Assignment 2

ETC1010-5510

Patricia Menéndez

Tuesday, May 18 2021

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

```r
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)

```r
tree_data <- tree_data0 %>%
  rename(Dateplanted = `Date Planted`,
        Yearplanted = `Year Planted`) %>%
  mutate(Dateplanted = dmy(Dateplanted),
         Year = year(Dateplanted))

head(tree_data)
```

```
## # A tibble: 6 x 20
##   `CoM ID` `Common Name` `Scientific Nam~ Genus Family `Diameter Breas~
##      <dbl> <chr>         <chr>            <chr> <chr>              <dbl>
## 1  1057605 White Poplar  Populus alba     Popu~ Salic~               NA
## 2  1028440 London Plane  Platanus x acer~ Plat~ Plata~               62
## 3  1058665 Small-leaved~ Tilia cordata    Tilia Malva~               19
## 4  1026352 Variegated E~ Ulmus minor      Ulmus Ulmac~               26
## 5  1038440 Canary Islan~ Pinus canariens~ Pinus Pinac~               91
## 6  1015128 London Plane  Platanus x acer~ Plat~ Plata~               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>
```

```r
#write_csv(tree_data, "Data/Assignment_data.csv")
```

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)

Yes, the encoding for 1900 has been converted to 2000 instead.

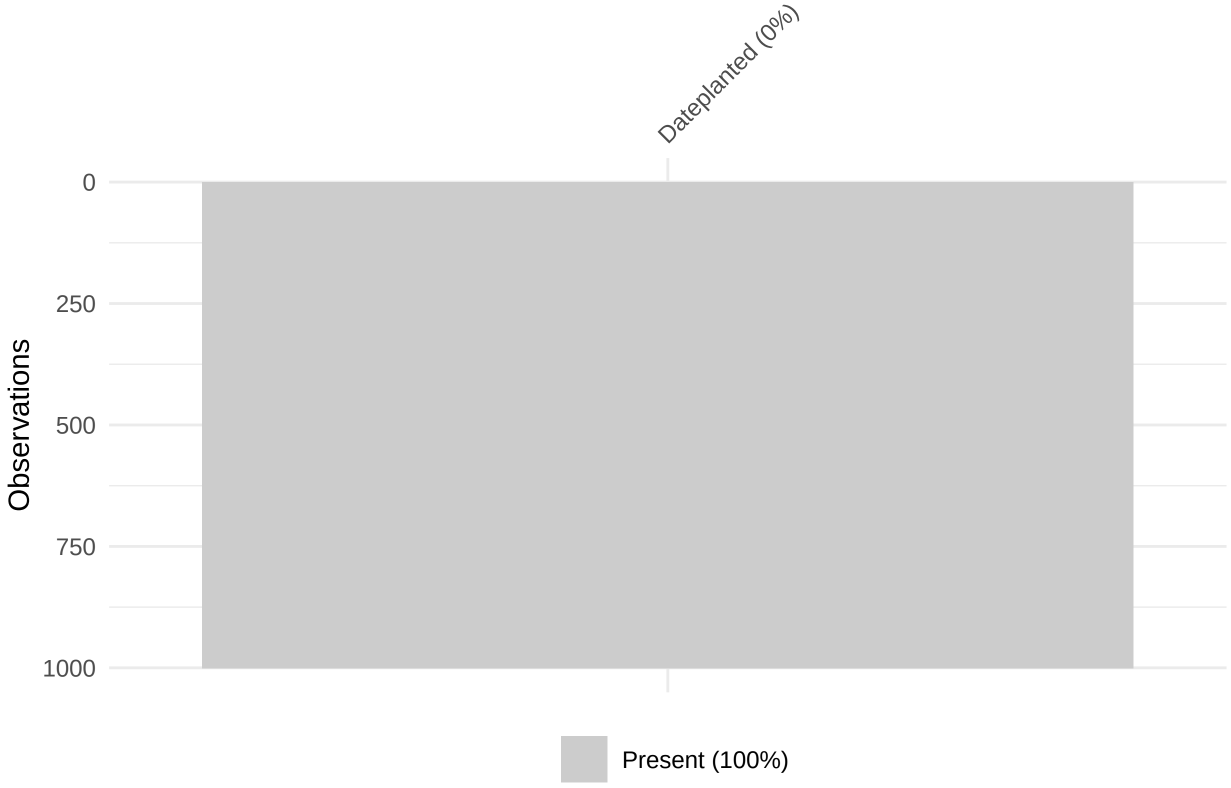
```r
length(which(tree_data$Year!=tree_data$Yearplanted))
```

