

Customer dropout membership*

Pedro Sobreiro, Javier Berrocal, Domingos Martinho, José Garcia Alonso

Extremadura University

3 July, 2021

Abstract

Abstract of the article.

Introduction

Research idea:

-

Context: An organization membership located in portugal. The organization offers an annual membership for the members, the service subscription has several payment options:

- Men with a annual fee of 10€
- Women annual fee of 6€
- Correspondent fee 6€
- Retired fee 5€
- Student fee 2.5€
- under-14 fee 1€

*Corresponding address: sobreiro@esdrm.ipsantarem.pt. The current template adapts part of the Rmd code by [Paul C. Bauer](#), Mannheim Centre for European Social Research.

```

library(dplyr)
library(dlookr)
library(ggplot2)

##eda_report(nlswork,output_dir = "C:/Users/mangelo.EEG/Documents/GitHub/prjs/reports/

## The data

names(airquality)

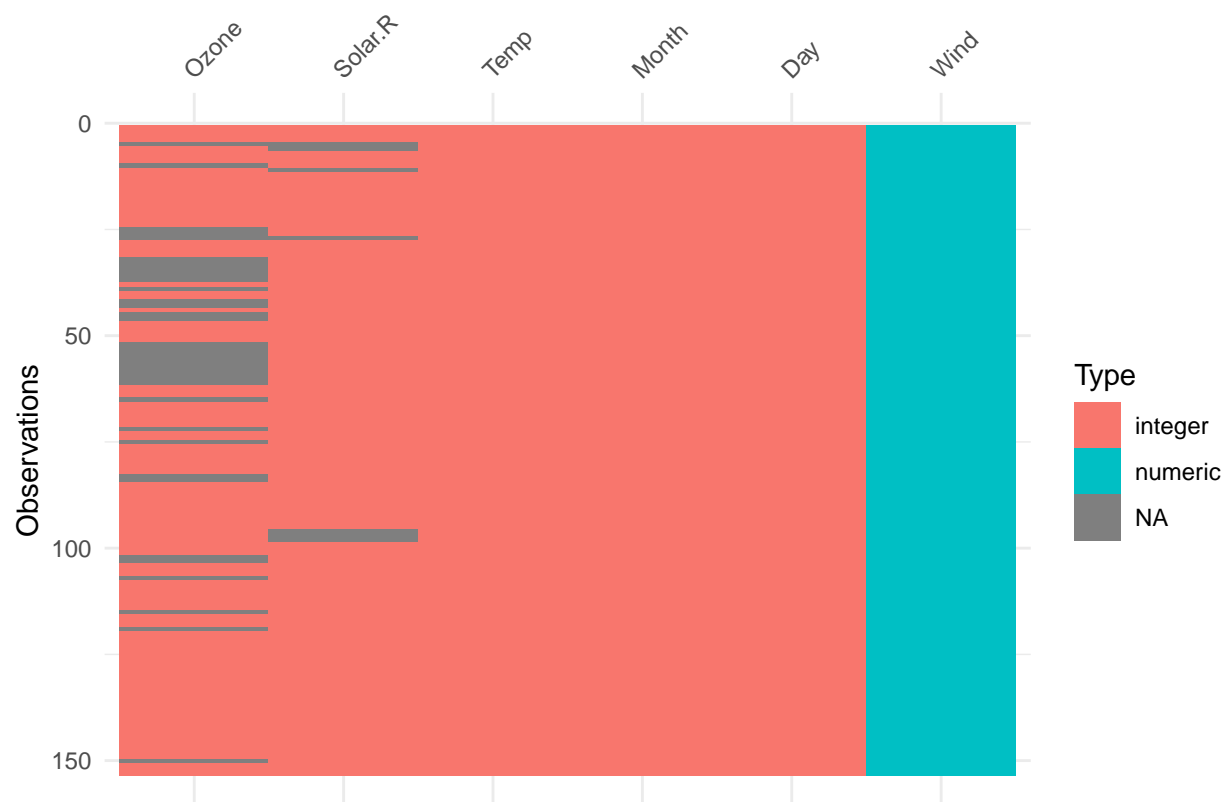
## [1] "Ozone"  "Solar.R" "Wind"    "Temp"    "Month"    "Day"

##summary(nlswork)

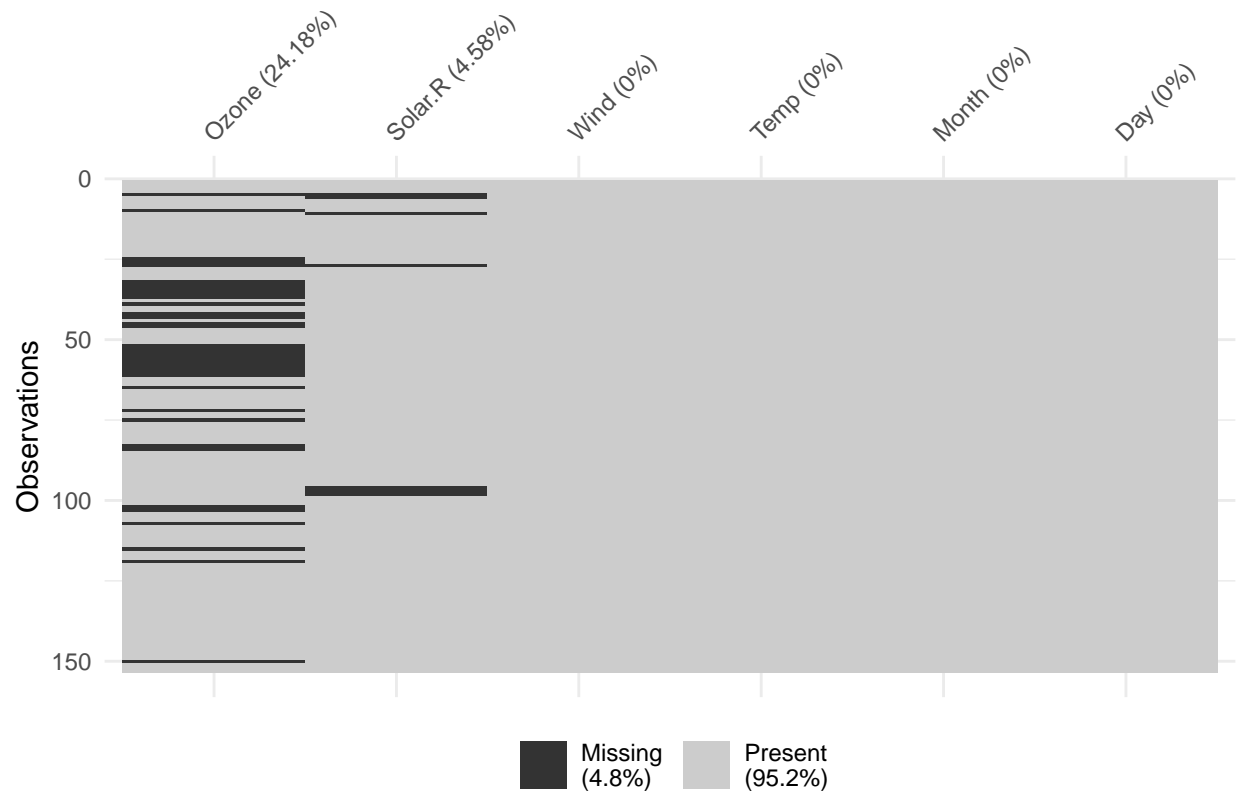
## Missing values

library(visdat)
vis_dat(airquality)

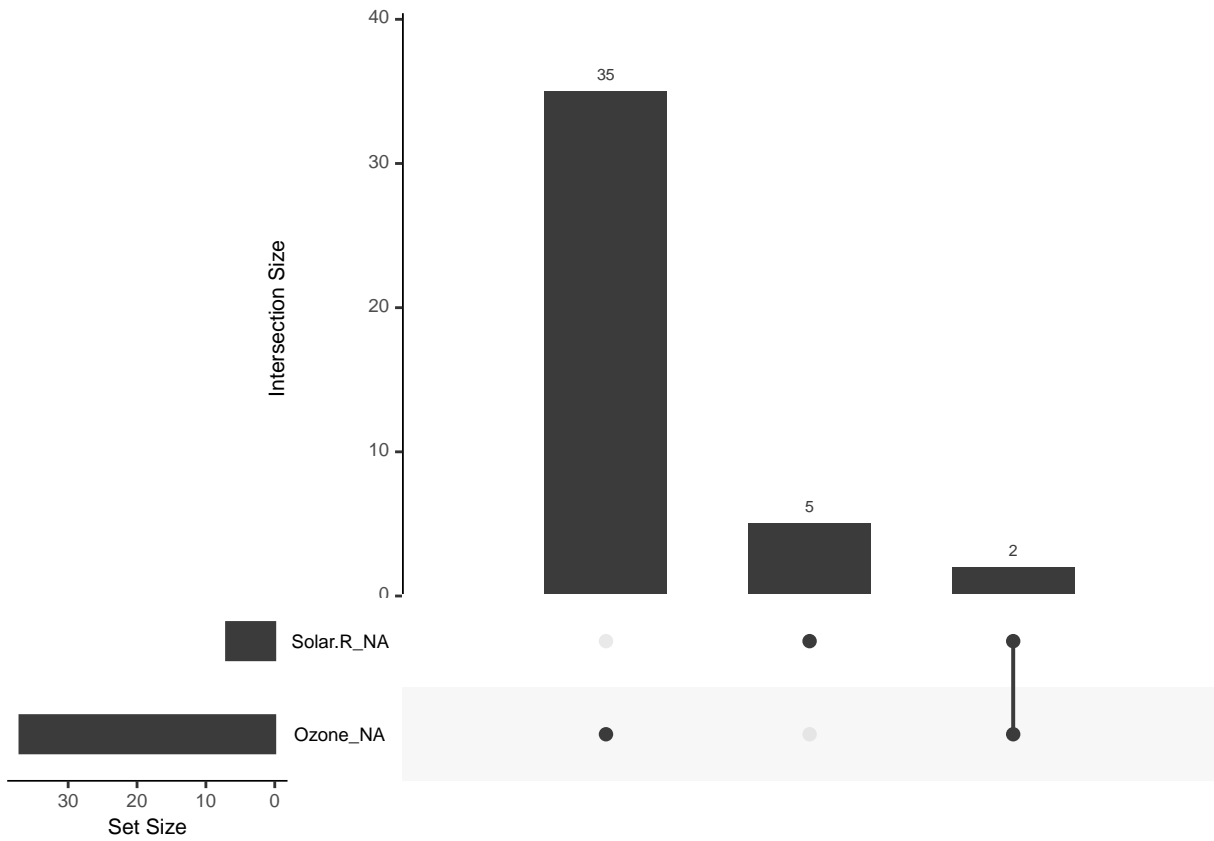
```



```
library(naniar)
vis_miss(airquality)
```



```
gg_miss_upset(airquality)
```



