hospital

```
# checking the missing values
dat[sum(is.na(hospital.name)), .N]
```

```
[1] 0
```

Checking heart.attack

```
# checking the missing values
dat[sum(is.na(heart.attack.name)), .N]
```

```
[1] 0
```

Checking death

```
# checking the missing values
dat[sum(is.na(death.name)), .N]
```

```
[1] 0
```

```
# checking the data type of each variable
str(dat)
```

```
Classes 'data.table' and 'data.frame': 2426922 obs. of 15 variables:
 $ id      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ begin   : int  0 14 18 19 31 39 49 55 58 62 ...
 $ end     : int  14 18 19 31 39 49 55 58 62 70 ...
 $ age     : int  55 55 55 55 55 55 55 55 55 55 ...
 $ diabetes : int  0 0 0 0 0 0 0 0 0 0 ...
 $ hypertension : int  1 1 1 1 1 1 1 1 1 1 ...
 $ kidney_disease: int  0 0 0 0 0 0 0 0 0 0 ...
 $ ace     : int  1 1 0 1 1 1 1 1 0 1 ...
 $ beta.blocker : chr  "0" "1" "0" "0" ...
 $ statin    : chr  "1" "1" "1" "1" ...
 $ hospital  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ heart.attack : int  0 0 0 0 0 0 0 0 0 0 ...
 $ death     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ record.index : int  1 2 3 4 5 6 7 8 9 10 ...
 $ gap_before  : logi FALSE FALSE FALSE FALSE FALSE ...
 - attr(*, ".internal.selfref")=<externalptr>
 - attr(*, "index")= atomic
 ..- attr(*, "__death")= int  1 2 3 4 5 6 7 8 9 10 ...
```

After checking, the data types for beta.blocker and statin are character, while they should be integer. Further checks are in need.

For beta.blocker:

```
dat[, sort(unique(get(beta.blocker.name)))]
```

```
[1] "0"      "1"      "didn't take" "False"    "Filled"
[6] "N"      "no"     "No"         "nope"     "Not filled"
[11] "Picked up" "True"   "Utilized"   "yeah"     "yes"
[16] "Yes"
```

Cleaning the characters to integers:

```
positive <- c("Filled", "Picked up", "True", "Utilized", "yeah", "yes", "Yes")
negative <- c("didn't take", "False", "N", "no", "No", "nope", "Not filled")

ps <- which(dat[, beta.blocker] %in% positive)
ng <- which(dat[, beta.blocker] %in% negative)

dat <- dat[ps, beta.blocker := 1]
dat <- dat[ng, beta.blocker := 0]

dat <- dat[, beta.blocker := as.numeric(beta.blocker)]
dat[, is.numeric(beta.blocker)]
```

```
[1] TRUE
```

```
dat[, sort(unique(beta.blocker))]
```

```
[1] 0 1
```

For statin:

```
dat[, sort(unique(get(statin.name)))]
```

```
[1] ""      "0"      "1"      "appt"    "Ask doctor"
[6] "BP Normal" "diabetes" "high risk"
```

Cleaning the characters to integers:

Since the records other than "0" and "1" are hard to be categorized to yes or no, and there are only 276 rows (0.011% of total records), I replaced these values by NA.

```
to_be_na <- c("", "appt", "Ask doctor", "BP Normal", "diabetes", "high risk")
nas <- which(dat[, statin] %in% to_be_na)
length(nas)
```

```
[1] 276
```

```
dat <- dat[nas, statin := NA]
dat <- dat[, statin := as.numeric(statin)]
dat[, is.numeric(statin)]
```

```
[1] TRUE
```

```
dat[, summary(statin)]
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.0000	1.0000	1.0000	0.7794	1.0000	1.0000	276

Patient-Level Summaries

For age, diabetes, hypertension, and kidney_disease, what are the average values and standard deviations at baseline? For age, this would be an average in years. For the disease states, this would be the percentage of the population who have these conditions.

