Práctica 2: Limpieza y validación de los datos

Pablo A. Delgado

6 de June, 2021

Contents

Detalles de la actividad

1. Descripción

En esta actividad se elabora un caso práctico, consistente en el tratamiento de un conjunto de datos, orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas.

2. Objetivos

Los objetivos concretos de esta práctica son:

- Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares.
- Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.
- Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos.
- Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.
- Actuar con los principios éticos y legales relacionados con la manipulación de datos en función del ámbito de aplicación.
- Desarrollar las habilidades de aprendizaje que les permitan continuar estudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo.
- Desarrollar la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

3. Competencias

En esta práctica se desarrollan las siguientes competencias del Máster de Data Science:

Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.

Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para su posterior análisis.

Resolucion

1. Descripcion del Dataset

Dado que actualmente la demanda de las empresas para atender los diferentes objetivos que se le plantean son cada vez mas criticos y contar con todo el personal es clave para resolverlos o llevarlos a cabo, ideal seria poder preveer cualquier situacion futura que impida contar con todos los recursos y poder gestionar los proyectos y evitar riesgos de entrega, implementacion o puesta en produccion de los mismos.

Es por eso que se desea a traves del dataset disponible descripto mas abajo poder determinar o predecir el ausentismo de los recursos humanos de una compania.

Incluso, bajando mas de nivel dando una ejemplo mas concreto, si consideramos las metodologias agiles, donde de antemano se prevee la capacidad con la que contara el equipo en el proximo sprint, que mejor que tener como ayuda para un scrum master una prevision de ausencias segun reglas prestablecidas de acuerdo al comportamiento general de un miembro del equipo?

Podriamos entender por ejemplos: porque se dan las ausencias? por temas personales? segun en que estacion del año estamos? podemos segmentar estos analisis segun las caracteristicas o habitos de ciertos grupos de empleados?

Pensemoslo no solo para establecer la capacidad del equipo, imaginemos si conocer estos posibles patrones de comportamiento podrian ayudar a RRHH a mejorar la seleccion de proximos candidatos a un puesto? o mismo la contratacion de personal temporal segun la demanda estacional de proyectos dentro de una empresa y contrastandolo con la cantidad posibles de horas de aunsencia del personal actual.

Luego intentaremos buscar a partir de los datos patrones que nos permitan tener mas informacion acerca del comportamiento de los empleados que nos den mas informacion de posibles cuestiones a tener en cuenta o patrones, como dijimos, en el comportamiento a la hora de ausentarse o la relacion que hay con la cantidad de horas en las que se ausentan.

Para ello con los algoritmos de regresion, con todo lo analizado previamente buscaremos predecir el comportamiento futuro del empleado o las horas que podran llegar a ausentarse en el futuro.

Los datos que utilizaremos para esta tarea sera el obtenido de descargado del UC Irvine Machine Learning Repository:

https://archive.ics.uci.edu/ml/datasets/Absenteeism+at+work

Abstract: The database was created with records of absenteeism at work from July 2007 to July 2010 at a courier company in Brazil.

Data Set Characteristics:	Multivariate, Time-Series	**Number of Instances:**	740	**Area:**	Business
Attribute Characteristics:	Integer, Real	**Number of Attributes:**	21	**Date Donated**	2018-04-05
Associated Tasks:	Classification, Clustering	**Missing Values?**	N/A	**Number of Web Hits:**	190993

Figure 1: DataSetDescription

Tal como se comenta en su descripcion este dataset permite realizar tareas de clasificacion y clustering.

- Data Set Information: The data set allows for several new combinations of attributes and attribute exclusions, or the modification of the attribute type (categorical, integer, or real) depending on the purpose of the research.The data set (Absenteeism at work - Part I) was used in academic research at the Universidade Nove de Julho - Postgraduate Program in Informatics and Knowledge Management.
- Attribute Information:

```
1. Individual identification (ID)
2. Reason for absence (ICD).
3. Month of absence
4. Day of the week
5. Seasons
6. Transportation expense
7. Distance from Residence to Work (kilometers)
8. Service time
9. Age
10. Work load Average/day
11. Hit target
```

12. Disciplinary failure
13. Education
14. Son (number of children)
15. Social drinker
16. Social smoker
17. Pet (number of pet)
18. Weight
19. Height
20. Body mass index
21. Absenteeism time in hours (target)

Todas los valores descriptivos o categoricos en las observaciones ya vienen convertidos a numericos en el dataset original. Aqui un detalle de los mismos:

Reason for absence

1. Certain infectious and parasitic diseases
2. II Neoplasms
3. Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism
4. Endocrine, nutritional and metabolic diseases
5. Mental and behavioural disorders
6. Diseases of the nervous system
7. Diseases of the eye and adnexa
8. Diseases of the ear and mastoid process
9. Diseases of the circulatory system
10. Diseases of the respiratory system
11. Diseases of the digestive system
12. Diseases of the skin and subcutaneous tissue
13. Diseases of the musculoskeletal system and connective tissue
14. Diseases of the genitourinary system
15. Pregnancy, childbirth and the puerperium
16. Certain conditions originating in the perinatal period
17. Congenital malformations, deformations and chromosomal abnormalities
18. Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
19. Injury, poisoning and certain other consequences of external causes
20. External causes of morbidity and mortality
21. Factors influencing health status and contact with health services.

And 7 categories without (CID):
- patient follow-up (22),
- medical consultation (23),
- blood donation (24),
- laboratory examination (25),
- unjustified absence (26),
- physiotherapy (27),
- dental consultation (28).

Day of the week

(Monday (2), Tuesday (3), Wednesday (4), Thursday (5), Friday (6))

Seasons

(summer (1), autumn (2), winter (3), spring (4))

education

(high school (1), graduate (2), postgraduate (3), master and doctor (4))

Disciplinary failure

(yes=1; no=0)

Social drinker

(yes=1; no=0)

Social smoker

(yes=1; no=0)

Dado que el archivo es un csv y solo tiene 740 observaciones, primero haremos una rapida inspeccion manual con un notepad++. Como hemos dicho anteriormente posee todos valores numericos, tenemos encabezado de columnas y todos los valores de cada fila estan separados por punto y coma. Todos los numeros son enteros a excepcion al parecer del work load average. Mientras que por ej la estatura y el peso estan expresados en cm y kilos respectivamente.

Dicho eso comenzaremos con el tratamiento del dataset.

2. Integración y selección de los datos de interés a analizar

En el caso de nuestro dataset no sera necesario realizar integracion con otros datasets ni tampoco realizar una seleccion o filtrado de los datos antes de analizarlos. Ya que todo esto fue realizado previamente por el equipo de personas que creo el proyecto en UCI.

Sin embargo realizaremos un primer analisis exploratorio de los datos (screening) para identificar si es necesario crear alguna nueva variable adicional en el dataset que nos ayude en nuestro objetivo.

Primero que nada importamos todas las librerias de R que estaremos usando o preveemos utilizar.

```
packages <- c("readr", "dplyr", "ggplot2","factoextra","gridExtra",
              "fpc", "reshape2", "stats","nortest", "car", "vcd", "pls")
new <- packages[!(packages %in% installed.packages()[,"Package"])]
if(length(new)) install.packages(new)
foo=lapply(packages, require, character.only=TRUE)
```

```
## Loading required package: readr
```

```
## Warning: package 'readr' was built under R version 4.0.5
```

```
## Loading required package: dplyr
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.0.5

## Loading required package: factoextra

## Warning: package 'factoextra' was built under R version 4.0.5

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

## Loading required package: gridExtra

## Warning: package 'gridExtra' was built under R version 4.0.5

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

## Loading required package: fpc

## Warning: package 'fpc' was built under R version 4.0.5

## Loading required package: reshape2

## Warning: package 'reshape2' was built under R version 4.0.5

## Loading required package: nortest

## Warning: package 'nortest' was built under R version 4.0.3

## Loading required package: car

## Warning: package 'car' was built under R version 4.0.5

## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.0.3

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## Loading required package: vcd

## Warning: package 'vcd' was built under R version 4.0.5

## Loading required package: grid

## Loading required package: pls

## Warning: package 'pls' was built under R version 4.0.5

##
## Attaching package: 'pls'

## The following object is masked from 'package:stats':
##
##     loadings
```

Leemos el archivo csv delimitado por ;

```
abs_df <- read_delim("Absenteeism_at_work.csv", col_names = TRUE, delim=';')
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double()
## )
## i Use `spec()` for the full column specifications.
```

```
# Dado que los nombres de las columnas tienes espacios, los quitamos agregando puntos:
names(abs_df)<-make.names(names(abs_df),unique = TRUE)

# Chequeamos ahora los nuevos nombres de columnas
colnames(abs_df)
```

```
##  [1] "ID"                              "Reason.for.absence"
##  [3] "Month.of.absence"                "Day.of.the.week"
##  [5] "Seasons"                         "Transportation.expense"
##  [7] "Distance.from.Residence.to.Work" "Service.time"
##  [9] "Age"                             "Work.load.Average.day."
## [11] "Hit.target"                      "Disciplinary.failure"
## [13] "Education"                       "Son"
## [15] "Social.drinker"                  "Social.smoker"
## [17] "Pet"                             "Weight"
## [19] "Height"                          "Body.mass.index"
## [21] "Absenteeism.time.in.hours"
```

```r
# chequeamos los tipos de datos las variables y vemos el sampleo de algunos de sus valores
str(abs_df)
```

```
## spec_tbl_df[,21] [740 x 21] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ID                               : num [1:740] 11 36 3 7 11 3 10 20 14 1 ...
##  $ Reason.for.absence               : num [1:740] 26 0 23 7 23 23 22 23 19 22 ...
##  $ Month.of.absence                 : num [1:740] 7 7 7 7 7 7 7 7 7 7 ...
##  $ Day.of.the.week                  : num [1:740] 3 3 4 5 5 6 6 6 2 2 ...
##  $ Seasons                          : num [1:740] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Transportation.expense           : num [1:740] 289 118 179 279 289 179 361 260 155 235 ...
##  $ Distance.from.Residence.to.Work: num [1:740] 36 13 51 5 36 51 52 50 12 11 ...
##  $ Service.time                     : num [1:740] 13 18 18 14 13 18 3 11 14 14 ...
##  $ Age                              : num [1:740] 33 50 38 39 33 38 28 36 34 37 ...
##  $ Work.load.Average.day.           : num [1:740] 240 240 240 240 240 ...
##  $ Hit.target                       : num [1:740] 97 97 97 97 97 97 97 97 97 97 ...
##  $ Disciplinary.failure             : num [1:740] 0 1 0 0 0 0 0 0 0 0 ...
##  $ Education                        : num [1:740] 1 1 1 1 1 1 1 1 1 3 ...
##  $ Son                              : num [1:740] 2 1 0 2 2 0 1 4 2 1 ...
##  $ Social.drinker                   : num [1:740] 1 1 1 1 1 1 1 1 1 0 ...
##  $ Social.smoker                    : num [1:740] 0 0 0 1 0 0 0 0 0 0 ...
##  $ Pet                              : num [1:740] 1 0 0 0 1 0 4 0 0 1 ...
##  $ Weight                           : num [1:740] 90 98 89 68 90 89 80 65 95 88 ...
##  $ Height                           : num [1:740] 172 178 170 168 172 170 172 168 196 172 ...
##  $ Body.mass.index                  : num [1:740] 30 31 31 24 30 31 27 23 25 29 ...
##  $ Absenteeism.time.in.hours        : num [1:740] 4 0 2 4 2 2 8 4 40 8 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ID = col_double(),
##   ..   `Reason for absence` = col_double(),
##   ..   `Month of absence` = col_double(),
##   ..   `Day of the week` = col_double(),
##   ..   Seasons = col_double(),
##   ..   `Transportation expense` = col_double(),
##   ..   `Distance from Residence to Work` = col_double(),
##   ..   `Service time` = col_double(),
##   ..   Age = col_double(),
##   ..   `Work load Average/day ` = col_double(),
##   ..   `Hit target` = col_double(),
##   ..   `Disciplinary failure` = col_double(),
##   ..   Education = col_double(),
##   ..   Son = col_double(),
##   ..   `Social drinker` = col_double(),
##   ..   `Social smoker` = col_double(),
##   ..   Pet = col_double(),
##   ..   Weight = col_double(),
##   ..   Height = col_double(),
##   ..   `Body mass index` = col_double(),
##   ..   `Absenteeism time in hours` = col_double()
##   .. )
```

Pero antes de comenzar a sacar conclusiones sobres los datos y empezar a cruzar variables podriamos discretizar variables como:

- Age

- Distance.from.Residence.to.Work
- Body.mass.index
- Absenteeism.time.in.hours

Ya que para age ya hemos visto que tenemos bastante dispersos las edades y mejor agruparlas por los clasicos rangos de edad. Con la distancia nos pasa algo similar.

Para Body mass index al ser un indice nos dice poco sin contrastarlo con algo que conozcamos asi que mejor agruparlo segun los criterios que se los suele analizar a nivel medico:

```
< 18.5 Underweight
18.5-25 Normal weight
25-30 Overweight
> 30  Obese
```

Mientras que para las horas de ausentismo podriamos tambien predefinir algunos grupos o rangos horarios de ausentismo para el analisis posterior:

```
0 h horas
1-3 horas
4-8 horas
9-16 horas
17-40 horas
+ 40 horas
```

```r
# Creamos Reason for Absence Desc
abs_df <- abs_df %>% mutate(Reason.for.absence.Desc = case_when
                    (Reason.for.absence == 0 ~ 'No Aplica',
                     Reason.for.absence == 1 ~ 'Infectious',
                     Reason.for.absence == 2 ~ 'Neoplasms',
                     Reason.for.absence == 3 ~ 'Immune System & Blood Issues',
                     Reason.for.absence == 4 ~ 'Metabolic diseases',
                     Reason.for.absence == 5 ~ 'Mental & Behavior disorders',
                     Reason.for.absence == 6 ~ 'nervous system diseases',
                     Reason.for.absence == 7 ~ 'eye and adnexa diseases',
                     Reason.for.absence == 8 ~ 'ear and mastoid diseases',
                     Reason.for.absence == 9 ~ 'circulatory diseases',
                     Reason.for.absence == 10 ~ 'respiratory diseases',
                     Reason.for.absence == 11 ~ 'digestive diseases',
                     Reason.for.absence == 12 ~ 'skin diseases',
                     Reason.for.absence == 13 ~ 'musculoskeletal diseases',
                     Reason.for.absence == 14 ~ 'genitourinary diseases',
                     Reason.for.absence == 15 ~ 'Pregnancy, and related',
                     Reason.for.absence == 16 ~ 'perinatal conditions',
                     Reason.for.absence == 17 ~ 'Congenital malformations',
                     Reason.for.absence == 18 ~ 'abnormal clinical findings',
                     Reason.for.absence == 19 ~ 'Injury, poisoning related',
                     Reason.for.absence == 20 ~ 'morbidity and mortality',
                     Reason.for.absence == 21 ~ 'Other Factors',
                     Reason.for.absence == 22 ~ 'patient follow-up',
                     Reason.for.absence == 23 ~ 'medical consultation',
                     Reason.for.absence == 24 ~ 'blood donation',
                     Reason.for.absence == 25 ~ 'laboratory examination',
```

```
                    Reason.for.absence == 26 ~ 'unjustified absence',
                    Reason.for.absence == 27 ~ 'physiotherapy',
                    Reason.for.absence == 28 ~ 'dental consultation'
                    )
                  )


# Creamos Seasons Desc
abs_df <- abs_df %>% mutate(Seasons.Desc = case_when
                  (Seasons == 1 ~ 'Summer',
                   Seasons == 2 ~ 'Autumn',
                   Seasons == 3 ~ 'Winter',
                   Seasons == 4 ~ 'Spring')
                  )

# Creamos Education Desc
abs_df <- abs_df %>% mutate(Education.Desc = case_when
                  (Education == 1 ~ 'HighSchool',
                   Education == 2 ~ 'Graduate',
                   Education == 3 ~ 'PostGraduate',
                   Education == 4 ~ 'Ms & Dr')
                  )

# Creamos Day of the week Desc
abs_df <- abs_df %>% mutate(Day.of.the.week.Desc = case_when (
                   Day.of.the.week == 2 ~ 'Monday',
                   Day.of.the.week == 3 ~ 'Tuesday',
                   Day.of.the.week == 4 ~ 'Wednesday',
                   Day.of.the.week == 5 ~ 'Thursday',
                   Day.of.the.week == 6 ~ 'Friday'
                    ))

# Creamos age range
abs_df["age_range"] <- as.factor(cut(abs_df$Age, breaks =
                                       c(0,10,20,30,40,50,60,70,100),
                                     labels = c(1, 2, 3, 4,5,6,7,8)))
# Donde se corresponden a los siguientes rangos respectivamente:
# c("0-9", "10-19", "20-29", "30-39","40-49","50-59","60-69","70-79")


# Creamos Distance range
abs_df["distance_range"] <- as.factor(cut(abs_df$Distance.from.Residence.to.Work,
                                          breaks = c(0,10,20,30,40,100),
                                          labels = c(1, 2, 3, 4, 5)))
# Donde se corresponden a los siguientes rangos respectivamente:
# c("10km", "20km", "30km", "40km", "+40km")


# Creamos BMI
abs_df["BMI"] <- as.factor(cut(abs_df$Body.mass.index,
                               breaks = c(0,18.5,25,30,100),
                               labels = c(1, 2, 3, 4)))
# Donde se corresponden a los siguientes rangos respectivamente:
```

```
# c("Underweight", "Normal", "Overweight", "Obese")

# Creamos absenteeism_range
abs_df <- abs_df %>% mutate(absenteeism_range = case_when
                   (Absenteeism.time.in.hours == 0 ~ '0 h',
                    between(Absenteeism.time.in.hours, 1, 3) ~ '1-3 h',
                    between(Absenteeism.time.in.hours, 4, 8) ~ '4-8 h',
                    between(Absenteeism.time.in.hours, 9, 16) ~  '9-16 h',
                    between(Absenteeism.time.in.hours, 17, 40) ~ '17-40 h',
                    Absenteeism.time.in.hours > 40 ~ '+ 40 h'
                   )
                )


# Mientras que variables como Disciplinary failure, Social drinker y Social
# smoker no haran falta convertirlas a descriptivas ya que es claro que 1 es Yes
# y 0 es No. Lo mismo sucede con Day of the week, es simple determinar de que
# estamos hablando cuando vemos los valores numericos. Caso similar con month,
# todos sabemos que mes representan los valores del 1 al 12, siendo el 0 lo que
# comentamos antes, valores no informados que no afectaran al modelo, aunque
# haremos nuestros testeos mas adelante en la practica.
```

Dicho eso y luego de haber agregado los campos descriptivos y discretizado sigamos analizando el resto de las variables en la siguiente seccion de limpieza. Donde analiceremos valores vacios, valores extremos, etc..