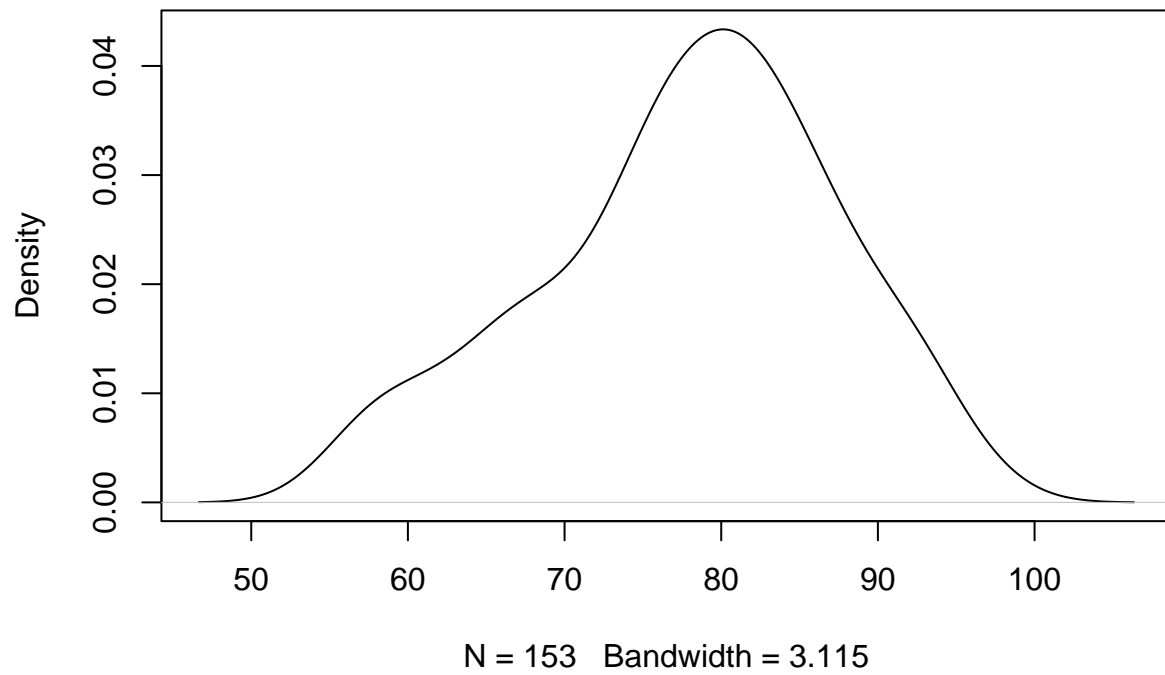
```
## GRAPHS
```

```
dplyr::glimpse(cars$Ozone)
```