```
# computing the mean and SD of the age variable
age_summary <- dat[get(t1.name) == 0, .(Item = eval(age.name),
  `Mean Value` = as.numeric(lapply(X = .SD, FUN = "mean")),
  `SD Value` = as.numeric(lapply(X = .SD, FUN = "sd")))]
age_summary <- age_summary[, lapply(X = .SD, FUN = "round.numerics", digits = 1)]

# computing the mean and SD of the diseases variable
dis_names <- c(diabetes.name, hypertension.name, kidney.disease.name)
dis_summary <- dat[get(t1.name) == 0, .(Item = eval(dis_names),
  `Mean Value` = as.numeric(lapply(X = .SD, FUN = "mean")),
  `SD Value` = as.numeric(lapply(X = .SD, FUN = "sd")))]
dis_summary <- dis_summary[, lapply(X = .SD, FUN = "round.numerics", digits = 1)]
dis_summary <- dis_summary[, `Mean Value` := paste(`Mean Value`, "%")]

# combining together and displaying in datatable
summaries <- rbind(age_summary, dis_summary)
datatable(summaries, rownames = FALSE)
```

Show entries

Search:

Item	Mean Value	SD Value
age	65.5	10
diabetes	19.4 %	0.4
hypertension	60 %	0.5
kidney_disease	10.2 %	0.3

Showing 1 to 4 of 4 entries

Previous Next

Counting Outcomes

Part A

How many heart attacks were there in follow-up? How many deaths occurred?

```
dat[, .(`Follow-up Heart Attacks Count` = sum(get(heart.attack.name)),
      `Deaths Count` = sum(get(death.name)))]
```

```
Follow-up Heart Attacks Count Deaths Count
1: 3262 6086
```

Part B

How many total hospitalizations occurred across all of the patients? - a single hospitalization may span multiple rows of data. Incorporate this count into the previous table. Compare the value calculated here to the number of rows with hospitalizations.

```
by_patient <- dat[, lapply(X = .SD, FUN = "count.hospitalization"), .SDcol = hospital.name,
by_patient[, lapply(X = .SD, FUN = "sum"), .SDcol = hospital.name]
```

```
hospital
1: 49642
```

Counting Outcomes by Medication Usage

Count the number of deaths, heart attacks, and hospitalizations *split by* medication usage. Show how many of these outcomes occurred while the patients were taking each medicine (ACE Inhibitors, Beta Blockers, and Statins) - and while they were not taking them.

```
outcomes_summaries <- rbind(sum.usage.positive(dat, ace.name), sum.usage.negative(dat, ace.name),
                             sum.usage.positive(dat, beta.blocker.name), sum.usage.negative(dat, beta.blocker.name),
                             sum.usage.positive(dat, statin.name), sum.usage.negative(dat, statin.name))
datatable(outcomes_summaries, rownames = FALSE)
```

Show 10 entries

Search:

Medication Usage	Deaths Count	Heart Attacks Count	Hospitalization Count
ace - taking	2639	1475	22400
ace - not taking	3447	1787	27242
beta.blocker - taking	2820	1333	20170
beta.blocker - not taking	3266	1929	29472
statin - taking	2605	1536	23413
statin - not taking	3479	1726	26229

Showing 1 to 6 of 6 entries

Previous 1 Next

Follow-Up

Each patient may spend some time in follow-up on the medication and other periods not using it. We want to get a sense of how much these medicines are used relative to the available time. A **person-year** is defined as one year of observation for one patient. 10 person-years can be accumulated by following one person for 10 years, two for 5 apiece, three for 2, 7, and 1, respectively, or other combinations. With this in mind, we want to study the utilization of medicines.

- How many total person-years of observation do we have in the records? What is the average number of years of observation per patient?