```
## [1] 5321
```

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

```r
tree_data_singlevariable <- tree_data_clean %>%
  dplyr::select(Dateplanted)

vis_miss(tail(tree_data_singlevariable, n = 1000) , warn_large_data = FALSE)
```



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) %>%
  arrange(-pct_miss)
```

```
## # A tibble: 20 x 3
##    variable                     n_miss pct_miss
##    <chr>                         <int>    <dbl>
##  1 Precinct                       6828 100
##  2 Diameter Breast Height         1454  21.3
##  3 Age Description                1454  21.3
##  4 Useful Life Expectency         1454  21.3
##  5 Useful Life Expectency Value   1454  21.3
##  6 Dateplanted                       2   0.0293
##  7 Year                              2   0.0293
##  8 Common Name                       1   0.0146
##  9 Located in                        1   0.0146
## 10 CoM ID                            0   0
## 11 Scientific Name                   0   0
## 12 Genus                             0   0
## 13 Family                            0   0
## 14 Yearplanted                       0   0
## 15 UploadDate                        0   0
## 16 CoordinateLocation                0   0
## 17 Latitude                          0    0
## 18 Longitude                         0    0
## 19 Easting                           0    0
## 20 Northing                          0    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)

Two missing values in the following rows:

```
 tree_data_clean %>%
  dplyr::filter(is.na(Dateplanted))
```

```
## # A tibble: 2 x 20
##   `CoM ID` `Common Name` `Scientific Nam~ Genus Family `Diameter Breas~
##      <dbl> <chr>         <chr>            <chr> <chr>             <dbl>
## 1  1024155 Cyprus Plane  Platanus orient~ Plat~ Plata~               22
## 2  1023092 London Plane  Platanus x acer~ Plat~ Plata~               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>
```

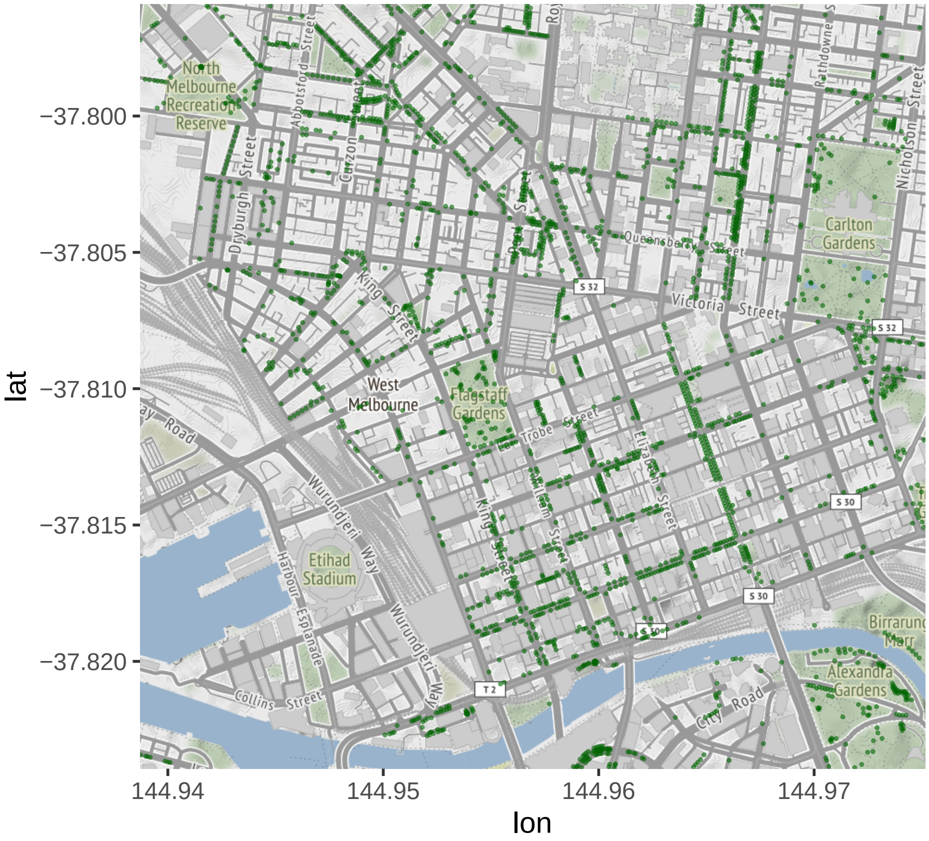
```
tree_data_clean1 <- tree_data_clean %>%
  dplyr::filter(!is.na(Dateplanted))

