Assignment 7

Problem 1

The following questions use datasets from the nycflights13 R package.

```
library(nycflights13)
library(tidyverse)
```

```
## -- 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(lubridate)
library(ggplot2)
```

a. Find the 48 hours (over the course of the whole year) that have the worst delays. Cross-reference with weather data. Can you see any patterns?

```
flights$hour <- flights$sched_dep_time %/% 100
avg_delay <- aggregate(dep_delay ~ origin + year + month + day + hour,
                       data = flights,
                       FUN = function(x) mean(x, na.rm = TRUE))
worst_48 <- avg_delay[order(-avg_delay$dep_delay), ][1:48, ]
worst_48
```

```
##       origin year month day hour dep_delay
## 18403    LGA 2013     7  28   21  279.6667
## 5695     EWR 2013     2   9   10  269.0000
## 4601     EWR 2013     2   9    9  266.0000
## 12033    LGA 2013     9   2   16  250.3077
## 14929    LGA 2013     7  22   18  246.4000
## 16237    LGA 2013     7  28   19  239.5000
## 17746    JFK 2013     4  10   21  237.0000
## 16754    LGA 2013     9  12   20  226.2500
## 7849     EWR 2013     3   8   12  224.8000
## 6676     LGA 2013    12   5   11  220.8571
## 13481    EWR 2013     9  12   17  217.1429
```

```
## 17834    LGA 2013     9  12   21  208.7500
## 15665    EWR 2013     9  12   19  208.5000
## 18659    LGA 2013     7   9   22  208.0000
## 18364    LGA 2013     6  27   21  204.0000
## 13867    LGA 2013     5  23   17  199.1667
## 16402    LGA 2013     9   2   20  199.0000
## 16666    JFK 2013     4  10   20  197.1538
## 15904    LGA 2013     4  19   19  197.0000
## 16667    LGA 2013     4  10   20  196.2000
## 14218    LGA 2013     9   2   18  188.8824
## 15106    LGA 2013     6  27   18  188.6667
## 16021    LGA 2013     7  22   19  188.2500
## 17102    LGA 2013     7  22   20  184.0000
## 16235    EWR 2013     7  28   19  180.6471
## 9843     LGA 2013     9   2   14  179.0588
## 17316    LGA 2013     7  28   20  177.8571
## 6788     EWR 2013     2   9   11  177.2500
## 18674    JFK 2013     4  10   22  176.4000
## 16676    LGA 2013     7  10   20  176.0000
## 5664     LGA 2013     3   8   10  173.8000
## 7770     LGA 2013    12   5   12  173.2727
## 11298    LGA 2013     9  12   15  172.0833
## 16948    LGA 2013     3  18   20  171.5714
## 5581     EWR 2013    12   5   10  170.9333
## 15580    LGA 2013     4  10   19  169.4615
## 13022    EWR 2013     6  30   16  167.6429
## 13024    LGA 2013     6  30   16  167.3333
## 16051    LGA 2013     5  23   19  167.2727
## 15579    JFK 2013     4  10   19  164.4375
## 16608    LGA 2013     8   8   20  161.0000
## 19161    JFK 2013     4  10   23  159.0000
## 18840    JFK 2013     4  19   22  157.0000
## 10039    EWR 2013     3   8   14  156.5385
## 8946     LGA 2013     3   8   13  156.0000
## 11136    LGA 2013     3   8   15  155.8125
## 15587    EWR 2013     7  10   19  155.3077
## 6757     LGA 2013     3   8   11  155.2000
```

```
# Add weather information
worst_48_weather <- left_join(worst_48, weather,
                              by = c("origin", "year", "month", "day", "hour"))
worst_48_weather
```

```
##    origin year month day hour dep_delay  temp  dewp  humid wind_dir wind_speed
## 1     LGA 2013     7  28   21  279.6667 73.94 69.08  84.80      180    8.05546
## 2     EWR 2013     2   9   10  269.0000 28.04 15.98  60.10      310   19.56326
## 3     EWR 2013     2   9    9  266.0000 26.96 17.06  65.84      310   13.80936
## 4     LGA 2013     9   2   16  250.3077 80.06 71.96  76.36        0    0.00000
## 5     LGA 2013     7  22   18  246.4000 80.06 73.04  79.20      130    9.20624
## 6     LGA 2013     7  28   19  239.5000 75.20 69.80  83.32      170   10.35702
## 7     JFK 2013     4  10   21  237.0000 53.06 50.00  89.31      110    9.20624
## 8     LGA 2013     9  12   20  226.2500 71.96 68.00  88.43      300   13.80936
## 9     EWR 2013     3   8   12  224.8000 33.80 32.00  95.75      320    9.20624
## 10    LGA 2013    12   5   11  220.8571 51.08 48.92  93.50      110    4.60312
```

```
## 11    EWR 2013     9  12   17  217.1429 84.02 73.04  69.64       180    5.75390
## 12    LGA 2013     9  12   21  208.7500 71.96 68.00  87.35       220    4.60312
## 13    EWR 2013     9  12   19  208.5000 78.80 71.60  93.47       260   18.41248
## 14    LGA 2013     7   9   22  208.0000 82.40 66.20  58.08       230    8.05546
## 15    LGA 2013     6  27   21  204.0000 77.00 71.06  84.92       150   16.11092
## 16    LGA 2013     5  23   17  199.1667 66.92 64.04  90.46       180   10.35702
## 17    LGA 2013     9   2   20  199.0000 77.00 71.96  84.46       170   10.35702
## 18    JFK 2013     4  10   20  197.1538 55.40 51.80  87.65       130   17.26170
## 19    LGA 2013     4  19   19  197.0000 62.06 57.02  83.54       170   17.26170
## 20    LGA 2013     4  10   20  196.2000 60.80 53.60  77.59        NA   11.50780
## 21    LGA 2013     9   2   18  188.8824 80.06 71.06  74.07         0    0.00000
## 22    LGA 2013     6  27   18  188.6667 78.98 69.08  71.72       140   10.35702
## 23    LGA 2013     