3. Limpieza de los datos

Hacemos las primeros chequeos del contenido del dataset:

```
# visualizamos las primeras 5 observaciones
head(abs_df,5)
```

```
## # A tibble: 5 x 29
##      ID Reason.for.absence Month.of.absence Day.of.the.week Seasons
##   <dbl>              <dbl>            <dbl>           <dbl>   <dbl>
## 1    11                 26                7               3       1
## 2    36                  0                7               3       1
## 3     3                 23                7               4       1
## 4     7                  7                7               5       1
## 5    11                 23                7               5       1
## # ... with 24 more variables: Transportation.expense <dbl>,
## #   Distance.from.Residence.to.Work <dbl>, Service.time <dbl>, Age <dbl>,
## #   Work.load.Average.day. <dbl>, Hit.target <dbl>, Disciplinary.failure <dbl>,
## #   Education <dbl>, Son <dbl>, Social.drinker <dbl>, Social.smoker <dbl>,
## #   Pet <dbl>, Weight <dbl>, Height <dbl>, Body.mass.index <dbl>,
## #   Absenteeism.time.in.hours <dbl>, Reason.for.absence.Desc <chr>,
## #   Seasons.Desc <chr>, Education.Desc <chr>, Day.of.the.week.Desc <chr>,
## #   age_range <fct>, distance_range <fct>, BMI <fct>, absenteeism_range <chr>
```

```
# Estadisticas basicas
summary(abs_df)
```

```
##        ID        Reason.for.absence Month.of.absence Day.of.the.week
##  Min.   : 1.00   Min.   : 0.00      Min.   : 0.000   Min.   :2.000
##  1st Qu.: 9.00   1st Qu.:13.00      1st Qu.: 3.000   1st Qu.:3.000
##  Median :18.00   Median :23.00      Median : 6.000   Median :4.000
##  Mean   :18.02   Mean   :19.22      Mean   : 6.324   Mean   :3.915
##  3rd Qu.:28.00   3rd Qu.:26.00      3rd Qu.: 9.000   3rd Qu.:5.000
##  Max.   :36.00   Max.   :28.00      Max.   :12.000   Max.   :6.000
##
##     Seasons      Transportation.expense Distance.from.Residence.to.Work
##  Min.   :1.000   Min.   :118.0          Min.   : 5.00
##  1st Qu.:2.000   1st Qu.:179.0          1st Qu.:16.00
##  Median :3.000   Median :225.0          Median :26.00
##  Mean   :2.545   Mean   :221.3          Mean   :29.63
##  3rd Qu.:4.000   3rd Qu.:260.0          3rd Qu.:50.00
##  Max.   :4.000   Max.   :388.0          Max.   :52.00
##
##   Service.time        Age        Work.load.Average.day.   Hit.target
##  Min.   : 1.00   Min.   :27.00   Min.   :205.9          Min.   : 81.00
##  1st Qu.: 9.00   1st Qu.:31.00   1st Qu.:244.4          1st Qu.: 93.00
##  Median :13.00   Median :37.00   Median :264.2          Median : 95.00
##  Mean   :12.55   Mean   :36.45   Mean   :271.5          Mean   : 94.59
##  3rd Qu.:16.00   3rd Qu.:40.00   3rd Qu.:294.2          3rd Qu.: 97.00
##  Max.   :29.00   Max.   :58.00   Max.   :378.9          Max.   :100.00
##
##  Disciplinary.failure   Education          Son        Social.drinker
##  Min.   :0.00000      Min.   :1.000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.00000      1st Qu.:1.000   1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.00000      Median :1.000   Median :1.000   Median :1.0000
##  Mean   :0.05405      Mean   :1.292   Mean   :1.019   Mean   :0.5676
##  3rd Qu.:0.00000      3rd Qu.:1.000   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.00000      Max.   :4.000   Max.   :4.000   Max.   :1.0000
##
##  Social.smoker          Pet             Weight          Height
##  Min.   :0.00000   Min.   :0.0000   Min.   : 56.00   Min.   :163.0
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.: 69.00   1st Qu.:169.0
##  Median :0.00000   Median :0.0000   Median : 83.00   Median :170.0
##  Mean   :0.07297   Mean   :0.7459   Mean   : 79.04   Mean   :172.1
##  3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.: 89.00   3rd Qu.:172.0
##  Max.   :1.00000   Max.   :8.0000   Max.   :108.00   Max.   :196.0
##
##  Body.mass.index Absenteeism.time.in.hours Reason.for.absence.Desc
##  Min.   :19.00   Min.   :  0.000           Length:740
##  1st Qu.:24.00   1st Qu.:  2.000           Class :character
##  Median :25.00   Median :  3.000           Mode  :character
##  Mean   :26.68   Mean   :  6.924
##  3rd Qu.:31.00   3rd Qu.:  8.000
##  Max.   :38.00   Max.   :120.000
##
##  Seasons.Desc       Education.Desc     Day.of.the.week.Desc   age_range
##  Length:740         Length:740         Length:740           4      :422
##  Class :character   Class :character   Class :character     3      :177
##  Mode  :character   Mode  :character   Mode  :character     5      :132
##                                                             6      :  9
##                                                             1      :  0
```

```
##                                                          2      :  0
##                                                          (Other):  0
##  distance_range BMI     absenteeism_range
##  1: 61          1:  0   Length:740
##  2:167          2:390   Class :character
##  3:223          3:146   Mode  :character
##  4: 79          4:204
##  5:210
##
##
```

```r
# Verificamos la estructura y contenido del conjunto de datos
str(abs_df)
```

```
## spec_tbl_df[,29] [740 x 29] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ID                               : num [1:740] 11 36 3 7 11 3 10 20 14 1 ...
##  $ Reason.for.absence               : num [1:740] 26 0 23 7 23 23 22 23 19 22 ...
##  $ Month.of.absence                 : num [1:740] 7 7 7 7 7 7 7 7 7 7 ...
##  $ Day.of.the.week                  : num [1:740] 3 3 4 5 5 6 6 6 2 2 ...
##  $ Seasons                          : num [1:740] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Transportation.expense           : num [1:740] 289 118 179 279 289 179 361 260 155 235 ...
##  $ Distance.from.Residence.to.Work: num [1:740] 36 13 51 5 36 51 52 50 12 11 ...
##  $ Service.time                     : num [1:740] 13 18 18 14 13 18 3 11 14 14 ...
##  $ Age                              : num [1:740] 33 50 38 39 33 38 28 36 34 37 ...
##  $ Work.load.Average.day.           : num [1:740] 240 240 240 240 240 ...
##  $ Hit.target                       : num [1:740] 97 97 97 97 97 97 97 97 97 97 ...
##  $ Disciplinary.failure             : num [1:740] 0 1 0 0 0 0 0 0 0 0 ...
##  $ Education                        : num [1:740] 1 1 1 1 1 1 1 1 1 3 ...
##  $ Son                              : num [1:740] 2 1 0 2 2 0 1 4 2 1 ...
##  $ Social.drinker                   : num [1:740] 1 1 1 1 1 1 1 1 1 0 ...
##  $ Social.smoker                    : num [1:740] 0 0 0 1 0 0 0 0 0 0 ...
##  $ Pet                              : num [1:740] 1 0 0 0 1 0 4 0 0 1 ...
##  $ Weight                           : num [1:740] 90 98 89 68 90 89 80 65 95 88 ...
##  $ Height                           : num [1:740] 172 178 170 168 172 170 172 168 196 172 ...
##  $ Body.mass.index                  : num [1:740] 30 31 31 24 30 31 27 23 25 29 ...
##  $ Absenteeism.time.in.hours        : num [1:740] 4 0 2 4 2 2 8 4 40 8 ...
##  $ Reason.for.absence.Desc          : chr [1:740] "unjustified absence" "No Aplica" "medical consultati
##  $ Seasons.Desc                     : chr [1:740] "Summer" "Summer" "Summer" "Summer" ...
##  $ Education.Desc                   : chr [1:740] "HighSchool" "HighSchool" "HighSchool" "HighSchool"
##  $ Day.of.the.week.Desc             : chr [1:740] "Tuesday" "Tuesday" "Wednesday" "Thursday" ...
##  $ age_range                        : Factor w/ 8 levels "1","2","3","4",..: 4 5 4 4 4 4 3 4 4 4 ...
##  $ distance_range                   : Factor w/ 5 levels "1","2","3","4",..: 4 2 5 1 4 5 5 5 2 2 ...
##  $ BMI                              : Factor w/ 4 levels "1","2","3","4": 3 4 4 2 3 4 3 2 2 3 ...
##  $ absenteeism_range                : chr [1:740] "4-8 h" "0 h" "1-3 h" "4-8 h" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ID = col_double(),
##   ..   `Reason for absence` = col_double(),
##   ..   `Month of absence` = col_double(),
##   ..   `Day of the week` = col_double(),
##   ..   Seasons = col_double(),
##   ..   `Transportation expense` = col_double(),
##   ..   `Distance from Residence to Work` = col_double(),
##   ..   `Service time` = col_double(),
```

```
##   ..   Age = col_double(),
##   ..   `Work load Average/day ` = col_double(),
##   ..   `Hit target` = col_double(),
##   ..   `Disciplinary failure` = col_double(),
##   ..   Education = col_double(),
##   ..   Son = col_double(),
##   ..   `Social drinker` = col_double(),
##   ..   `Social smoker` = col_double(),
##   ..   Pet = col_double(),
##   ..   Weight = col_double(),
##   ..   Height = col_double(),
##   ..   `Body mass index` = col_double(),
##   ..   `Absenteeism time in hours` = col_double()
##   .. )
```

Si bien, segun la descripcion del dataset en UCI y con la inspeccion visual no hay missing values, realicemos un chequeo rapido:

```
# Estadísticas de valores vacíos
colSums(is.na(abs_df))
```

```
##                              ID              Reason.for.absence
##                               0                               0
##                Month.of.absence                 Day.of.the.week
##                               0                               0
##                         Seasons           Transportation.expense
##                               0                               0
## Distance.from.Residence.to.Work                    Service.time
##                               0                               0
##                             Age           Work.load.Average.day.
##                               0                               0
##                      Hit.target             Disciplinary.failure
##                               0                               0
##                       Education                             Son
##                               0                               0
##                  Social.drinker                    Social.smoker
##                               0                               0
##                             Pet                          Weight
##                               0                               0
##                          Height                 Body.mass.index
##                               0                               0
##       Absenteeism.time.in.hours         Reason.for.absence.Desc
##                               0                               0
##                    Seasons.Desc                  Education.Desc
##                               0                               0
##            Day.of.the.week.Desc                       age_range
##                               0                               0
##                  distance_range                             BMI
##                               0                               0
##               absenteeism_range
##                               0
```

```r
# y ahora los missing
colSums(abs_df=="")
```

```
##                              ID               Reason.for.absence
##                               0                                0
##                 Month.of.absence                  Day.of.the.week
##                               0                                0
##                         Seasons           Transportation.expense
##                               0                                0
## Distance.from.Residence.to.Work                     Service.time
##                               0                                0
##                             Age           Work.load.Average.day.
##                               0                                0
##                      Hit.target             Disciplinary.failure
##                               0                                0
##                       Education                              Son
##                               0                                0
##                  Social.drinker                   Social.smoker
##                               0                                0
##                             Pet                           Weight
##                               0                                0
##                          Height                 Body.mass.index
##                               0                                0
##       Absenteeism.time.in.hours         Reason.for.absence.Desc
##                               0                                0
##                    Seasons.Desc                  Education.Desc
##                               0                                0
##            Day.of.the.week.Desc                       age_range
##                               0                                0
##                  distance_range                             BMI
##                               0                                0
##               absenteeism_range
##                               0
```

Ahora analicemos algunas variables numerica y visualmente para entender mejor los datos:

```r
# Veamos los valores distintos de cada atributo
sapply(abs_df, function(x) length(unique(x)))
```

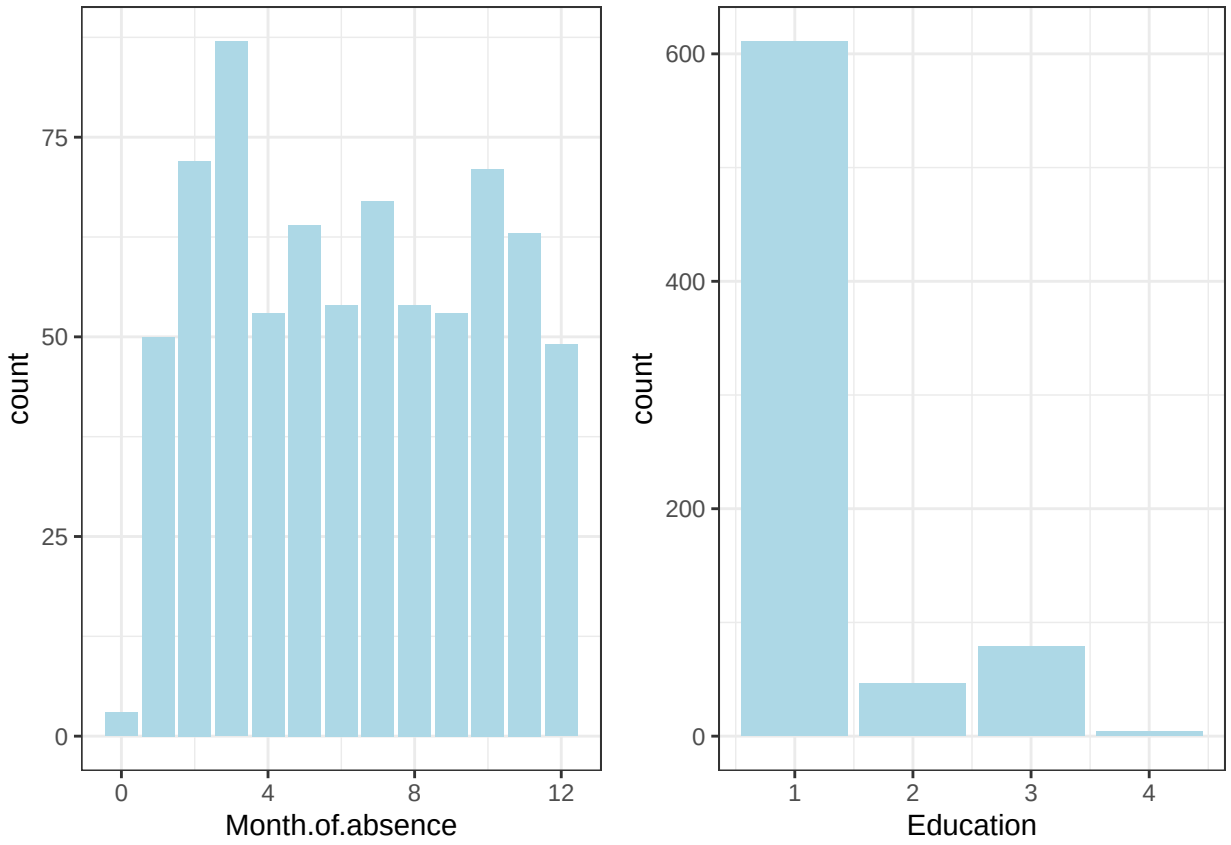
```
##                              ID               Reason.for.absence
##                              36                               28
##                 Month.of.absence                  Day.of.the.week
##                              13                                5
##                         Seasons           Transportation.expense
##                               4                               24
## Distance.from.Residence.to.Work                     Service.time
##                              25                               18
##                             Age           Work.load.Average.day.
##                              22                               38
##                      Hit.target             Disciplinary.failure
##                              13                                2
##                       Education                              Son
```

```
##                                4                                5
##                  Social.drinker                   Social.smoker
##                                2                                2
##                              Pet                           Weight
##                                6                               26
##                           Height                 Body.mass.index
##                               14                               17
##       Absenteeism.time.in.hours         Reason.for.absence.Desc
##                               19                               28
##                    Seasons.Desc                   Education.Desc
##                                4                                4
##            Day.of.the.week.Desc                        age_range
##                                5                                4
##                  distance_range                              BMI
##                                5                                3
##               absenteeism_range
##                                6
```

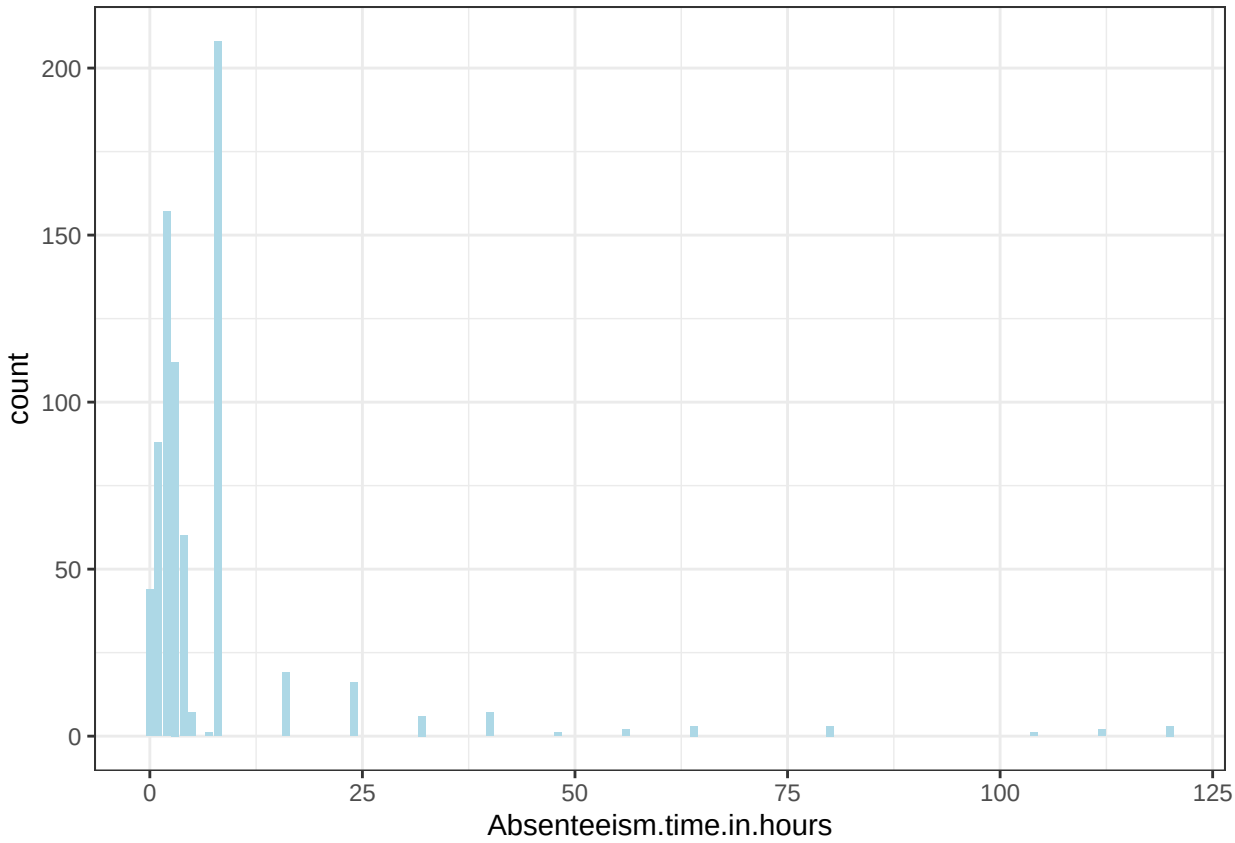
Aqui vemos que tenemos 13 meses informados de ausencia, arranquemos analizando esa variable con graficas simples.

