Assignment 2

ETC1010_5510

Jason Ching Yuen Siu

Sunday, May 16 2021

```
library(naniar)
library(broom)
library(ggmap)
library(knitr)
library(lubridate)
library(timeDate)
library(tsibble)
library(here)
library(readr)
library(tidyverse)
library(ggResidpanel)
library(gridExtra)
```

```
##please set your own working directory if such issues are encountered
tree_data0 <- read_csv("Data/Assignment_data.csv")
```

Part I

Question 1: Rename the variables *Date Planted* and *Year Planted* to *Dateplanted* and *Yearplanted* using the *rename()* function. Make sure *Dateplanted* is defined as a date variable. Then extract from the variable *Dateplanted* the year and store it in a new variable called *Year*. Display the first 6 rows of the data frame. (5pts)

```
#using rename () to rename  *Date Planted* and *Year Planted* to *Dateplanted* and *Yearplanted*
tree_data <- tree_data0 %>%
  rename(Dateplanted = `Date Planted`,
         Yearplanted = `Year Planted` )
#*Dateplanted* is defined as a **date variable**.
tree_data$Dateplanted <- dmy(tree_data$Dateplanted)
# new variable as "year"
tree_data <- tree_data %>% mutate(Year = year( tree_data$Dateplanted))
#display the 1st 6 rows
head(tree_data, 6)
```

```
## # A tibble: 6 x 20
##   `CoM ID` `Common Name`   `Scientific Name`  Genus  Family   `Diameter Breast ~
##      <dbl> <chr>           <chr>              <chr>  <chr>                  <dbl>
## 1  1057605 White Poplar    Populus alba       Popul~ Salicac~                  NA
## 2  1028440 London Plane    Platanus x acerif~ Plata~ Platana~                  62
## 3  1058665 Small-leaved L~ Tilia cordata      Tilia  Malvace~                  19
## 4  1026352 Variegated Elm  Ulmus minor        Ulmus  Ulmaceae                  26
## 5  1038440 Canary Island ~ Pinus canariensis  Pinus  Pinaceae                  91
## 6  1015128 London Plane    Platanus x acerif~ Plata~ Platana~                  99
## # ... with 14 more variables: Yearplanted <dbl>, Dateplanted <date>,
## #   Age Description <chr>, Useful Life Expectency <chr>,
## #   Useful Life Expectency Value <dbl>, Precinct <lgl>, Located in <chr>,
## #   UploadDate <chr>, CoordinateLocation <chr>, Latitude <dbl>,
## #   Longitude <dbl>, Easting <dbl>, Northing <dbl>, Year <dbl>
```

Question 2: Have you noticed any differences between the variables *Year* and *Yearplanted*? Why is that? Demonstrate your claims using R code. Fix the problem if there is one (Hint: Use *ifelse* inside a mutate function to fix the problem and store the data in *tree_data_clean*). After this question, please use the data in *tree_data_clean* to proceed. (3pts)

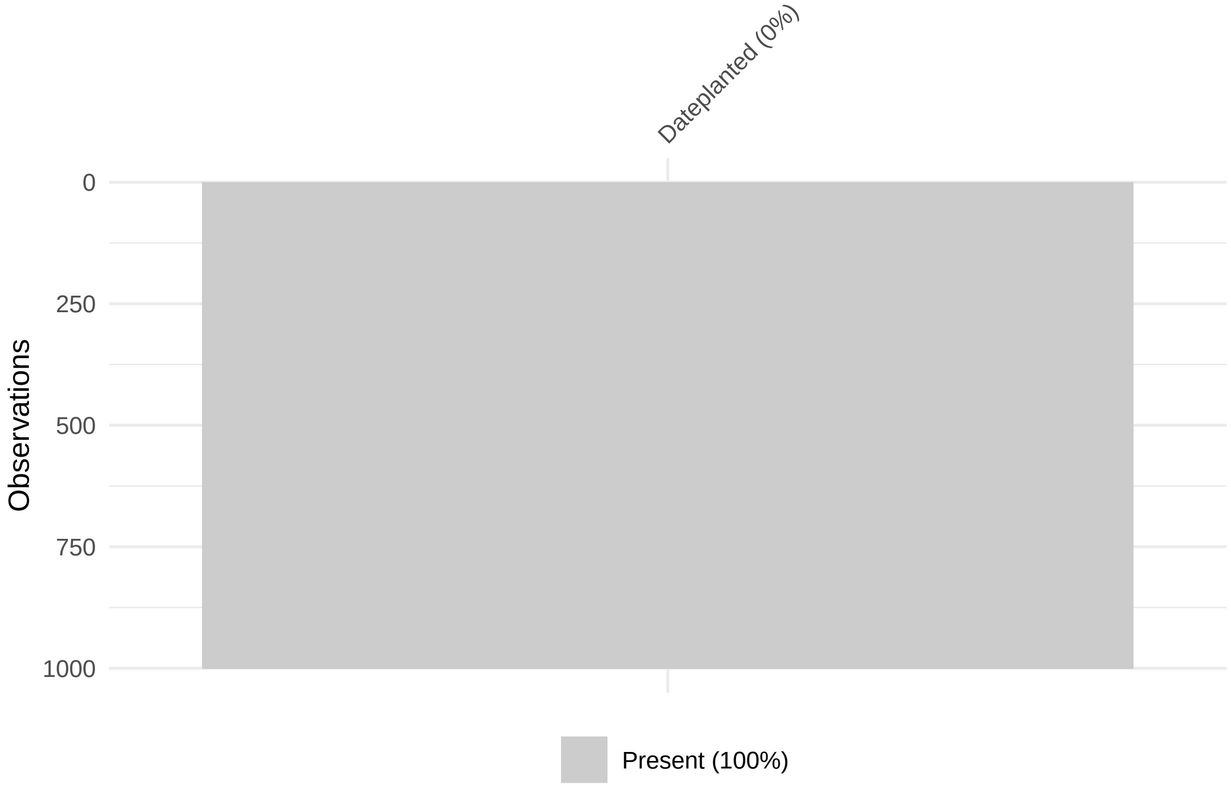
Why ? possibility 1 :Perhaps people mis-input the value 2000 as 1900, therefore, the value of 1900 does not reflect the date planted