7  22   19  188.2500 80.60 73.40  79.12       140   11.50780
## 24    LGA 2013     7  22   20  184.0000 78.08 73.04  84.53       140    6.90468
## 25    EWR 2013     7  28   19  180.6471 73.40 71.60  94.10        NA    6.90468
## 26    LGA 2013     9   2   14  179.0588 75.02 69.08  81.79       240    4.60312
## 27    LGA 2013     7  28   20  177.8571 73.94 69.08  84.80       190    5.75390
## 28    EWR 2013     2   9   11  177.2500 28.94 15.98  57.92       320   28.76950
## 29    JFK 2013     4  10   22  176.4000 55.94 53.06  90.03       340   11.50780
## 30    LGA 2013     7  10   20  176.0000 86.00 68.00  55.04       230    9.20624
## 31    LGA 2013     3   8   10  173.8000 33.80 28.94  84.56       350   20.71404
## 32    LGA 2013    12   5   12  173.2727 53.96 51.98  93.60       180    6.90468
## 33    LGA 2013     9  12   15  172.0833 82.04 68.00  62.53       180   10.35702
## 34    LGA 2013     3  18   20  171.5714 32.00 28.94  88.32        70   13.80936
## 35    EWR 2013    12   5   10  170.9333 53.60 51.98 100.00         0    0.00000
## 36    LGA 2013     4  10   19  169.4615 60.08 53.06  77.59       310   33.37262
## 37    EWR 2013     6  30   16  167.6429 80.06 73.04  79.20       160    8.05546
## 38    LGA 2013     6  30   16  167.3333 75.92 71.60  88.59       170   13.80936
## 39    LGA 2013     5  23   19  167.2727 66.92 62.96  88.18       170   13.80936
## 40    JFK 2013     4  10   19  164.4375 60.80 48.92  71.64       330   31.07106
## 41    LGA 2013     8   8   20  161.0000 75.02 69.98  84.34        NA    4.60312
## 42    JFK 2013     4  10   23  159.0000 55.40 53.96  96.14       170    8.05546
## 43    JFK 2013     4  19   22  157.0000 57.02 55.94  96.17       180   29.92028
## 44    EWR 2013     3   8   14  156.5385 33.98 33.08  96.46       320    8.05546
## 45    LGA 2013     3   8   13  156.0000 35.96 28.94  75.39       360   21.86482
## 46    LGA 2013     3   8   15  155.8125 37.94 28.94  69.73       360   16.11092
## 47    EWR 2013     7  10   19  155.3077 86.00 69.08  57.12       250   13.80936
## 48    LGA 2013     3   8   11  155.2000 33.98 28.94  81.57       350   19.56326
##    wind_gust precip pressure visib           time_hour
## 1         NA   0.00   1015.0  8.00 2013-07-28 21:00:00
## 2   28.76950   0.00   1016.5 10.00 2013-02-09 10:00:00
## 3   28.76950   0.00   1015.2 10.00 2013-02-09 09:00:00
## 4         NA   0.01   1006.6  5.00 2013-09-02 16:00:00
## 5         NA   0.00   1010.5 10.00 2013-07-22 18:00:00
## 6         NA   0.01       NA  8.00 2013-07-28 19:00:00
## 7         NA   0.01   1010.5  9.00 2013-04-10 21:00:00
## 8         NA   0.30       NA  9.00 2013-09-12 20:00:00
## 9         NA   0.06       NA  1.00 2013-03-08 12:00:00
## 10        NA   0.00       NA  2.50 2013-12-05 11:00:00
## 11        NA   0.00   1005.5  9.00 2013-09-12 17:00:00
## 12        NA   0.03   1006.0 10.00 2013-09-12 21:00:00
## 13        NA   0.23       NA  8.00 2013-09-12 19:00:00
## 14        NA   0.00       NA 10.00 2013-07-09 22:00:00
## 15  23.01560   0.07       NA 10.00 2013-06-27 21:00:00
```

```
## 16        NA   0.00   1009.5  7.00 2013-05-23 17:00:00
## 17        NA   0.00   1006.3  7.00 2013-09-02 20:00:00
## 18        NA   0.14       NA 10.00 2013-04-10 20:00:00
## 19  25.31716   0.00       NA  2.50 2013-04-19 19:00:00
## 20        NA   0.02       NA 10.00 2013-04-10 20:00:00
## 21        NA   0.00   1006.3  8.00 2013-09-02 18:00:00
## 22        NA   0.00   1005.1 10.00 2013-06-27 18:00:00
## 23        NA   0.00       NA 10.00 2013-07-22 19:00:00
## 24        NA   0.00   1010.1 10.00 2013-07-22 20:00:00
## 25        NA   0.08       NA  7.00 2013-07-28 19:00:00
## 26        NA   0.01   1007.4  6.00 2013-09-02 14:00:00
## 27        NA   0.03   1014.2  8.00 2013-07-28 20:00:00
## 28  35.67418   0.00   1016.8 10.00 2013-02-09 11:00:00
## 29        NA   0.10       NA  8.00 2013-04-10 22:00:00
## 30        NA   0.00   1010.0 10.00 2013-07-10 20:00:00
## 31  33.37262   0.02       NA  1.00 2013-03-08 10:00:00
## 32        NA   0.00       NA  7.00 2013-12-05 12:00:00
## 33  18.41248   0.00   1007.0 10.00 2013-09-12 15:00:00
## 34        NA   0.12   1020.2  1.25 2013-03-18 20:00:00
## 35        NA   0.00       NA  0.50 2013-12-05 10:00:00
## 36        NA   0.14       NA  7.00 2013-04-10 19:00:00
## 37        NA   0.00   1009.8  8.00 2013-06-30 16:00:00
## 38        NA   0.00       NA  3.00 2013-06-30 16:00:00
## 39        NA   0.00       NA  8.00 2013-05-23 19:00:00
## 40        NA   0.11       NA 10.00 2013-04-10 19:00:00
## 41        NA   0.00   1016.5 10.00 2013-08-08 20:00:00
## 42        NA   0.15       NA  4.00 2013-04-10 23:00:00
## 43  39.12652   0.00   1005.2  0.25 2013-04-19 22:00:00
## 44        NA   0.05       NA  1.75 2013-03-08 14:00:00
## 45  29.92028   0.01       NA  6.00 2013-03-08 13:00:00
## 46  26.46794   0.00   1017.2 10.00 2013-03-08 15:00:00
## 47  18.41248   0.00   1010.3 10.00 2013-07-10 19:00:00
## 48        NA   0.01       NA  5.00 2013-03-08 11:00:00
```

```r
# It's hard to tell if there's a correlation b/w weather and delays.
```

