Incomplete Data Analysis, 2020/2021
R and missing data and exploratory plots

Vanda Inacio

In `R` missing values are coded as `NA`, which stands for *not available*, and which should not be confunded with `NaN`, which stands for *not a number*. Out of curiosity, in the popular Pima Indians Diabetes dataset, available in several `R` packages, missing values (in variables as such body mass index and blood pressure) were coded originally with the numerical value zero! Ouch!

Let us start inspecting how some of the most used built-in `R` functions handle data with missing values (i.e., with `NA` values). For example, suppose that the vector y contains three numbers, e.g., 1, 3, and 5. If we are to compute the mean of y, we simply do

```
y <- c(1, 3, 5)
mean(y)
```

```
## [1] 3
```

Now, let us check what happens if the last number is missing.

```
y <- c(1, 3, NA)
mean(y)
```

```
## [1] NA
```

The mean is now undefined. If we look at the help of the function `mean`, by typing `help(mean)`, we notice the extra argument `na.rm`, a logical value indicating whether `NA` values should be removed before the computation proceeds. The default value is set to `FALSE`. If we instead set it to true, the missing values will be removed before the computation of the mean.

```
y <- c(1, 3, NA)
mean(y, na.rm = TRUE)
```

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

This makes possible to compute a result but, of course, the set os observations on which the computations are based has changed. I reiterate that disregarding the missing values may cause problems when conducting statistical inference.

Let us now investigate the popular `lm` function, where `lm` stands for linear model/modelling. Only for illustrative purposes, let us use the built-in dataset `airquality`, which contains 153 observations on 6 variables. Suppose we want to predict daily ozone concentration (measure in parts per billion) from wind speed (measured in miles per hour), that is, we will be considering the model

$$\text{Ozone}_i = \beta_0 + \beta_1 \text{Wind}_i + \varepsilon_i, \quad \varepsilon \overset{\text{iid}}{\sim} \text{N}(0, \sigma^2), \quad i = 1, \ldots, 153.$$

We start by investigating whether each of the variables has any missing values.

```
data(airquality)
names(airquality)
```

```
## [1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Month"   "Day"
```

```
sum(is.na(airquality$Ozone))
```

```
## [1] 37
```

```
sum(is.na(airquality$Wind))
```

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

The variable ozone has 37 missing values, whereas the variable wind does not have any missing values. We then fit the regression model

```
fit <- lm(Ozone ~ Wind, data = airquality)
fit
```

```
##
## Call:
## lm(formula = Ozone ~ Wind, data = airquality)
##
## Coefficients:
## (Intercept)         Wind
##      96.873       -5.551
```

We got no error message. This is because the `lm` function, unlike the `mean` function, automatically excludes missing values before fitting the model. We can check this by typing

```
deleted <- na.action(fit)
naprint(deleted)
```

```
## [1] "37 observations deleted due to missingness"
```

The main message from these two simple examples is that it is mandatory, before using any built-in function, to check how it handles missing values.

As for any analysis, the first thing to do with missing values is to do some exploratory analyses, mainly descriptive statistics, including visualisations. Some basic statistics can be obtained via the command `summary`

```
summary(airquality)
```

