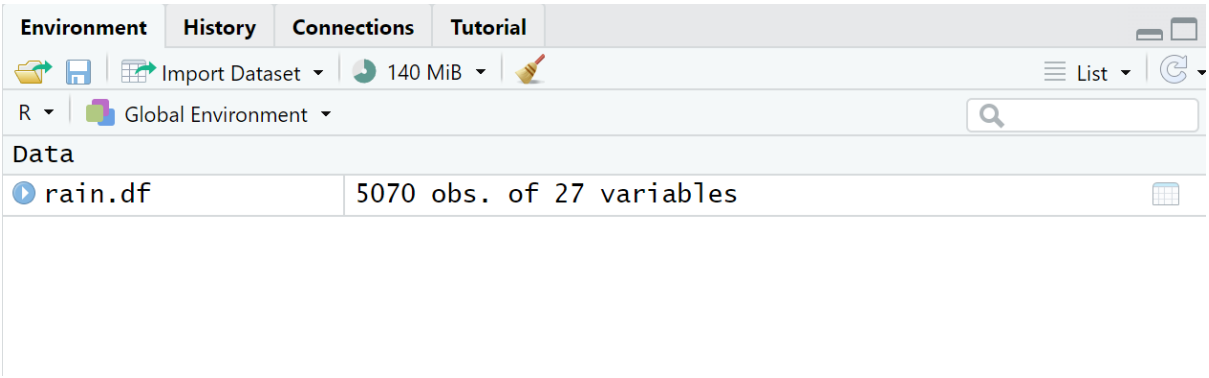
1. Rainfall data

a)



위 environment tab에 저장된 rain.df 변수의 정보를 통해 5070개의 row와 27개의 column이 있음을 알 수 있다.

b)

```
> names(rain.df)
 [1] "V1"  "V2"  "V3"  "V4"  "V5"  "V6"  "V7"  "V8"  "V9"  "V10" "V11" "V12" "V13" "V14"
[15] "V15" "V16" "V17" "V18" "V19" "V20" "V21" "V22" "V23" "V24" "V25" "V26" "V27"
>
```

names(rain.df) 를 통해 column들의 name은 v1 부터 v27 까지 있음을 알 수 있다.

c)

```
> rain.df[1,3]
[1] 1
```

위처럼 행과 열의 위치를 알면 value를 파악할 수 있다. 해당 row, column의 value는 1이다

d)

```
> rain.df[1,]
  V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20 V21 V22 V23 V24
1 60  4  1  0  0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  V25 V26 V27
1   0   0   0
```

첫 열의 전체 값을 출력할 수 있다.

e)

```
> names(rain.df) <- c("year", "month", "day", seq(0,23))
> rain.df
   year month day 0 1 2 3 4 5  6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
1    60     4   1 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
2    60     4   2 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
3    60     4   3 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
4    60     4   4 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
5    60     4   5 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
6    60     4   6 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
7    60     4   7 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
8    60     4   8 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
9    60     4   9 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Name 함수를 통해 rain.df의 column의 name들을 바꿀 수 있다. (row 에 해당하는 것은 현재 index이다.) 데이터가 1960년부터 1980년 까지의 강수량 데이터이므로 그에 맞게 해당되는 이름들을 부여했음을 볼 수 있다.

f)

```
> rain.df$rainsum<-apply(rain.df[,4:27],1,sum)
> rain.df
   year month day 0 1 2 3 4 5  6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 rainsum
1    60     4   1 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0       0
2    60     4   2 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0       0
3    60     4   3 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0       0
4    60     4   4 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0       0
5    60     4   5 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0       0
6    60     4   6 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0       0
7    60     4   7 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0       0
8    60     4   8 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0       0
9    60     4   9 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0       0
10   60     4  10 0 0 0 0 0 0  0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0       0
```

apply(rain.df[,4:27],1,sum) 을 이용해 하루 강수량 합계를 구할 수 있었다.

2. Data types

a) max(x)의 결과는 "5" 이다. 5, 15, 250, 2023 중에서 앞자리가 제일 큰 것이 "5"이기 때문이다. 중요한 것은 숫자들이 문자열로 인식되고 있다는 점이다. 따라서 sort(x)의 결과도 문자열의 순서대로 15, 2023, 250, 5 가 나온다. Sum(x)는 오류를 발생시키는데 문자열끼리의 합을 구하라고 해서 합을 구할 수가 없다. Is.factor(x)는 False를 반환하는데, 이는 categorical한 value가 아니기 때문이다.

b) y의 첫번째 원소가 character type이기 때문에, 뒤의 두 원소도 character type으로 변경되어서 벡터 y에 저장된다. 따라서 y[2]+y[3]은 오류를 발생시킨다. Length(y)는 원소의 개수이므로 3을 반환한다. Y는 벡터이므로 dim(y)는 null을 반환한다. Dim 함수는 array, matrix, dataframe에서만 작동한다. Is.factor(y) 또한 False를 반환한다.

3. Working with functions and operators

a)