b. Add the latitude and longitude of the origin and destination airports to the flights dataset. Is it easier to rename the columns before or after the join?

```r
flights_with_coords <- flights %>%
  left_join(airports %>% rename(origin_lat = lat, origin_lon = lon), by = c("origin" = "faa")) %>%
  left_join(airports %>% rename(dest_lat = lat, dest_lon = lon), by = c("dest" = "faa"))
flights_with_coords
```

```
## # A tibble: 336,776 x 33
##     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
##  1  2013     1     1      517            515         2      830            819
##  2  2013     1     1      533            529         4      850            830
##  3  2013     1     1      542            540         2      923            850
##  4  2013     1     1      544            545        -1     1004           1022
##  5  2013     1     1      554            600        -6      812            837
##  6  2013     1     1      554            558        -4      740            728
```

```
##  7  2013     1     1      555            600        -5      913            854
##  8  2013     1     1      557            600        -3      709            723
##  9  2013     1     1      557            600        -3      838            846
## 10  2013     1     1      558            600        -2      753            745
## # i 336,766 more rows
## # i 25 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>, name.x <chr>, origin_lat <dbl>,
## #   origin_lon <dbl>, alt.x <dbl>, tz.x <dbl>, dst.x <chr>, tzone.x <chr>,
## #   name.y <chr>, dest_lat <dbl>, dest_lon <dbl>, alt.y <dbl>, tz.y <dbl>,
## #   dst.y <chr>, tzone.y <chr>
```

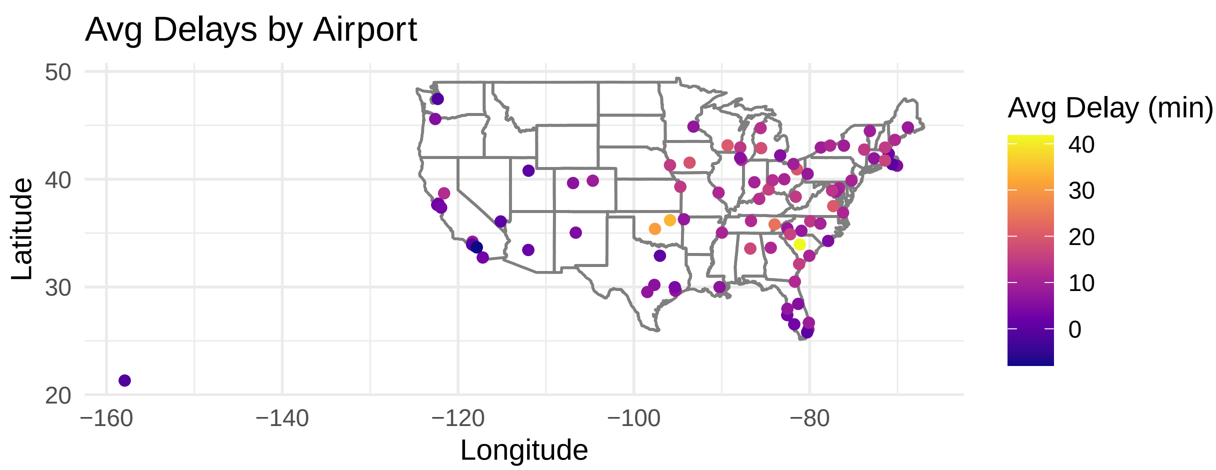
```r
# It's easier to rename the columns before the join.
# Renaming after joining is messier.
```