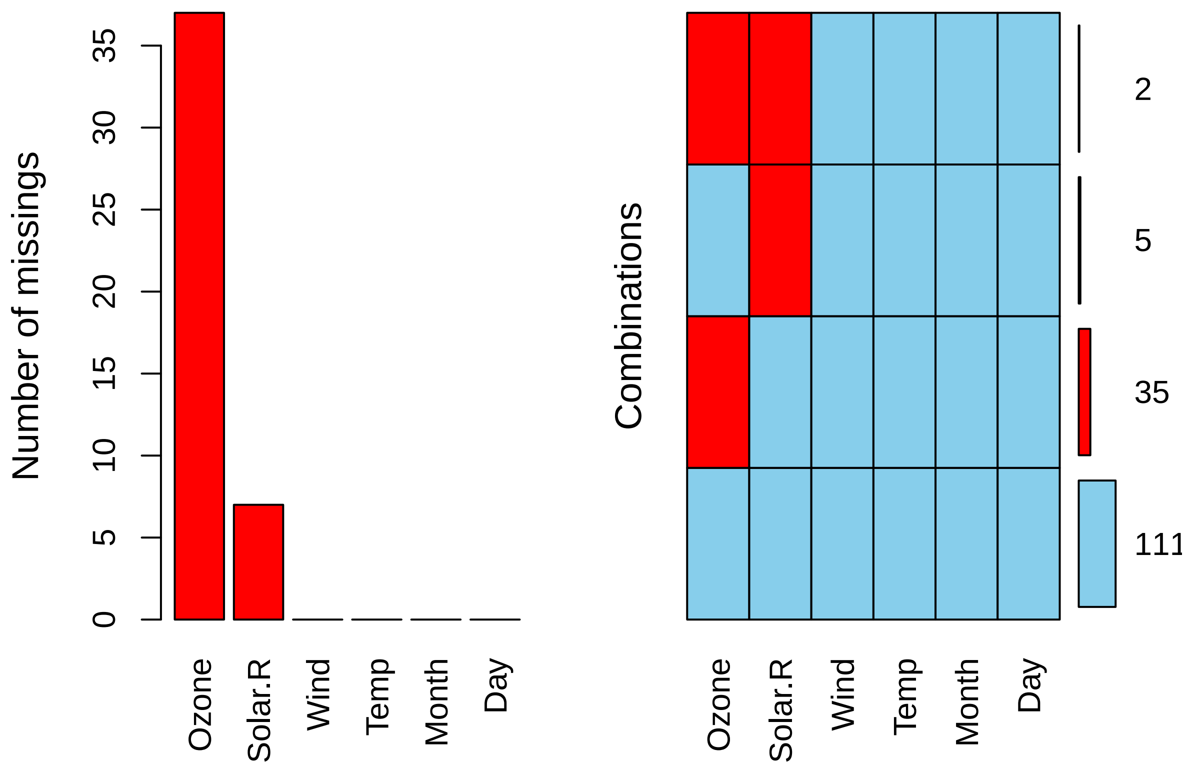
```
##      Ozone           Solar.R           Wind             Temp
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00
##  1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00
##  Median : 31.50   Median :205.0   Median : 9.700   Median :79.00
##  Mean   : 42.13   Mean   :185.9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

In particular `summary` gives us the number of missing values on each variable (we could have also used `summary` earlier when checking whether there were missing values for the variables ozone and wind). There are several packages that have very helpful functions to perform visualisations. In particular, I mention `VIM` and **naniar** (**mice** and **Amelia** have also a few functions for missing data exploration). First of all, we need to load the packages (if you have not the packages installed, you need to install them first).

```
require(VIM)
require(naniar)
require(mice)
require(Amelia)
```

The aggregation plot (function `aggr`) in the `VIM` package is a good starting point.

```
#from VIM
v <- aggr(airquality, plot = FALSE)
plot(v, prop = FALSE, numbers = TRUE)
```



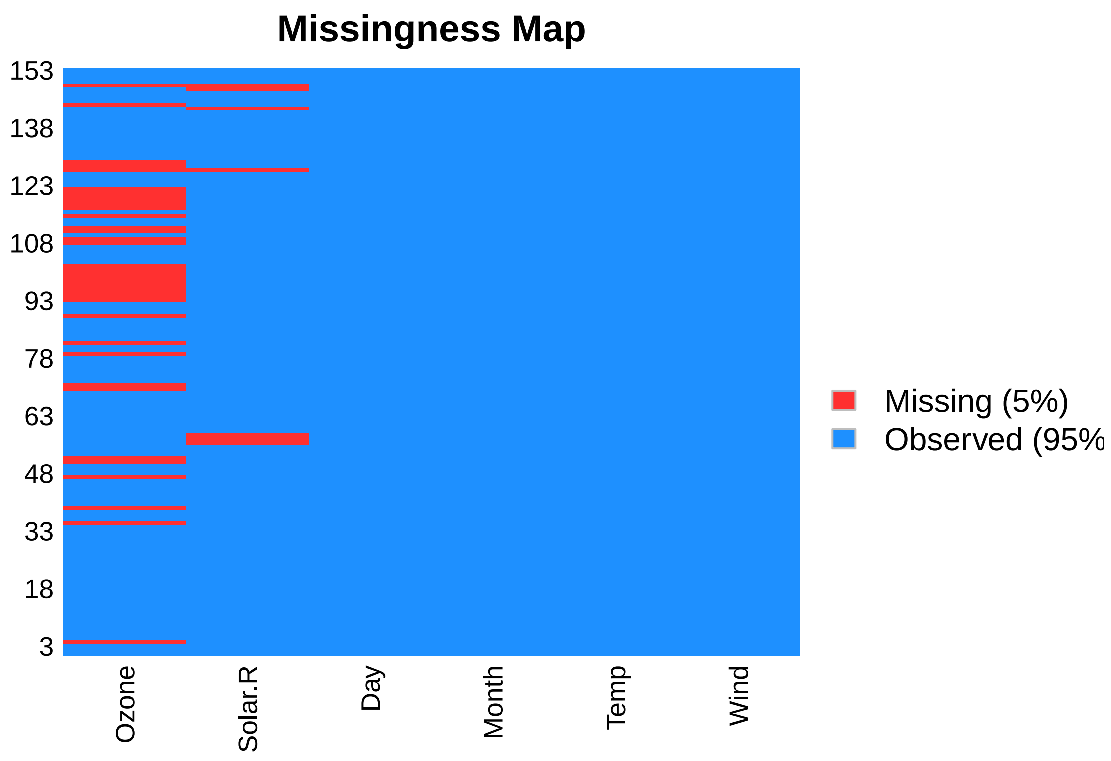
```
summary(v)
```

