MATH2405 TP4, 2024

Assignment 1

Michael Teixeira S4133975

Staff salary and wages budget 2014-18

Introduction

In this report, we will work with a database provided by the **City of Melbourne Council**[1]. It contains staff salary and wages plan and budget for the financial years 2014 to 2018.

Setup

This chunk is where we load the packages required for producing the report, if not previously installed we need to install them individually by using the function: install.packages("name_of_package") and load it by using library(name_of_package)

```
# The following packages were used to build this report:

library(tidyverse) # [2] Tidyverse collection of packages. It assists with data import, tidying, manipulation, and data visualisation. For example:"dplyr", "readr", "tidyr", # and magrittr for pipe operators.
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(magrittr) # Package from tidyverse for pipe operators.
```

```
##
## Attaching package: 'magrittr'
##
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

Data Description

This dataset shows the summary of planned Human Resources expenditure across permanent and temporary council staff. It also includes a breakdown of expenditure and number of staff for each category across the organisation's structure. It contains 6 variables:

- As at Date - The date when this financial record was calculated.
- Year
- Actual/Budget/Plan - A category value to indicate if the figure represents actual, budgeted or planned values.
- Category - Organisational branch for this financial measure.
- Description - Category value (one of: Permanent full time, Permanent part time, Total casuals and other) representing the contractual arrangement for each category.
- Value ($,000) - The value in dollars (AUD).

Import Data

The file was downloaded from the source[1] in *.csv* file format to our local disk and a copy was made with the name from *"staff-salary-and-wages-budget-2014-15-2"* to *"Assigment1"*. Now we will be importing to Rstudio.

```
# Load file into a variable df, using read.csv.

df <- read.csv("Assigment1.csv", sep=";") # Data separated by semicolon
```

Inspect dataset and variables

With the following functions we can analyse part of the data frame, identify the structure of the data, which contains 52 rows and 6 columns, and get statistics numbers that can help us to get to know better the data, as well identify possible outliers.

Inspecting the structure of the df:

```
head(df) # First 6 rows, analyse of relevant variables and their types.
```

```
##       as_at_date           year actual_budget_plan
## 1 2014-07-01 2014-15                      BUDGET
## 2 2014-07-01 2014-15                      BUDGET
## 3 2014-07-01 2014-15                      BUDGET
## 4 2014-07-01 2014-15                      BUDGET
## 5 2014-07-01 2015-16                        PLAN
```

```
## 6 2014-07-01 2015-16                    PLAN
##                                         category                description value_000
## 1                CITY PLANNING AND INFRASTRUCTURE Permanent full time 37143.97
## 2                             CORPORATE BUSINESS Permanent full time 19769.99
## 3                                  CITY BUSINESS Permanent full time 17260.93
## 4 CITY GOVERNANCE, LEARNING AND DEVELOPMENT Permanent full time          5975.22
## 5       CORPORATE BUSINESS Permanent part time 766.27 ## 6       COMMUNITY DEVELOPMENT
Permanent full time 36404.90
```

```
str(df) # Structure of data, 52 observations and 6 variables. Five char variables and one numeric.
```

```
## $ category              : chr "CITY PLANNING AND INFRASTRUCTURE" "CORPORATE BUSINESS" "CITY BUSINESS" "
## $ description           : chr "Permanent full time" "Permanent full time" "Permanent full time" "Perman
## $ value_000             : num 37144 19770 17261 5975 766 ...
## 'data.frame':       52 obs. of 6 variables:
## $ as_at_date                         : chr "2014-07-01" "2014-07-01" "2014-07-01" "2014-07-01" ...
## $ year                  : chr "2014-15" "2014-15" "2014-15" "2014-15" ...
## $ actual_budget_plan: chr "BUDGET" "BUDGET" "BUDGET" "BUDGET" ...
```

```
summary(df) # Statistics numbers, detecting possible outliers.
```

```
##      as_at_date           year               actual_budget_plan     category
## Length:52            Length:52             Length:52             Length:52
## Class :character     Class :character      Class :character      Class :character
## Mode :character      Mode :character       Mode :character       Mode :character
##
##
##
## description           value_000
## Length:52            Min.     :      16.93
## Class :character     1st Qu.: 757.15
## Mode :character      Median : 4090.15
##                      Mean     :11039.04
##                      3rd Qu.:19147.20
##                      Max.     :42590.38
```

Identifying possible data entry errors, misspellings:

```
# unique function to check how many unique values each column contain.

 unique(df$year)
```

```
## [1] "2014-15" "2015-16" "2016-17" "2017-18"
```

```
unique(df$actual_budget_plan)
```

```
## [1] "BUDGET" "PLAN"
```

```
unique(df$category)
```

```
## [1] "CITY PLANNING AND INFRASTRUCTURE"
## [2] "CORPORATE BUSINESS"
## [3] "CITY BUSINESS"
## [4] "CITY GOVERNANCE, LEARNING AND DEVELOPMENT"
## [5] "COMMUNITY DEVELOPMENT"
## [6] "CITY DESIGN" ## [7]
"CASUALS AND OTHER"
```

```
unique(df$description)
```

```
## [1] "Permanent full time"                "Permanent part time"
## [3] "Total casuals and other"
```

Tidy data

Previously, with the `unique()` function, we checked that there were no misspellings in the data, but we noticed that the variables were in the wrong type of data and the names of the columns could be clearer. At this stage, we proceed with tidying the data.