c. Compute the average delay by destination, then join on airports and show the spatial distribution of delays.

```r
avg_delay_by_dest <- flights %>%
  group_by(dest) %>%
  summarise(avg_arr_delay = mean(arr_delay, na.rm = TRUE), n = n()) %>%
  filter(n > 100)

delay_map_data <- avg_delay_by_dest %>%
  inner_join(airports, by = c("dest" = "faa"))

ggplot(delay_map_data, aes(x = lon, y = lat, color = avg_arr_delay)) +
  borders("state") +
  geom_point() +
  coord_quickmap() +
  theme_minimal() +
  scale_color_viridis_c(option = "C") +
  labs(
    title = "Avg Delays by Airport",
    x = "Longitude", y = "Latitude",
    color = "Avg Delay (min)"
  )
```



d. From flights, compare air_time with the duration between departure and arrival.

```
flights %>%
  filter(!is.na(dep_time), !is.na(arr_time), !is.na(air_time)) %>%
  mutate(dep_minutes = (dep_time %/% 100) * 60 + (dep_time %% 100),
         arr_minutes = (arr_time %/% 100) * 60 + (arr_time %% 100),
         sched_duration = (arr_minutes - dep_minutes) %% (24 * 60),
         diff = sched_duration - air_time) %>%
  summarise(mean_difference = mean(diff, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   mean_difference
##             <dbl>
## 1           -13.4
```

```
# dep_time and arr_time does not account for potential time differences.
# air_time is the actual duration in flight.
```

Problem 2

For each of the following dates, intervals, and times, show how you'd convert them to ISO format using a lubridate function.

```r
library(lubridate)
d1 <- "January 1, 2010"
d2 <- "2015-Mar-07"
d3 <- "06-Jun-2017"
d4 <- c("August 19 (2015)", "July 1 (2015)") # This is an interval
d5 <- "12/30/14" # Dec 30, 2014
t1 <- "1705" # time 17:05
t2 <- "11:15:10 PM"

# Conversion to ISO format
d1_iso <- mdy(d1)
d1_iso
```

```
## [1] "2010-01-01"
```

```r
d2_iso <- ymd(d2)
d2_iso
```

```
## [1] "2015-03-07"
```

```r
d3_iso <- dmy(d3)
d3_iso
```

```
## [1] "2017-06-06"
```

```r
d4_interval_start <- mdy(d4[1])
d4_interval_end <- mdy(d4[2])
cat('"', as.character(d4_interval_start), '" to "', as.character(d4_interval_end), '"\n', sep="")
```

```
## "2015-08-19" to "2015-07-01"
```

```r
d5_iso <- mdy(d5)
d5_iso
```

```
## [1] "2014-12-30"
```

```r
t1_iso <- format(strptime(t1, format = "%H%M"), format = "%H:%M:%S")
t1_iso
```

```
## [1] "17:05:00"
```

```r
t2_iso <- format(strptime(t2, format = "%I:%M:%S %p"), format = "%H:%M:%S")
t2_iso
```

```
## [1] "23:15:10"
```

Problem 3

Write a function that takes a birthday (as a date) as input and outputs how old you are in years.

```r
calculate_age <- function(birthday) {
  current_date <- Sys.Date()

  # Calculate age
  age <- as.numeric(format(current_date, "%Y")) - as.numeric(format(birthday, "%Y"))

  # Cases where birthday hasn't occurred yet in the year
  if (format(current_date, "%m-%d") < format(birthday, "%m-%d")) {
    age <- age - 1
  }
  return(age)
}

# Example
calculate_age(as.Date("2003-03-30"))
```

```
## [1] 22
```

```r
calculate_age(as.Date("2003-05-01"))
```

```
## [1] 21
```