```
##
##  Missings per variable:
##  Variable Count
##     Ozone    37
##   Solar.R     7
##      Wind     0
##      Temp     0
##     Month     0
##       Day     0
##
##  Missings in combinations of variables:
##  Combinations Count   Percent
##   0:0:0:0:0:0   111 72.549020
##   0:1:0:0:0:0     5  3.267974
##   1:0:0:0:0:0    35 22.875817
##    1:1:0:0:0:0     2  1.307190
```

The plot in the left shows the number of missing values occurring in each variable, while the plot in the right shows the combinations of missing values in the different variables. For instance, we can see that there are 111 observations with complete data on all variables, 35 observations with missing values on the ozone variable, 5 observations with missing values in the solar radiation variable and 2 observations with missing values in both the ozone and solar radiation variable.

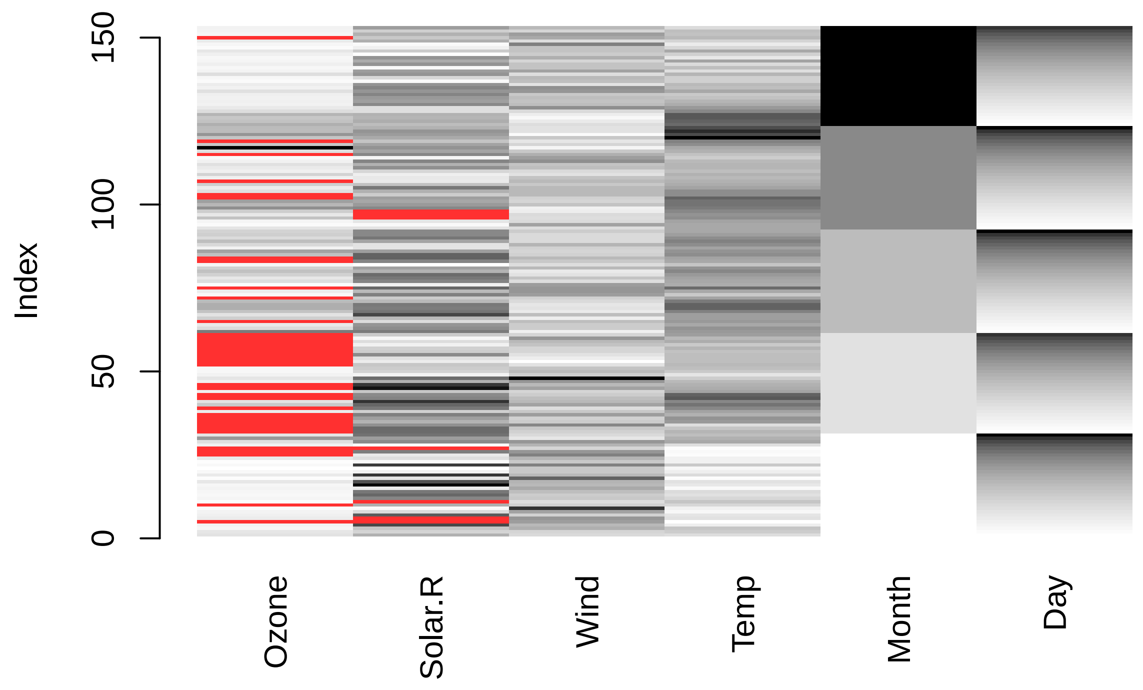
The function `missmap` in the package `Amelia` provides similar information, showing the variables against the observations, with a different color for when missing values occur.