Probemos tambien dos mas, la educacion de los empleados con la cantidad de horas de ausentimos

```
a = ggplot(abs_df,aes(x=Month.of.absence,fill=Month.of.absence))+
  geom_bar(fill="lightblue")+
  theme_bw()

b = ggplot(abs_df,aes(x=Education,fill=Education))+
  geom_bar(fill="lightblue")+
  theme_bw()

c = ggplot(abs_df,aes(x=Absenteeism.time.in.hours,fill=Absenteeism.time.in.hours))+
  geom_bar(fill="lightblue")+theme_bw()

grid.arrange(a, b, nrow = 1, ncol = 2)
```




```
grid.arrange(c, nrow = 1)
```




```
# vemos que los datos para ese mes no informado, son pocos, solo 3 obs.
table(abs_df$Month.of.absence)
```

```
##
##  0  1  2  3  4  5  6  7  8  9 10 11 12
##  3 50 72 87 53 64 54 67 54 53 71 63 49
```

```
# Chequeemos esos datos:
select(filter(abs_df, Month.of.absence ==0),
       ID,Reason.for.absence,Day.of.the.week,Education,Absenteeism.time.in.hours)
```

```
## # A tibble: 3 x 5
##      ID Reason.for.absence Day.of.the.week Education Absenteeism.time.in.hours
##   <dbl>              <dbl>           <dbl>     <dbl>                     <dbl>
## 1     4                  0               3         1                         0
## 2     8                  0               4         1                         0
## 3    35                  0               6         1                         0
```

En cuanto a los datos con mes no informado, luego podriamos eliminar estos registros ya que como vemos:

1 tienen razon de ausentismo en 0 y ademas la cantidad de horas informadas es cero, eso indica que son personas que nunca han faltado y por eso no hay ninguna razon de ausentismo???

2 son datos del nivel de educacion 1 que corresponde al que mayor cantidad de registros tenemos, por lo que quitar 3 filas de el no deberia representar un sesgo.

Mientras que vemos en la grafica que para la variable education predominan las de valor 1. Pero claro si bien tener las variables numericas nos permite utilizar varias funciones y aplicar algoritmos, para un analisis exploratorio inicial seria bueno tener los valores descriptivos reales para entender "mas facil" los datos.

Asi que iremos agregando variables al dataset con las descripciones de estas variables numericas.

(high school (1), graduate (2), postgraduate (3), master and doctor (4))

Antes de comenzar a agregar columnas descriptivas, chequemos el punto 1 que acabamos de comentar.

```
# Validamos para los reason 0, si hay horas de ausencia
table(filter(abs_df, Reason.for.absence ==0)$Absenteeism.time.in.hours)
```

```
##
##  0
## 43
```

```
# Validamos para los que no tienen horas informadas que tengan 0 en reason
table(filter(abs_df, Absenteeism.time.in.hours == 0)$Reason.for.absence)
```

```
##
##  0 27
## 43  1
```

Bueno, al parecer el reason 0 se corresponde a los que no se han ausentando al trabajo, salvo solo 1 obversacion donde tiene un reason de ausencia pero no ha informado horas. Por lo no afectara a nuestros analisis estos registros. Con esto tambien revalidamos que necesitamos tener disponibles los valores descriptivos para este tipo de analisis exploratorio que estamos haciendo ya que sino tendriamos que ir a buscar cada vez a que descripcion corresponde cada ID de variable.

Otra forma de verificar valores extremos o fuera de rango es con boxplot.stats y en particular con la variable out que directamente nos da los valores extremos. Veamoslo

```
boxplot.stats(abs_df$Month.of.absence)$out
```

```
## numeric(0)
```

```
boxplot.stats(abs_df$Reason.for.absence)$out
```

```
## numeric(0)
```

```
boxplot.stats(abs_df$Education)$out
```

```
##   [1] 3 2 3 3 2 2 3 3 3 2 2 2 3 2 2 3 3 2 3 2 3 2 2 2 3 3 3 3 3 3 3 3 2 3 3 3
##  [38] 3 3 3 3 3 2 3 3 2 2 3 2 2 2 3 3 3 3 2 3 3 2 2 2 2 2 3 3 3 3 2 3 3 2 2 3
##  [75] 3 3 2 2 3 2 2 3 3 3 4 2 3 2 2 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 4 3 3 4 4 3
## [112] 3 2 3 2 2 3 3 2 3 2 3 2 3 3 2 2 2 3
```

```
boxplot.stats(abs_df[abs_df$Absenteeism.time.in.hours!=0,]$Absenteeism.time.in.hours)$out
```

```
##  [1]  40  40  32  32  40  24  64  56  40  40  24  24  24  56  24  24  24  24  80
## [20]  32  24  32  40  64 120  32  24 120  40  24 112  24  32  80  24 112  24 104
## [39]  24  64  48  24 120  80
```

Lo que vemos aqui por ej para education es que si bien esos valores parecen extremos, al estar fuera del IQR, en realidad se trata de pocas observaciones respecto al resto de la poblacion y terminan pareciendo valores erroneos. Pero no lo son, corresponden como sabemos a personas con un nivel de educacion superior al habitual en este grupo en particular. Por lo que no deben descartarse ni aplicar ningun tipo de transformacion.

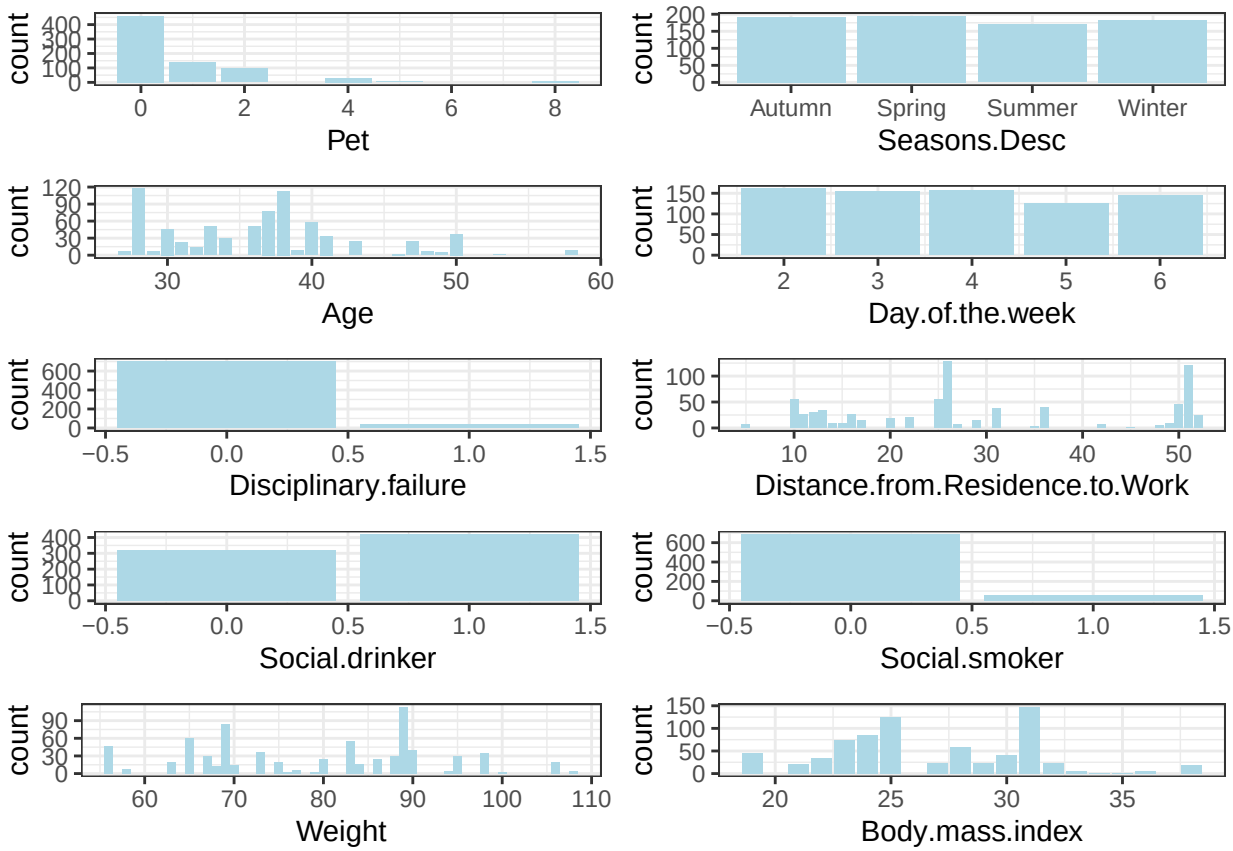
Mientras que para las horas de ausencia sucede algo similar, siempre habra casos particulares donde personas se ausenten por razones imprevistas y seran los menos. Y es justamente una de los objetivos de este estudio. Entender no solo las posibles horas de ausencia habituales, sino las de incluso casos extraños.

4. Análisis de los datos

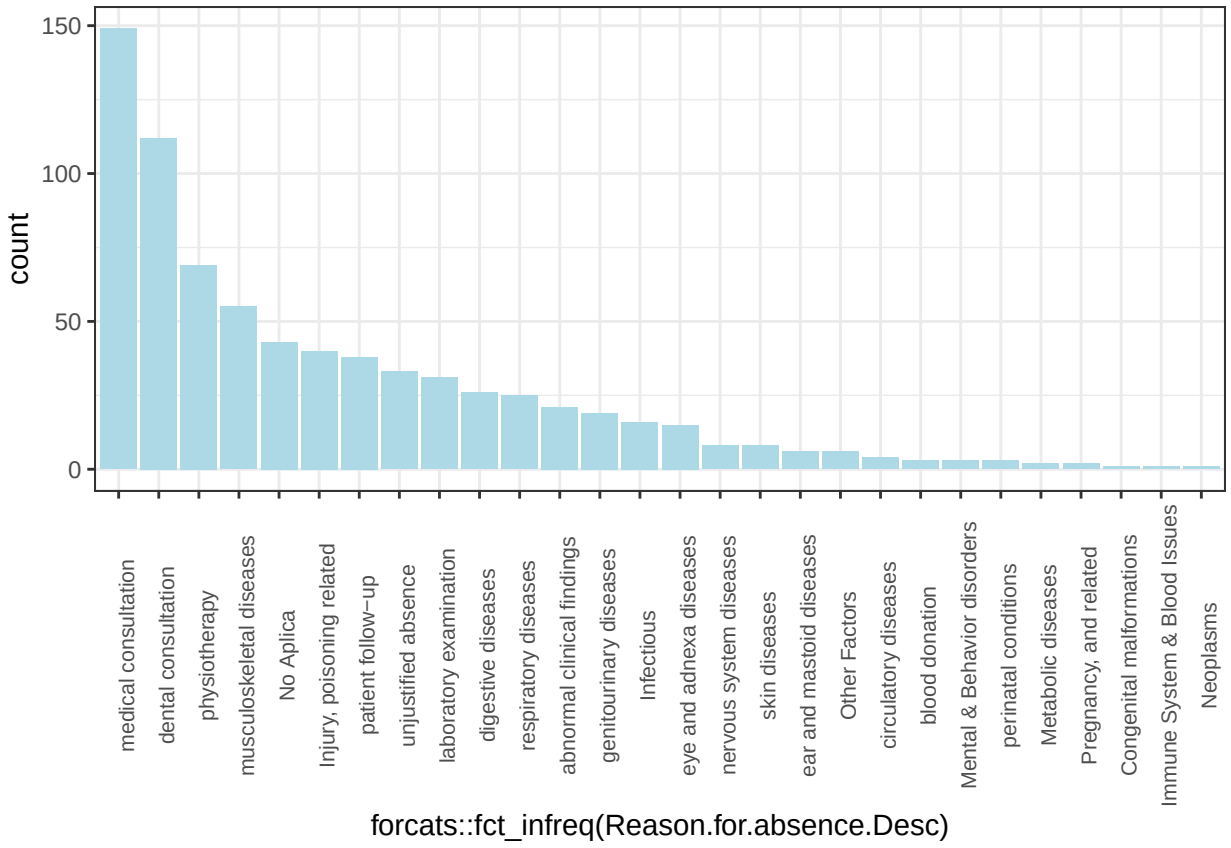
4.1. Analisis descriptivo

Esta vez arranquemos realizando un EDA mas exhaustivo:

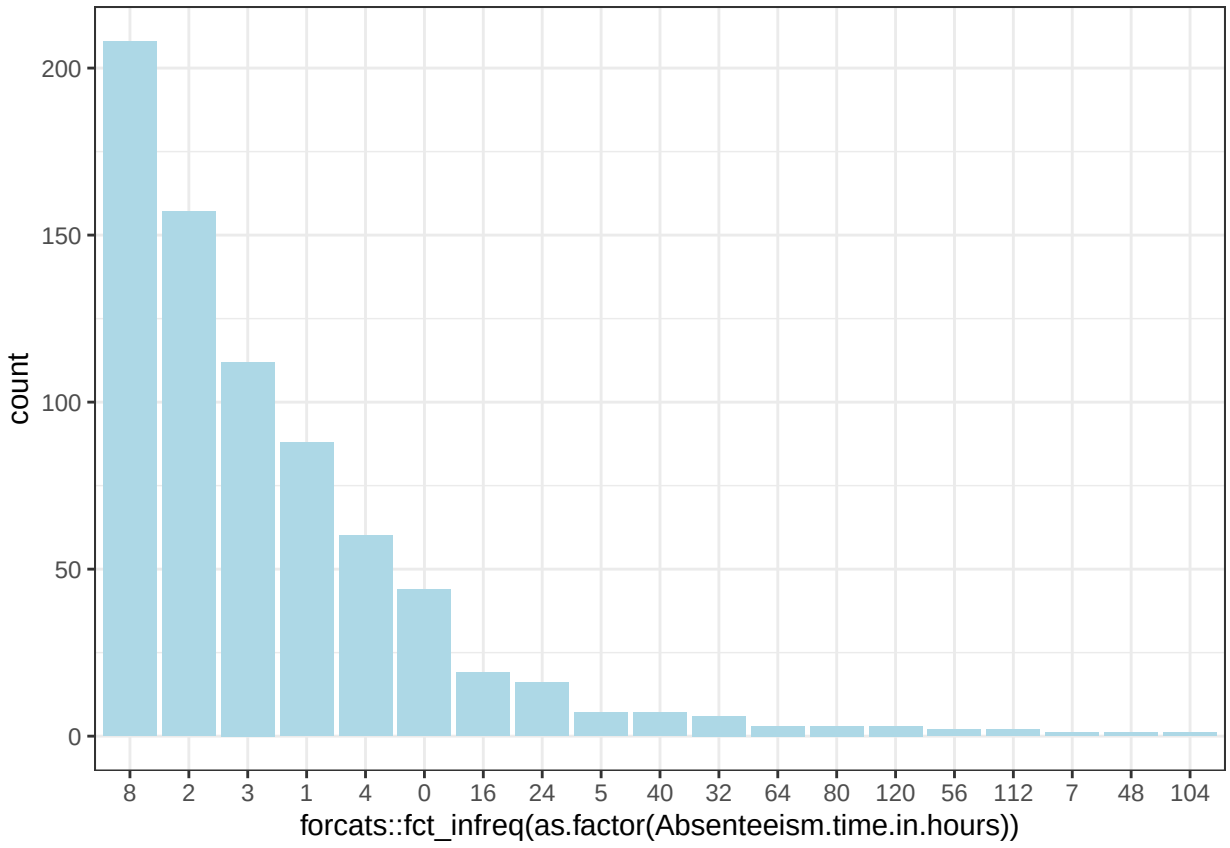
```
pet = ggplot(abs_df,aes(x=Pet,fill=Pet))+geom_bar(fill="lightblue")+
  theme_bw()
Seasons = ggplot(abs_df,aes(x=Seasons.Desc,fill=Seasons.Desc))+
  geom_bar(fill="lightblue")+theme_bw()

age = ggplot(abs_df,aes(x=Age,fill=Age))+geom_bar(fill="lightblue")+
  theme_bw()
day = ggplot(abs_df,aes(x=Day.of.the.week,fill=Day.of.the.week))+
  geom_bar(fill="lightblue")+theme_bw()

disciplinary = ggplot(abs_df,aes(x=Disciplinary.failure,
                                 fill=Disciplinary.failure))+
  geom_bar(fill="lightblue")+
  theme_bw()

distance = ggplot(abs_df,aes(x=Distance.from.Residence.to.Work,
                             fill=Distance.from.Residence.to.Work))+
  geom_bar(fill="lightblue")+
  theme_bw()

drinker = ggplot(abs_df,aes(x=Social.drinker,fill=Social.drinker))+
  geom_bar(fill="lightblue")+
  theme_bw()

smoker = ggplot(abs_df,aes(x=Social.smoker,fill=Social.smoker))+
  geom_bar(fill="lightblue")+
  theme_bw()


weight = ggplot(abs_df,aes(x=Weight,fill=Weight))+
  geom_bar(fill="lightblue")+
  theme_bw()
mass = ggplot(abs_df,aes(x=Body.mass.index,fill=Body.mass.index))+
  geom_bar(fill="lightblue")+theme_bw()

grid.arrange(pet,Seasons, age, day, disciplinary, distance, drinker, smoker,
             weight, mass, nrow = 5, ncol = 2)
```




```
reason = ggplot(abs_df) +
    geom_bar(aes(x = forcats::fct_infreq(Reason.for.absence.Desc)
                 #,fill="Education"
                 ), fill="lightblue")+
  theme_bw()+theme(axis.text.x=element_text(size=8,angle=90))

grid.arrange(reason, nrow = 1)
```

