Data Assignment4

RuiCao_300423011

9/19/2019

a.Find the number of vehicles in registered in TLA Wellington City that are used as private passenger vehicles. [3 marks]

```
library(ggthemes)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tibble)
library(ggplot2)
library(tidyr)

mo <- read.csv("motor_vehicle_reduced.csv",stringsAsFactors = FALSE)
f <- filter(mo, TLA %in% "WELLINGTON CITY" & VEHICLE_USAGE %in% "PRIVATE PASSENGER")
NROW(f)
```

```
## [1] 158
```

b.Find out how many white or silver vehicles have been imported from Japan and registered in New Zealand in 2000 or later. [3 marks]

```
# I am not sure the original contry's meaning, if the mean is like registered country, then the
 code should be like above. if not it should be like second code.
ws <- filter(mo,(BASIC_COLOUR %in%  "WHITE" | BASIC_COLOUR %in% "SILVER") & ORIGINAL_COUNTRY %in
% "JAPAN" & ORIGINAL_COUNTRY %in% "NEW ZEALAND" & FIRST_NZ_REGISTRATION_YEAR >= 2000 )
nrow(ws)
```

```
## [1] 0
```

```
# second code of the same question.
ws2 <- filter(mo,(BASIC_COLOUR %in%  "WHITE" | BASIC_COLOUR %in% "SILVER") & ORIGINAL_COUNTRY %i
n% "JAPAN" & FIRST_NZ_REGISTRATION_YEAR >= 2000 )
nrow(ws2)
```

```
## [1] 964
```

c.Produce a table summarising all the different Volkswagen models in the dataset with non-zero gross vehicle mass. Give the model name, mean gross vehicle mass and the earliest and latest vehicle year for each model. [5 marks]

```
vo1 <- filter(mo, MAKE %in% "VOLKSWAGEN" & GROSS_VEHICLE_MASS >0)
new_ta <- select(vo1, MODEL, GROSS_VEHICLE_MASS,  VEHICLE_YEAR)
g <- group_by(new_ta,MODEL)
summarise(g, MEAN_MASS = mean(GROSS_VEHICLE_MASS), EARLIEST = min(VEHICLE_YEAR), LATEST = max(VE
HICLE_YEAR))
```

```
## # A tibble: 12 x 4
##    MODEL   MEAN_MASS EARLIEST LATEST
##    <chr>       <dbl>    <int>  <int>
##  1 AMAROK      3040      2011   2012
##  2 BEETLE      1560      2000   2002
##  3 BORA        1565      2001   2001
##  4 CADDY       2365      2011   2011
##  5 EOS         2020      2010   2010
##  6 GOLF        1710.     1997   2012
##  7 LUPO        1275      2005   2005
##  8 PASSAT      1973.     2003   2012
##  9 POLO        1417.     2000   2011
## 10 T5          2800      2011   2012
## 11 TIGUAN      2250      2010   2012
## 12 TOUAREG     2688.     2003   2012
```

d.Produce a contingency table giving the number of vehicles for every combination of make and import status. Restrict the table to the 10 makes with the most new vehicles, and show all import statuses for those makes. [3 marks]

```
tenVel <- select(mo,MAKE , IMPORT_STATUS, VEHICLE_YEAR)
gby <- group_by(tenVel, MAKE, IMPORT_STATUS)
con_table <- as_tibble(summarise(gby ,  TOTAL = n(), YEAR = max(VEHICLE_YEAR)))
arra <- arrange(con_table, desc(YEAR))
head(arra,10)
```

```
## # A tibble: 10 x 4
##    MAKE      IMPORT_STATUS TOTAL  YEAR
##    <chr>     <chr>         <int> <int>
##  1 BRIFORD   NEW              29  2013
##  2 FORD      NEW             414  2013
##  3 HOLDEN    NEW             280  2013
##  4 HOMEBUILT NEW              22  2013
##  5 HONDA     NEW             156  2013
##  6 HYUNDAI   NEW             100  2013
##  7 MAZDA     NEW             216  2013
##  8 NISSAN    NEW             136  2013
##  9 PINTO     NEW               7  2013
## 10 TRAILER   NEW             256  2013
```