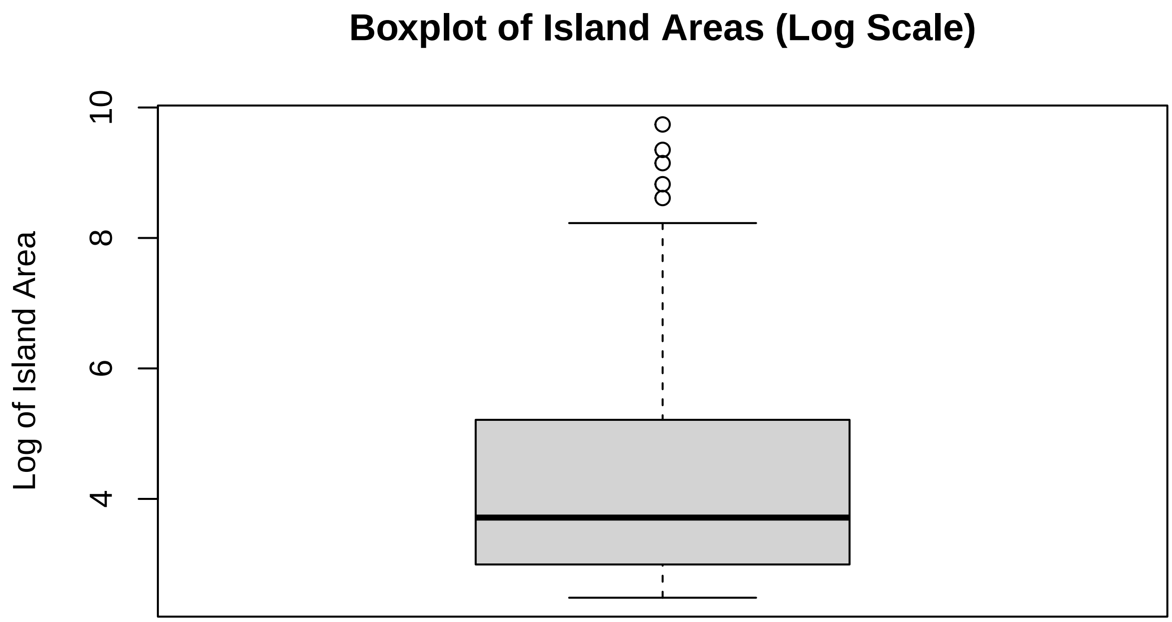
Problem 4

Consider the islands vector discussed in class

a. Construct a boxplot for the islands dataset on both a log scale and the original scale.

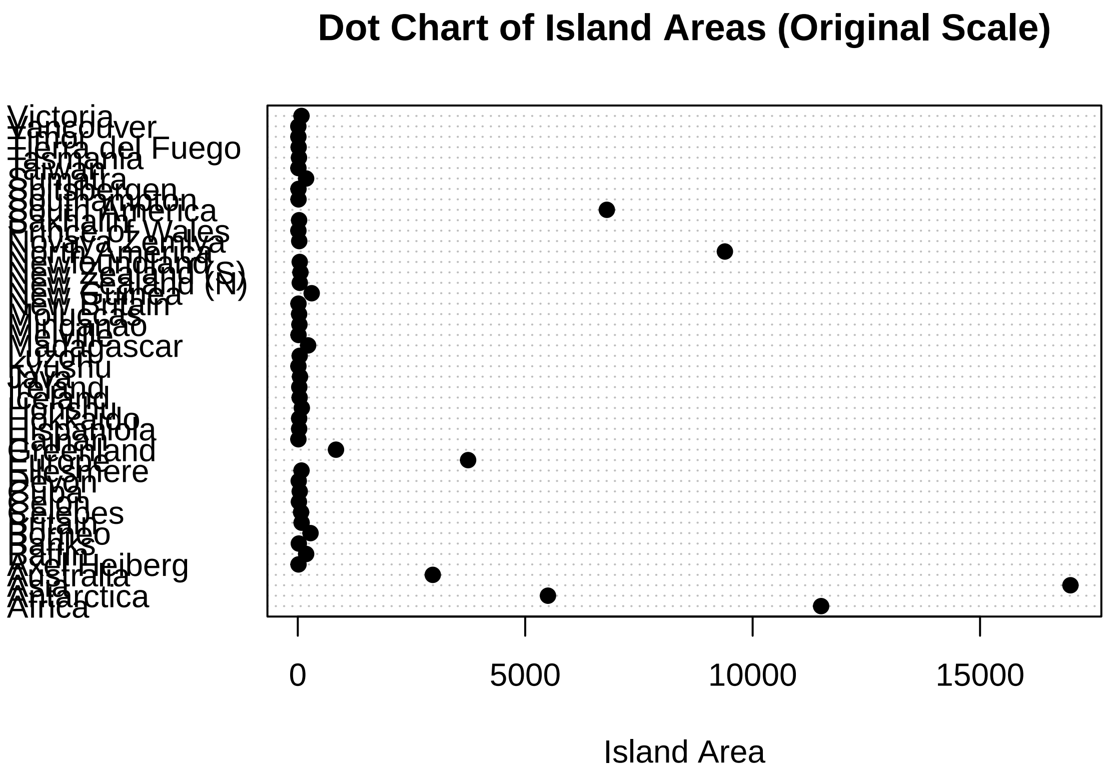
```r
boxplot(islands,
        main="Boxplot of Island Areas (Original Scale)",
        ylab="Island Area")
```


```r
boxplot(log(islands),
        main="Boxplot of Island Areas (Log Scale)",
        ylab="Log of Island Area")
```