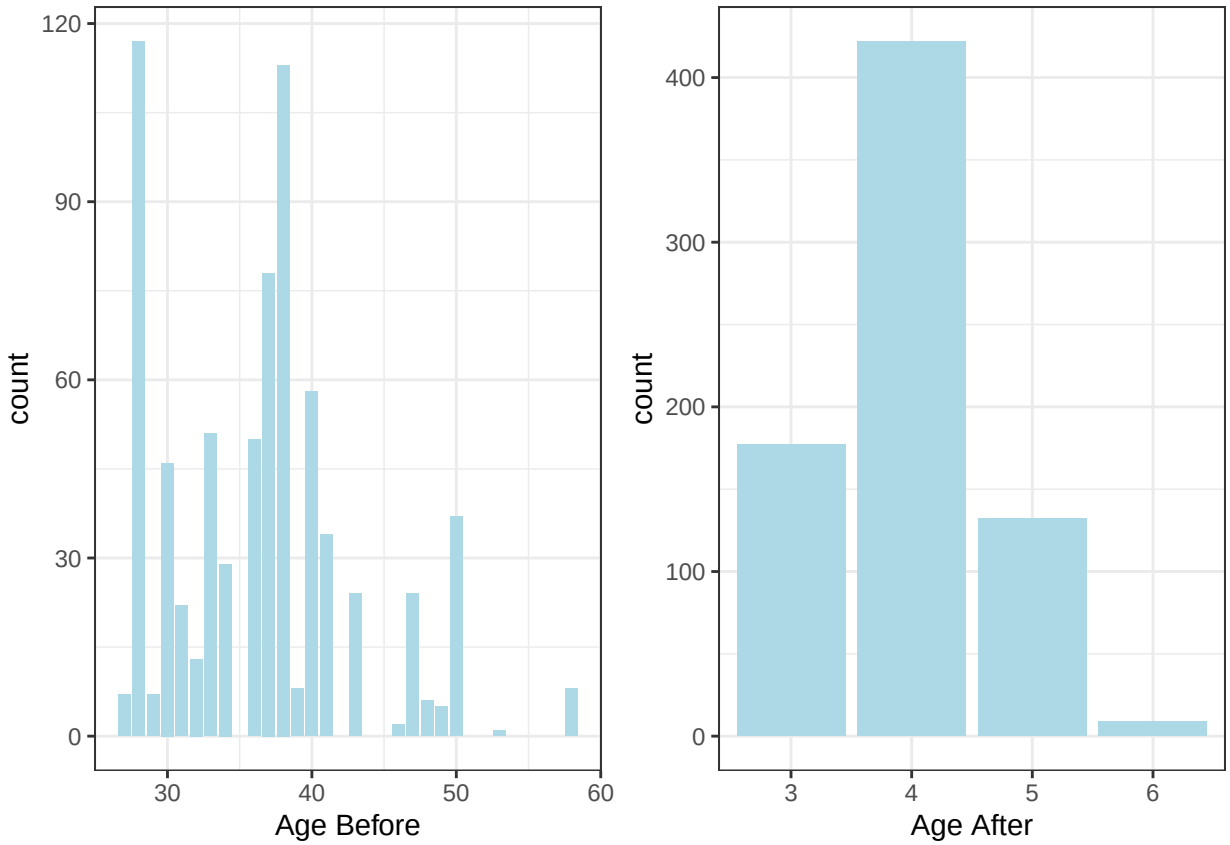



```
absenteeism = ggplot(abs_df)+
  geom_bar(aes(x=forcats::fct_infreq(as.factor(Absenteeism.time.in.hours))),
           fill="lightblue")+
  theme_bw()

grid.arrange(absenteeism, nrow = 1)
```

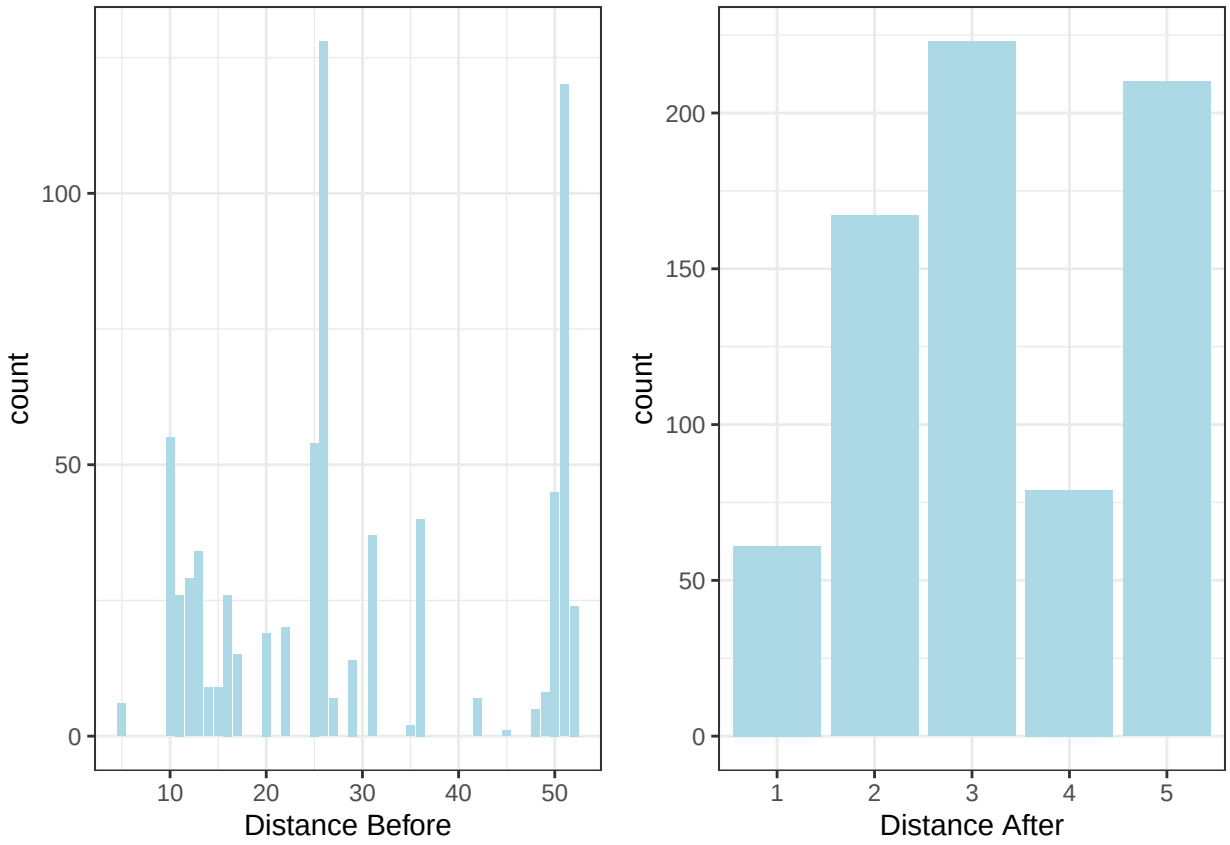


Vemos ahora como quedaron distribuidos los datos de estas variables post discretizacion

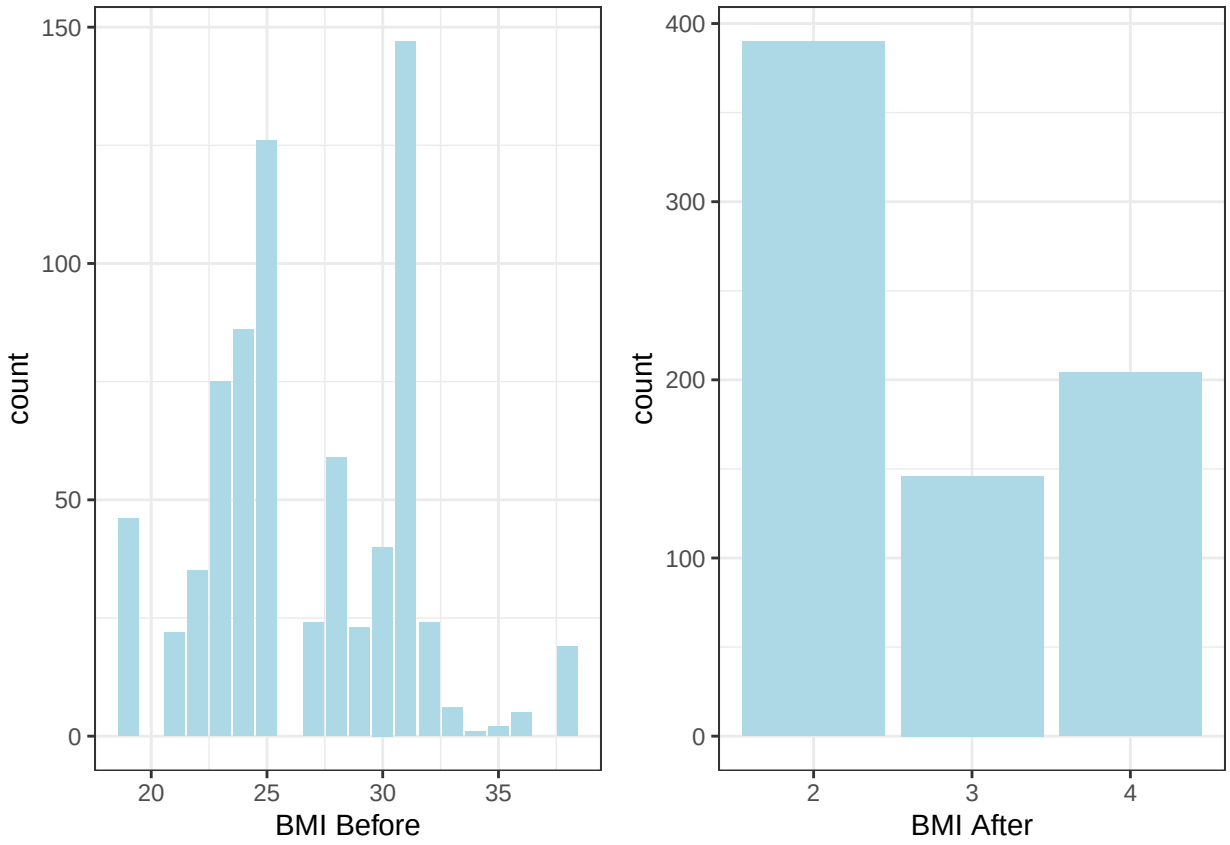
```
# Comparacion age y age discretizada
ant = ggplot(abs_df,aes(x=Age,fill=Age))+geom_bar(fill="lightblue")+
  xlab("Age Before")+theme_bw()

desp = ggplot(abs_df,aes(x=age_range,fill=age_range))+
  geom_bar(fill="lightblue")+xlab("Age After")+theme_bw()
grid.arrange(ant,desp, nrow = 1, ncol = 2)
```



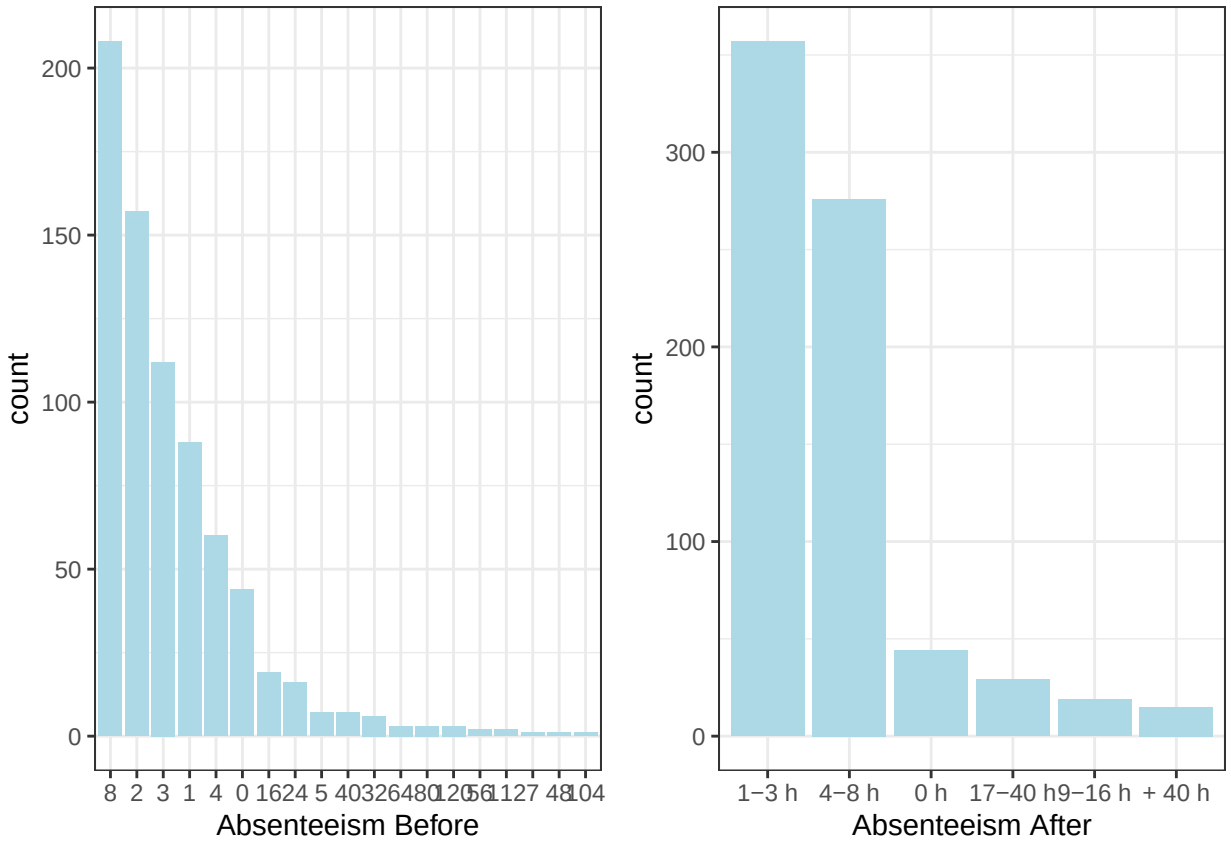

```
# Distancia y Distancia discretizada
ant = ggplot(abs_df,aes(x=Distance.from.Residence.to.Work,
                        fill=Distance.from.Residence.to.Work))+
  geom_bar(fill="lightblue")+xlab("Distance Before")+theme_bw()

desp = ggplot(abs_df,aes(x=distance_range,fill=distance_range))+
  geom_bar(fill="lightblue")+xlab("Distance After")+theme_bw()

grid.arrange(ant,desp, nrow = 1, ncol = 2)
```




```
# BMI y BMI discretizada
ant = ggplot(abs_df,aes(x=Body.mass.index,fill=Body.mass.index))+
  geom_bar(fill="lightblue")+xlab("BMI Before")+theme_bw()

desp = ggplot(abs_df,aes(x=BMI,fill=BMI))+geom_bar(fill="lightblue")+
  xlab("BMI After")+theme_bw()

grid.arrange(ant,desp, nrow = 1, ncol = 2)
```