Possibility 2 : When the programmer wrangle th data, they did not notice this syntax error

```
tree_data_clean <- tree_data %>%
  mutate(Yearplanted = ifelse(Yearplanted ==Year, Yearplanted, Year))
```

Question 3: Investigate graphically the missing values in the variable *Dateplanted* for the last 1000 rows of the data set. What do you observe? (max 30 words) (2pts)

```
tree_data_singlevariable <- tree_data_clean %>%
  select(Dateplanted) %>%
  tail(1000)
vis_miss(tree_data_singlevariable)
```



There are no missing values (NA) for last 1000 rows

Question 4: What is the proportion of missing values in each variable in the tree data set? Display the results in descending order of the proportion. (2pts)

```
miss_var_summary(tree_data_clean) %>%
  select(variable,pct_miss)%>%
  arrange(desc(pct_miss))
```

```
## # A tibble: 20 x 2
##    variable                     pct_miss
##    <chr>                           <dbl>
##  1 Precinct                     100
##  2 Diameter Breast Height        21.3
##  3 Age Description               21.3
##  4 Useful Life Expectency        21.3
##  5 Useful Life Expectency Value  21.3
##  6 Yearplanted                    0.0293
##  7 Dateplanted                    0.0293
##  8 Year                           0.0293
##  9 Common Name                    0.0146
## 10 Located in                     0.0146
## 11 CoM ID                         0
## 12 Scientific Name                0
## 13 Genus                          0
## 14 Family                         0
## 15 UploadDate                    0
## 16 CoordinateLocation            0
## 17 Latitude                      0
## 18 Longitude                     0
## 19 Easting                       0
## 20 Northing                      0
```

Question 5: How many observations have a missing value in the variable *Dateplanted*? Identify the rows and display the information in those rows. Remove all the rows in the data set of which the variable *Dateplanted* has a missing value recorded and store the data in *tree_data_clean1*. Display the first 4 rows of *tree_data_clean1*. Use R inline code to complete the sentense below. (6pts)

```
#How many observations have a missing value in the variable *Dateplanted*?
tree_data_clean %>% select(Dateplanted) %>% n_miss()
```

```
## [1] 2
```

```
# there are 2 missing values
tree_data_clean %>% subset(is.na(tree_data_clean$Dateplanted))
```

```
## # A tibble: 2 x 20
##   `CoM ID` `Common Name` `Scientific Name`   Genus  Family   `Diameter Breast H~
##      <dbl> <chr>         <chr>               <chr>  <chr>                  <dbl>
## 1  1024155 Cyprus Plane  Platanus orientalis Plata~ Platana~                  22
## 2  1023092 London Plane  Platanus x acerifo~ Plata~ Platana~                  29
## # ... with 14 more variables: Yearplanted <dbl>, Dateplanted <date>,
## #   Age Description <chr>, Useful Life Expectency <chr>,
## #   Useful Life Expectency Value <dbl>, Precinct <lgl>, Located in <chr>,
## #   UploadDate <chr>, CoordinateLocation <chr>, Latitude <dbl>,
## #   Longitude <dbl>, Easting <dbl>, Northing <dbl>, Year <dbl>
```

```
#so remove them
tree_data_clean1 <- tree_data_clean %>% drop_na(Dateplanted)
head(tree_data_clean1,4)
```

```
## # A tibble: 4 x 20
##   `CoM ID` `Common Name`   `Scientific Name`  Genus  Family   `Diameter Breast ~
##      <dbl> <chr>           <chr>              <chr>  <chr>                 <dbl>
## 1  1057605 White Poplar    Populus alba       Popul~ Salicac~                 NA
## 2  1028440 London Plane    Platanus x acerif~ Plata~ Platana~                 62
## 3  1058665 Small-leaved L~ Tilia cordata      Tilia  Malvace~                 19
## 4  1026352 Variegated Elm  Ulmus minor        Ulmus  Ulmaceae                 26
## # ... with 14 more variables: Yearplanted <dbl>, Dateplanted <date>,
## #   Age Description <chr>, Useful Life Expectency <chr>,
## #   Useful Life Expectency Value <dbl>, Precinct <lgl>, Located in <chr>,
## #   UploadDate <chr>, CoordinateLocation <chr>, Latitude <dbl>,
## #   Longitude <dbl>, Easting <dbl>, Northing <dbl>, Year <dbl>
```