head(tree_data_clean1, 4)
```

```
## # A tibble: 4 x 20
##   `CoM ID` `Common Name` `Scientific Nam~ Genus Family `Diameter Breas~
##      <dbl> <chr>         <chr>            <chr> <chr>             <dbl>
## 1  1057605 White Poplar  Populus alba     Popu~ Salic~               NA
## 2  1028440 London Plane  Platanus x acer~ Plat~ Plata~               62
## 3  1058665 Small-leaved~ Tilia cordata    Tilia Malva~               19
## 4  1026352 Variegated E~ Ulmus minor      Ulmus Ulmac~               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)

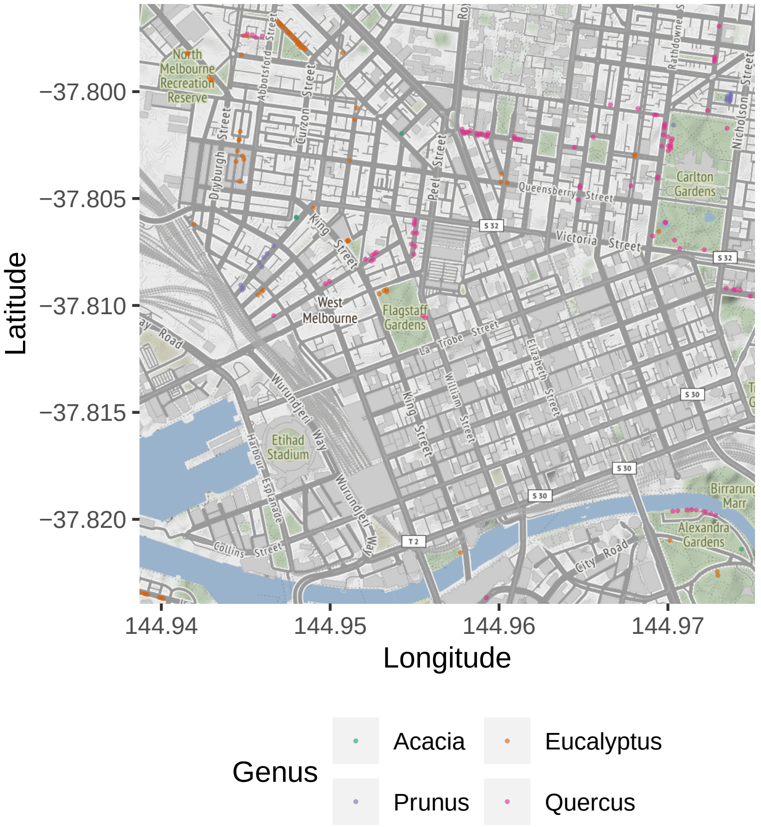
```r
# We have created the map below for you
melb_map <- read_rds(here::here("Data/melb-map.rds"))

# 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)
```



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)

```r
selected_group <- tree_data_clean1 %>%
  dplyr::filter(Genus %in% c("Eucalyptus",
                             "Macadamia",
                             "Prunus",
                             "Acacia",
                             "Quercus")) %>%
  droplevels()
```

```
ggmap(melb_map) +
  geom_point(data = selected_group,
             aes(x = Longitude,
                 y = Latitude,
              color = Genus),
             alpha = 0.6,
             size = 0.2) +
  labs(x = "Longitude",
       y = "Latitude") +
  scale_colour_brewer(palette = "Dark2",
                      name = "Genus") +
  guides(col = guide_legend(nrow = 2,
                            byrow = TRUE)) +
  theme(legend.position = "bottom")
```