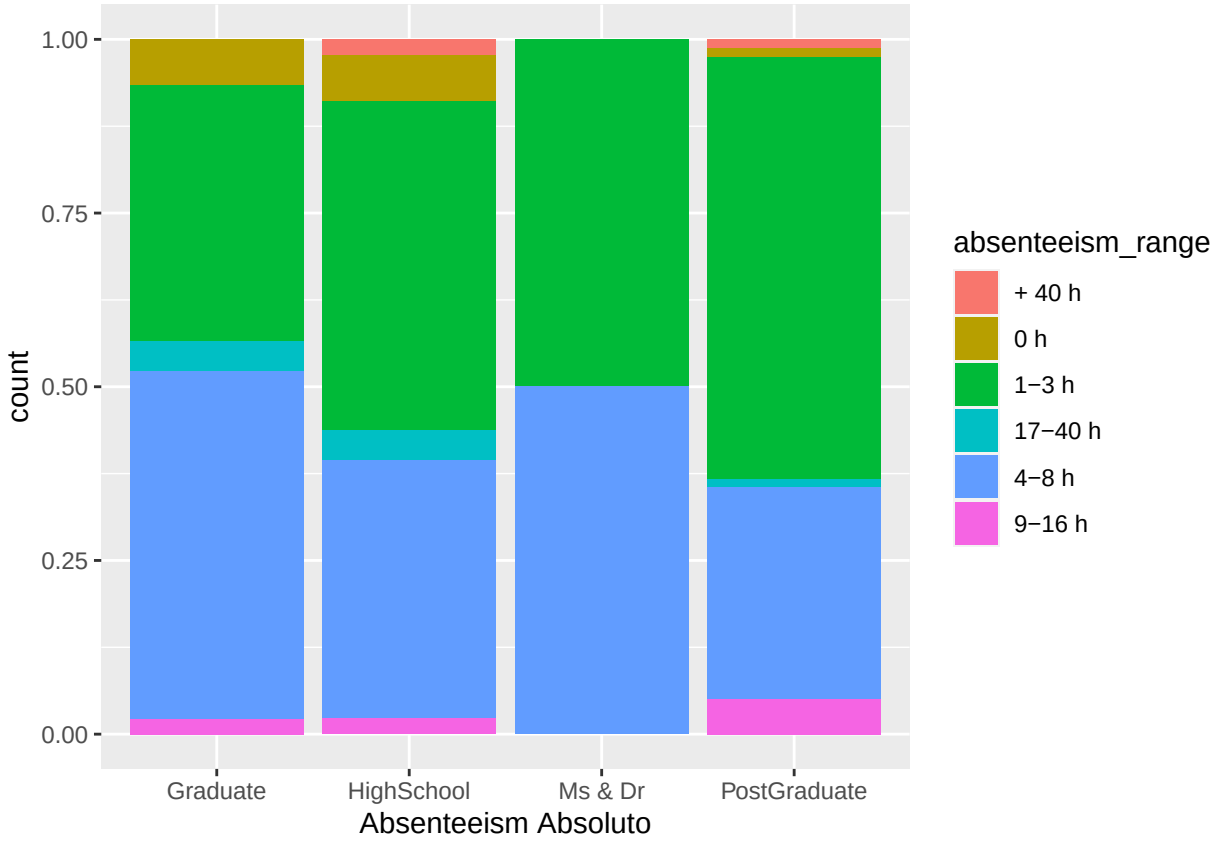

```
# Horas de Ausentimos discretizada
ant = ggplot(abs_df)+
  geom_bar(aes(x=forcats::fct_infreq(as.factor(Absenteeism.time.in.hours) )),
           fill="lightblue")+xlab("Absenteeism Before")+theme_bw()

desp = ggplot(abs_df)+
  geom_bar(aes(x=forcats::fct_infreq(as.factor(absenteeism_range) )),
           fill="lightblue")+xlab("Absenteeism After")+theme_bw()

grid.arrange(ant,desp, nrow = 1, ncol = 2)
```



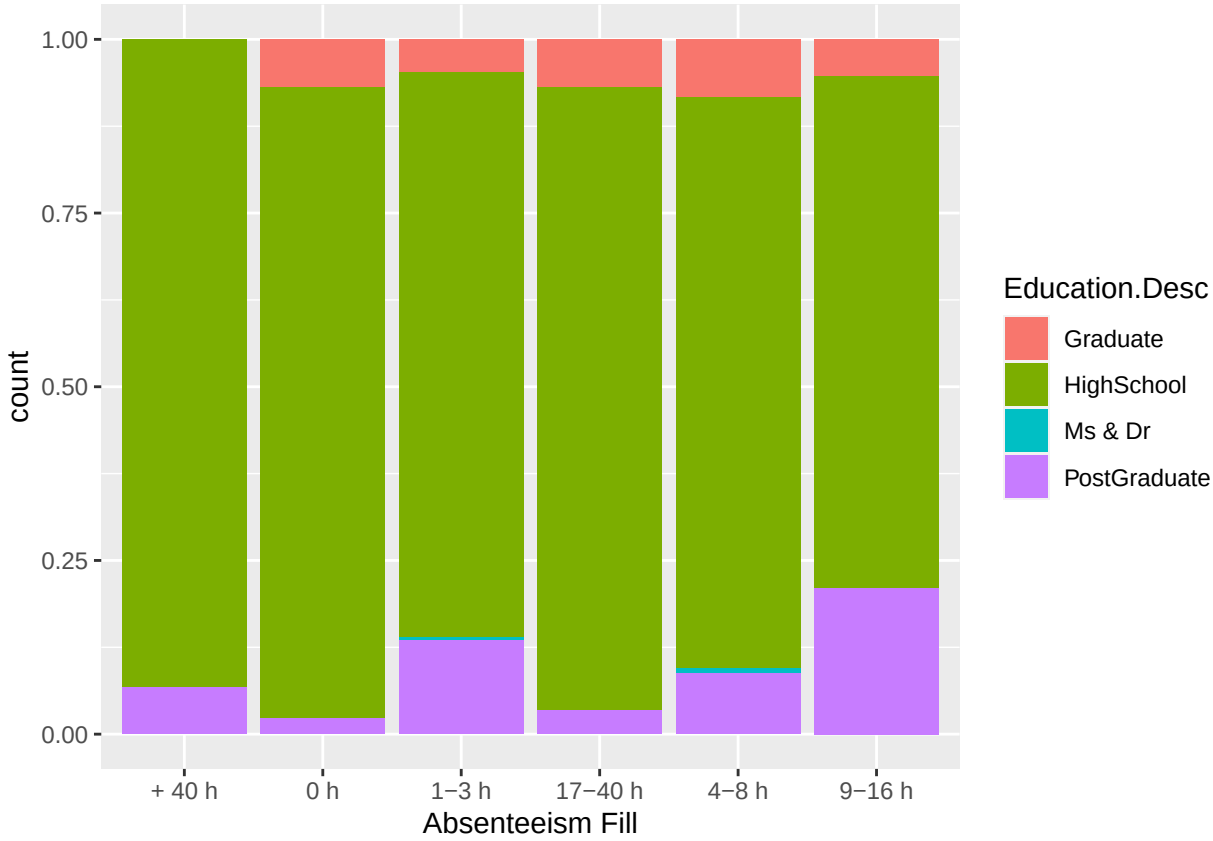
Comencemos ahora a contrastar variables, encontrar correlaciones, etc.. en definitiva entender aun mejor los datos.

- Education y Ausentismo

```
# Segun el nivel educativo no cambia mucho el % de horas de ausentismo es parejo
# en todos los niveles, el mayor porcentaje ronda entre unas horas y un dia
# entero. No mucho mas que eso.

ggplot(abs_df,aes(x=Education.Desc,fill=absenteeism_range))+
  geom_bar(position="fill")+xlab("Absenteeism Absoluto")
```



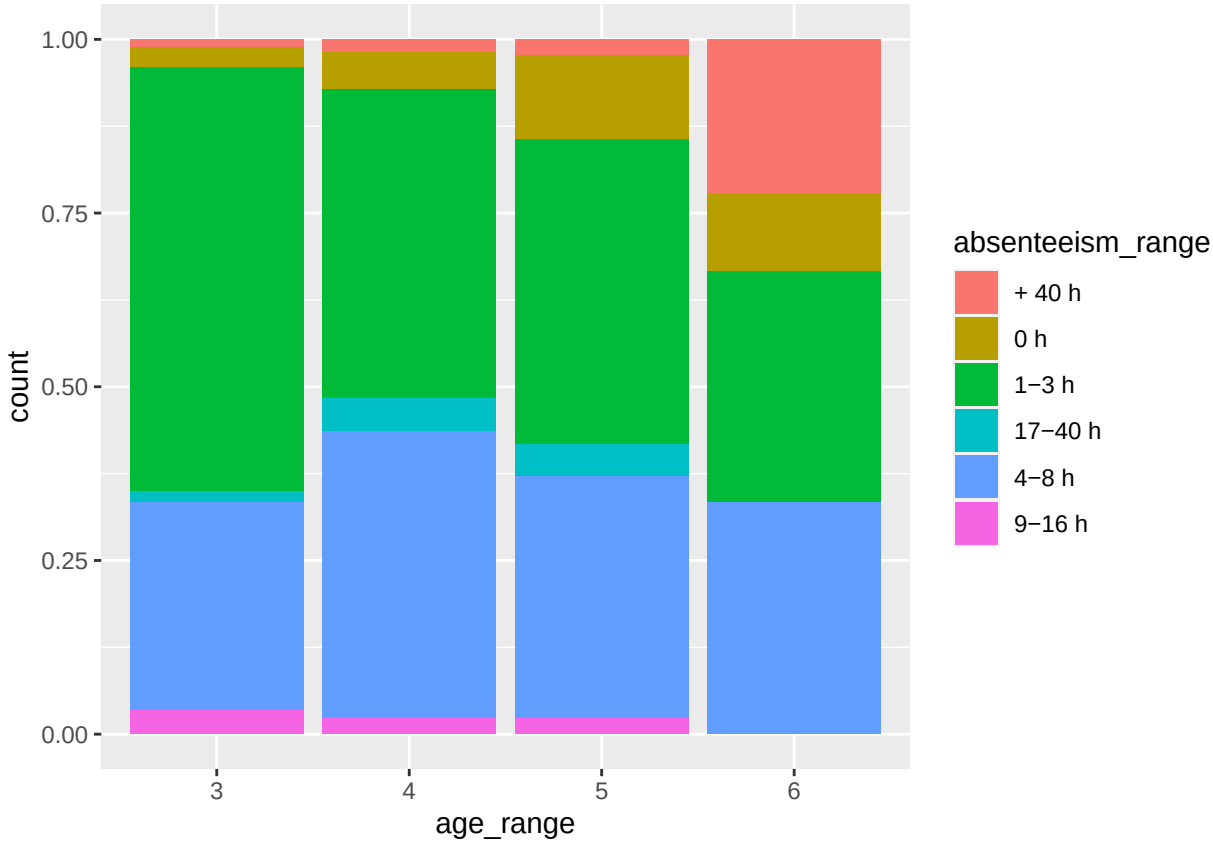

```
# Pero ahora, si es muy evidente que la mayor cantidad de personas que se
# ausentan son las que alcanzaron solo el High School, en el resto de los niveles
# el ausentismo se reduce notablemente.
ggplot(abs_df,aes(x=absenteeism_range,fill=Education.Desc))+
  geom_bar(position="fill")+xlab("Absenteeism Fill")
```



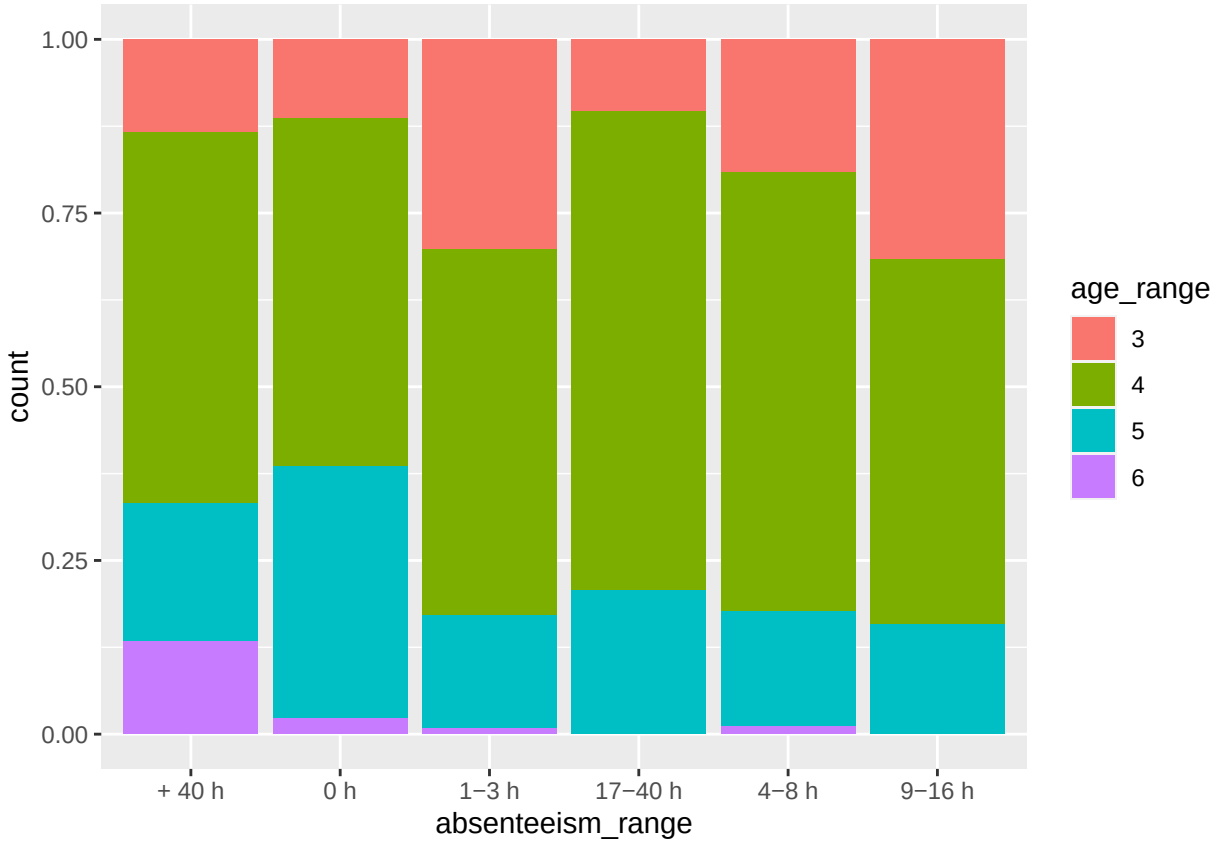
Esto no implica que la gente con menor educacion falte mucho o no, pero al menos en los datos que tenemos es un punto a considerar y tenerlo presente. Pero si quizas tenga que ver con la edad tal vez? veamos...

- Age y Ausentismo

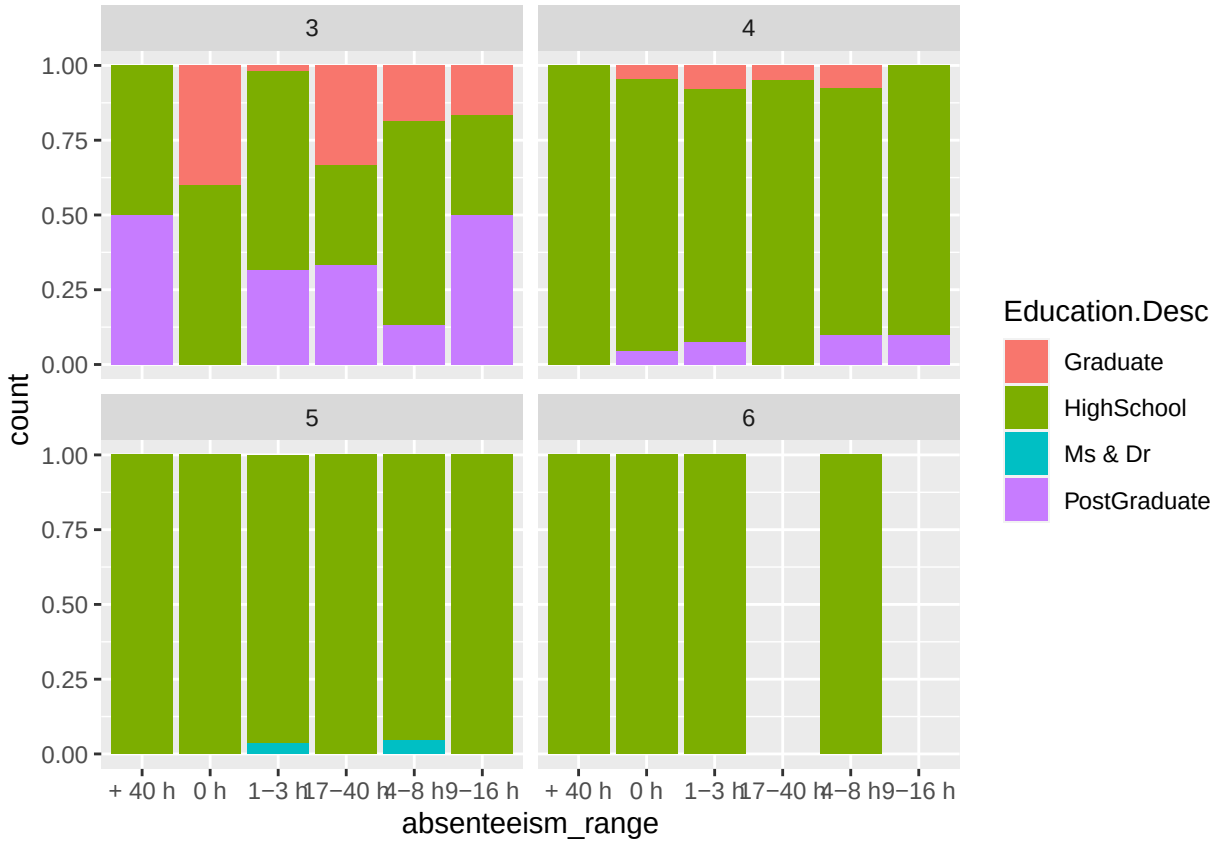
```
#Ahora analicemos la edad, el ausentismo y como dijimos antess combinemoslo con
# la educacion, en particular para los de high school.

# En general en "porcentaje" se ausentan mas veces los de 20 años, pero si vemos
# a que medida que aumenta la edad, aumenta tambien la cantidad de horas que se
# ausentan los empleados, se ve esa minima tendencia aqui:
ggplot(abs_df,aes(x=age_range,fill=absenteeism_range))+
  geom_bar(position="fill")
```