```
#from Amelia
missmap(airquality, col = c("firebrick1", "dodgerblue"))
```



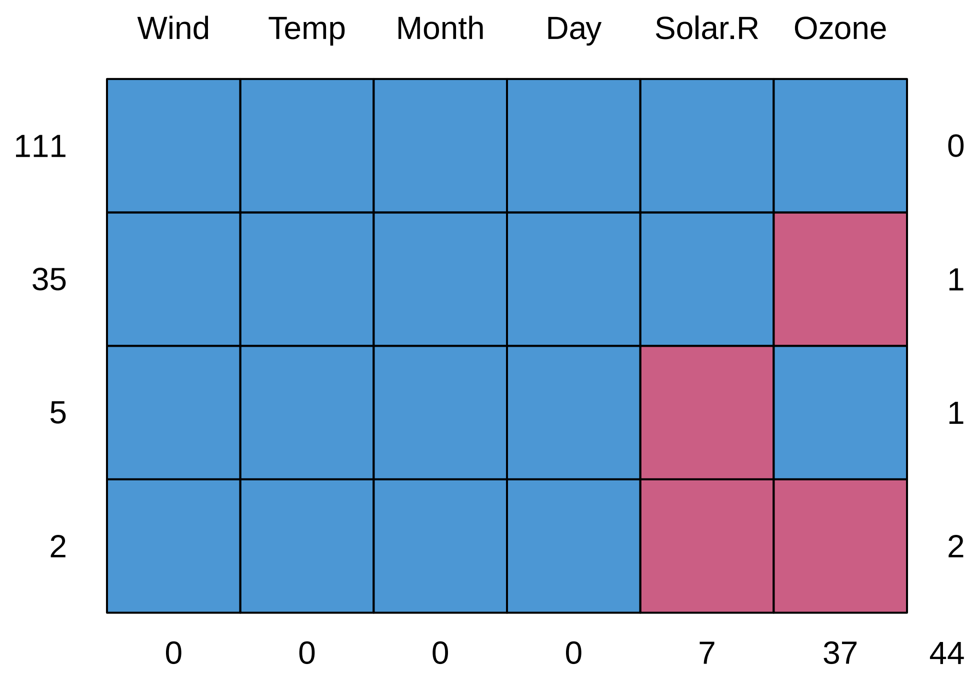
An almost similar plot can be obtained using the function `matrixplot` from the `VIM` package.

```
#from VIM
matrixplot(airquality, col = c("firebrick1", "dodgerblue"))
```



The package `mice`, which we will be using later in the course when learning about multiple imputation contains the function `md.pattern` which provides insight about the pattern of missing values (the same can be obtained with the `aggr` function that we have illustrated before and which I think is more helpful).

```
#from mice
md.pattern(airquality)
```



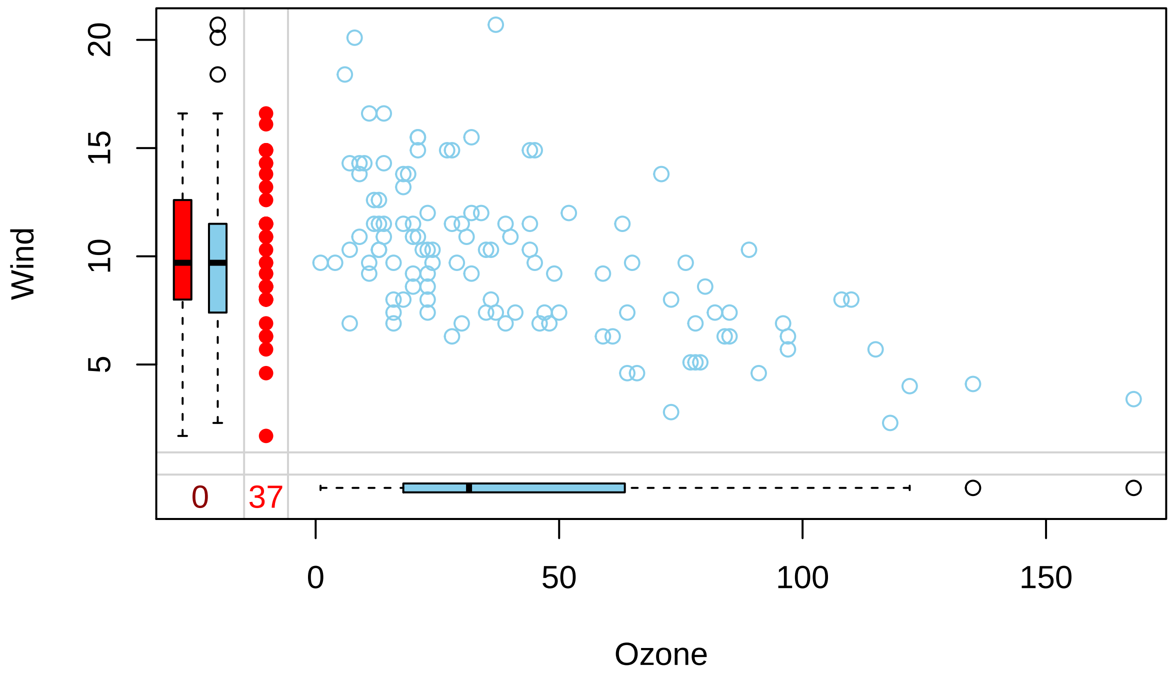
```
##     Wind Temp Month Day Solar.R Ozone
## 111    1    1     1   1       1     1  0
## 35     1    1     1   1       1     0  1
## 5      1    1     1   1       0     1  1
## 2      1    1     1   1       0     0  2
##          0     0      0    0        7     37 44
```

The function `marginplot` in `VIM` displays, in addition to a standard scatterplot, information about missing in the plot margins.