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)

```
# This will order the trees from the most recent planted to the older onces
# because we are using descending order for the Year variable
tree_data_clean_filter <- tree_data_clean1 %>%
  select("Year",
         "Located in",
         "Common Name") %>%
  arrange(desc(Year))

head(tree_data_clean_filter, 4)
```

```
## # A tibble: 4 x 3
##    Year `Located in` `Common Name`
##   <dbl> <chr>        <chr>
## 1  2000 Street       Small-leaved Linden
## 2  2000 Street       Spotted Gum
## 3  2000 Street       Drooping sheoak
## 4  2000 Park         Kanooka
```

```
# To find out the older trees you could simple look at the tail of the data
# created in the previous R code chunk using  --> tail(tree_data_clean_filter, 4)
# the function tail() will show you the end of the data.

# Alternatively you can simply re-do the same steps as above and
# arrange the variable Year from smaller to larger as follows:
tree_data_clean_filter2 <- tree_data_clean1 %>%
  select("Year",
         "Located in",
         "Common Name") %>%
  arrange(Year)

head(tree_data_clean_filter2, 4)
```

```
## # A tibble: 4 x 3
##    Year `Located in` `Common Name`
##   <dbl> <chr>        <chr>
## 1  1900 Park         White Poplar
## 2  1900 Park         London Plane
## 3  1900 Street       Variegated Elm
## 4  1900 Park         Canary Island Pine
```

The first tree was planted in 1900 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)

```
tree_data_clean1 %>%
  dplyr::filter(`Located in` %in% c("Park", "Street")) %>%
  count(`Located in`) %>%
  kable()
```

Located in	n
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)

```
tree_data_clean1 %>%
  count(Family, sort = TRUE) %>%
  head(n = 5)
```

```
## # A tibble: 5 x 2
##   Family          n
##   <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)

```
 tree_data_clean1 %>%
  dplyr::filter(`Common Name` %in% c("Ironbark",
                                     "Olive",
                                     "Plum",
                                     "Oak",
                                     "Elm")) %>%
  group_by(Yearplanted, `Common Name`) %>%
  count(`Common Name`, sort = TRUE) %>%
  kable()
```

Yearplanted	Common Name	n
1900	Elm	179
1900	Ironbark	29
2000	Ironbark	23
2000	Elm	18
1900	Olive	17
2000	Oak	9
1900	Oak	4

The oldest most abundant tree was 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)

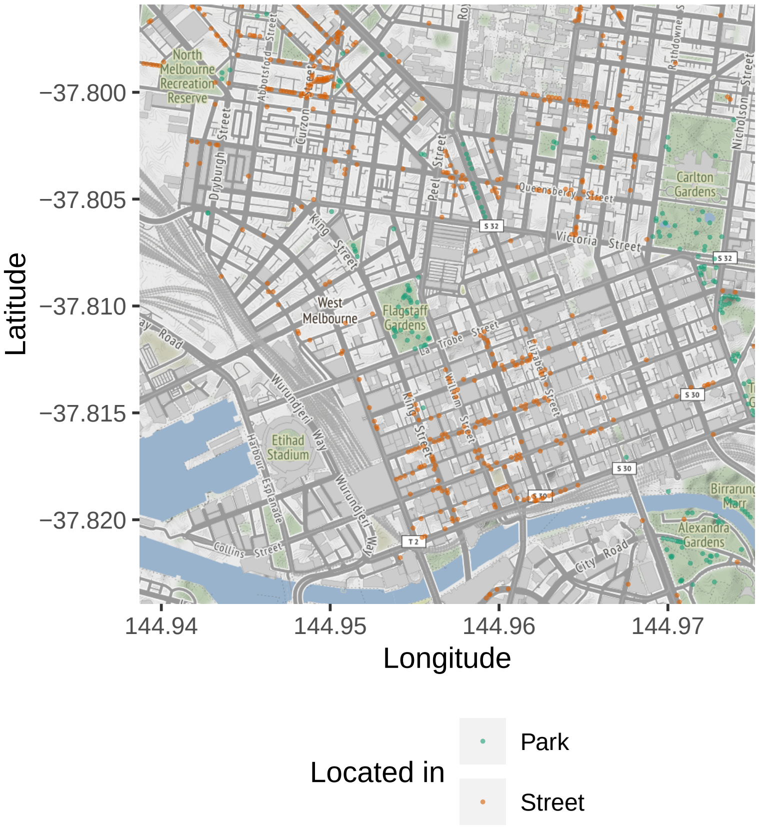
```r
large_trees_data <- tree_data_clean1 %>%
  dplyr::filter(`Diameter Breast Height` > 40 ,
                `Diameter Breast Height` < 100) %>%
  count(`Located in`)
```

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)