```
# Changing name of variables for a clear description:

 df %<>% rename("date_calc" = "as_at_date","financial_year" = "year",

"contract_type" = "description", "value_aud" = "value_000")

# Updating data type format of variables:

df$date_calc <- as.Date(df$date_calc)

# Created a order for the factor year
 df$financial_year <- factor(c("2014-15","2015-16","2016-17","2017-18"),
levels = c("2014-15","2015-16","2016-17","2017-18"))

levels(df$financial_year)
```

```
## [1] "2014-15" "2015-16" "2016-17" "2017-18"
```

```
df$actual_budget_plan <- as.factor(df$actual_budget_plan)

df$category <- as.factor(df$category)

df$contract_type <- as.factor(df$contract_type)

class(df$value_aud) # value_aud is already numeric type
```

```
## [1] "numeric"
```

```
# Checking for duplicates.


duplicated(df) # All results return FALSE, so no duplicated values.
```

Summary Statistics

The data has been cleaned; we are now ready to get statistics from the values.

We start by getting the stats of values that have been used in each financial year, in each category:

```
# Statistics values grouped first by financial year and then by category,  and assigned to
value_stats variable.

value_stats <- df %>%
group_by(financial_year, category) %>%
 summarise(Min_value = min(value_aud, na.rm = TRUE),
                   Q1_value = quantile(value_aud, probs = 0.25, na.rm = TRUE),
                   Median_value = median(value_aud, na.rm = TRUE),
                   Q3_value = quantile(value_aud, probs = 0.75, na.rm = TRUE),
                   Max_value = max(value_aud, na.rm = TRUE),
                   Mean_value = mean(value_aud, na.rm = TRUE),
                   SD_value = sd(value_aud, na.rm = TRUE),
                    Total_value = sum(value_aud),
                   Missing_values = sum(is.na(value_aud)),
                   n_reports = n())

                    # n_reports = total number of observations.
```

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

```
head(value_stats)
```

```
## # A tibble: 6 x 12
## # Groups:   financial_year [1]
##   financial_year category   Min_value Q1_value Median_value Q3_value Max_value
##   <fct>          <fct>          <dbl>    <dbl>        <dbl>    <dbl>     <dbl>
## 1 2014-15        CITY BUSINE~     537.     537.         537.     537.      537.
## 2 2014-15        CITY DESIGN    10976.   10976.       10976.   10976.    10976.
## 3 2014-15        CITY GOVERN~      39.7     40.2         40.7     41.2      41.7
## 4 2014-15        CITY PLANNI~     929.   19036.       37144.   39867.    42590.
## 5 2014-15        COMMUNITY D~    4180.   11803.       19426.   27049.    34671.
## 6 2014-15        CORPORATE B~     730.     757.         802.    5817.     20758.
## # i 5 more variables: Mean_value <dbl>, SD_value <dbl>, Total_value <dbl>,
## #   Missing_values <int>, n_reports <int>
```

We now analyse yearly values of overall category, by group the data by financial year:

```
# Values grouped by financial year to compare costs in different financial years.

value_stats_year <- df %>%
 group_by(financial_year) %>%
 summarise(Min_value = min(value_aud, na.rm = TRUE),
                        Q1_value = quantile(value_aud, probs = 0.25, na.rm = TRUE),
                        Median_value = median(value_aud, na.rm = TRUE),
                        Q3_value = quantile(value_aud, probs = 0.75, na.rm = TRUE),
                        Max_value = max(value_aud, na.rm = TRUE),
                        Mean_value = mean(value_aud, na.rm = TRUE),
                        SD_value = sd(value_aud, na.rm = TRUE),
                         Total_value = sum(value_aud),
                        Missing_values = sum(is.na(value_aud)),
                        n_reports = n())

head(value_stats_year)
```

```
## # A tibble: 4 x 11
##   financial_year Min_value Q1_value Median_value Q3_value Max_value Mean_value
##   <fct>              <dbl>    <dbl>        <dbl>    <dbl>     <dbl>      <dbl>
## 1 2014-15             39.7     730.         929.   20758.    42590.     11862.
## 2 2015-16             18.6    3359.       10051.   22669.    40756.     15796.
## 3 2016-17             17.8     801.        9572    18124.    39001.     10959.
## 4 2017-18             16.9     847.        3527.    5975.    38043.      5539.
## # i 4 more variables: SD_value <dbl>, Total_value <dbl>, Missing_values <int>,
## #   n_reports <int>
```

Next we will try to understand how the money has been distributed by category:

```
# Comparing costs from different categories in total over the four financial years.