```
#from VIM
names(airquality)
```

```
## [1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Month"   "Day"
```

```
marginplot(airquality[c(1,3)])
```

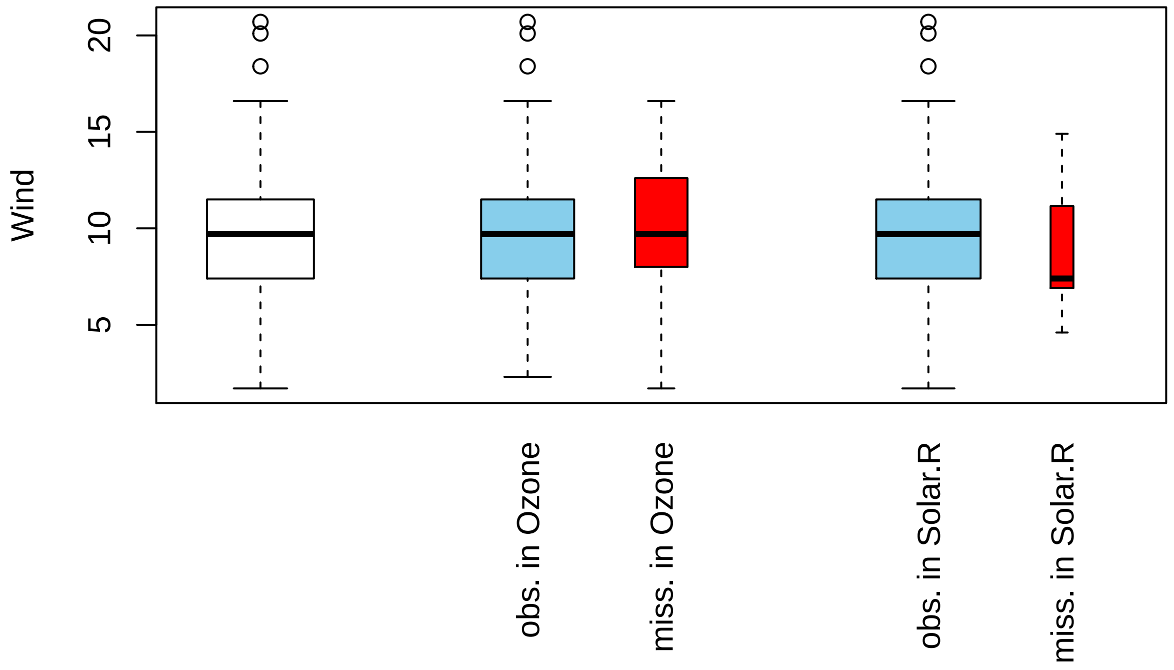


The blue points in the 'data area' correspond to observations for which both ozone and wind were observed. The red dots in the left margin correspond to the records for which wind is observed and ozone is missing. The points are drawn at the known values of wind. The bottom horizontal margin does not contain any red points because wind is fully observed. Furthermore, the left margin contains two boxplots, a blue and a red one. The blue boxplot in the left margin summarises the marginal distribution of wind observtions for which the corresponding ozone value is observed, while the red boxplot summarises the marginal distribution of wind observations for which the corresponding ozone value is missing. Rememeber that if missing in ozone is MCAR these two distributions are expected to be identical. Obviously here we are constrained at plotting two variables at a time only, but nevertheless we can gather some interesting insights.

The function `pbox`(from `VIM`) consists of several boxplots as shown below.