```r
large_trees_data_parks <- tree_data_clean1 %>%
  dplyr::filter(`Diameter Breast Height` > 40 ,
                `Diameter Breast Height` < 100)
```

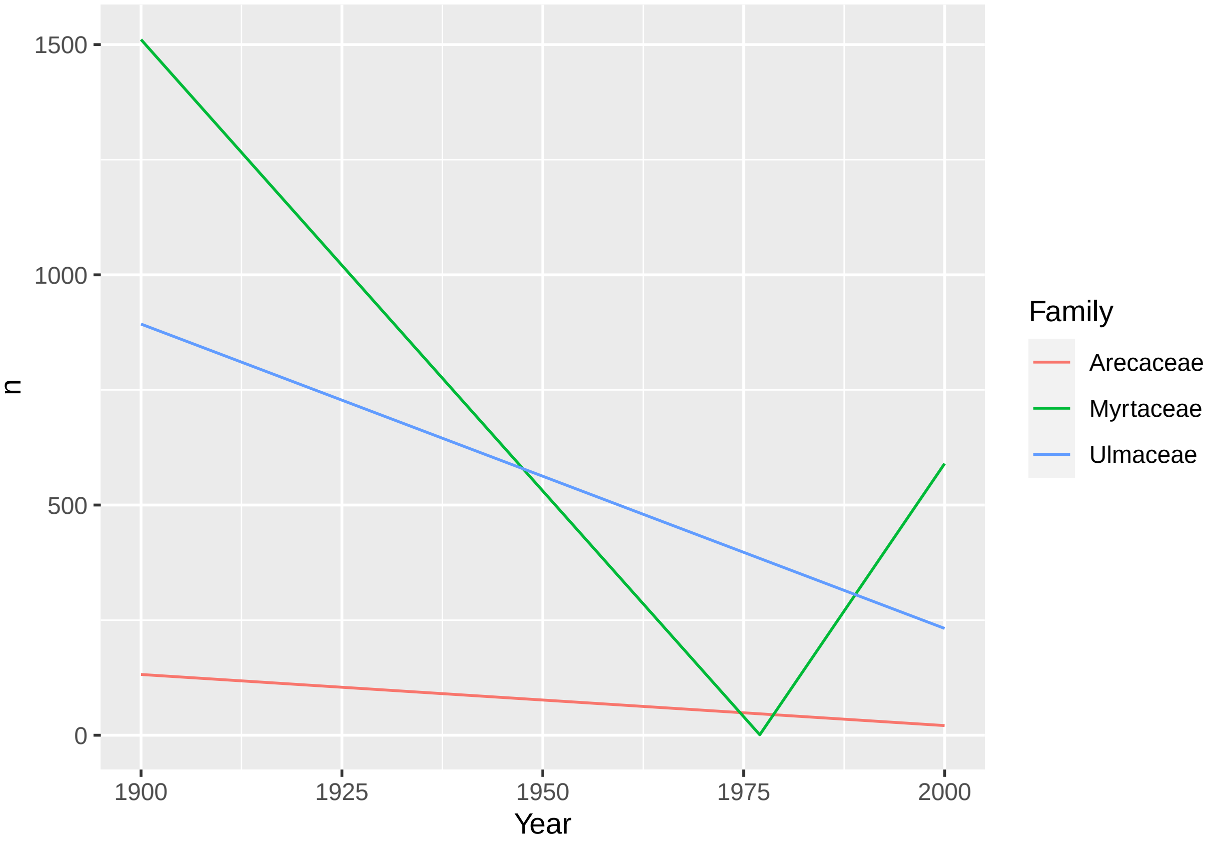
Large trees seem to be concentrated on certain streets.