```
> seq(from = 1, to = 10000, by = 372)
 [1]    1  373  745 1117 1489 1861 2233 2605 2977 3349 3721 4093 4465 4837 5209 5581 5953
[18] 6325 6697 7069 7441 7813 8185 8557 8929 9301 9673
```

```
> seq(from = 1, to = 10000, length.out = 50)
 [1]     1.0000   205.0612   409.1224   613.1837   817.2449  1021.3061  1225.3673
 [8]  1429.4286  1633.4898  1837.5510  2041.6122  2245.6735  2449.7347  2653.7959
[15]  2857.8571  3061.9184  3265.9796  3470.0408  3674.1020  3878.1633  4082.2245
[22]  4286.2857  4490.3469  4694.4082  4898.4694  5102.5306  5306.5918  5510.6531
[29]  5714.7143  5918.7755  6122.8367  6326.8980  6530.9592  6735.0204  6939.0816
[36]  7143.1429  7347.2041  7551.2653  7755.3265  7959.3878  8163.4490  8367.5102
[43]  8571.5714  8775.6327  8979.6939  9183.7551  9387.8163  9591.8776  9795.9388
[50] 10000.0000
```

b)

```
> rep(1:3, times = 3)
[1] 1 2 3 1 2 3 1 2 3
> rep(1 : 3, each = 3)
[1] 1 1 1 2 2 2 3 3 3
>
```

Times는 순차대로 반복하는 것이고, each는 각 원소를 반복하고 다음으로 넘어가는 차이가 있다.

4. Working with airquality dataset

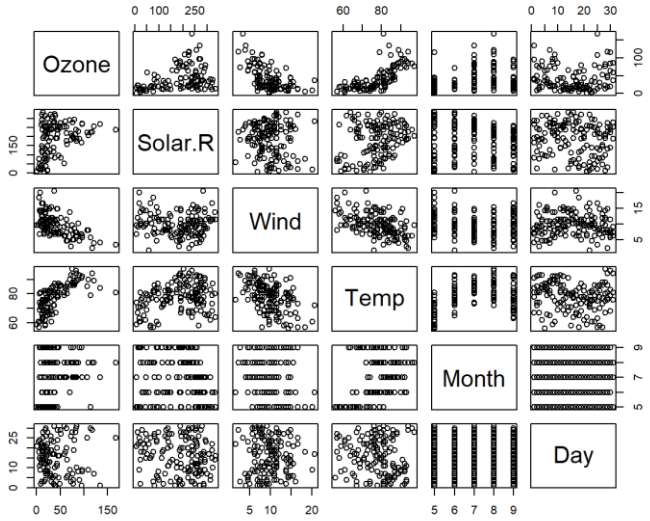
a)

```
> data(airquality)
> summary(airquality)
     Ozone           Solar.R          Wind             Temp           Month
 Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00   Min.   :5.000
 1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00   1st Qu.:6.000
 Median : 31.50   Median :205.0   Median : 9.700   Median :79.00   Median :7.000
 Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88   Mean   :6.993
 3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00   3rd Qu.:8.000
 Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00   Max.   :9.000
 NA's   :37       NA's   :7
      Day
 Min.   : 1.0
 1st Qu.: 8.0
 Median :16.0
 Mean   :15.8
 3rd Qu.:23.0
 Max.   :31.0
```