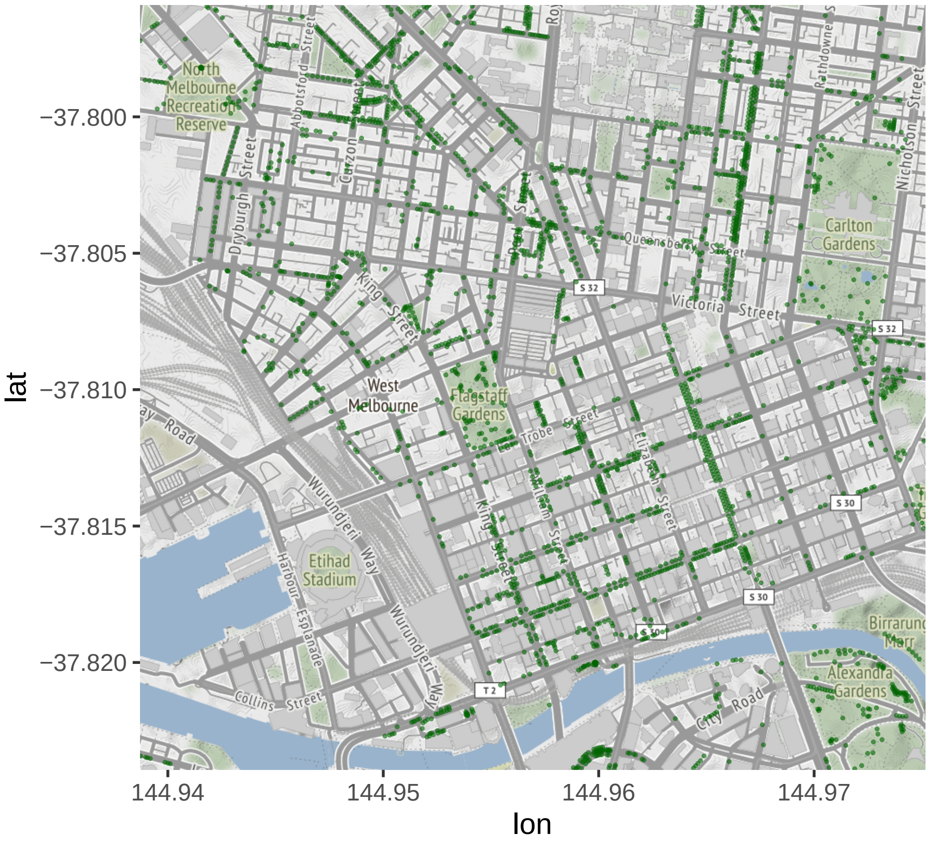
The number of rows in the cleaned data set are 6826 and the number of columns are 20

Question 6: Create a map with the tree locations in the data set. (2pts)

```
# We have created the map below for you
melb_map <- read_rds(here::here("Data/melb-map.rds"))

# Here you just need to add the location for each tree into the map.
ggmap(melb_map) +
  geom_point(data = tree_data_clean1,
             aes(x =Longitude ,
                 y = Latitude),
             colour = "#006400",
             alpha = 0.6,
             size = 0.2)
```

```
## Warning: Removed 4378 rows containing missing values (geom_point).
```



Question 7: Create another map and draw trees in the *Genus* groups of Eucalyptus, Macadamia, Prunus, Acacia, and Quercus. Use the "Dark2" color palette and display the legend at the bottom of the plot. (8pts)

```
#Create another map and draw trees in the *Genus* groups of Eucalyptus, Macadamia, Prunus, Acacia, and
GenusSelected <- c("Eucalyptus","Macadamia","Prunus","Acacia","Quercus")
```