```
#from VIM
pbox(airquality, pos = 3)
```

```
## Warning in createPlot(main, sub, xlab, ylab, labels, ca$at): not enough space to
## display frequencies
```

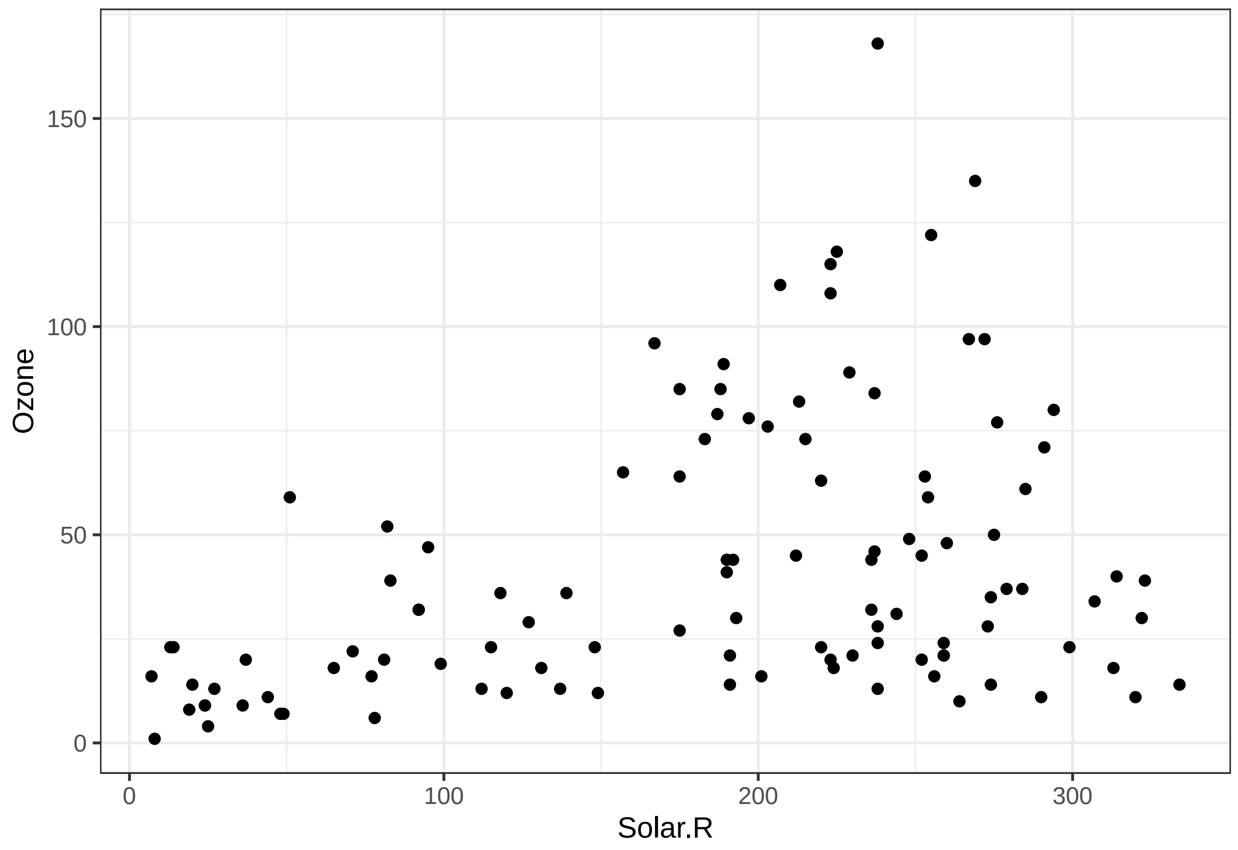


This plot consists of several boxplots. First, a standard boxplot of the variable of interest, in this case and only for illustratrive examples, wind (column 3 in the dataset `airquality`) is produced. Second, boxplots grouped by observed and missing values, induced by the two variables with missing values (ozone and solar radiation), in this variable are shown. This is extremely useful when trying to check MCAR against MAR.

Finally, the package `naniar` has a variety of different plots to explore missing data. It is based on `ggplot2` and tidy data principles. This is not required in the course, but I am illustrating here some of the plots, so you know they exist! First, as a matter of caution, note that plain `ggplot` plots do not handle missing values by default and automatically remove them, as illustrated below.

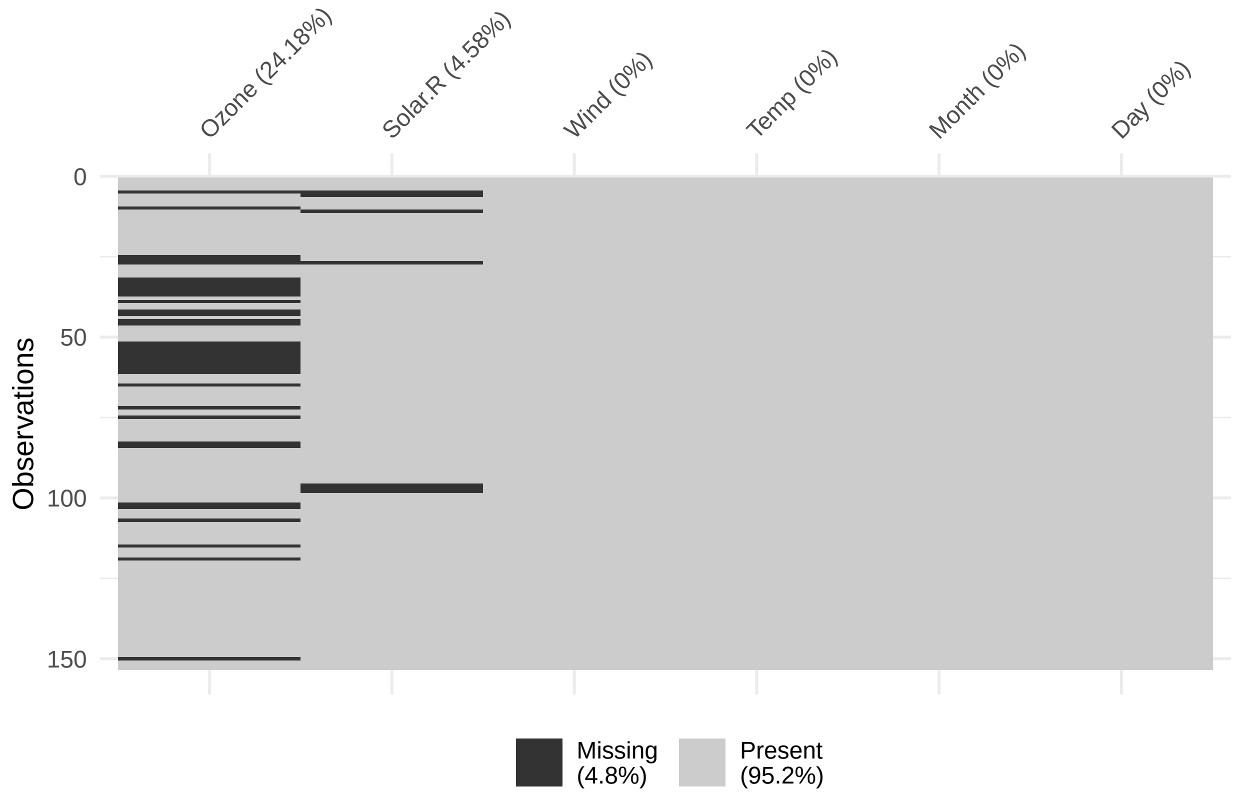
```r
require(ggplot2)
ggplot(airquality,
       aes(x = Solar.R,
           y = Ozone)) +
  geom_point() +
  theme_bw()
```