```
person_year <- dat[get(ace.name) == 1 | get(beta.blocker.name) == 1 | get(statin.name) == 1,
                  .(`Person-year Count` = sum(get(t2.name) - get(t1.name))/ one.year), by =
person_year_summary <- person_year[, .(`Total Person-year` = sum(get("Person-year Count")),
                                       `Mean Person-year` = mean(get("Person-year Count")))]
person_year_summary[, lapply(X = .SD, FUN = "round.numerics", digits = 1)]
```

```
Total Person-year Mean Person-year
1: 36386.7 3.6
```

Utilization

- How many person-years did the patients spend on each medicine – ACE Inhibitors, Beta Blockers, and Statins? How much time was spent off of them? How much time was missing from observation?

```
utilization_summaries <- rbind(utilization(dat, ace.name), utilization(dat, beta.blocker.name),
datatable(utilization_summaries, rownames = FALSE))
```

Show 10 entries

Search:

Medicine	With-medicine Person-year	Without-medicine Person-year	Gap Person-year
ace	26965.8	10095.4	0
beta.blocker	22329.4	14731.8	0
statin	28867.4	8180.6	13.1

Showing 1 to 3 of 3 entries

Previous 1 Next

Crude Event Rates

We compared the counts for the outcomes of hospitalization, heart attacks, and death against the relative follow-up time. Compute the *crude rates* (the mean number of outcomes) per 100 person years of follow-up - showing the overall amount of follow-up time, the number of events for each outcome, and their ratio in units of events per 100 person years. (Defined one year as 365.25 days.)

```
crude_rates <- rbind(crude_event(dat, hospital.name), crude_event(dat, heart.attack.name),
datatable(crude_rates, rownames = FALSE))
```

Show 10 entries

Search:

Outcomes	Overall Follow-up Time (by units of 100 years)	Number of Events	Crude Rate
hospital	370.6	49642	133.9
heart.attack	370.6	3262	8.8
death	370.6	6086	16.4

Showing 1 to 3 of 3 entries

Previous 1 Next

Crude Event Rates By Medication Usage

How do the crude rates of hospitalization, heart attacks, and death per 100 person-years of follow-up differ depending on medication usage? Show the number of events and crude rates while taking and not taking each medicine:

- ACE Inhibitors

```
crude_ace <- rbind(crude_ace(dat, hospital.name), crude_no_ace(dat, hospital.name),
crude_ace(dat, heart.attack.name), crude_no_ace(dat, heart.attack.name),
crude_ace(dat, death.name), crude_no_ace(dat, death.name))
datatable(crude_ace, rownames = FALSE)
```

Show 10 entries

Search:

Outcomes	Medication	Number of Events	Overall Follow-up Time (by units of 100 years)	Crude Rate
hospital	With ace	22400	269.7	83.1
hospital	Without ace	27242	101	269.8
heart.attack	With ace	1475	269.7	5.5
heart.attack	Without ace	1787	101	17.7
death	With ace	2639	269.7	9.8
death	Without ace	3447	101	34.1

Showing 1 to 6 of 6 entries

Previous 1 Next

- Beta Blockers

```
crude_beta <- rbind(crude_beta(dat, hospital.name), crude_no_beta(dat, hospital.name),
                   crude_beta(dat, heart.attack.name), crude_no_beta(dat, heart.attack.name),
                   crude_beta(dat, death.name), crude_no_beta(dat, death.name))
datatable(crude_beta, rownames = FALSE)
```

Show 10 entries

Search:

Outcomes	Medication	Nubmer of Events	Overall Follow-up Time (by units of 100 years)	Crude Rate
hospital	With beta.blocker	20170	223.3	90.3
hospital	Without beta.blocker	29472	147.3	200.1
heart.attack	With beta.blocker	1333	223.3	6
heart.attack	Without beta.blocker	1929	147.3	13.1
death	With beta.blocker	2820	223.3	12.6
death	Without beta.blocker	3266	147.3	22.2

Showing 1 to 6 of 6 entries

Previous 1 Next

- Statins