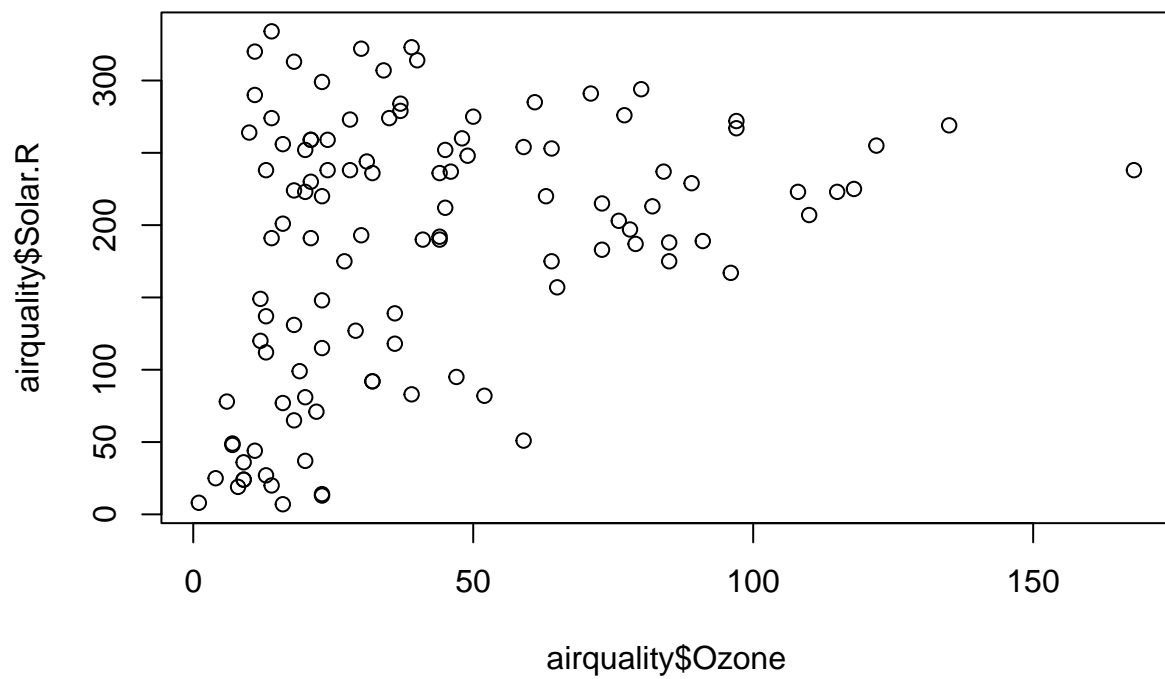
```
## NULL
```

```
d <- density(airquality$Temp)
plot(d)
```

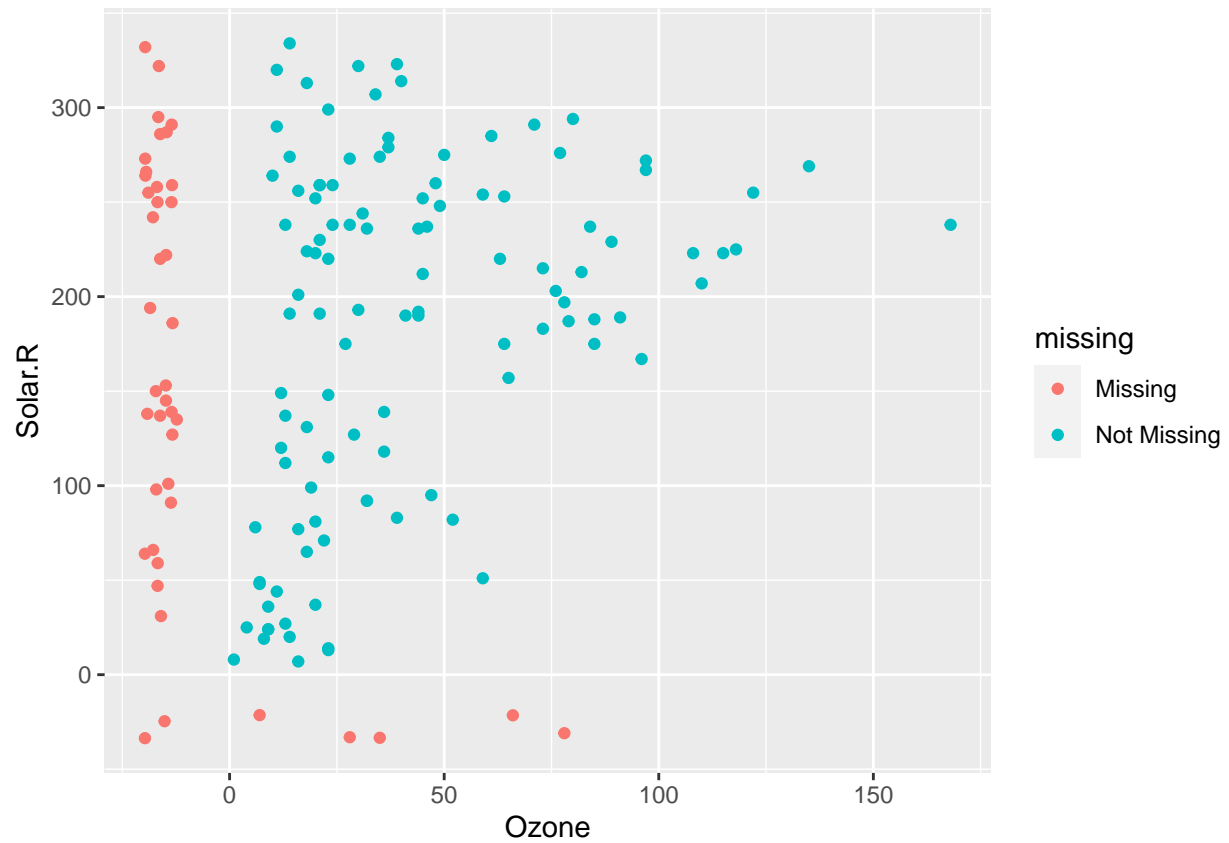
density.default(x = airquality\$Temp)