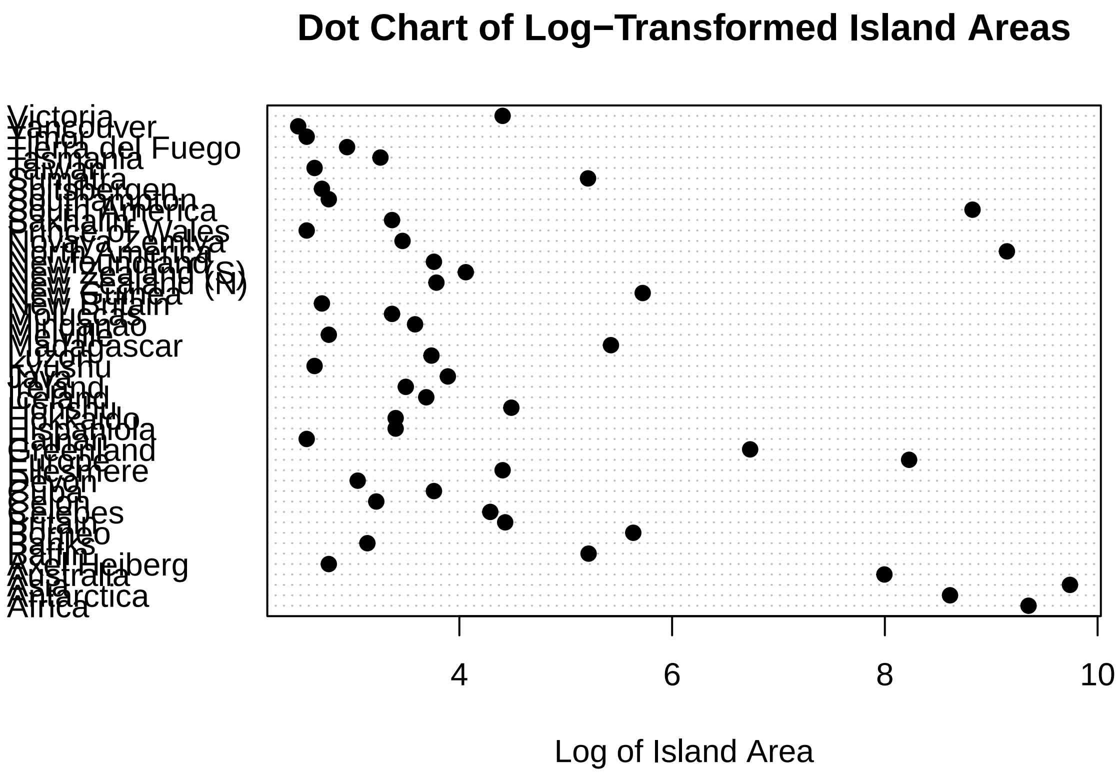


b. Construct a dot chart of the island areas. Is a log transformation needed here?

```
dotchart(islands,
         main="Dot Chart of Island Areas (Original Scale)",
         xlab="Island Area",
         pch=19)
```




```r
dotchart(log(islands),
         main="Dot Chart of Log-Transformed Island Areas",
         xlab="Log of Island Area",
         pch=19)
```



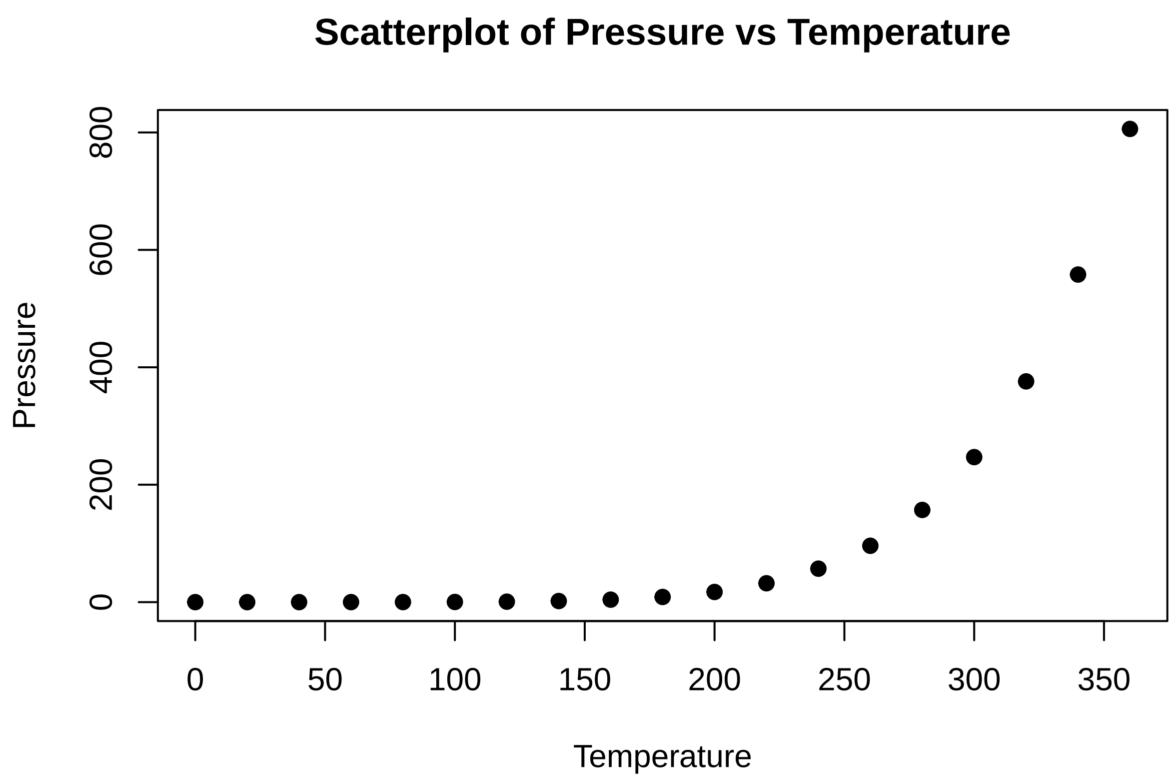

```
# The dot chart with original scale has extreme outliers.
# Hence, log transformation is needed to have a better idea of the distribution.
```

Problem 5

The pressure dataset contains two columns: temperature and pressure

a. Construct a scatterplot with pressure on the vertical axis and temperature on the horizontal axis. Are the variables related linearly or nonlinearly?

```
plot(pressure$temperature, pressure$pressure,
     xlab = "Temperature",
     ylab = "Pressure",
     main = "Scatterplot of Pressure vs Temperature",
     pch = 19)
```

