

Task 1 - Generate Alarms Data

The following code snippet was used to generate the mock-up alarm frequency and regional information, which that were used as examples in the lecture last week. Run the code with your student ID as seed variable and generate your own (random) data

```
In [162... # load library
library(tidyverse)

# set random seed
seed <- 686249907 # <Replaced '765' with my student ID here.>
set.seed(seed)

# generate alarm id, dates and region vectors
alarms_id.vt <- paste0('alarms_', sample(LETTERS, size = 10, replace = FALSE))
alarms_dates.vt <- paste0('d.', seq(Sys.Date()-60, Sys.Date(), by = '1 day')) # 60 day
alarms_region.vt <- c('AKL_North', 'AKL_Central', 'Waiheke', 'AKL_South', 'AKL_Others')

# generate random alarm frequency counts
alarms_count.mt <- matrix(round(runif(length(alarms_id.vt) * length(alarms_dates.vt)) *
                                nrow = length(alarms_id.vt), ncol=length(alarms_dates.vt))
colnames(alarms_count.mt) <- alarms_dates.vt

# set up data frames
alarms_count.df <- data.frame(alarm_id = alarms_id.vt, alarms_count.mt)
alarms_info.df <- data.frame(alarm_ID = alarms_id.vt, alarms_region = alarms_region.vt)
```

Task 2 - {tidyverse} Operations

Use the data generated to answer the following question: what is the average number of alarms per day in each region?

```
In [163... alarms_insight.df = alarms_count.df %>%
  pivot_longer(-alarm_id, names_to='date', values_to='frequency') %>%
  left_join(alarms_info.df, by=c('alarm_id' = 'alarm_ID')) %>%
  group_by(date, alarms_region) %>%
  summarise(avg_count = mean(frequency)) %>%
  separate(date, c('prefix', 'year', 'month', 'day')) %>%
  select(-prefix)
```

`summarise()` has grouped output by 'date'. You can override using the `.groups` argument.

```
In [164... alarms_insight.df
```

A tibble: 305 × 5

year	month	day	alarms_region	avg_count
<chr>	<chr>	<chr>	<chr>	<dbl>
2022	01	07	AKL_Central	404216498
2022	01	07	AKL_North	390631160
2022	01	07	AKL_Others	321501920
2022	01	07	AKL_South	273605540
2022	01	07	Waiheke	545564040
2022	01	08	AKL_Central	524049658
2022	01	08	AKL_North	316742309
2022	01	08	AKL_Others	171393187
2022	01	08	AKL_South	459933812
2022	01	08	Waiheke	566158320
2022	01	09	AKL_Central	357655234
2022	01	09	AKL_North	310978298
2022	01	09	AKL_Others	525591878
2022	01	09	AKL_South	254867997
2022	01	09	Waiheke	378910289
2022	01	10	AKL_Central	313730518
2022	01	10	AKL_North	537904401
2022	01	10	AKL_Others	555072654
2022	01	10	AKL_South	584490574
2022	01	10	Waiheke	162342754
2022	01	11	AKL_Central	588575370
2022	01	11	AKL_North	395243417
2022	01	11	AKL_Others	544542323
2022	01	11	AKL_South	405684703
2022	01	11	Waiheke	288114078
2022	01	12	AKL_Central	363421186
2022	01	12	AKL_North	366368832
2022	01	12	AKL_Others	342411466
2022	01	12	AKL_South	444587801
2022	01	12	Waiheke	532102250
⋮	⋮	⋮	⋮	⋮
2022	03	03	AKL_Central	442880208
2022	03	03	AKL_North	509858014
2022	03	03	AKL_Others	236514590
2022	03	03	AKL_South	666186214
2022	03	03	Waiheke	512126668
2022	03	04	AKL_Central	409359428
2022	03	04	AKL_North	463078036
2022	03	04	AKL_Others	464134552
2022	03	04	AKL_South	108520557

year	month	day	alarms_region	avg_count
<chr>	<chr>	<chr>	<chr>	<dbl>
2022	03	04	Waiheke	149532680
2022	03	05	AKL_Central	310662183
2022	03	05	AKL_North	66314590
2022	03	05	AKL_Others	233997706
2022	03	05	AKL_South	294177968
2022	03	05	Waiheke	179080156
2022	03	06	AKL_Central	288127110
2022	03	06	AKL_North	526912454
2022	03	06	AKL_Others	372609197
2022	03	06	AKL_South	95036126
2022	03	06	Waiheke	376845210
2022	03	07	AKL_Central	297490214
2022	03	07	AKL_North	452086944
2022	03	07	AKL_Others	324125840
2022	03	07	AKL_South	348098438
2022	03	07	Waiheke	377673714
2022	03	08	AKL_Central	529988500
2022	03	08	AKL_North	176984974
2022	03	08	AKL_Others	244552858
2022	03	08	AKL_South	496592344
2022	03	08	Waiheke	363324694

Task 3: Date operation using {lubridate}

Use the data generated to answer the following question: which days of the week has the highest and the lowest average number of alarms across the entire Auckland?

Hint: guess what, there is a cheat sheet.

```
In [165... library(lubridate)
```

```
In [166... avg_alarms_per_weekday <- alarms_insight.df %>%
  unite(date, year, month, day, sep='-') %>%
  mutate(weekday = wday(date, label=T, abbr=F)) %>%
  group_by(weekday) %>%
  summarise(max_avg=max(avg_count), min_avg=min(avg_count))
```

Average number of alarms across all regions, sorted by **maximum** average.

```
In [167... avg_alarms_per_weekday[order(avg_alarms_per_weekday$max_avg),]
```

A tibble: 7 × 3

weekday	max_avg	min_avg
<ord>	<dbl>	<dbl>
Sunday	526912454	36545567
Saturday	566158320	44884958
Wednesday	621904651	75038223
Monday	635515190	105728908
Friday	636532446	22162170
Tuesday	647938858	61954680
Thursday	666186214	52402932

Average number of alarms across all regions, sorted by **minimum** average.

In [168...

```
avg_alarms_per_weekday[order(avg_alarms_per_weekday$min_avg), ]
```

A tibble: 7 × 3

weekday	max_avg	min_avg
<ord>	<dbl>	<dbl>
Friday	636532446	22162170
Sunday	526912454	36545567
Saturday	566158320	44884958
Thursday	666186214	52402932
Tuesday	647938858	61954680
Wednesday	621904651	75038223
Monday	635515190	105728908

Task 4: Retrospective

Write a short paragraph summarising your experience and comments about using {tidyverse} for data wrangling tasks.

I had a fantastic experience using tidyverse for these data wrangling tasks. R and tidyverse provide a much smoother and faster experience than other data analysis tools that I have experience with, like Excel and Python. It seems that the basic functions provided by tidyverse fill the same niche that I have used SQL and SQL-like languages for in the past. I like that the piping format of tidyverse operations matches the SQL `select` function and its downstream modifiers. The various tidyverse documentation sites and Stackoverflow forum questions make troubleshooting easy.

EOF