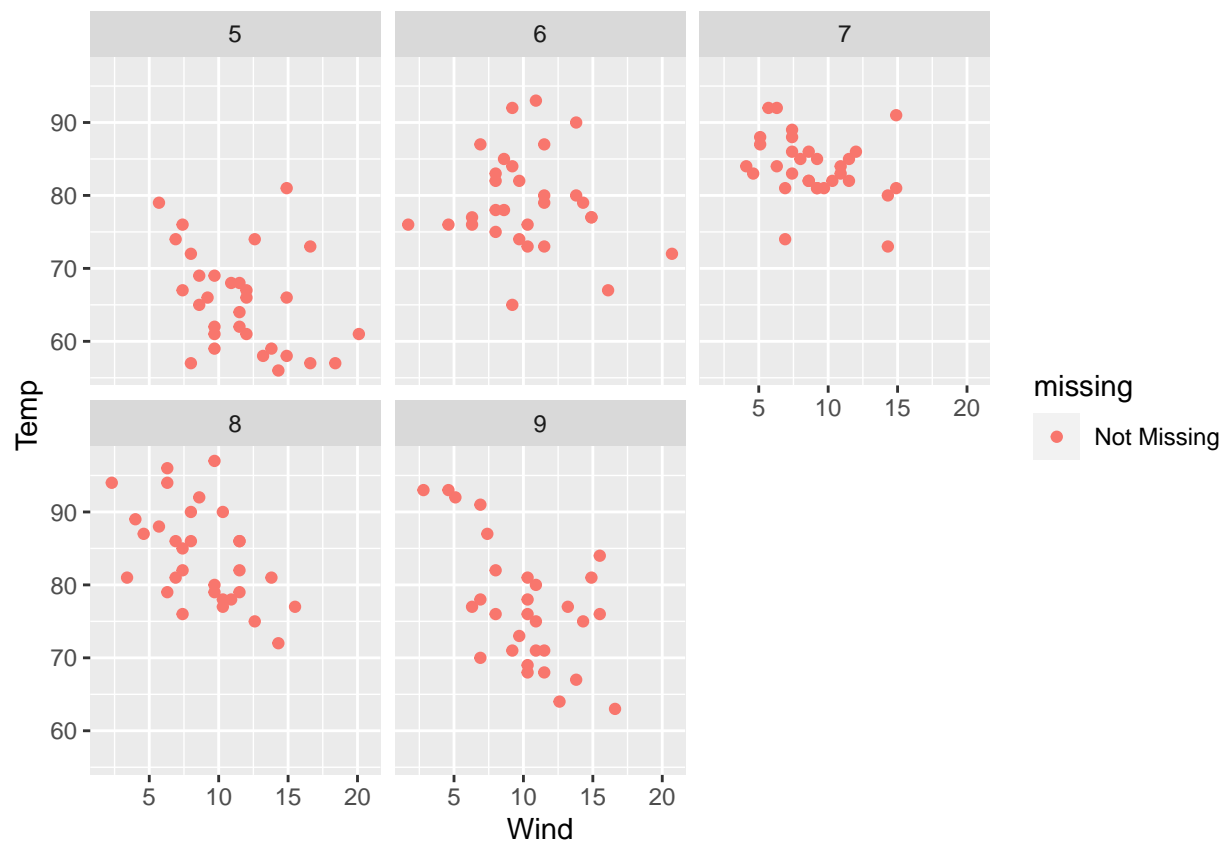
```
plot(airquality$Ozone, airquality$Solar.R)
```



```
ggplot(airquality, aes(x = Ozone, y = Solar.R)) +  
geom_miss_point()
```



```
ggplot(airquality, aes(x = Wind, y = Temp)) +  
  geom_miss_point() +  
  facet_wrap(vars(Month))
```



```
stats <- summary(airquality$Temp)
stats
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      56.00  72.00   79.00   77.88  85.00   97.00
```

```
describe(airquality)
```

```
## # A tibble: 6 x 26
##   variable      n    na  mean    sd se_mean  IQR skewness kurtosis  p00
##   <chr>    <int> <int> <dbl> <dbl>  <dbl> <dbl>    <dbl>    <dbl> <dbl>
## 1 Ozone      116    37  42.1  33.0   3.06  45.2   1.24      1.29    1
## 2 Solar.R    146     7 186.   90.1   7.45  143   -0.428   -0.968    7
## 3 Wind       153     0   9.96  3.52   0.285  4.1   0.348    0.111   1.7
## 4 Temp       153     0  77.9   9.47   0.765  13   -0.378   -0.404   56
## 5 Month      153     0   6.99  1.42   0.115  2   -0.00239 -1.30     5
## 6 Day        153     0  15.8   8.86   0.717  15    0.00265 -1.20     1
## # ... with 16 more variables: p01 <dbl>, p05 <dbl>, p10 <dbl>, p20 <dbl>,
## #   p25 <dbl>, p30 <dbl>, p40 <dbl>, p50 <dbl>, p60 <dbl>, p70 <dbl>,
## #   p75 <dbl>, p80 <dbl>, p90 <dbl>, p95 <dbl>, p99 <dbl>, p100 <dbl>
```


The average age in our data is 77.9.