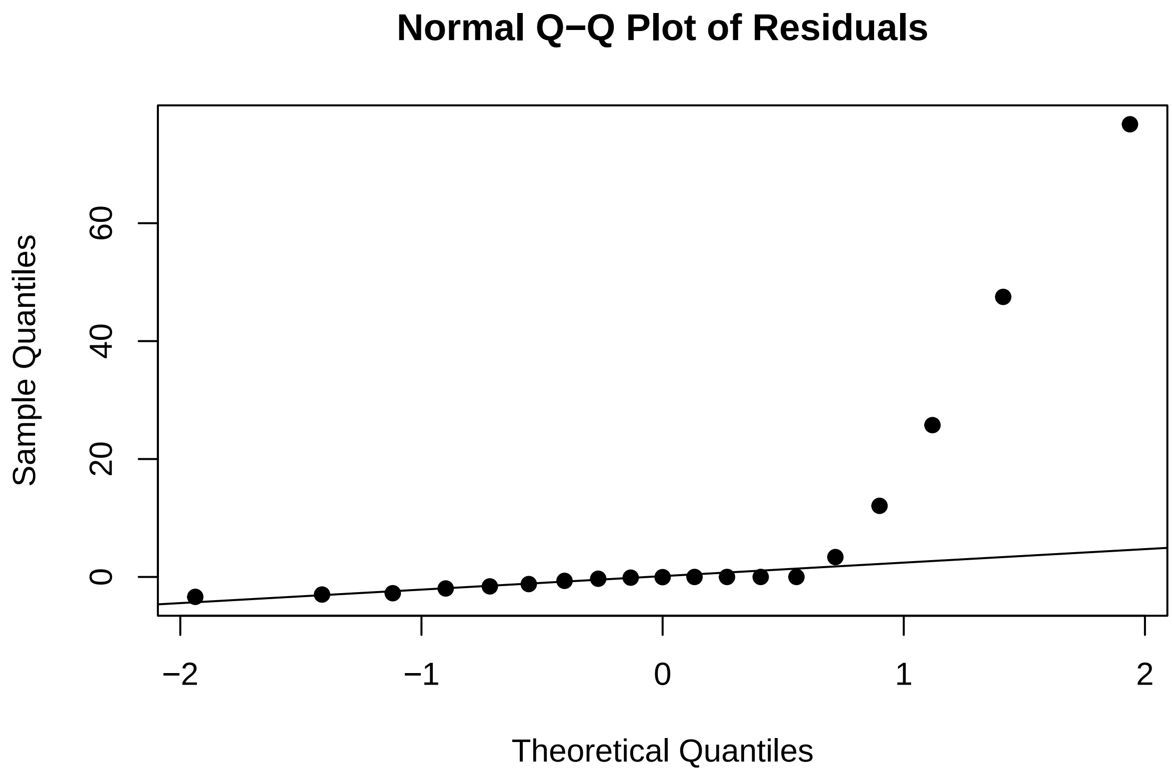



```
# As temperature increases, pressure increases as well, but not linearly.
```

b. Construct a normal QQ-plot of these residuals and determine whether they follow a normal or skewed distribution.

```
residuals <- with(pressure, pressure - (0.168 + 0.007 * temperature)^(20/3))

qqnorm(residuals,
       main = "Normal Q-Q Plot of Residuals",
       pch = 19)
qqline(residuals)
```

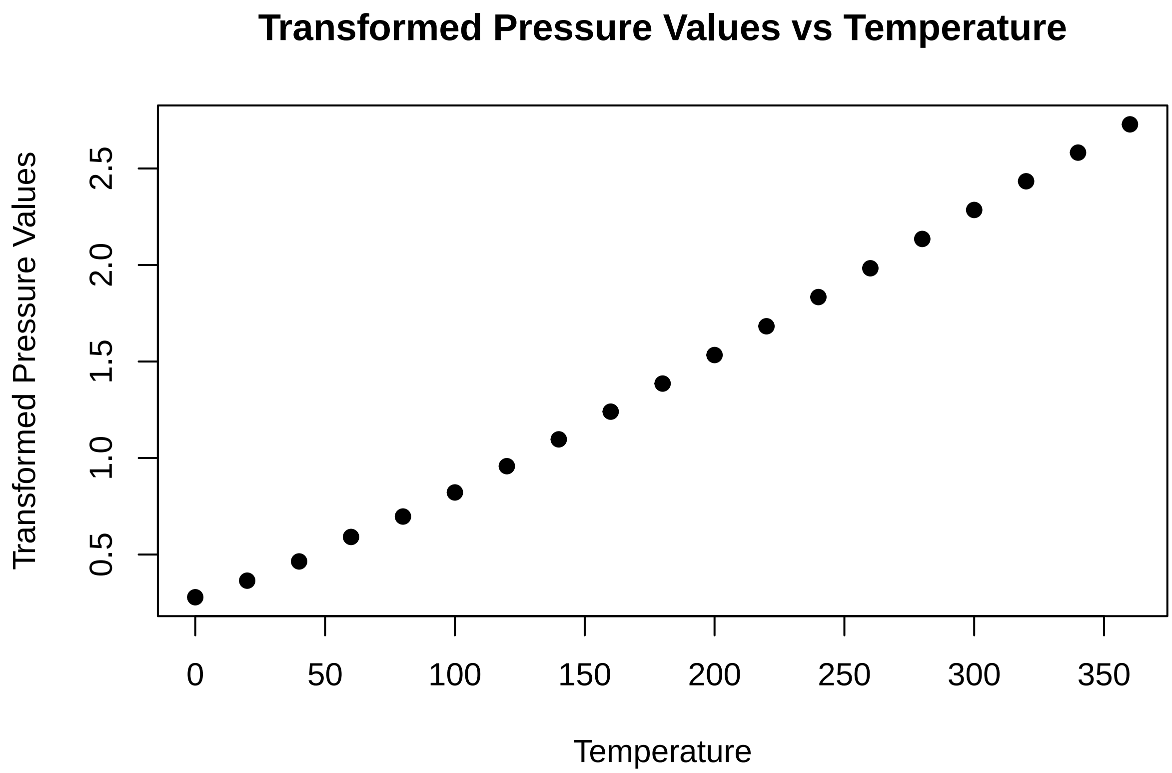



```
# Residuals follow the normal QQ-line only up until quantile 1.
# After quantile 1, the residuals deviate significantly.
# So, the distribution must be skewed.
```