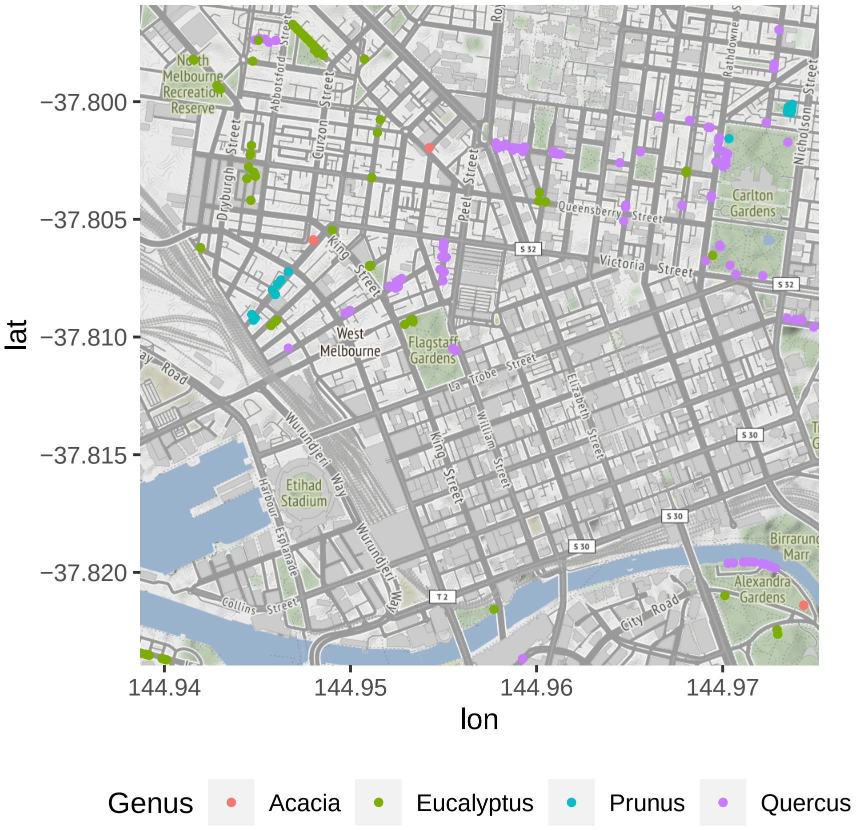
```
selected_group <- tree_data_clean1 %>% filter(Genus %in% GenusSelected)

melb_map <- read_rds(here::here("Data/melb-map.rds"))

# Here you just need to add the location for each tree into the map.
ggmap(melb_map) +
  geom_point(data = selected_group,
             aes(x = Longitude,
                 y = Latitude,
                 colour = Genus),
             palette = "Dark2",
             alpha = 1,
             size = 1)+
  theme(legend.position = "bottom")
```

```
## Warning: Ignoring unknown parameters: palette

## Warning: Removed 911 rows containing missing values (geom_point).
```



Question 8: Filter the data *tree_data_clean1* so that only the variables *Year*, *Located in*, and *Common Name* are displayed. Arrange the data set by *Year* in descending order and display the first 4 lines. Call this new data set *tree_data_clean_filter*. Then answer the following question using inline R code: When (*Year*), where (*Located in*) and what tree (*Common Name*) was the first tree planted in Melbourne according to this data set? (8pts)

```
#Filter the data *tree_data_clean1* so that only the variables *Year*, *Located in*, and *Common Name*
tree_data_clean_filter <- tree_data_clean1 %>%
  select(Year, `Located in`,`Common Name`) %>%
  arrange(desc(Year))
#display the first 4 lines
head(tree_data_clean_filter,4)
```

```
## # A tibble: 4 x 3
##    Year `Located in` `Common Name`
##   <dbl> <chr>        <chr>
## 1  2000 Park         White Poplar
## 2  2000 Park         London Plane
## 3  2000 Street       Small-leaved Linden
## 4  2000 Street       Variegated Elm
```

The first tree was planted in 2000 at a Park and the tree name is White Poplar

Question 9: How many trees were planted in parks and how many in streets? Tabulate the results (only for locations in parks and streets) using the function *kable()* from the *kableExtra* R package. (3pts)

```
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
kable(
  tree_data_clean1 %>%
    group_by(`Located in`) %>%
    filter(`Located in` !="NA") %>%
    summarise(Count=n())
  )
```