Q2.

a.Reduce the dataset to only items measured in kg, and check this by displaying a list of the first 10 unique item names in the reduced dataset. [2 marks]

```
food <- read.csv("food_prices_yearmonth.csv", stringsAsFactors = FALSE)
fil_kg <- filter(food, grepl(".kg", Item))
se_kg <- select(fil_kg, Item)
dis_food <- distinct(se_kg)
head(dis_food,10)
```

```
##              Item
## 1    Oranges, 1kg
## 2    Bananas, 1kg
## 3     Apples, 1kg
## 4  Kiwifruit, 1kg
## 5    Lettuce, 1kg
## 6   Broccoli, 1kg
## 7    Cabbage, 1kg
## 8   Tomatoes, 1kg
## 9    Carrots, 1kg
## 10 Mushrooms, 1kg
```

b.Make a new data frame/tibble containing only the January values. [1 mark]

```
# all values in January
only_Jan <- filter(food, Month %in% "January")
table_Jan <- as_tibble(only_Jan)
head(table_Jan,5)
```

```
## # A tibble: 5 x 7
##   Item.ID      Data_value Units   Item          Year Month_num Month
##   <chr>             <dbl> <chr>   <chr>        <int>     <int> <chr>
## 1 CPIM.SAP0100       3.18 Dollars Oranges, 1kg  2007         1 January
## 2 CPIM.SAP0100       3.16 Dollars Oranges, 1kg  2008         1 January
## 3 CPIM.SAP0100       4.48 Dollars Oranges, 1kg  2009         1 January
## 4 CPIM.SAP0100       3.47 Dollars Oranges, 1kg  2010         1 January
## 5 CPIM.SAP0100       3.72 Dollars Oranges, 1kg  2011         1 January
```

```
# just kg values in January
Jan_kg <- filter(fil_kg, Month %in% "January")
table_Jan_kg <- as_tibble(Jan_kg)
tail(table_Jan_kg,5)
```

```
## # A tibble: 5 x 7
##   Item.ID     Data_value Units   Item                Year Month_num Month
##   <chr>            <dbl> <chr>   <chr>              <int>     <int> <chr>
## 1 CPIM.SAP02~       13.7 Dollars Ham, sliced or sha~  2015         1 Janua~
## 2 CPIM.SAP02~       12.7 Dollars Ham, sliced or sha~  2016         1 Janua~
## 3 CPIM.SAP02~       12.7 Dollars Ham, sliced or sha~  2017         1 Janua~
## 4 CPIM.SAP02~       12.5 Dollars Ham, sliced or sha~  2018         1 Janua~
## 5 CPIM.SAP02~       12.7 Dollars Ham, sliced or sha~  2019         1 Janua~
```