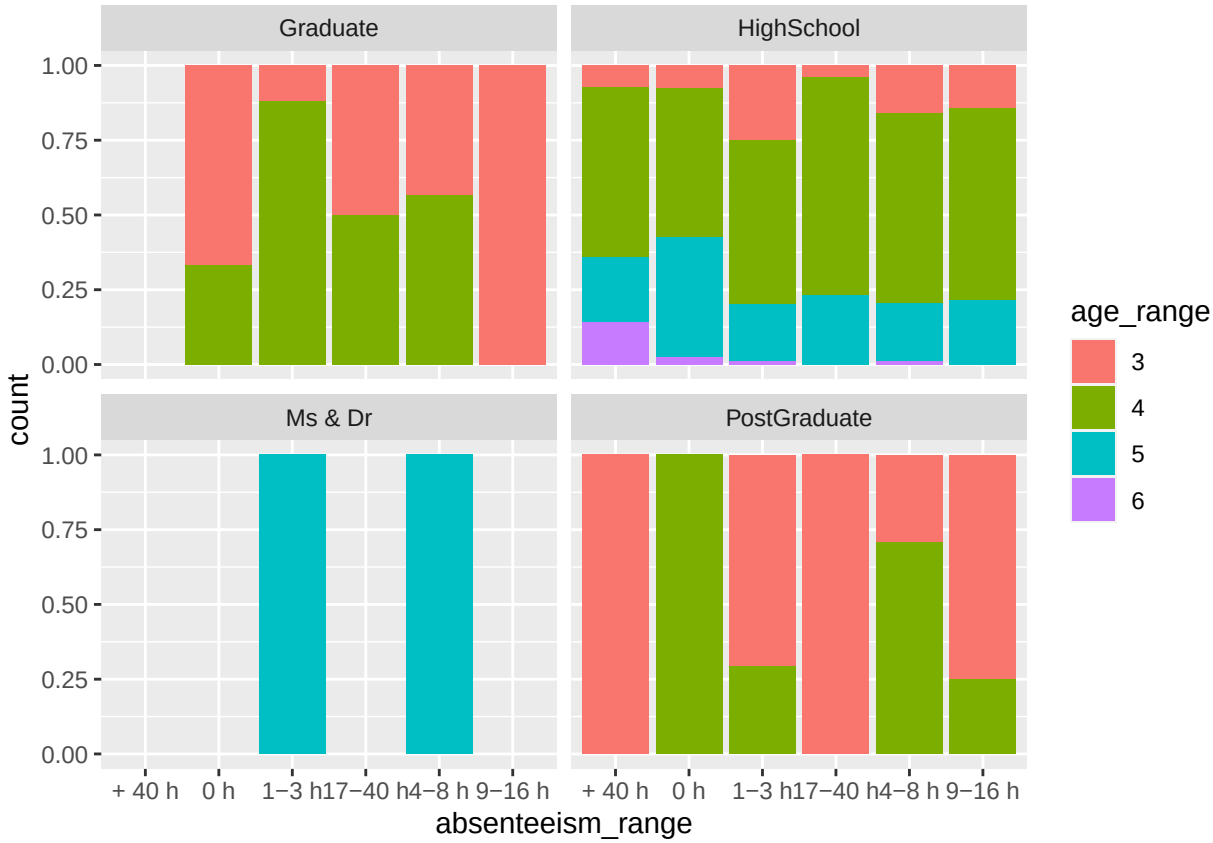

```
# Y por lo que terminamos viendo aqui, los que mas se ausentan en realidad son
# los del rango de los 30 años.
ggplot(abs_df,aes(x=absenteeism_range,fill=age_range))+
  geom_bar(position="fill")
```




```
# Y con este ultima grafica pareciera que los mas faltan son los que llegaron
# hasta el high school, y son mayores de 30 años.
ggplot(abs_df,aes(x=absenteeism_range,fill=Education.Desc))+
  geom_bar(position="fill")+facet_wrap(~age_range)
```

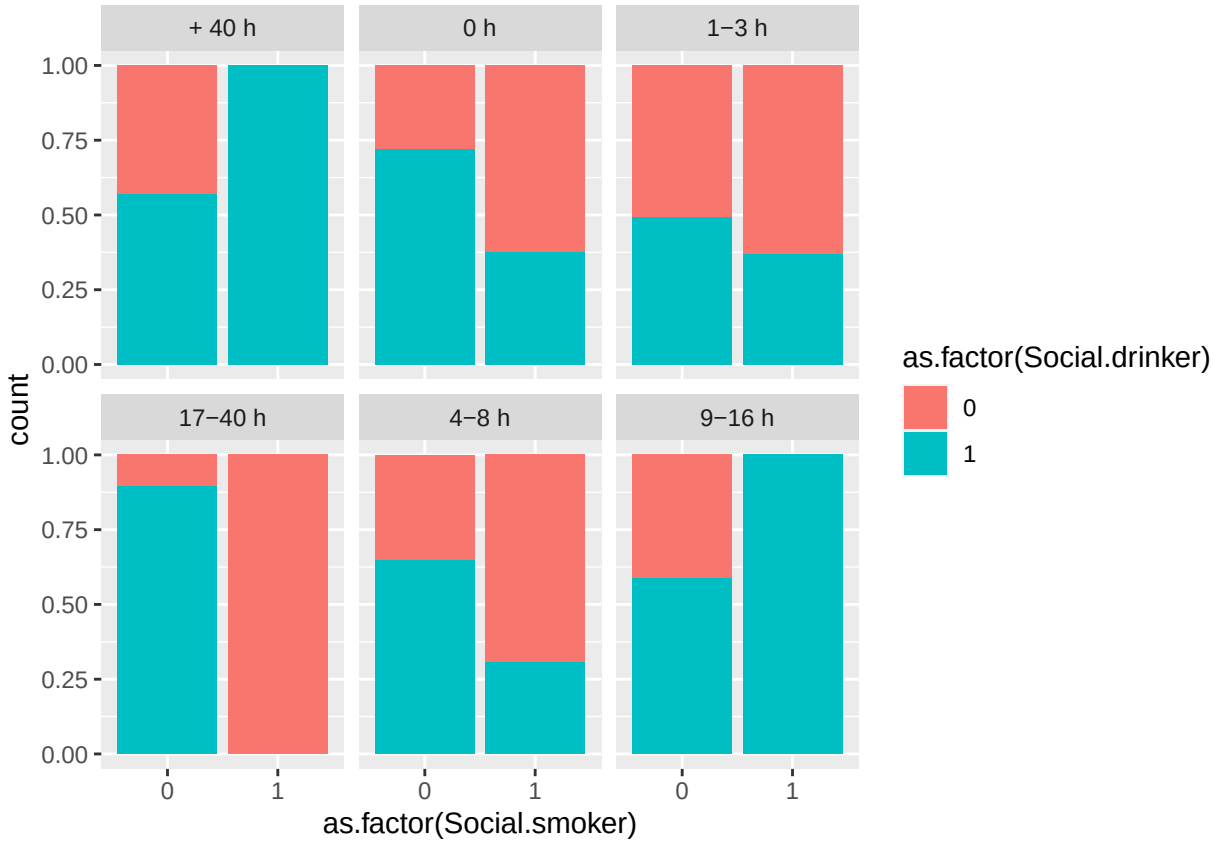



```
# pero no podemos decir lo mismo si lo vemos de esta forma, ya que en porcentajes
# tanto graduados como tambien postgraduados tienen altos porcentajes de
# ausentismo incluso en el rango de los 20 años.
ggplot(abs_df,aes(x=absenteeism_range,fill=age_range))+
  geom_bar(position="fill")+facet_wrap(~Education.Desc)
```

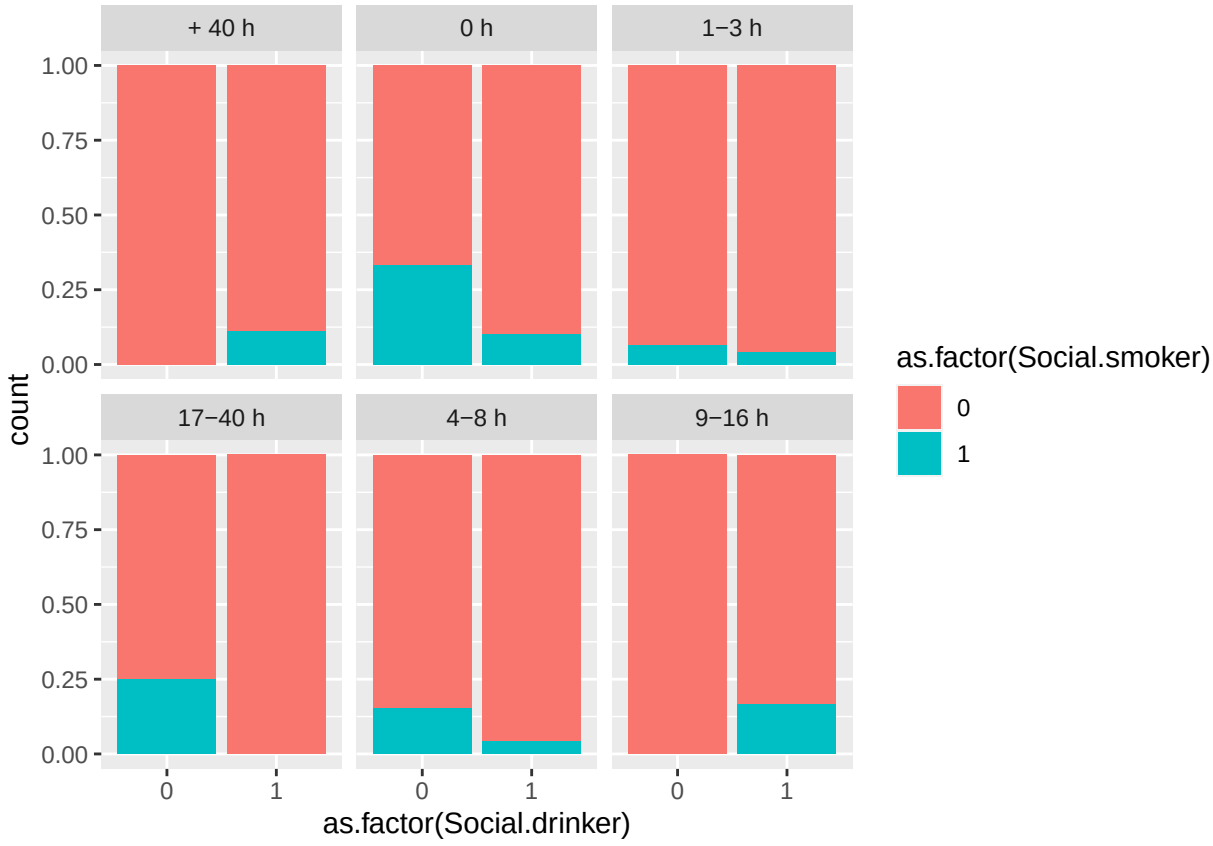


- Smokers & Drinkers

```
# Con estas dos comparaciones podemos ver que el fumador tiene menos peso que el
# bebedor, ya que un fumador puede fumar o no, podemos decir que hay un 50 y 50.
# Pero es claro viendolo desde el punto de vista de los bebedores sociales: hay
# muy pocos fumadores, sea cual sea su condicion. Eso es bueno mas alla del
# dataset, un vicio menos!

smoker = ggplot(abs_df,aes(x=as.factor(Social.smoker),
                           fill=as.factor(Social.drinker)))+
  geom_bar(position="fill")+
  facet_wrap(~absenteeism_range)

drinker = ggplot(abs_df,aes(x=as.factor(Social.drinker),
                            fill=as.factor(Social.smoker)))+
  geom_bar(position="fill")+
  facet_wrap(~absenteeism_range)

smoker
```


+ 40 h
0 h
1−3 h
17−40 h
4−8 h
9−16 h
count
1.00
0.75
0.50
0.25
0.00
0
1
as.factor(Social.smoker)
as.factor(Social.drinker)
0
1

```
drinker
```

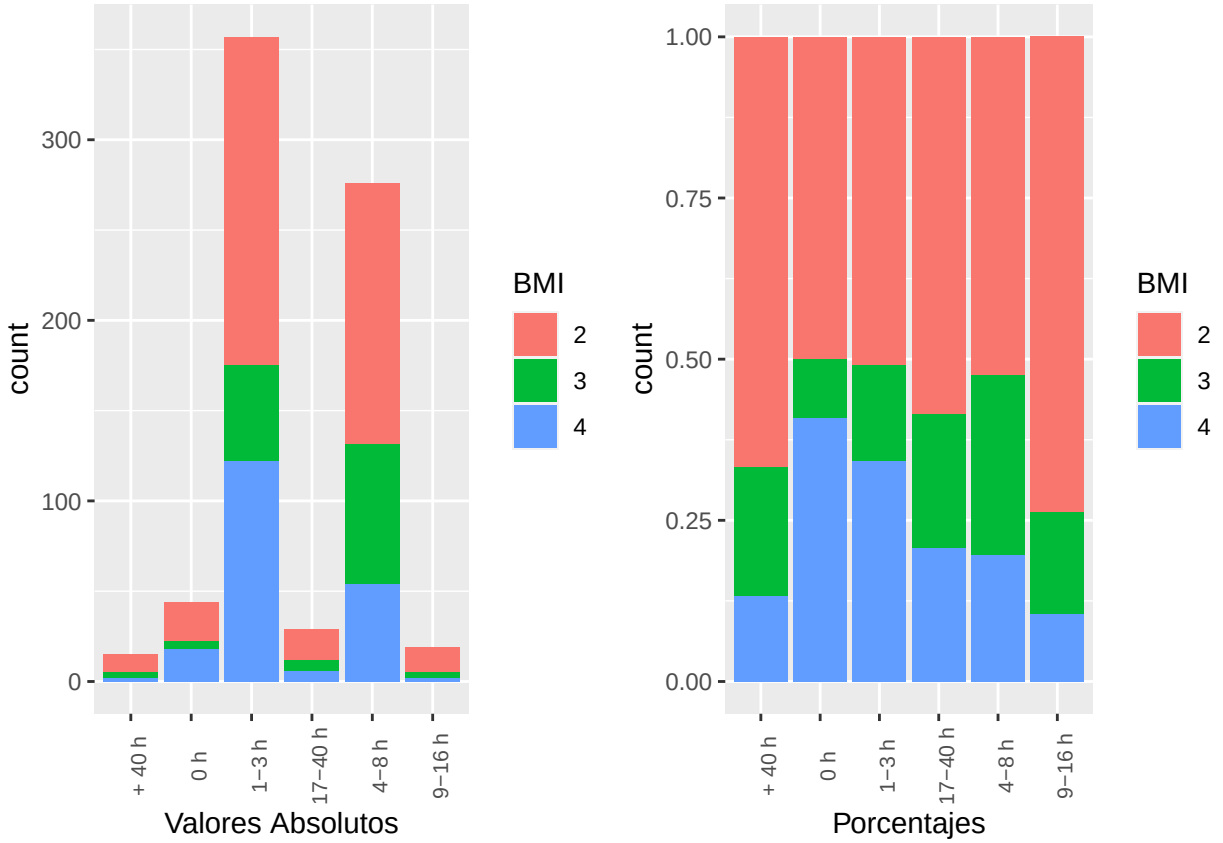



```
# Digamos que tienen mas tendencia a faltar los bebedores sociales y no asi
# los fumadores.
```

Pero que pasa ahora con el indice de masa corporal y la distancia al trabajo?

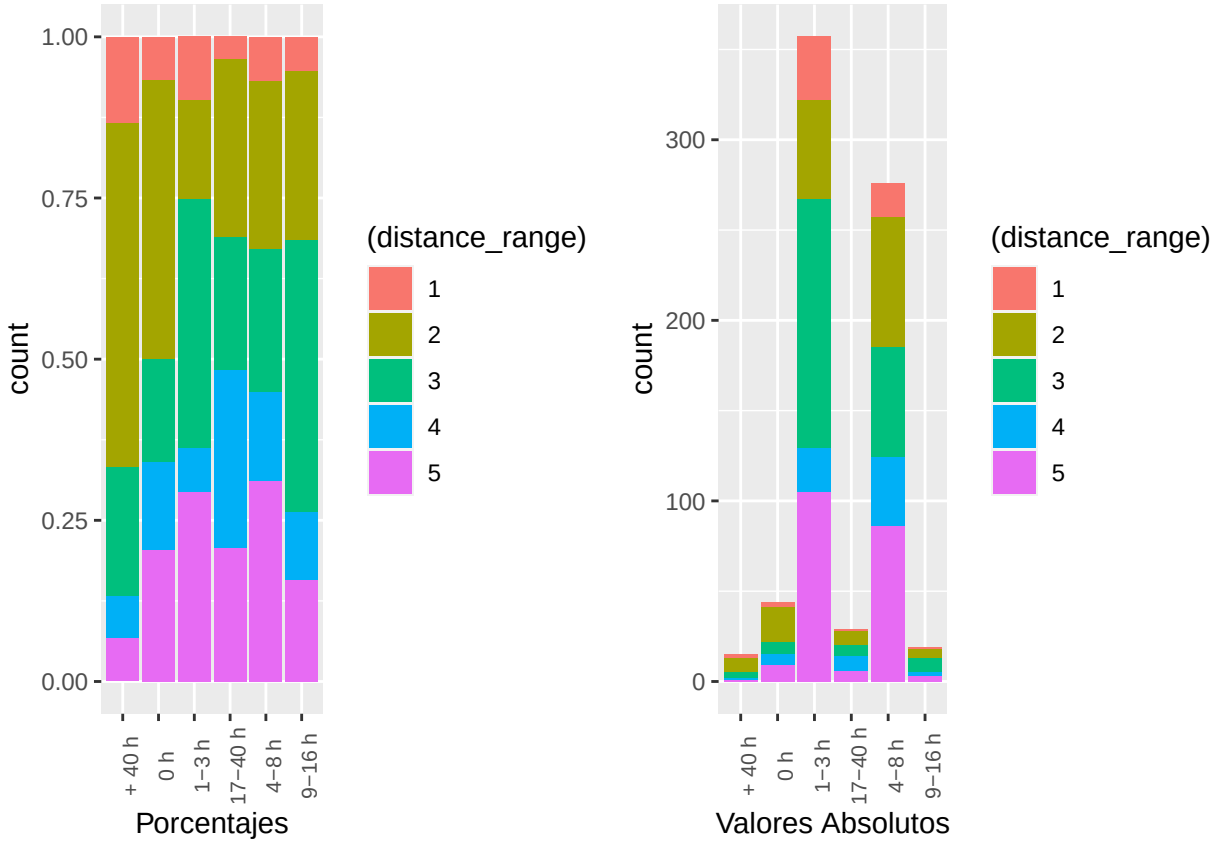
- BMI y Ausentismo

```
# en valores absolutos parece no haber una gran diferencia entre las personas
# de peso normal vs los empleados con sobrepeso u obesos
abs = ggplot(abs_df,aes(x=absenteeism_range,fill=BMI))+
  geom_bar()+
  theme(axis.text.x=element_text(size=8,angle=90))+
  xlab("Valores Absolutos")

# Pero si lo vemos en porcentajes, parece que en general los empleados con peso
# normal faltaran mas, por lo que tener sobrepeso no es algo que afecte al presentimo.
per = ggplot(abs_df,aes(x=absenteeism_range,fill=BMI))+
  geom_bar(position="fill")+
  xlab("Porcentajes")+
  theme(axis.text.x=element_text(size=8,angle=90))

grid.arrange(abs,per, nrow = 1, ncol = 2)
```



- Distancia y Ausentismo

```
g1 = ggplot(abs_df,aes(x=absenteeism_range,fill=(distance_range)))+
  geom_bar(position="fill")+xlab("Porcentajes")+
  theme(axis.text.x=element_text(size=8,angle=90))

g2 = ggplot(abs_df,aes(x=absenteeism_range,fill=(distance_range)))+
  geom_bar()+xlab("Valores Absolutos")+
  theme(axis.text.x=element_text(size=8,angle=90))

# Con los rangos de distancia que armamos no parece haber una gran diferencia o
# algo que nos indique que rango falta mas o menos, pero si hubieramos agrupado
# en 2 rangos entre menos de 20km vs mas de 20km, ahi veriamos claramente una
# diferencia. Hay una gran diferencia entre ambas rangos de distancia. Cuanto mas
# lejos vivan parece que hubiera una tendencia a que se ausenten mas.

grid.arrange(g1,g2, nrow = 1, ncol = 2)
```



Podriamos seguir combinando variables, e incluso comenzar a usar una variable mas que importante: Reason.for.absence, para contrastarla con las horas y todas las demas variables, pero como sabemos son muchas en nuestro dataset para hacer esto secuencialmente.