Located in	Count
Park	2737
Street	4088

Question 10: How many trees are there in each of the Family groups in the data set *tree_data_clean1* (display the first 5 lines of the results in descending order)? (2pt)

```
head(
  tree_data_clean1 %>%
    group_by(Family) %>%
    summarise(Count = n()) %>%
    arrange(desc(Count))
      ,5)
```

```
## # A tibble: 5 x 2
##   Family      Count
##   <chr>       <int>
## 1 Myrtaceae    2102
## 2 Platanaceae  1512
## 3 Ulmaceae     1125
## 4 Fabaceae      327
## 5 Fagaceae      254
```

Question 11: Create a markdown table displaying the number of trees planted in each year (use variable *Yearplanted*) with common names Ironbark, Olive, Plum, Oak, and Elm (Hint: Use kable() from the gridExtra R package). What is the oldest most abundant tree in this group? (8pts)

```
#displaying the number of trees planted in each year
#with common names Ironbark, Olive, Plum, Oak, and Elm
cNameSelected <- c("Ironbark","Olive","Plum","Oak","Elm")

tree_year<-tree_data_clean1 %>%
   filter(`Common Name` %in% cNameSelected)%>%
   select(Yearplanted,`Common Name`) %>%
   group_by(Yearplanted,`Common Name`) %>%
   count(Yearplanted,`Common Name`)

kable(tree_year)
```

Yearplanted	Common Name	n
2000	Elm	197
2000	Ironbark	52
2000	Oak	13
2000	Olive	17

```
tree_year %>% arrange(desc(n))%>%  select(`Common Name`) %>% head(1)
```

```
## Adding missing grouping variables: `Yearplanted`
```

```
## # A tibble: 1 x 2
## # Groups:   Yearplanted, Common Name [1]
##   Yearplanted `Common Name`
##         <dbl> <chr>
## 1        2000 Elm
```

The oldest most abundant tree in this group is Elm

Question 12: Select the trees with diameters (Diameter Breast Height) greater than 40 cm and smaller 100 cm and comment on where the trees are located (streets or parks). (max 25 words) (3pts)

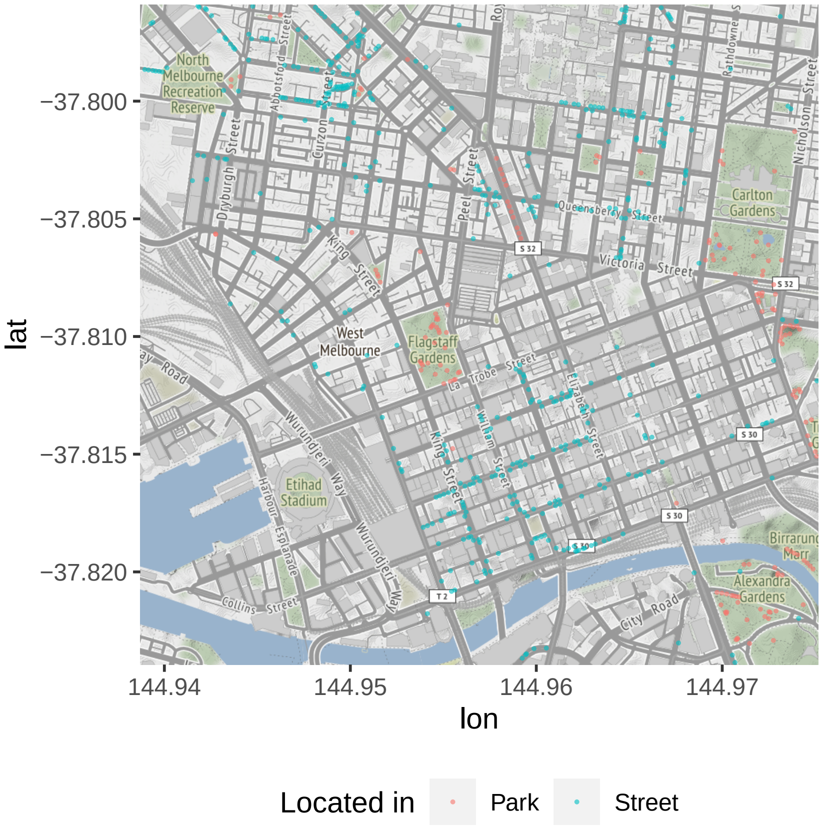
```
large_trees_data <- tree_data_clean1 %>%
  filter(`Diameter Breast Height`>40 & `Diameter Breast Height`<100) %>% group_by(`Located in`) %>% summ
```

There are 795 located on Park, 800 located on Street

Question 13: Plot the trees within the diameter range that you have selected in Question 12, which are located in parks and streets on a map using 2 different colours to differentiate their locations (streets or parks). (6pts)

```
large_trees_data_parks <- tree_data_clean1 %>%
  filter(`Diameter Breast Height`>40 & `Diameter Breast Height`<100)
```

```
ggmap(melb_map) +
             geom_point(data = large_trees_data_parks,
             aes(x = Longitude ,
                 y = Latitude,
                 color =`Located in` ),
             alpha = 0.6,
           size = 0.2,
           palette="Dark2")+
  theme(legend.position = "bottom")
```