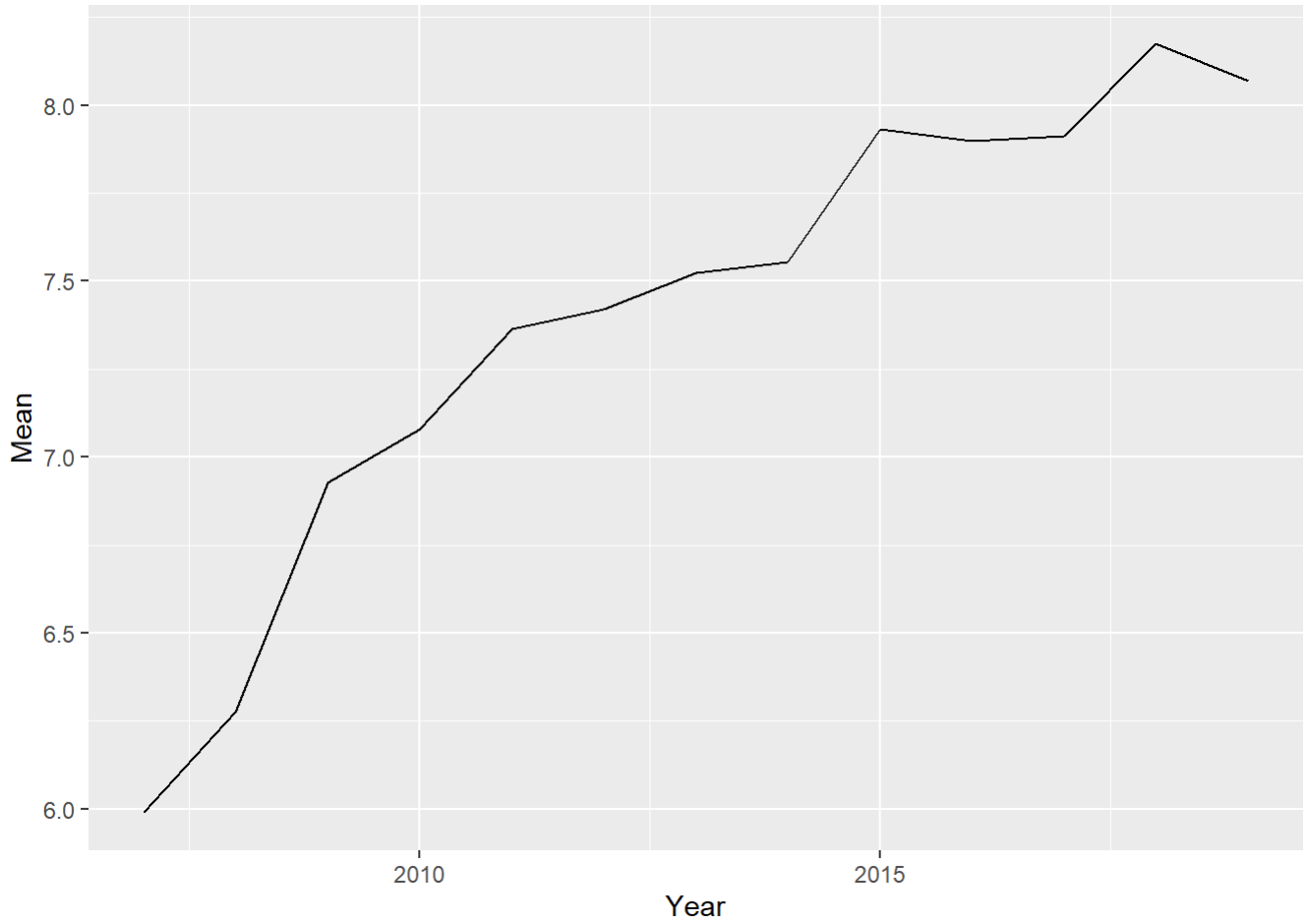
c.Create a table that gives the mean price of all the kg-valued items for each year in January, display the table, and keep that table as a new tibble/data frame. You don’t need to convert the prices into dollar prices, just display them as numbers. [2 marks]

```
group_Jan <- group_by(table_Jan_kg, Year)
mean_Jan <- as_tibble(summarise(group_Jan, Mean = mean(Data_value)))
head(mean_Jan,5)
```

```
## # A tibble: 5 x 2
##    Year  Mean
##   <int> <dbl>
## 1  2007  5.99
## 2  2008  6.28
## 3  2009  6.93
## 4  2010  7.08
## 5  2011  7.36
```

d.Use the table you just created to produce a time-series line plot of mean price by year, using the ggplot2 package. Make sure to label the plot axes correctly. [3 marks]

```
ggplot(mean_Jan,
       aes(x=Year,
           y=Mean,
           ))+
  geom_line()
```