```
## Warning: Removed 42 rows containing missing values (geom_point).
```

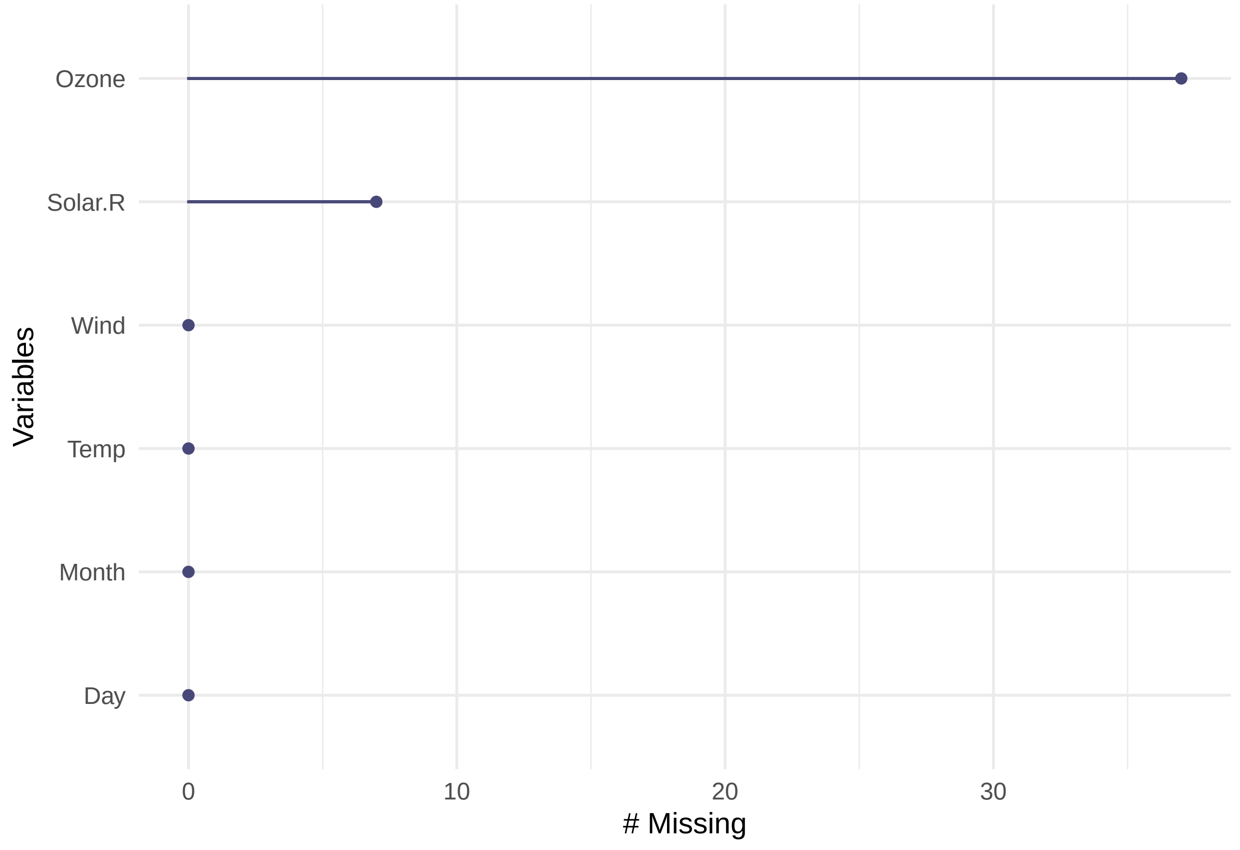


There are several functions in `naniar` providing similar information to those from `VIM`.