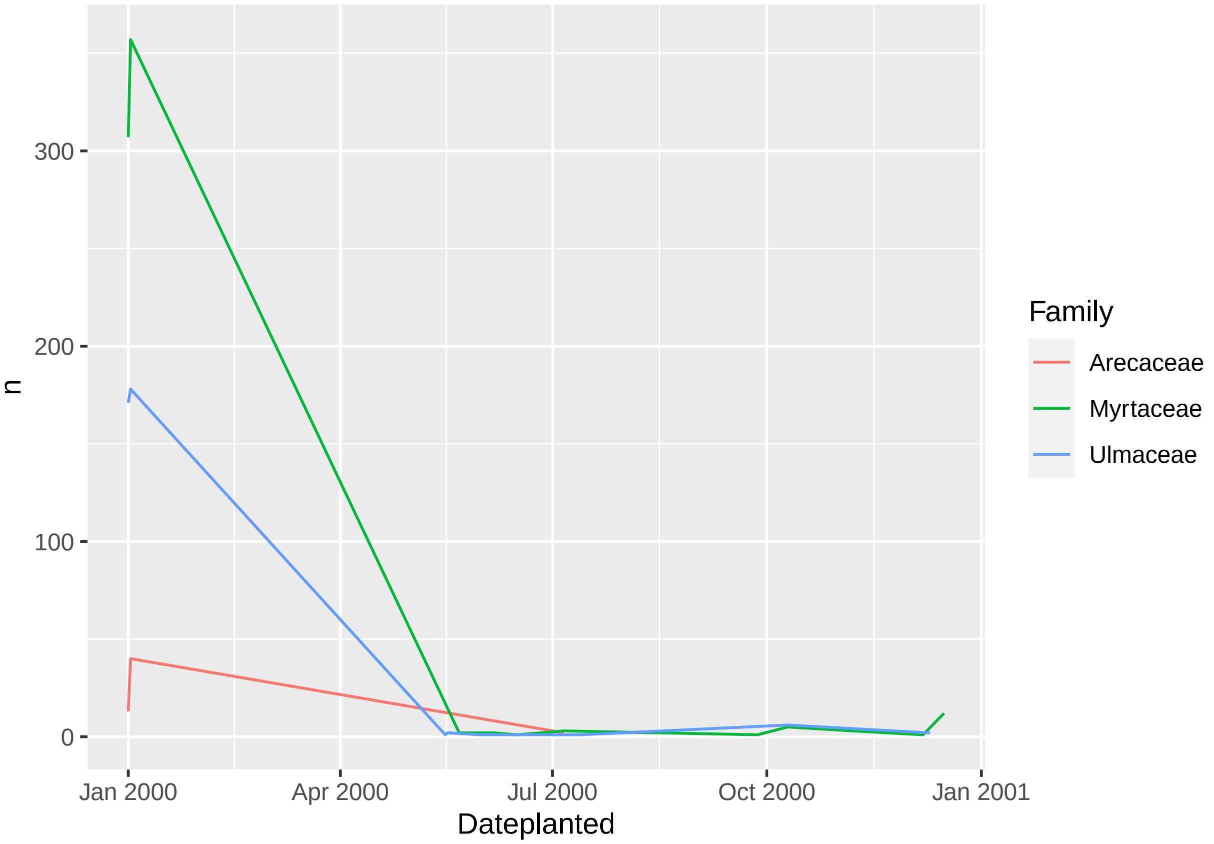


Question 14: Create a time series plot (using geom_line) that displays the total number of trees planted per year in the data set *tree_data_clean1* that belong to the Families: Myrtaceae, Arecaceae, and Ulmaceae. What do you observe from the plot? (6pts)

```
Fig_data <- tree_data_clean1 %>% select(Dateplanted,Family)%>% filter(Family == c('Myrtaceae', 'Arecacea
```

```
## Warning in Family == c("Myrtaceae", "Arecaceae", "Ulmaceae"): longer object
## length is not a multiple of shorter object length
```

```
ggplot(Fig_data,aes(x = Dateplanted,y = n,color = Family))+geom_line()
```