value_stats_cat <- df %>%
group_by(category) %>%
 summarise(Min_value = min(value_aud, na.rm = TRUE),
                        Q1_value = quantile(value_aud, probs = 0.25, na.rm = TRUE),
                        Median_value = median(value_aud, na.rm = TRUE),
                        Q3_value = quantile(value_aud, probs = 0.75, na.rm = TRUE),
                        Max_value = max(value_aud, na.rm = TRUE),
                        Mean_value = mean(value_aud, na.rm = TRUE),
                        SD_value = sd(value_aud, na.rm = TRUE),
                         Total_value = sum(value_aud),
                        Missing_values = sum(is.na(value_aud)),
                         n_reports = n()) # Total number of observations.

head(value_stats_cat)
```

```
## # A tibble: 6 x 11
##   category                Min_value Q1_value Median_value Q3_value Max_value Mean_value
##   <fct>                       <dbl>    <dbl>        <dbl>    <dbl>     <dbl>      <dbl>
## 1 CASUALS AND OTH~            3359.    3485.        3607.    3728.     3852.      3606.
## 2 CITY BUSINESS                490.     531.        8911.   18328.    19792.      9527.
## 3 CITY DESIGN                   16.9     18.4       4796.   10164.    10976.      5147.
## 4 CITY GOVERNANCE~              39.7     43.1       3010.    6345.     6851.      3228.
## 5 CITY PLANNING A~             847.     919.       19057.   39440.    42590.     20391.
## 6 COMMUNITY DEVEL~            3810.    4135.       19520.   36814.    39755.     20654.
## # i 4 more variables: SD_value <dbl>, Total_value <dbl>, Missing_values <int>,
## #   n_reports <int>
```

We can come to the conclusion that, depending on our business task goal, grouping the data according to different variables can offer different perspectives on the data.

Create a list

Creating a list with values number that each will represent each level of the category variable. Levels from category factor associated and labels created as 1 to 7.

```
# Created a list that contains a numeric value for each response to the category variable.


str(df$category)   # 7 levels.
```

```
## Factor w/ 7 levels "CASUALS AND OTHER",..: 5 7 2 4 7 6 6 4 6 3 ...
```

```
category_list = factor(df$category, labels = c(1:7)) str(category_list)
```

```
## Factor w/ 7 levels "1","2","3","4",..: 5 7 2 4 7 6 6 4 6 3 ...
```

On the previous code chunk, a vector has been created with the same levels of `category`, and assigned labels from 1 to 7 represented by the default levels order that R attributives in alphabetical order.

Join the list

category_list added as a new column to df as “category_n”.

```
df$category_n <- category_list        # New column added to df.
```

Subsetting (10 observations)

On this chunk, we subset the ten first rows of the data frame with all columns and then we convert it to a matrix.

```
# Subset the data and convert it to a matrix.

df_subset<- df[1:10, ]        # Subset-ting data frame using square brackets.

df_matrix <- matrix(df_subset) # Data frame converted to matrix.

 is.matrix(df_matrix)         # Checking if conversion was completed.
```

```
## [1] TRUE
```

```
str(df_matrix)                              # Structure of the matrix.
```

```
## List of 7
## $ : Date[1:10], format: "2014-07-01" "2014-07-01" ...
## $ : Factor w/ 4 levels "2014-15","2015-16",..: 1 2 3 4 1 2 3 4 1 2 ## $ : Factor w/ 2 levels
"BUDGET","PLAN": 1 1 1 1 2 2 2 2 2 2
## $ : Factor w/ 7 levels "CASUALS AND OTHER",..: 5 7 2 4 7 6 6 4 6 3 ## $ : Factor w/ 3 levels
"Permanent full time",..: 1 1 1 1 2 1 2 1 2 2 ## $ : num [1:10] 37144 19770 17261 5975 766 ...
## $ : Factor w/ 7 levels "1","2","3","4",..: 5 7 2 4 7 6 6 4 6 3
## - attr(*, "dim")= int [1:2] 7 1
```

Function matrix() assigned numeric values to all of the factor type of data. Changed the table to wide containing now 7 lists with 10 observations each. This is because the length and the size of the matrix needs to be equal.

Subsetting (first and last variable)

Extracting first and last variables of the df and then save data as RData format:

```
# Subset your data and saving it to an R object file

df_subset2 <- df[ ,c(1, 7)]   # Sub-setting all rows of the first and last column.

head(df_subset2)# Checking result.
```

```
##          date_calc category_n
```

```
## 1 2014-07-01          5
## 2 2014-07-01          7
## 3 2014-07-01          2
## 4 2014-07-01          4
## 5 2014-07-01          7
## 6 2014-07-01          6
```

```
save(df_subset2, file = "DfSubset2.RData") # Saving file in RData format, in working directory.
```

Reference

[1] City of Melbourne (2015) *Staff salary and wages budget 2014-15*, melbourne.vic.gov.au website, accessed 14 July 2024. https://data.gov.au/dataset/ds-melbourne-staff-salary-and-wages-budget-2014-15/details?q=salaries

[2] Wickham et al., (2019). *Welcome to the tidyverse.* Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686

[3] R Core Team (2019). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.