```r
vis_miss(airquality)
```

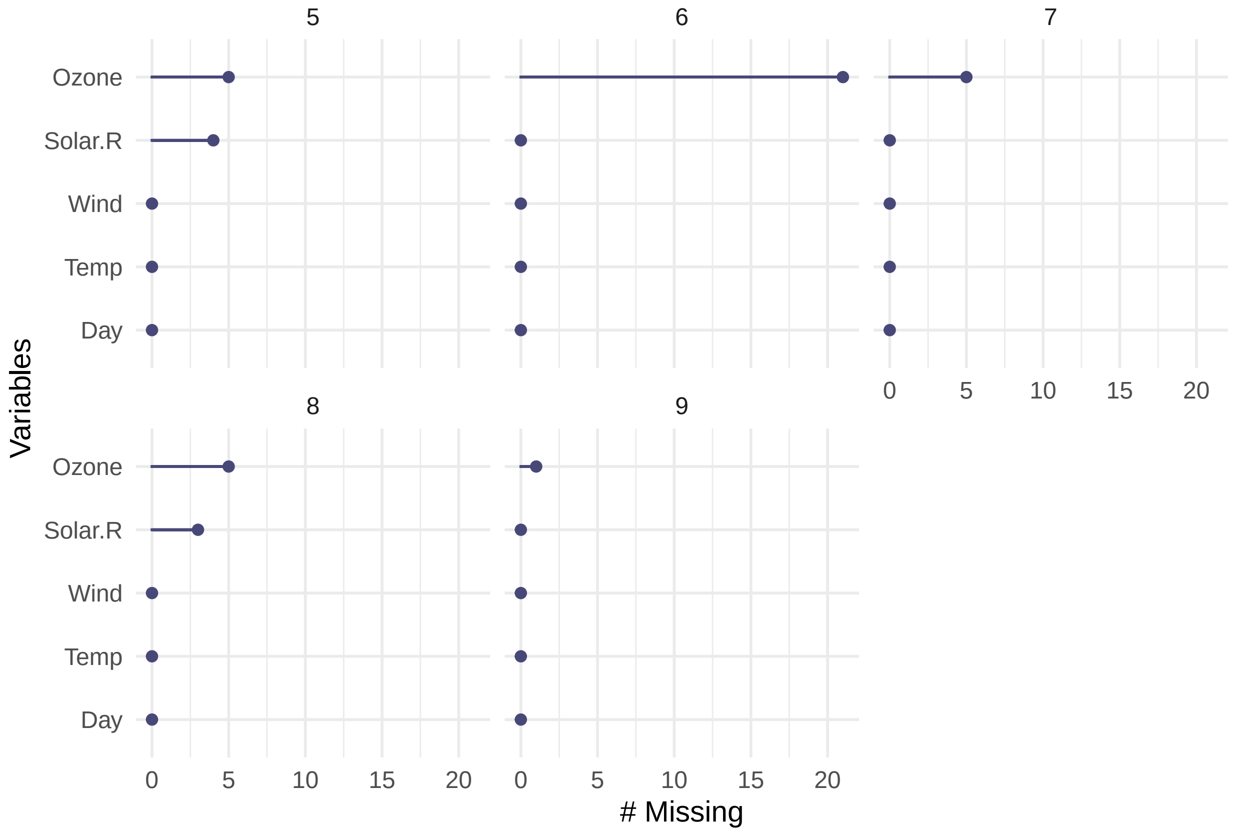


```
gg_miss_var(airquality)
```



The package has also a function that allows to look at missingness when stratifying by another variable, as illustrated in the plot below

```r
gg_miss_var(airquality, facet = Month)
```



The supporting articles for the `VIM` and `naniar` packages are on Learn. There is also a vignette, available in the link below, illustrating several of the plots that can be obtained with the `naniar` package.

`https://naniar.njtierney.com/articles/naniar-visualisation.html`