Part 2: Simulation Exercise

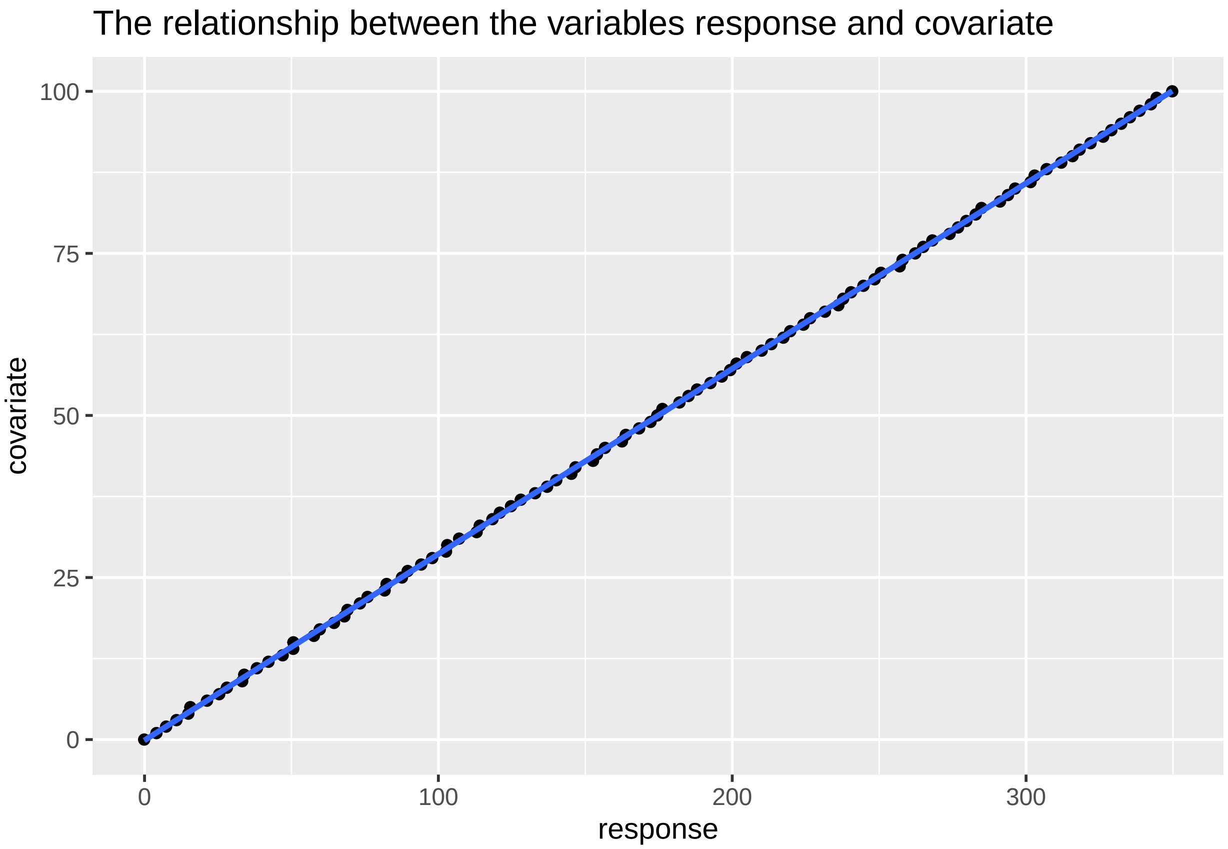
== # Question 15: Create a data frame called *simulation_data* that contains 2 variables with names *response* and *covariate*. Generate the variables according to the following model: $response = 3.5 \times covariate + epsilon$ where *covariate* is a variable that takes values $0, 1, 2, \ldots, 100$ and ϵ is generated according to a Normal distribution (Hint: Use the function *rnorm()* to generate *epsilon*.) (3pts)

```
set.seed(2021)
covariate <- c(0:100)
epsilon <- rnorm(101)
response <- 3.5 * covariate + epsilon
simulation_data <- tibble(response,covariate)
```

Question 16: Display graphically the relationship between the variables *response* and *covariate* (1pt) using a point plot. Which kind of relationship do you observe? (2pts)

```
ggplot(simulation_data,aes(response,covariate))+
  geom_point()+
  ggtitle("The relationship between the variables response and covariate")+
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```


The relationship between the variables response and covariate

```
#there is a positive relationship
```

Question 17: Fit a linear model between the variables *response* and *covariate* that you generate in Question 15 and display the model summary. (2pts)

```
fit <- lm(response~covariate, data = simulation_data)
summary(fit)
```

```
##
## Call:
## lm(formula = response ~ covariate, data = simulation_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.07431 -0.71466  0.05844  0.64196  2.25176
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.135896   0.199948    0.68    0.498
## covariate   3.493775   0.003455 1011.35   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.012 on 99 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 1.023e+06 on 1 and 99 DF,  p-value: < 2.2e-16
```

Question 18: What are the values for the intercept and the slope in the estimated model in Question 17 (Hint: Use the function *coef()*)? How do these values compare with the values in the simulation model? (max 50 words) (2pts)