```r
ggmap(melb_map) +
  geom_point(data = large_trees_data_parks ,
             aes(x = Longitude,
                 y = Latitude,
                 color = `Located in`),
             alpha = 0.6,
             size = 0.2) +
  labs(x = "Longitude",
       y = "Latitude") +
  scale_colour_brewer(palette = "Dark2") +
  guides(col = guide_legend(nrow = 2,
                            byrow = TRUE)) +
  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 %>%
 dplyr::filter(Family %in% c("Myrtaceae", "Arecaceae", "Ulmaceae")) %>%
 mutate(Family = as.factor(Family)) %>%
 group_by(Year, Family) %>%
 count(Family, sort = TRUE)

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



With time less arecaceae and ulmaceae family trees have been planted. After 1977 there was an increase in the number of myrtaceae family trees planted.

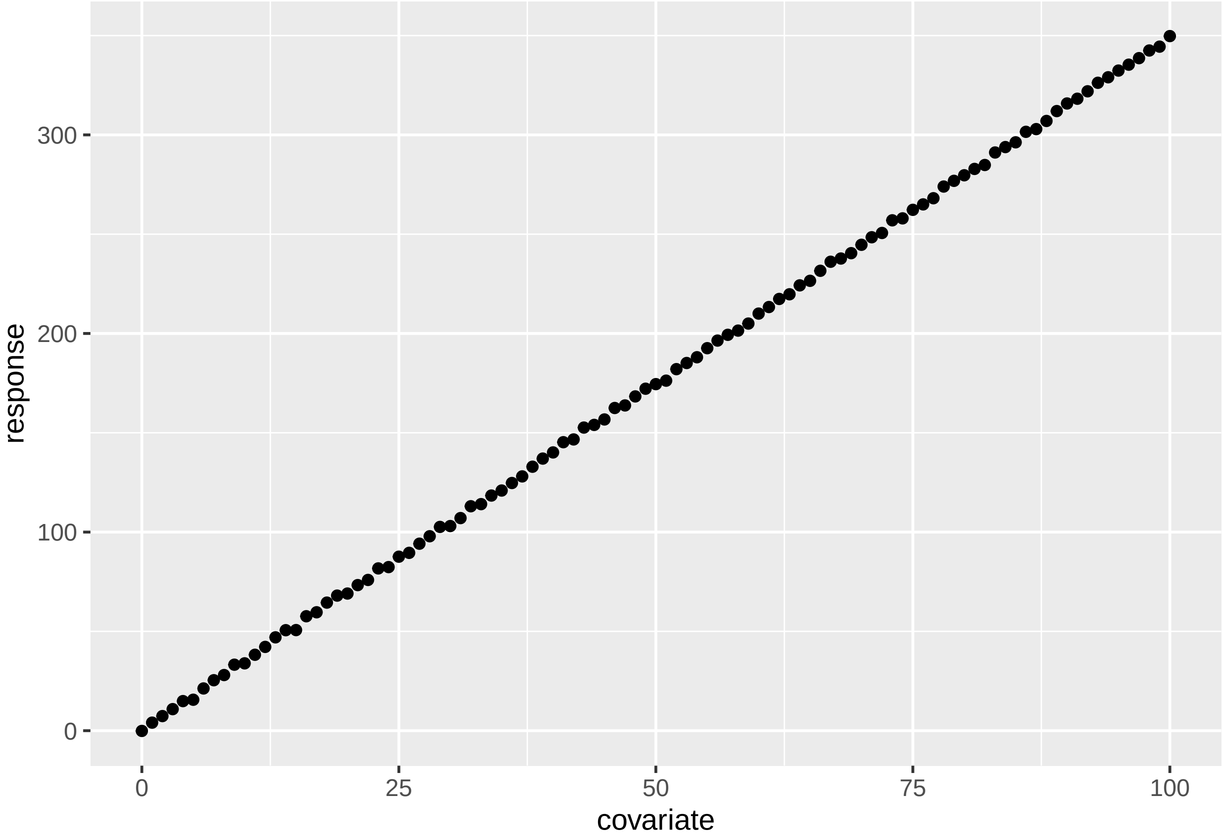
Part 2: Simulation Exercise

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)
simulation_data <- data.frame( covariate = c(0:100),
                               response = 3.5*c(0:100) + rnorm(101))
```

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(x = covariate, y = response)) +
  geom_point()
```