Tables

R Markdown PDF is now able to produce good tables with our output. For **stargazer** the label is contained in the function, while for **kable** it's contained in the chunk name.

stargazer(): Summary and regression tables

Table @ref(tab1) shows data's summary statistics.¹ **stargazer()** is an excellent solution to export outputs.

```
library(stargazer)
stargazer(cars,
  title = "Summary table with stargazer",
  label="tab1cars",
  table.placement = "h",
  header=FALSE)
```

Table 1: Summary table with stargazer

| Statistic | N | Mean | St. Dev. | Min | Pctl(25) | Pctl(75) | Max |
|-----------|----|--------|----------|-----|----------|----------|-----|
| speed | 50 | 15.400 | 5.288 | 4 | 12 | 19 | 25 |
| dist | 50 | 42.980 | 25.769 | 2 | 26 | 56 | 120 |

Table @ref(tab2) reports regression outputs. Name the models as you can refer to their names in the text (M1, M2, M3).

```
library(stargazer)
model1 <- lm(speed ~ dist, data = cars)
model2 <- lm(speed ~ dist, data = cars)
model3 <- lm(dist ~ speed, data = cars)
stargazer(model1, model2, model3,
  title = "Regression table with stargazer",
  label="tab2",
  table.placement = "h",
  column.labels = c("M1", "M2", "M3"),
  model.numbers = FALSE,
  header=FALSE)
```

¹You can reference the table as @ref(tab1cars).

Table 2: Regression table with stargazer

| | <i>Dependent variable:</i> | | |
|-------------------------------|----------------------------|---------------------|----------------------|
| | speed M1 | dist M2 | dist M3 |
| dist | 0.166*** (0.017) | 0.166*** (0.017) | |
| speed | | | 3.932*** (0.416) |
| Constant | 8.284*** (0.874) | 8.284*** (0.874) | -17.579** (6.758) |
| Observations | 50 | 50 | 50 |
| R ² | 0.651 | 0.651 | 0.651 |
| Adjusted R ² | 0.644 | 0.644 | 0.644 |
| Residual Std. Error (df = 48) | 3.156 | 3.156 | 15.380 |
| F Statistic (df = 1; 48) | 89.567*** | 89.567*** | 89.567*** |

Note:

*p<0.1; **p<0.05; ***p<0.01

Figures

Graphs with R

You can insert figures like this. One would like to produce and insert them on the fly in the .rmd file. Figure @ref(fig:fig-1) is such an example.

```
plot(cars$speed, cars$dist)
```

However, in some cases it does not work.

Example: ggplot2 graphs

See the ggplot2 output reported in Figure @ref(fig:fig-2).

```
mtcars$cyl <- as.factor(mtcars$cyl) # Convert cyl to factor
library(ggplot2)
ggplot(mtcars, aes(x=wt, y=mpg, shape=cyl)) + geom_point() +
  labs(x="Weight (lb/1000)", y = "Miles/(US) gallon",
       shape="Number of \n Cylinders") + theme_classic()
```