```
coef(fit)[2] #slope
```

```
## covariate
##  3.493775
```

```
coef(fit)[1] #intercept
```

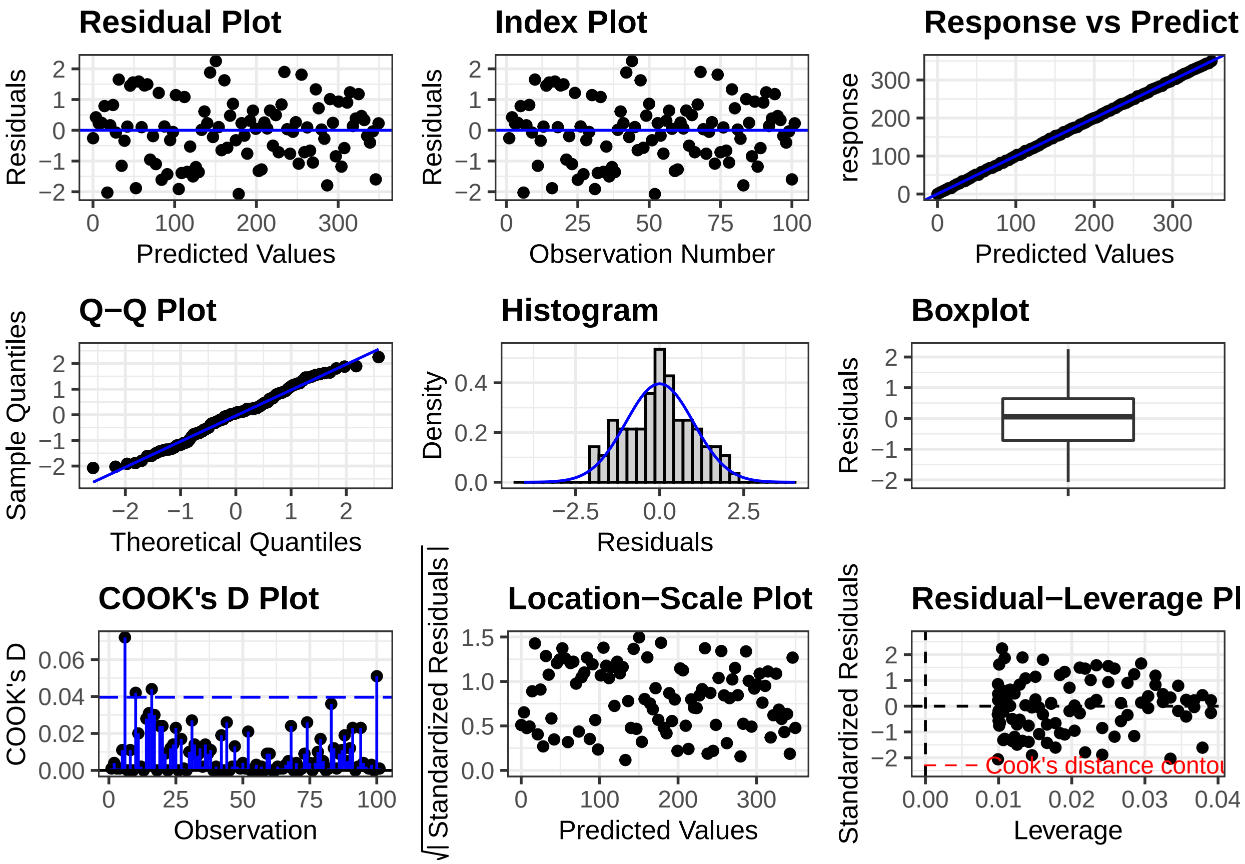
```
## (Intercept)
##   0.1358957
```

The values of the intercept and slope are `coef(fit)[1]` and `coef(fit)[2]`.

The formula y = 3.4937754x + 0.1358957 gives us fitted values that are closest to the actual values in the sense that the length of the residual vector is the smallest possible.

Question 19: Create a figure to display the diagnostic plots of the linear model that you fit in Question 17. Comment on the diagnostic plots (max 50 words). Is this a good/bad model and why? (max 30 words) (4pts)

```
library(ggResidpanel)
resid_panel(fit, plots = "all")
```



It is a good model given that : 1. The points in Residual plots are of no patterns 2. The points in QQ plot lies on the blue line, meaning normal distribution 3. There are no points lying below the cook's distance contour line 4. All the predicted values are same as the actual values

Question 20: Report R2, Radjusted, AIC, and BIC. Is this a good/bad model? Please explain your answer. (max 30 words) (2pts)

```
glance(fit) %>% select()
```

```
## # A tibble: 1 x 0
```

It is a generally a good linear model given that : 1. perfect score for R2 (99%)

However, since there is no other models to compare, using AIC and BIC, We do not know if this model is good enough.