The relationship between the variables response and covariate is linear

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

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

```
##
## 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)

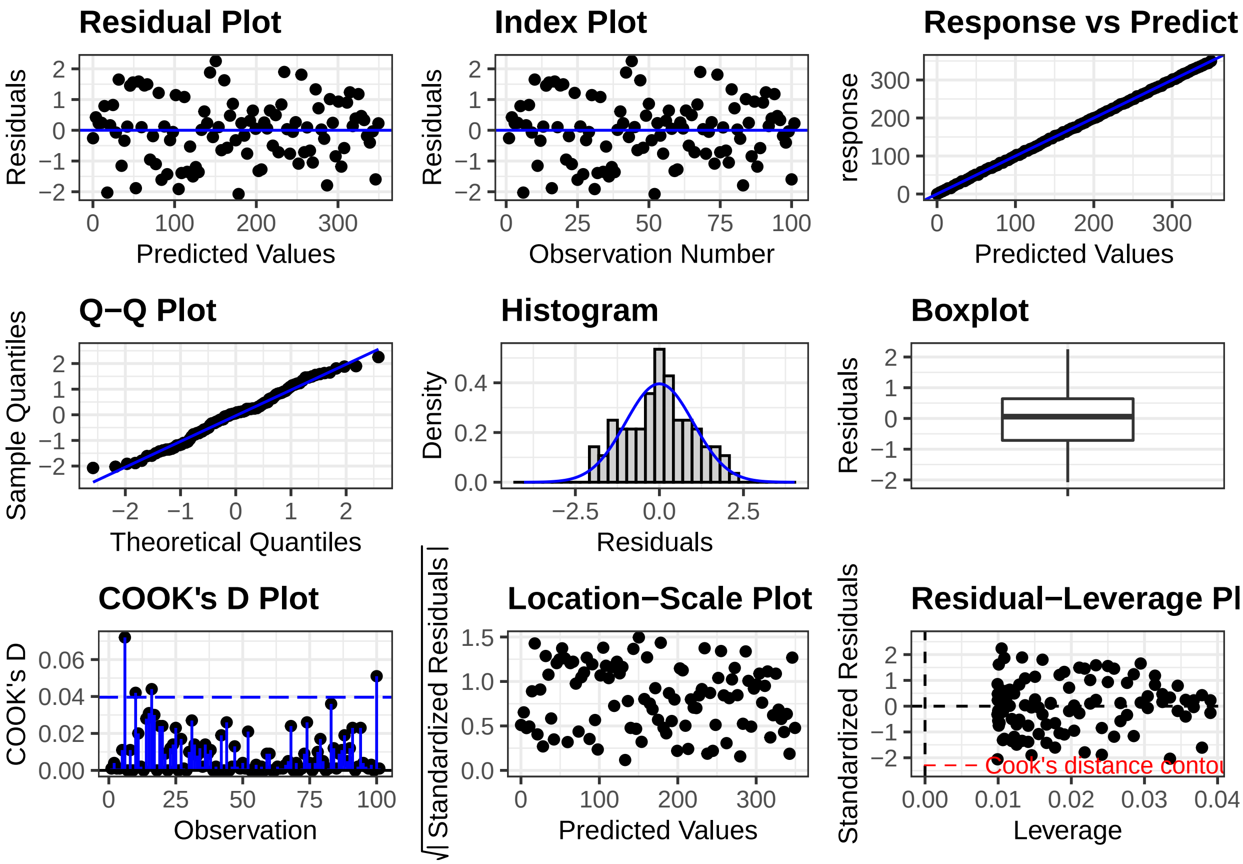
```
coef(mod)
```

```
## (Intercept)   covariate
##   0.1358957   3.4937754
```

The slope is estimated well and the intercept in this model takes the value of epsilon when covariate takes value 0.

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)

```
resid_panel(mod, plots = "all")
```



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

```
broom::glance(mod)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic   p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>     <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1      1.00          1.00  1.01  1022819. 1.58e-200     1  -144.  293.  301.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```