# assignment_1

2024-02-23

**Import Packages** The code below imports the packages used in this assignment.

```
# Import Packages

suppressMessages({
  library(tidyverse)
})
print('loaded')
```

```
## [1] "loaded"
```

### Q1 - HDB Resale Flat Prices

Before starting on any of the given questions, we scope through the dataset to understand the structure and feel of the data. We also conduct a very basic check for null values.

The code below imports the downloaded data.

```
# Import Data

suppressMessages({
  resale_flat_prices_2017 <- read_csv(
    'data/ResaleflatpricesbasedonregistrationdatefromJan2017onwards.csv'
    )
})
print('loaded')
```

```
## [1] "loaded"
```

### Dataset Overview

Display the top 5 rows to get a general feel of the data.

```
# View Data

head(resale_flat_prices_2017)
```

```
## # A tibble: 6 x 11
##   month town  flat_type block street_name storey_range floor_area_sqm flat_model
##   <chr> <chr> <chr>     <chr> <chr>       <chr>                <dbl> <chr>
## 1 2017~ ANG ~ 2 ROOM    406   ANG MO KIO~ 10 TO 12                44 Improved
## 2 2017~ ANG ~ 3 ROOM    108   ANG MO KIO~ 01 TO 03                67 New Gener~
## 3 2017~ ANG ~ 3 ROOM    602   ANG MO KIO~ 01 TO 03                67 New Gener~
## 4 2017~ ANG ~ 3 ROOM    465   ANG MO KIO~ 04 TO 06                68 New Gener~
```

```
## 5 2017~ ANG ~ 3 ROOM    601   ANG MO KIO~ 01 TO 03                 67 New Gener~
## 6 2017~ ANG ~ 3 ROOM    150   ANG MO KIO~ 01 TO 03                 68 New Gener~
## # i 3 more variables: lease_commence_date <dbl>, remaining_lease <chr>,
## #   resale_price <dbl>
```

**Observation** The dataset is arranged in ascending numerical and alphabetical order, starting with the year 2017 and with Ang Mo Kio.

Next we check the last 5 entries of the dataset to understand how updated it is.

```
tail(resale_flat_prices_2017,5)
```

```
## # A tibble: 5 x 11
##   month town  flat_type block street_name storey_range floor_area_sqm flat_model
##   <chr> <chr> <chr>     <chr> <chr>       <chr>                <dbl> <chr>
## 1 2024~ YISH~ EXECUTIVE 387   YISHUN RIN~ 04 TO 06               142 Apartment
## 2 2024~ YISH~ EXECUTIVE 355   YISHUN RIN~ 01 TO 03               154 Maisonette
## 3 2024~ YISH~ EXECUTIVE 606   YISHUN ST ~ 10 TO 12               142 Apartment
## 4 2024~ YISH~ EXECUTIVE 824   YISHUN ST ~ 07 TO 09               146 Maisonette
## 5 2024~ YISH~ MULTI-GE~ 666   YISHUN AVE~ 04 TO 06               164 Multi Gen~
## # i 3 more variables: lease_commence_date <dbl>, remaining_lease <chr>,
## #   resale_price <dbl>
```

**Observation** The latest entry is in February 2024, of which the current date is 23 February 2024. A quick check from the beta.data.gov.sg website shows this dataset was last updated 8 hours ago, thus it can be assumed that this dataset is most updated up to this current week.

**Dataset Size** We obtain the shape of the dataset to understand the size of the dataset we will be working with.

```
# Shape of dataset

dim(resale_flat_prices_2017)
```

```
## [1] 173334     11
```

**Observation** The dataset has 173,334 rows and 11 columns, considered quite a large dataset to deal with.

**Dataset Columns** Next we print all column names to understand the variables we will be working with.

```
names(resale_flat_prices_2017)
```

```
##  [1] "month"              "town"               "flat_type"
##  [4] "block"              "street_name"        "storey_range"
##  [7] "floor_area_sqm"     "flat_model"         "lease_commence_date"
## [10] "remaining_lease"    "resale_price"
```

**Observation** The dataset can be broken down into 5 main categories namely: time, location, flat details, lease and price.

**Dataset Summary** Next we print a summary of the dataset to understand the datatypes of each column.

```
summary(resale_flat_prices_2017)
```

```
##     month               town            flat_type            block
##  Length:173334      Length:173334      Length:173334      Length:173334
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  street_name        storey_range       floor_area_sqm    flat_model
##  Length:173334      Length:173334      Min.   : 31.00   Length:173334
##  Class :character   Class :character   1st Qu.: 82.00   Class :character
##  Mode  :character   Mode  :character   Median : 93.00   Mode  :character
##                                        Mean   : 97.24
##                                        3rd Qu.:112.00
##                                        Max.   :249.00
##  lease_commence_date remaining_lease     resale_price
##  Min.   :1966        Length:173334      Min.   : 140000
##  1st Qu.:1985        Class :character   1st Qu.: 368000
##  Median :1996        Mode  :character   Median : 463000
##  Mean   :1996                           Mean   : 493357
##  3rd Qu.:2009                           3rd Qu.: 587000
##  Max.   :2022                           Max.   :1568888
```

**Observation** The dataset has 2 data types, strings and numerical values. However, it looks like some data transformation is needed later on for some columns to convert its string value to a numerical value (eg. remaining_lease).

### Check for Null Values

The code below checks the dataset for null values.

```
# null value check

na_check <- colSums(is.na(resale_flat_prices_2017)) > 0
print(na_check)
```

```
##               month                town           flat_type               block
##               FALSE               FALSE               FALSE               FALSE
##         street_name        storey_range      floor_area_sqm         flat_model
##               FALSE               FALSE               FALSE               FALSE
## lease_commence_date     remaining_lease        resale_price
##               FALSE               FALSE               FALSE
```

**Observation** The dataset does not look to have any null values for now. However, we cannot assume that the values are free of error of course. It could be the same where null values are filled with 0 or 9999.

After the above cursory check of the dataset, we are ready to begin the assignment.

**Question 1A-1** In 2021, there have been 261 HDB flats transacted at or more than $1m. Compute how many such transactions there were in the last year.

**Aim** Find the number of more than $1m transactions in 2023.

**Approach** This would involve trimming the dataset down based on several conditional statements.

**Extract Transcations in 2023** The code below extracts the list of transcations that occurred in 2023.

```r
transcations_2023 <- resale_flat_prices_2017 %>%
  filter(grepl("2023", month))
```

**Extract Transcations >= $1m in 2023** The code below filters the dataframe for transcations where resale_price >= $1m

```r
transcations_2023_million <- transcations_2023 %>%
  filter(resale_price >= 1000000)
```

**Extract Total Count** The code below gets the total count of the transcations_2023_million dataframe.

```r
count_2023_million <- nrow(transcations_2023_million)
print(count_2023_million)
```

```
## [1] 470
```

**Answer** There were **470** resale transcations in 2023 with prices greater than or equal to $1m.

**Question 1A-2** HDB resale prices rose 0.8 per cent in December 2021 from the previous month. Do the same comparison for the same period in the last year.