Tal vez para acotar mejor el analisis y determinar mas facil clusters u obtener reglas o patrones mas sencillos debamos acotar el nro de variables aplicando tecnicas de reduccion dimensionalidad.

En este caso usaremos PCA, tecnica que nos permite fusionar o crear nuevos atributos a partir de los existentes. A grandes rasgos, PCA es una transformación lineal de las variables. Cada nueva variable de un registro se contruye a partir de la antiguas variables de ese mismo registro a través de una transformación lineal fija.

Veamoslo en detalle.

4.2 Reduccion de Dimensionalidad: PCA

Si bien a nivel macro los pasos para aplicar PCA son:

Estandarizar variables

Calcular la matriz de covarianzas MC

Generar los eigenvectores y eigenvalores a partir de la matriz MC

Ordenar los eigenvectores a partir de los eigenvalores de manera descendente y quedarnos con los TOP k eigenvectores

Contruir la matriz de projecciones MP, a partir de los eigenvectores seleccionados

Transformar el dataset original a partir de la matriz MP para obtener las nuevas dimensiones.

Nos apoyaremos en las funciones existentes en R que permiten calcular directamente las componentes principales y los principal component scores de cada observación sin tener que ir paso por paso.

Comencemos analizando la normalidad y homocesticidad de las variables:

4.2.1. Normalidad y Homocedasticidad Tanto la normalidad como la homocedasticidad puede validar con los siguientes test:

- Test normalidad de Anderson-Darling.
- leveneTest para variables normales o fligner.test para las que no los son.

```
alpha = 0.05
col.names = colnames(abs_df)
for (i in 2:21) {
  if (i == 2) cat("Variables que no siguen una distribución normal:\n")
    if ( is.integer(unlist(abs_df[,i])) | is.numeric(unlist(abs_df[,i])) ) {
      p_val = ad.test(as.vector(unlist(abs_df[,i])))$p.value
      if (p_val < alpha) {
        cat(col.names[i])
        # Format output
        if (i < 21)
            cat(", ")
        if (i %% 3 == 0)
            cat("\n")
      }
    }
}
```

```
## Variables que no siguen una distribución normal:
## Reason.for.absence, Month.of.absence,
## Day.of.the.week, Seasons, Transportation.expense,
## Distance.from.Residence.to.Work, Service.time, Age,
## Work.load.Average.day., Hit.target, Disciplinary.failure,
## Education, Son, Social.drinker,
## Social.smoker, Pet, Weight,
## Height, Body.mass.index, Absenteeism.time.in.hours
```

Como vemos la mayoria no siguen una distribucion normal segun el test, pero recordemos que por el TLC, teorema del limite central, podemos asumirla dada la cantidad de observaciones que tenemos en nuestro dataset.

Pero que pasa con la varianza?

Validemos algunas de las variables que hemos estando analizando visualmente antes. Donde contrastaremos la cantidad de horas de ausentismo vs diferentes cantidad de horas de servicio, de hijos, de fallos disciplinarios, etc…

```
leveneTest(Absenteeism.time.in.hours  ~ as.factor(abs_df$Service.time), abs_df)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group  17  0.9019 0.5718
##       722
```

```
leveneTest(Absenteeism.time.in.hours  ~ as.factor(abs_df$Son), abs_df)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value   Pr(>F)
## group   4  3.6116 0.006325 **
##       735
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Absenteeism.time.in.hours  ~ as.factor(abs_df$Disciplinary.failure), abs_df)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value  Pr(>F)
## group   1  6.4085 0.01156 *
##       738
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Absenteeism.time.in.hours  ~ as.factor(abs_df$Age), abs_df)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value   Pr(>F)
## group  21  2.8861 1.84e-05 ***
##       718
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Absenteeism.time.in.hours  ~ as.factor(abs_df$Transportation.expense), abs_df)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group  23  2.7765 1.915e-05 ***
##       716
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Como vemos en algunos casos las homocedasticidad se cumple como en otros no. Por lo que podemos afirmar que no para todas las variables se cumple la homogeneidad de varianzas.

Dada esta situacion deberiamos estandarizar las variables para que tengan media cero y desviación estándar 1 antes de realizar el estudio PCA y reducir la dimensionalidad de nuestro dataset para la posterior creacion de un modelo predictivo solo con las variables relevantes o de mayor peso.

Pero este paso podemos "saltearlo" haciendo True el parametro scale de la funcion prcomp como vemos a continuacion:

```
set.seed(123)

# Con esta funcion prcomp podemos estandarizar las variables y hacer que la
# desviacion estandar sea 1 con el parametro scale=TRUE
pca <- prcomp(abs_df[,2:20], scale = TRUE)
names(pca)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

```
# Donde la variable center contiene la media de cada variable antes de estandarizar
sort(pca$center, decreasing =TRUE)
```

```
##          Work.load.Average.day.          Transportation.expense
##                      271.49023514                      221.32972973
##                          Height                        Hit.target
##                      172.11486486                       94.58783784
##                          Weight                               Age
##                       79.03513514                       36.45000000
## Distance.from.Residence.to.Work                 Body.mass.index
##                       29.63108108                       26.67702703
##               Reason.for.absence                    Service.time
##                       19.21621622                       12.55405405
##                 Month.of.absence                   Day.of.the.week
##                        6.32432432                        3.91486486
##                           Seasons                         Education
##                       2.54459459                        1.29189189
##                               Son                               Pet
##                       1.01891892                        0.74594595
##                    Social.drinker                    Social.smoker
##                       0.56756757                        0.07297297
##             Disciplinary.failure
##                       0.05405405
```

```
# Mientras que rotation contiene el valor de los loadings  para cada componente
# (eigenvector). El número de componentes principales se corresponde con el
# mínimo(n-1,p), que en este caso es  min(740,19)=19
pca_rot = pca$rotation

# veamos algunas registros:
head(pca_rot,5)
```

```
##                                  PC1         PC2         PC3         PC4
## Reason.for.absence     -0.02819557  0.10824296 -0.50294205  0.23071901
## Month.of.absence       -0.01133352 -0.29010416  0.27854472  0.45861647
## Day.of.the.week         0.04348821 -0.05700793 -0.13294692 -0.08425603
## Seasons                 0.02069323 -0.14699657  0.30468248  0.25516352
## Transportation.expense  0.18242140 -0.46028241 -0.05169362 -0.11065066
##                                  PC5         PC6         PC7         PC8
## Reason.for.absence     -0.1359912  0.19269674 -0.1363563  0.27511166
## Month.of.absence       -0.2078847  0.11853275 -0.2914655  0.06887394
## Day.of.the.week        -0.3718443  0.26059590  0.1373242 -0.40101547
## Seasons                -0.1954020  0.24341794  0.2297925  0.08018879
## Transportation.expense  0.1471083 -0.03179337 -0.0911628 -0.02284105
##                                  PC9        PC10         PC11        PC12
## Reason.for.absence     -0.08490446  0.07318197 -0.037927415  0.05527255
## Month.of.absence       -0.03198799 -0.01454097 -0.156032913  0.01569713
## Day.of.the.week        -0.32233988  0.64648994  0.001324067  0.18287631
## Seasons                -0.53357274 -0.36525114 -0.191210020 -0.06023595
## Transportation.expense -0.07738452  0.03194276  0.224638871 -0.41531744
##                                 PC13        PC14        PC15        PC16
```

```
## Reason.for.absence      0.43972888 -0.31328031  0.41617094 -0.20989640
## Month.of.absence       -0.10901176  0.18732626 -0.17312517 -0.61274975
## Day.of.the.week        -0.03285673  0.14298624 -0.06871459  0.04450177
## Seasons                 0.06573209 -0.09645378  0.12410946  0.41550227
## Transportation.expense  0.25904157  0.48463950  0.28591388  0.03242163
##                               PC17         PC18          PC19
## Reason.for.absence      0.074603559  0.004108610  0.0003328799
## Month.of.absence       -0.035818727 -0.035482319 -0.0073819026
## Day.of.the.week         0.010469203 -0.024097950 -0.0045097737
## Seasons                 0.004841727  0.057524380  0.0074277598
## Transportation.expense  0.305975639 -0.009977971 -0.0025019038
```

```
# Ahora si las varianzas estan mas igualadas post estandarizacion:
apply(X = pca_rot, MARGIN = 2, FUN = var)
```

```
##        PC1        PC2        PC3        PC4        PC5        PC6        PC7
## 0.05077273 0.04942371 0.05543370 0.05235815 0.05547919 0.05241982 0.05549947
##        PC8        PC9       PC10       PC11       PC12       PC13       PC14
## 0.05554494 0.04295815 0.05403742 0.05101497 0.04686775 0.05426368 0.05334386
##       PC15       PC16       PC17       PC18       PC19
## 0.05504891 0.04935615 0.05542957 0.05543467 0.05531318
```

Entender el vector de loadings que forma cada componente nos puede ayudar a interpretar que clase de información capta cada componente, por ejemplo si miramos la componente 1:

```
# lo que vemos es que la primer componente capta mas informacion de la variable
# Education, cantidad de mascotas y gastos de viaje positivamente y negativamente
# del BMI, Weight y Service Time.
sort(pca_rot[,'PC1'], decreasing =TRUE)
```

```
##                         Education                               Pet
##                        0.25698073                        0.19723000
##          Transportation.expense                      Social.smoker
##                        0.18242140                        0.08099409
##                        Hit.target                               Son
##                        0.05532888                        0.05256902
##                  Day.of.the.week           Work.load.Average.day.
##                        0.04348821                        0.03205467
##                           Seasons                 Month.of.absence
##                        0.02069323                       -0.01133352
##               Reason.for.absence             Disciplinary.failure
##                       -0.02819557                       -0.03835907
##                            Height Distance.from.Residence.to.Work
##                       -0.03857923                       -0.05859953
##                    Social.drinker                               Age
##                       -0.29997218                       -0.37866379
##                      Service.time                            Weight
##                       -0.43010409                       -0.45566890
##                  Body.mass.index
##                       -0.46123912
```

```
# y asi podriamos seguir analizando el resto de las componentes para entender
# que variables tienen mas pesos sobre otras.


# Otra de las variables que genera la funcion prcomp es la matriz x, la cual es
# el resultado de multiplicar los datos por los loadings
head(pca$x)
```

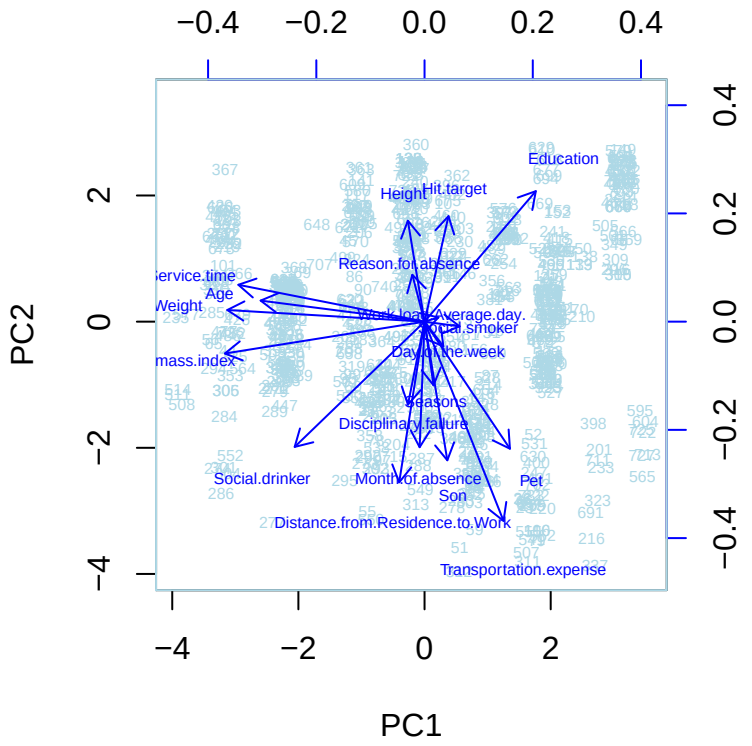
```
##             PC1        PC2        PC3         PC4        PC5         PC6
## [1,] -0.8018106 -0.9387058 -1.5527538 -0.19948102  1.2054350 -0.25893097
## [2,] -3.3699259  0.6018233  3.3403943 -1.90361666  0.9849275 -1.50934070
## [3,] -2.2076221  0.2074964 -1.7339801  0.20268274 -0.1276618 -0.83570947
## [4,]  0.5803615 -0.2707334  0.7169593 -3.11341328 -1.4862371 -0.46870314
## [5,] -0.7306018 -1.0574092 -1.5608717 -0.40008494  0.7307035  0.03912549
## [6,] -2.1464432  0.1272981 -1.9210087  0.08415207 -0.6507692 -0.46910535
##              PC7         PC8        PC9       PC10       PC11        PC12
## [1,] -1.31825553 -0.09549691  0.1149397 -0.2300605  0.5334458 -0.20528475
## [2,] -0.69812000 -2.30798782  0.5443390 -0.1275991  0.3325918  1.26159439
## [3,] -0.13697022 -0.91449311  0.5849612 -0.1386459 -0.7011554  0.07235169
## [4,] -2.11974336  0.36596138  0.4799788  0.4643850 -0.4768062  0.37149344
## [5,] -1.07656320 -0.75750705 -0.3083224  0.6533832  0.5488004  0.03232210
## [6,]  0.05621635 -1.47863829  0.1314962  0.7708307 -0.6992927  0.32962055
##            PC13        PC14       PC15       PC16        PC17        PC18
## [1,]  0.6830515  0.09658612 -0.3572178 -0.6749090  0.07316205 -0.38506799
## [2,] -0.4191963 -0.93995706  0.5979066 -1.4366324 -0.21531602 -0.24248623
## [3,] -0.1207314  0.28038899 -0.4409258 -0.7891401  0.13431385 -0.07381479
## [4,]  0.3896311  2.10169489 -0.5674894  0.2124616 -1.14009612 -1.07938200
## [5,]  0.4804050  0.40918058 -0.6019286 -0.5376382  0.06135145 -0.42043033
## [6,] -0.1669539  0.48154082 -0.5375929 -0.7265354  0.14904183 -0.10771558
##             PC19
## [1,]  0.05811001
## [2,]  0.04350669
## [3,] -0.02809902
## [4,]  0.02311105
## [5,]  0.05164728
## [6,] -0.03444333
```

```
# 740 valores x 19 componentes
dim(pca$x)
```

```
## [1] 740  19
```

Veamos como se ven en un grafico de dos dimesiones al menos las primeras dos componentes:

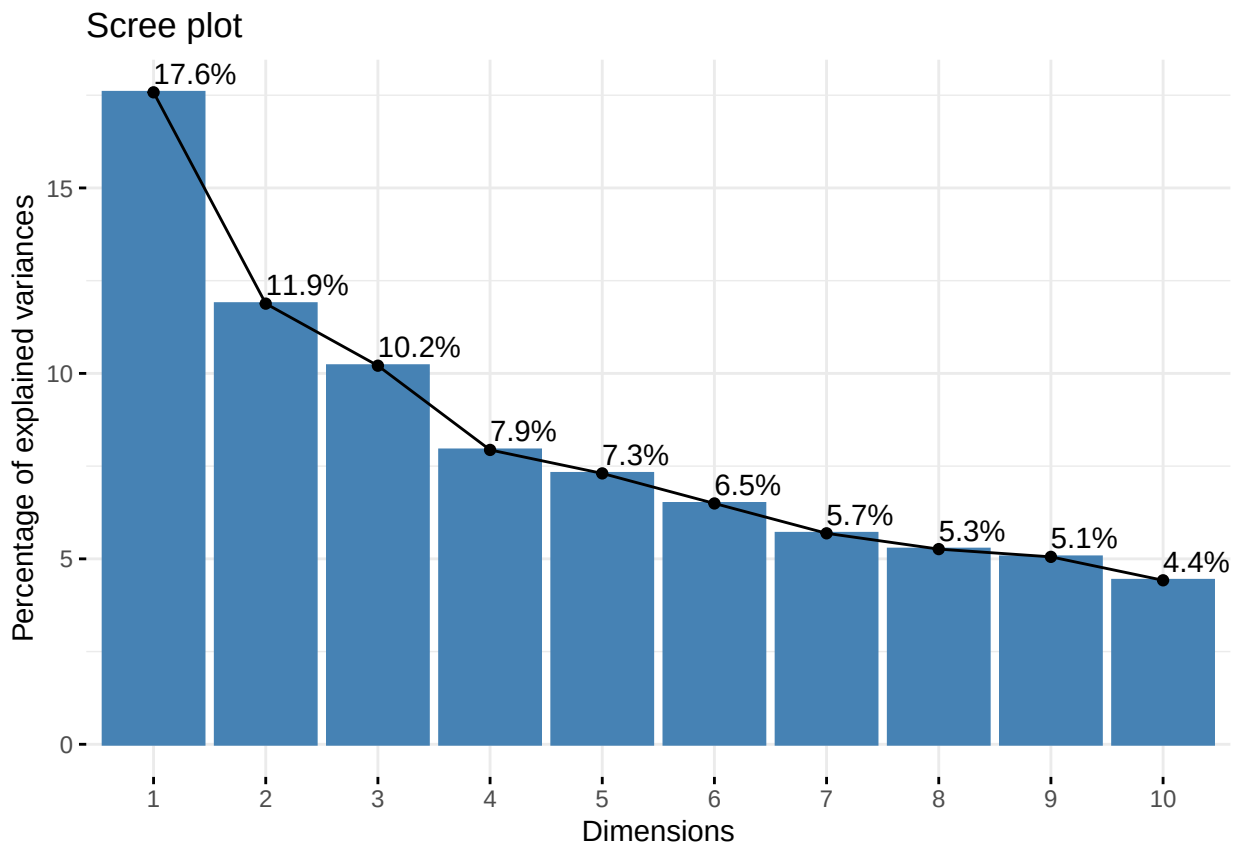
```
biplot(x = pca, scale = 0, cex = 0.5, col = c("lightblue", "blue"))
```




```
#fviz_pca_var(pca,col.var='blue')
```

Ya obtenidas las componentes principales, se puede saber cual es la varianza explicada por cada una de ellas, la proporción respecto a la varianza total y la proporción acumulada de la varianza.