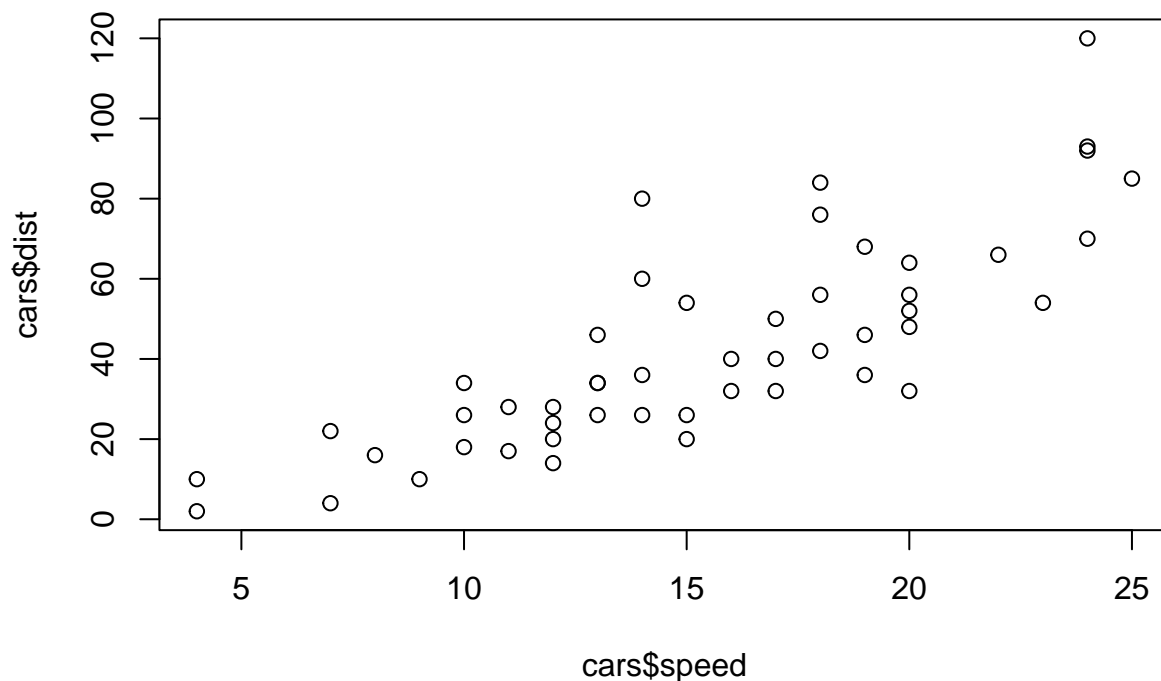


Figure 1: Scatterplot of Speed and Distance

Another example using Plotly

With Plotly we can produce interactive graphs which play well, for example, once can embed in html webpages (drop by [here](#) for an example). One can insert this type of graphs in R Markdown PDF using `Orca` (it generates static images from Plotly graphs). Go [here](#) to check how to install it. See Figure @ref(fig:fig-3) for an example.

```
library(plotly)
p <- plot_ly(cars, type = "scatter", mode="markers",
             x=~speed,
             y=~dist)
#Sinc setenv('MAPRNY_TOKEN' = '121231231') # set arbitrary token
# Lets create a value for example

media <- mean(cars$speed)
```

The criminal rate is 15.4%.

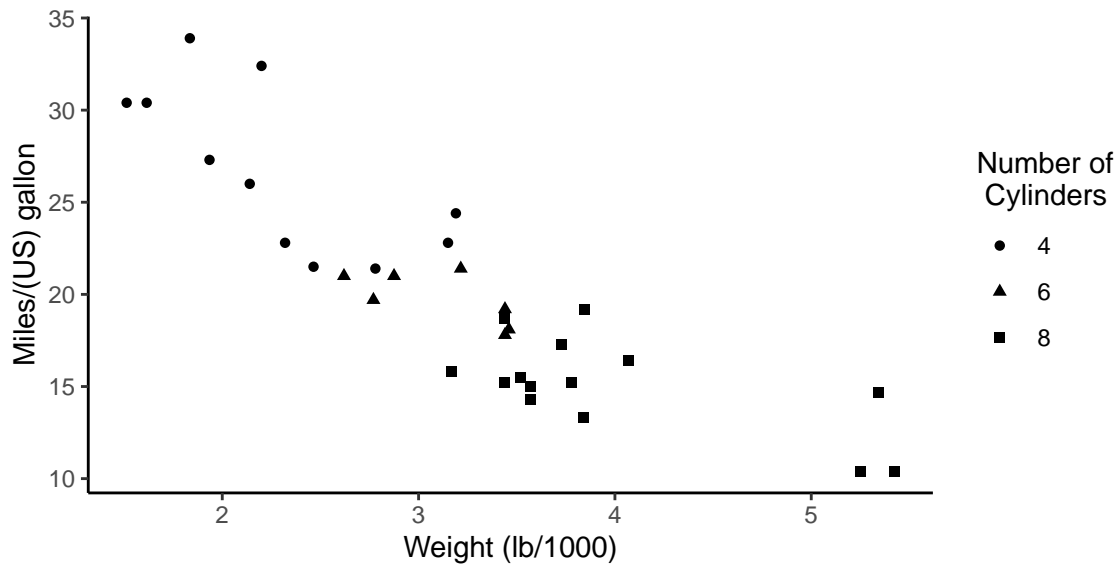


Figure 2: Miles per gallon according to the weight

Miguel's tests

R

Example of an equation

$$\int_0^{2\pi} \sin x \, dx$$

Example of a matrix

$$\mathbf{X} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

or

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (1)$$

See Equation @ref(eq:binom).

$$y_{ijt} = \beta x_{ijt} + \eta_i + \gamma_j + \lambda_t + \varepsilon_{ijt} \quad (2)$$

```
library(stargazer)
stargazer(cars,
  title = "Summary 24",
  label="tab24",
  table.placement = "ht",
  header=FALSE)
```

Table 3: Summary 24

| Statistic | N | Mean | St. Dev. | Min | Pctl(25) | Pctl(75) | Max |
|-----------|----|--------|----------|-----|----------|----------|-----|
| speed | 50 | 15.400 | 5.288 | 4 | 12 | 19 | 25 |
| dist | 50 | 42.980 | 25.769 | 2 | 26 | 56 | 120 |