e.Go

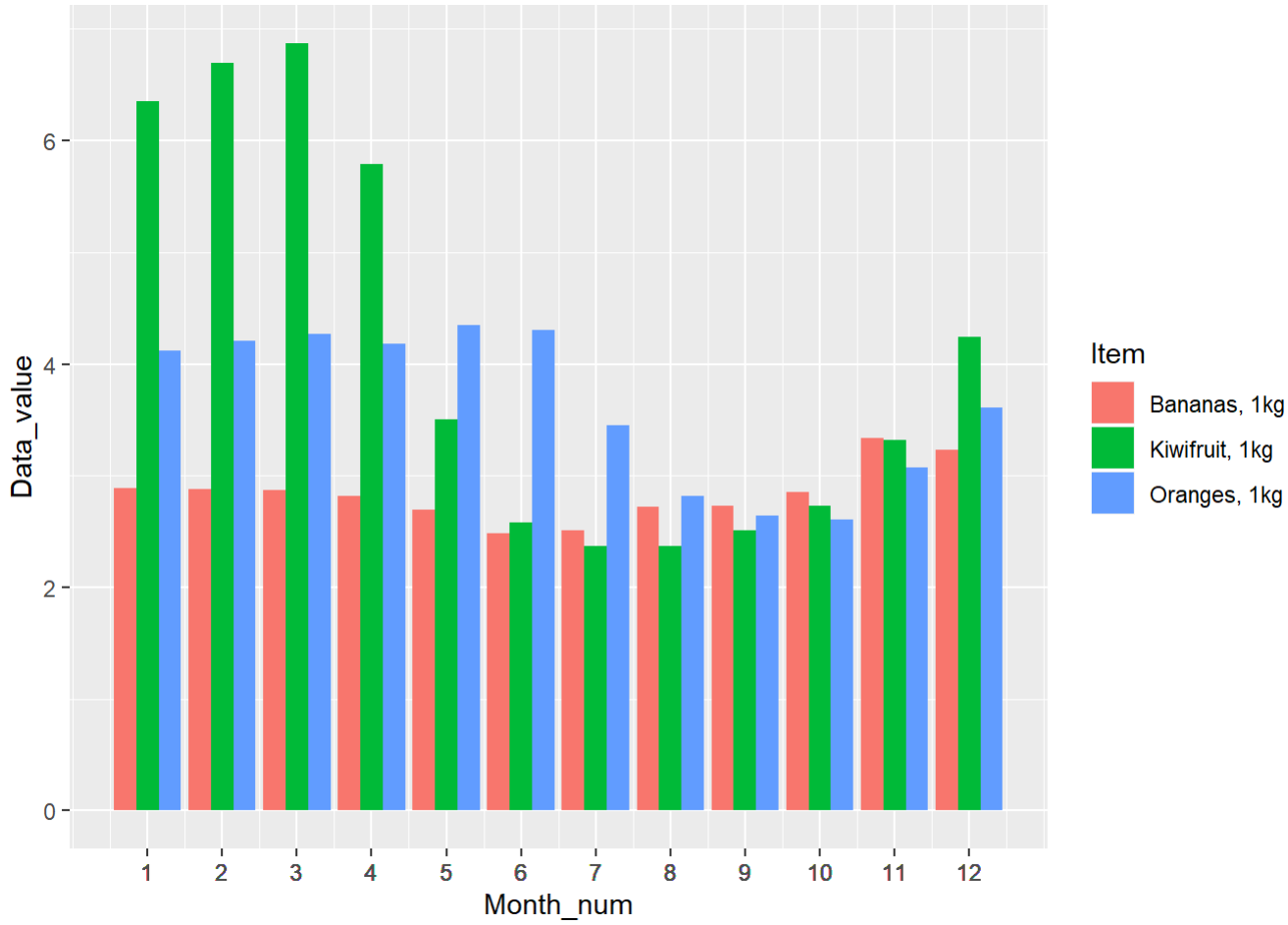
back to the full kg-weighted items dataset for all months. Reduce it to just the 2018 data. [1 mark]

```
kg_2018 <- filter(fil_kg, Year %in% 2018)
head(kg_2018,5)
```

```
##        Item.ID Data_value   Units         Item Year Month_num    Month
## 1 CPIM.SAP0100       4.12 Dollars Oranges, 1kg 2018         1  January
## 2 CPIM.SAP0100       4.21 Dollars Oranges, 1kg 2018         2 February
## 3 CPIM.SAP0100       4.27 Dollars Oranges, 1kg 2018         3    March
## 4 CPIM.SAP0100       4.18 Dollars Oranges, 1kg 2018         4    April
## 5 CPIM.SAP0100       4.35 Dollars Oranges, 1kg 2018         5      May
```