```
> sum(is.na(airquality))
[1] 44
```

Summary를 통해 airquality가 ozone, solar.R, wind, temp, month, day 총 6개의 variable를 갖고 있고, 153개의 observation이 있음을 확인할 수 있다. 또한 최소값, 중앙값, 평균, 1,3사분위 등과 결측치들의 개수도 확인할 수 있다.

b)

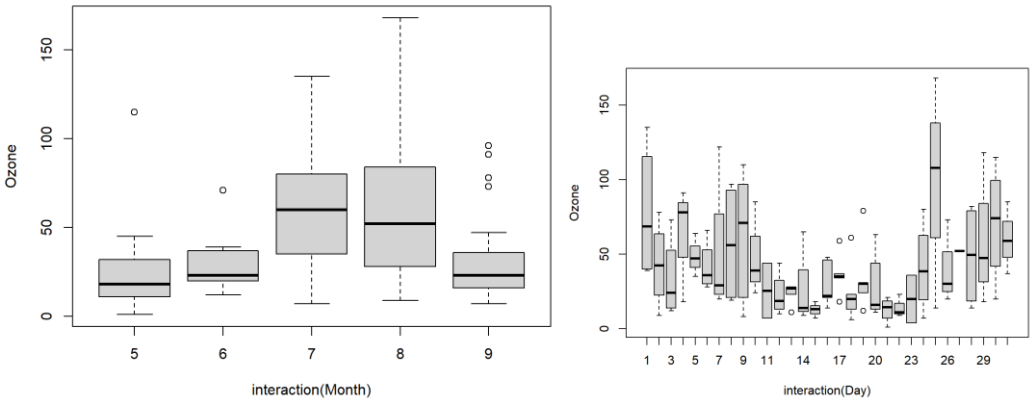


관계가 있다고 보이는 것은 wind와 temp, temp와 ozone, wind와 ozone 정도인 것 같다. 이 중 wind와 temp, temp와 ozone은 양의 상관관계를, wind와 ozone은 음의 상관관계를 갖는 것으로 보인다.

c)

```
help(boxplot)
boxplot(Ozone ~ interaction(Month), data = subset(airquality, !is.na(Ozone)))

boxplot(Ozone ~ interaction(Day), data = subset(airquality, !is.na(Ozone)))
```

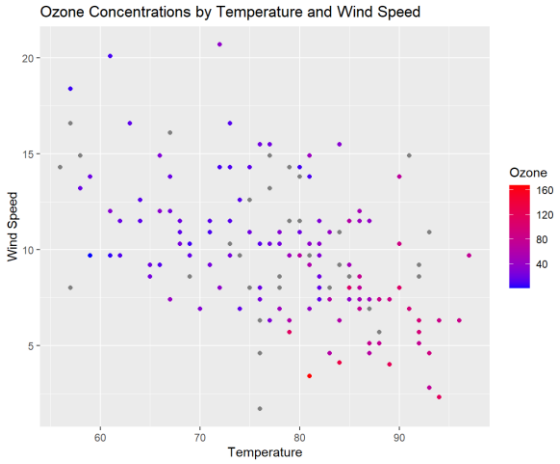


Month에 따른 boxplot을 보면 7,8월에는 ozone이 다소 높아짐을 볼 수 있다.

Day에 따라서는 월말과 월초에 다소 높은 것을 볼 수 있다.

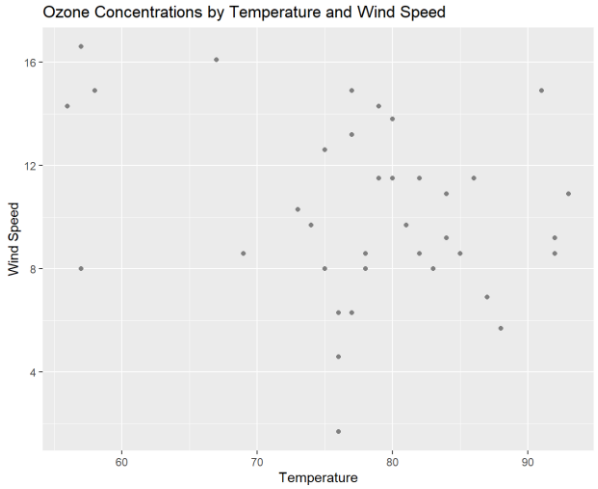
d)

```r
scatter_plot <- ggplot(airquality, aes(x = Temp, y = Wind, z = Ozone)) +
  geom_point(aes(color = Ozone)) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "Ozone Concentrations by Temperature and Wind Speed",
       x = "Temperature", y = "Wind Speed", z = "Ozone Concentration")
```



시각화를 통해 temperature가 높고 wind speed가 낮을수록 ozone의 농도가 높아짐을 확인할 수 있다.

e)



결측치만 따로 뽑아서 확인해본 결과, 온도가 70~90 사이에서 결측치가 많이 확인됐다.

	FALSE	TRUE
5	83.870968	16.129032
6	30.000000	70.000000
7	83.870968	16.129032
8	83.870968	16.129032
9	96.666667	3.333333

또한 6월에 유독 Ozone에 대한 결측치가 높았다. 기온과 바람에 많은 영향을 받는 오존의 특성, 그리고 오존의 관측치의 최대가 160 정도였다는 점에서, 아마도 결측치가 발생한 것은 너무 높아

서, 혹은 너무 낮아서 발생한 것이 아닐까 싶다.