```
crude_sta <- rbind(crude_sta(dat, hospital.name), crude_no_sta(dat, hospital.name),
                   crude_sta(dat, heart.attack.name), crude_no_sta(dat, heart.attack.name),
                   crude_sta(dat, death.name), crude_no_sta(dat, death.name))
datatable(crude_sta, rownames = FALSE)
```

Show 10 entries

Search:

Outcomes	Medication	Nubmer of Events	Overall Follow-up Time (by units of 100 years)	Crude Rate
hospital	With statin	23413	288.7	81.1
hospital	Without statin	26229	81.8	320.6
heart.attack	With statin	1536	288.7	5.3
heart.attack	Without statin	1726	81.8	21.1
death	With statin	2605	288.7	9
death	Without statin	3479	81.8	42.5

Showing 1 to 6 of 6 entries

Previous 1 Next

Unadjusted Odds Ratios

What is the impact of each medication? One way to calculate their impact is with the *unadjusted odds ratio*, which compares the rate of outcomes while taking the medicine to the rate without taking the medicine. For reference, an odds ratio less than 1 demonstrates that a factor is associated with a reduction in an outcome, a value greater than 1 shows that the factor is associated with an increase in an outcome, and a value close to 1 shows no association. For each medicine, we computed the unadjusted odds ratios for hospitalization, heart attacks, and death.

- ACE Inhibitors

```
ace_uor <- crude_ace[, .(`Unadjusted Odds Ratios` = as.numeric(.SD[1,4])/as.numeric(.SD[2,4]))
datatable(ace_uor[, lapply(X = .SD, FUN = "round.numerics", digits = 2)], rownames = FALSE)
```

Show 10 entries

Search:

Outcomes	Unadjusted Odds Ratios
hospital	0.31
heart.attack	0.31
death	0.29

Showing 1 to 3 of 3 entries

Previous 1 Next

Based on the results, ACE Inhibitors do have obvious impact on reducing the numbers of hospitalization, heart attack, and the mortality. We could also see that the impact is slightly larger on reducing the number of death based on the given dataset. (But note that every impact here does not indicate any casuations between this medication and the outcomes.

- Beta Blockers

```
beta_uor <- crude_beta[, .(`Unadjusted Odds Ratios` = as.numeric(.SD[1,4])/as.numeric(.SD[2,4]),
  datatable(beta_uor[, lapply(X = .SD, FUN = "round.numerics", digits = 2)], rownames = FALSE)
```

Show 10 entries

Search:

Outcomes	Unadjusted Odds Ratios
hospital	0.45
heart.attack	0.46
death	0.57

Showing 1 to 3 of 3 entries

Previous 1 Next

Based on the results, Beta Blockers do have obvious impact on reducing the numbers of hospitalization, heart attack, and the mortality, while the impact is relatively smaller on reducing the number of death based on the given dataset. In addition, comparing the unadjusted odds ratios of this medicine with the other two, Beta Blockers have the highest ratios in all the crude events, which means the performance of this treatment might not be as good as the other two. (But note that every impact here does not indicate any casuations between this medication and the outcomes.

- Statins

```
sta_uor <- crude_sta[, .(`Unadjusted Odds Ratios` = as.numeric(.SD[1,4])/as.numeric(.SD[2,4]),
  datatable(sta_uor[, lapply(X = .SD, FUN = "round.numerics", digits = 2)], rownames = FALSE)
```

Show 10 entries

Search:

Outcomes	Unadjusted Odds Ratios
hospital	0.25
heart.attack	0.25
death	0.21

Showing 1 to 3 of 3 entries

Previous 1 Next

Based on the results, statins do have obvious impact on reducing the numbers of hospitalization, heart attack, and the mortality. We could also see that the impact is slightly larger on reducing the number of death based on the given dataset. Moreover, this medicine has the best performance among the other two, if we only look at the unadjusted odds ratios here. (But note that every impact here does not indicate any casuations between this medication and the outcomes.