Final remarks

Check the replication package for Bonhomme, Lamadon and Manresa (2019): <https://github.com/tlamadon/blm-replicate>

References

Appendix: Chunk options

Software versioning

R

```
cat(paste("#", capture.output(sessionInfo()), "\n", collapse = ""))

## # R version 4.1.0 (2021-05-18)
## # Platform: x86_64-pc-linux-gnu (64-bit)
## # Running under: Ubuntu 20.04.2 LTS
## #
## # Matrix products: default
## # BLAS: /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## # LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3
## #
## # locale:
## #  [1] LC_CTYPE=C.UTF-8      LC_NUMERIC=C          LC_TIME=C.UTF-8
## #  [4] LC_COLLATE=C.UTF-8    LC_MONETARY=C.UTF-8   LC_MESSAGES=C.UTF-8
## #  [7] LC_PAPER=C.UTF-8      LC_NAME=C             LC_ADDRESS=C
## # [10] LC_TELEPHONE=C        LC_MEASUREMENT=C.UTF-8 LC_IDENTIFICATION=C
## #
## # attached base packages:
## # [1] stats      graphics  grDevices  utils      datasets  methods    base
## #
## # other attached packages:
## # [1] plotly_4.9.4.1  stargazer_5.2.2  nanianr_0.6.1    visdat_0.5.3
## # [5] ggplot2_3.3.5   dlookr_0.4.5     dplyr_1.0.7      rmarkdown_2.9
## # [9] nvimcom_0.9-115
## #
## # loaded via a namespace (and not attached):
## #  [1] webshot_0.5.2      RColorBrewer_1.1-2  http_1.4.2
## #  [4] UpSetR_1.4.0        tools_4.1.0         backports_1.2.1
## #  [7] utf8_1.2.1          R6_2.5.0            rpart_4.1-15
## # [10] lazyeval_0.2.2      Hmisc_4.5-0         nortest_1.0-4
## # [13] DBI_1.1.1           colorspace_2.0-2    nnet_7.3-16
## # [16] withr_2.4.2         tidyselect_1.1.1    gridExtra_2.3
## # [19] curl_4.3.2          compiler_4.1.0      extrafontdb_1.0
## # [22] cli_3.0.0           rvest_1.0.0         htmlTable_2.2.1
```

```
## # [25] xml2_1.3.2          sandwich_3.0-1      labeling_0.4.2
## # [28] scales_1.1.1        checkmate_2.0.0    mvtnorm_1.1-2
## # [31] proxy_0.4-26        RcmdrMisc_2.7-1    systemfonts_1.0.2
## # [34] stringr_1.4.0       digest_0.6.27      foreign_0.8-81
## # [37] svglite_2.0.0       rio_0.5.27         base64enc_0.1-3
## # [40] jpeg_0.1-8.1        pkgconfig_2.0.3    htmltools_0.5.1.1
## # [43] extrafont_0.17      highr_0.9          htmlwidgets_1.5.3
## # [46] rlang_0.4.11        readxl_1.3.1       rstudioapi_0.13
## # [49] prettydoc_0.4.1     farver_2.1.0       generics_0.1.0
## # [52] jsonlite_1.7.2      zoo_1.8-9          crosstalk_1.1.1
## # [55] zip_2.2.0           car_3.0-11         magrittr_2.0.1
## # [58] kableExtra_1.3.4    Formula_1.2-4      Matrix_1.3-4
## # [61] Rcpp_1.0.6          munsell_0.5.0      fansi_0.5.0
## # [64] abind_1.4-5         gdtools_0.2.3      partykit_1.2-13
## # [67] lifecycle_1.0.0     stringi_1.6.2      yaml_2.2.1
## # [70] inum_1.0-4          carData_3.0-4      MASS_7.3-54
## # [73] plyr_1.8.6          grid_4.1.0         hrbrthemes_0.8.0
## # [76] forcats_0.5.1       crayon_1.4.1       lattice_0.20-44
## # [79] haven_2.4.1         splines_4.1.0      hms_1.1.0
## # [82] knitr_1.33          pillar_1.6.1       glue_1.4.2
## # [85] evaluate_0.14       latticeExtra_0.6-29 data.table_1.14.0
## # [88] png_0.1-7           vctrs_0.3.8        Rttf2pt1_1.3.8
## # [91] cellranger_1.1.0    tidyr_1.1.3        gtable_0.3.0
## # [94] purrr_0.3.4         assertthat_0.2.1   xfun_0.24
## # [97] openxlsx_4.2.4      libcoin_1.0-8      e1071_1.7-7
## # [100] class_7.3-19        survival_3.2-11    viridisLite_0.4.0
## # [103] tibble_3.1.2        tinytex_0.32       cluster_2.1.2
## # [106] corrplot_0.90       ellipsis_0.3.2
```

```
# or use message() instead of cat()
```

Figure 3: Example: export a Plotly figure using ‘orca’