f.Select three of the kg-weighted items. Again using ggplot2, plot a bar chart showing prices for each month, and showing those three items side-by-side in different colours. Label the y-axis as dollar price. Make sure the months are in the correct order – you may need to set Month to a factor with the right order of levels. Also make sure the month labels are not displayed overlapping. [8 marks]

```
mon <- filter(kg_2018, Item %in% c("Oranges, 1kg", "Bananas, 1kg", "Kiwifruit, 1kg"), Data_valu
e)
ggplot(mon) +geom_bar(aes(x=Month_num, y=Data_value,fill = Item), stat="identity",position ="dod
ge") + scale_x_continuous(breaks = kg_2018$Month_num, labels = kg_2018$Month_num)
```


Data_value
0
2
4
6
1
2
3
4
5
6
7
8
9
10
11
12
Month_num
Item
Bananas, 1kg
Kiwifruit, 1kg
Oranges, 1kg

Q3.

```
library(tidyr)
library(dplyr)
library(tibble)
vehicles <- as_tibble(read.csv("motor_vehicle_reduced.csv"))
summarise_vehicles <- function(region, type, max_axles, earliest_year = min(vehicles$VEHICLE_YEA
R)) {
    if (!(region %in% unique(vehicles$TLA))) stop(paste(region, "is not in the list of TLAs (reg
ions) in the dataset."))
    if (!(type %in% unique(vehicles$VEHICLE_TYPE))) stop(paste(type, "is not in the list of vehi
cle types in the dataset."))

    vehicles_sub <- filter(vehicles, TLA==region &
                             VEHICLE_TYPE==type &
                               NUMBER_OF_AXLES <= max_axles &
                               VEHICLE_YEAR >= earliest_year
                           # does not have " NUMBER_OF_DOORS " attributes in the table.
                            #&NUMBER_OF_DOORS > 3
                           # we can change to number of seat > 3
                           & NUMBER_OF_SEATS >3
                              ) %>%
                    mutate(VEHICLE_DECADE = floor(VEHICLE_YEAR/10)*10)

    vehicles_sub <- filter(vehicles_sub, GROSS_VEHICLE_MASS < 0)
    # No data available in table
    vehicles_sub <- group_by(vehicles_sub, MAKE, VEHICLE_DECADE) %>%
                    arrange(VEHICLE_decade) %>%
                    select(BASIC_COLOUR, BODY_TYPE, MODEL, MAKE,
                           VEHICLE_DECADE, CC_RATING, GROSS_VEHICLE_MASS)

    result <- summarise(vehicles_sub, N=n(), Mean_CC_Rating=mean(CC_RATING))

    filter(result, N > 10)
}
summarise_vehicles("AUCKLAND","PASSENGER CAR/VAN",2,2000)
```

```
## Warning: Factor `MAKE` contains implicit NA, consider using
## `forcats::fct_explicit_na`

## Warning: Factor `MAKE` contains implicit NA, consider using
## `forcats::fct_explicit_na`
```

```
## # A tibble: 0 x 4
## # Groups:   MAKE [1]
## # ... with 4 variables: MAKE <fct>, VEHICLE_DECADE <dbl>, N <int>,
## #   Mean_CC_Rating <dbl>
```