c. Apply the power transformation y^{3/20} to the pressure values and plot them against temperature. Is the relationship now linear or nonlinear?

```
transformed_pressure <- pressure$pressure^(3/20)

plot(pressure$temperature, transformed_pressure,
     xlab = "Temperature",
     ylab = "Transformed Pressure Values",
     main = "Transformed Pressure Values vs Temperature",
     pch = 19)
```



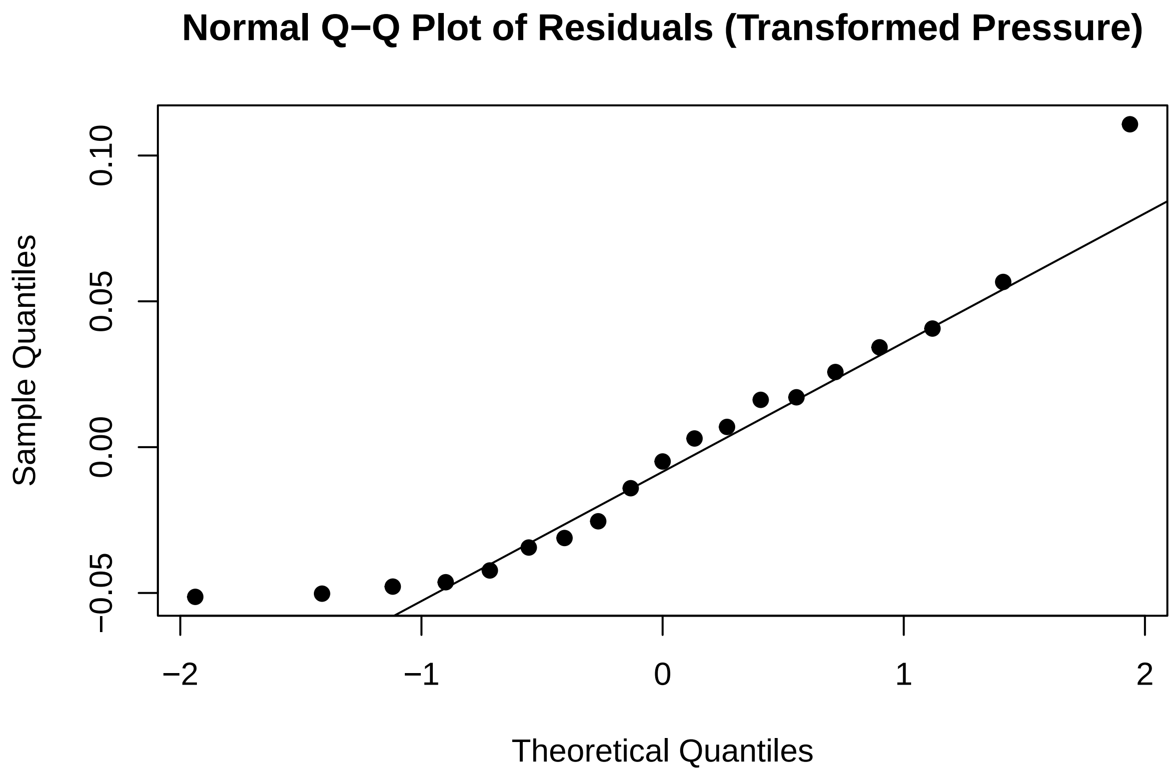

```
# The relationship now is linear
```

d. Calculate residuals for the difference between the transformed pressure values and those predicted by the straight line. Obtain a normal QQ-plot and decide whether the residuals follow a normal distribution.

```
y <- 0.168 + 0.007 * pressure$temperature
residuals_transformed <- transformed_pressure - y
residuals_transformed
```

```
##  [1]  0.110711250  0.056650814  0.016218760  0.002973747 -0.031154698
##  [6] -0.046316503 -0.050234499 -0.051330515 -0.047812375 -0.042289841
## [11] -0.034418307 -0.025419874 -0.014076446 -0.004917943  0.006936429
## [16]  0.017096489  0.025762397  0.034231233  0.040665226
```

```
qqnorm(residuals_transformed,
       main = "Normal Q-Q Plot of Residuals (Transformed Pressure)",
       pch = 19)
qqline(residuals_transformed)
```




```
# Points follow the line better than before, but with some outliers.
# The residuals are approximately normally distributed.
```