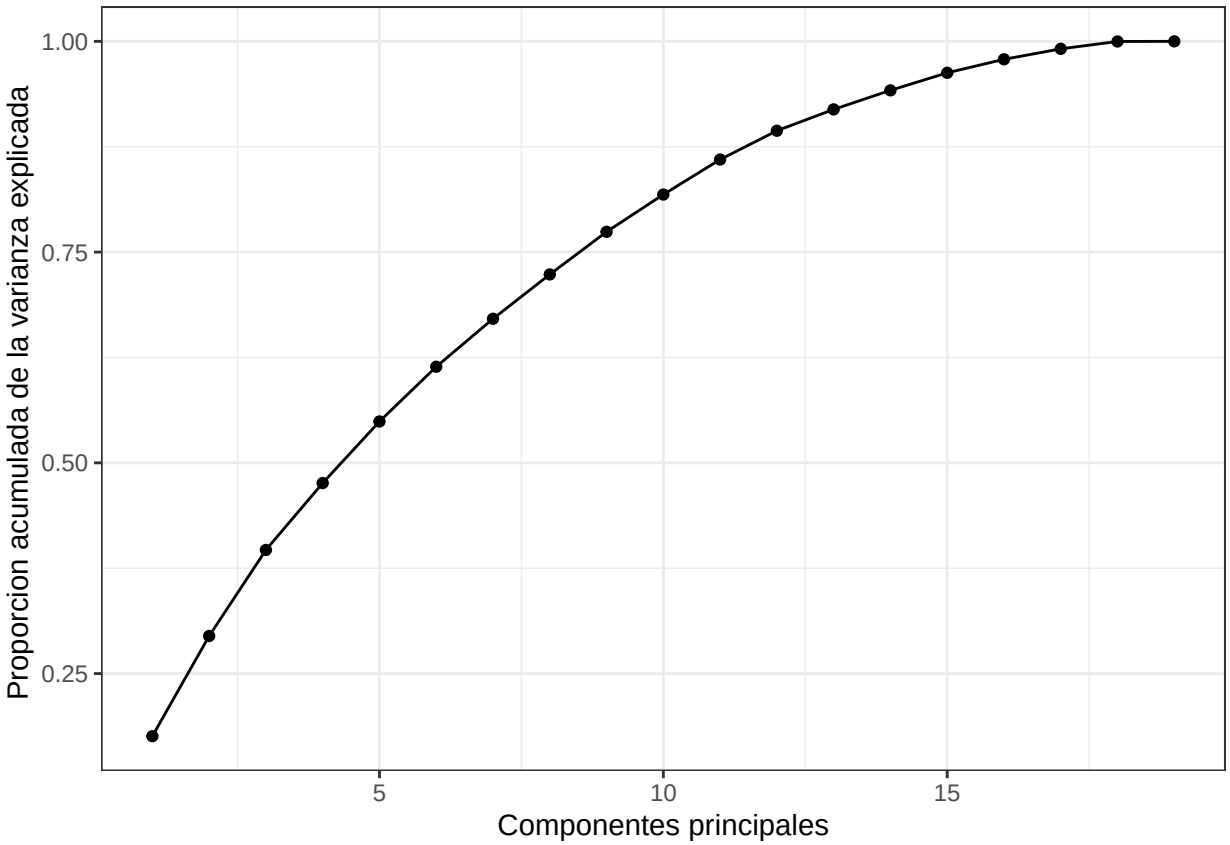
```
# Proporcion de la varianza explicada
fviz_eig(pca,addlabels=T)
```




```
# Proporcion acumulada de la varianza explicada
proporcion_varianza <- pca$sdev^2 / sum(pca$sdev^2)
proporcion_varianza_acum <- cumsum(proporcion_varianza)
proporcion_varianza_acum
```

```
##  [1] 0.1757789 0.2945901 0.3966667 0.4760345 0.5490732 0.6140134 0.6709011
##  [8] 0.7235278 0.7740651 0.8182839 0.8597975 0.8939602 0.9192645 0.9418648
## [15] 0.9626482 0.9786858 0.9910924 0.9998594 1.0000000
```

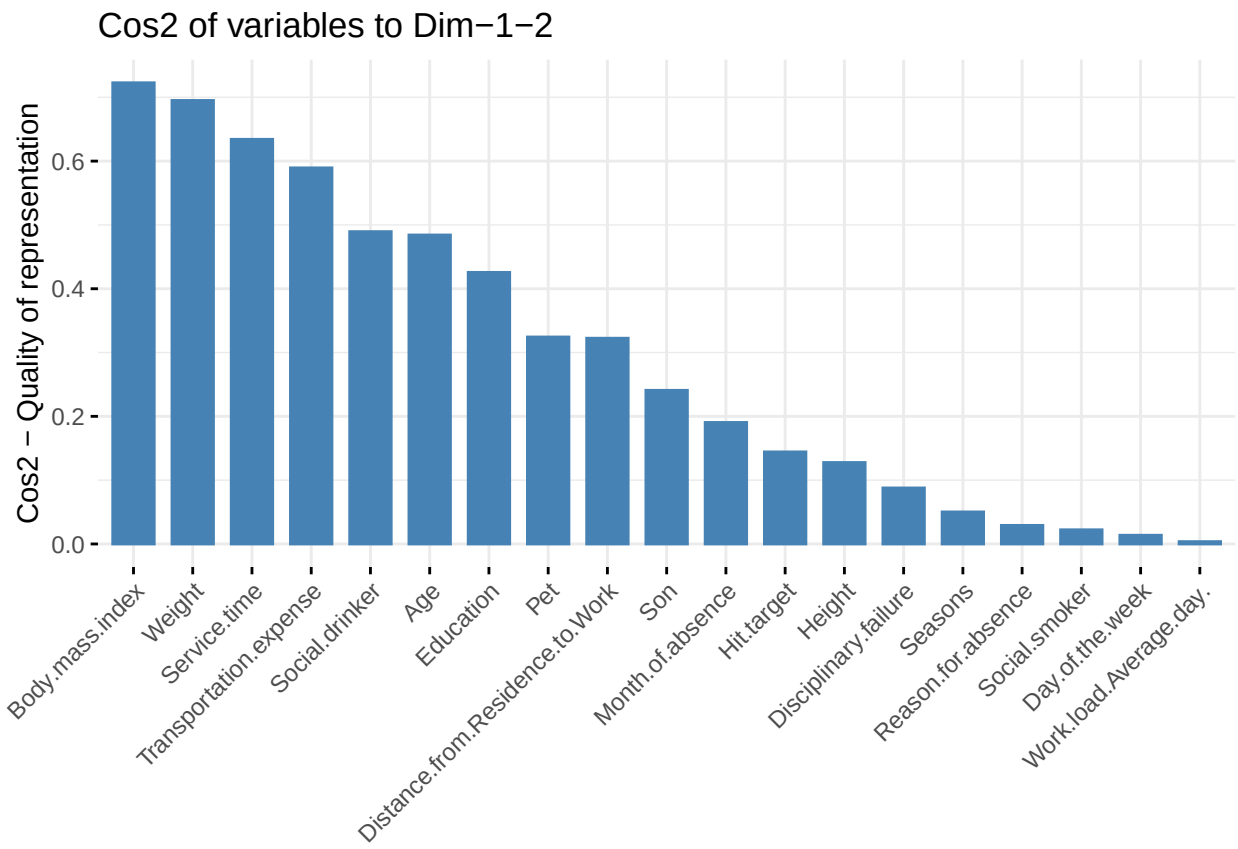
```
ggplot(data = data.frame(proporcion_varianza_acum, pc = 1:19),
       aes(x = pc, y = proporcion_varianza_acum, group = 1)) +
  geom_point() +
  geom_line() +
  theme_bw() +
  labs(x = "Componentes principales",
       y = "Proporcion acumulada de la varianza explicada ")
```



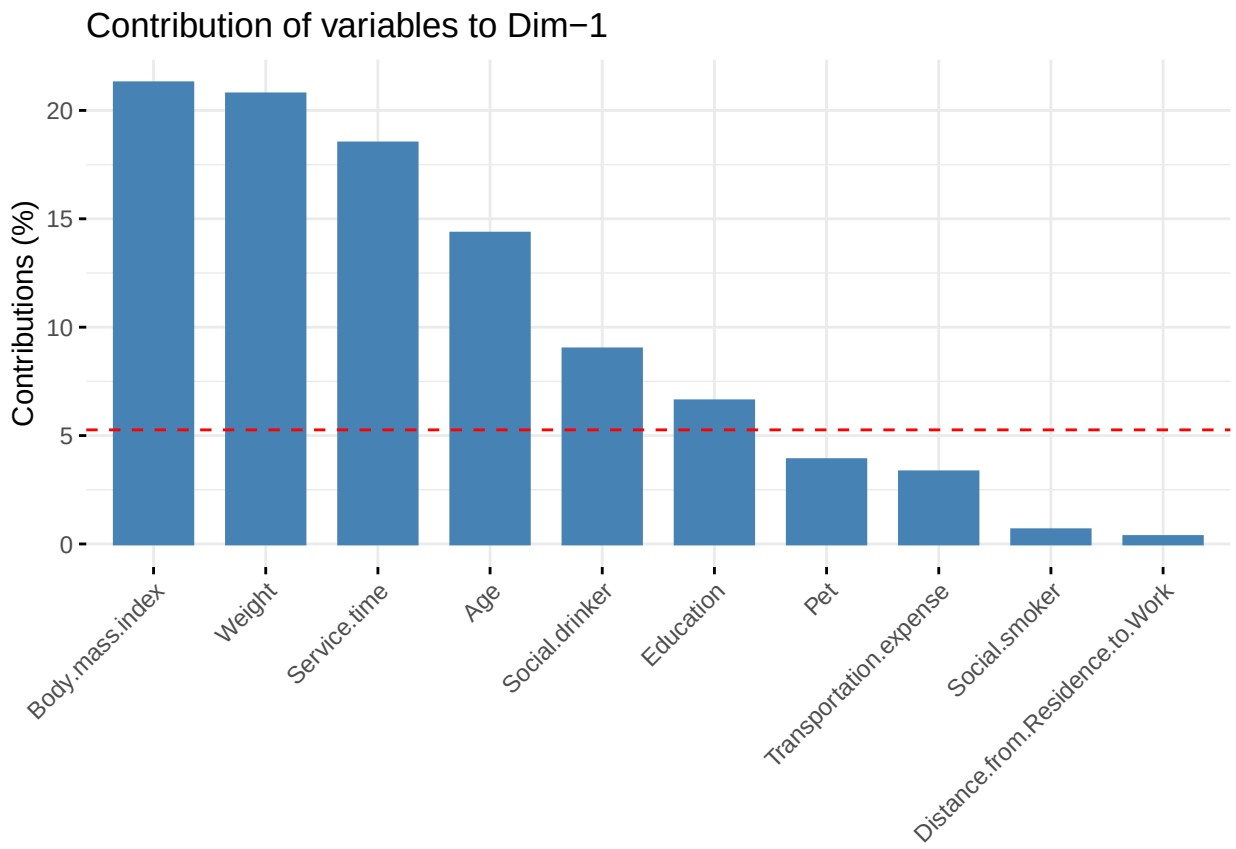
Dicho todo esto, si quisieramos por ejemplo explicar al menos un minimo de 70% de la varianza deberiamos usar las primeras 8 componentes principales. o 10 si quisieramos explicar el 80%. El objetivo siempre es buscar aquellas componentes que explican la maxima varianza, esto es porque, queremos retener la mayor cantidad de informacion posible usando estas componentes. Entonces, cuanto mayor es la varianza explicada, mayor sera la informacion contenida en estas componentes.

Veamos o determinemos para proximos analisis cuales son las variables mas relevantes en las primeras dos componentes, ya que cuando contamos con muchas variables, podríamos decidir mostrar solo aquellas con mayor contribución.

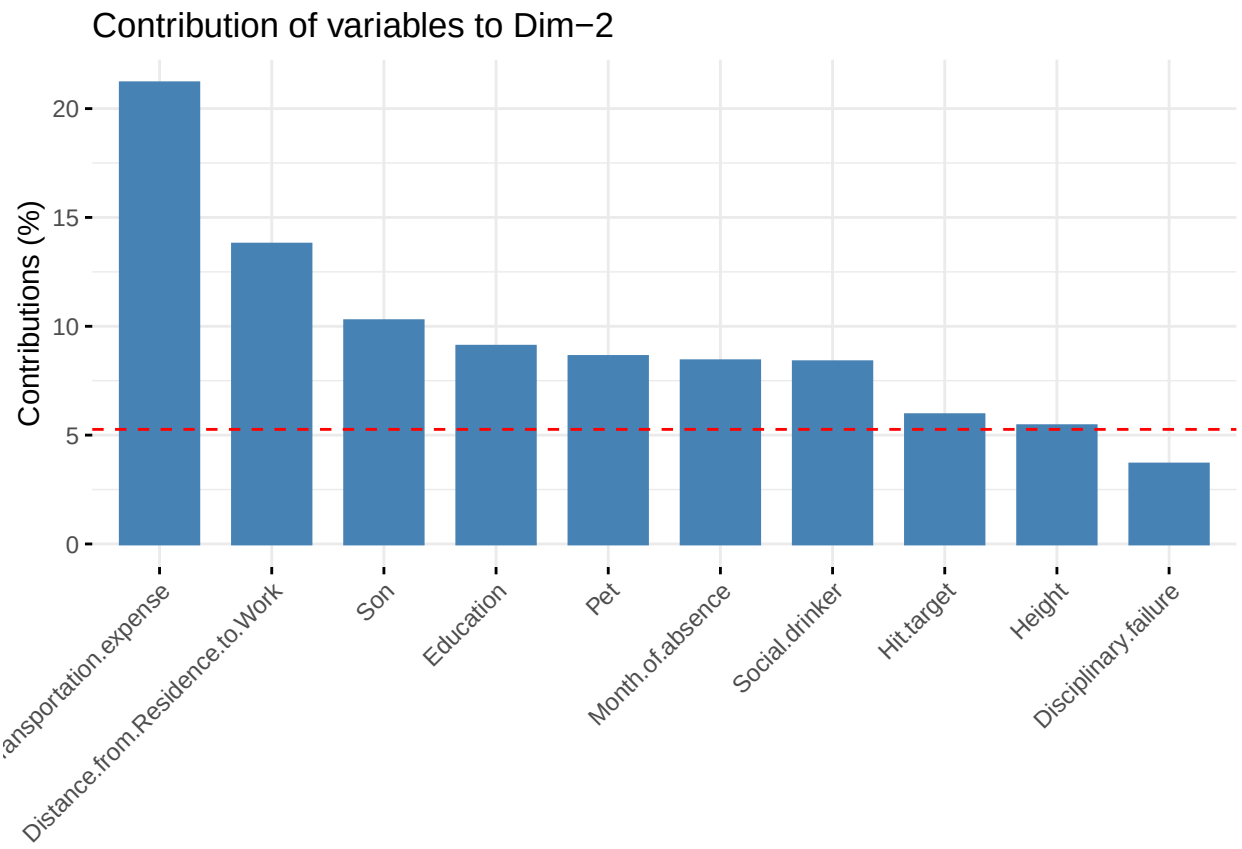
```
# Calidad de presentación de variables en un correlograma.
fviz_cos2(pca,choice='var',axes=1:2)
```




```
# Contribución de las variables a los respectivos componentes principales
fviz_contrib(pca,choice='var',axes=1, top = 10) #componente 1
```




```
fviz_contrib(pca,choice='var',axes=2, top = 10) #componente 2
```



La línea roja discontinua indica el valor medio de contribución. Para una determinada componente, una variable con una contribución mayor a este límite puede considerarse importante a la hora de contribuir a esta componente. En las representaciones anteriores, la variable Body.Mass.Index es la que más contribuye a la PC1.

4.3. Pruebas Estadisticas

4.3.1. Contraste de Hipotesis Anteriormente en los analisis visuales que hemos hecho vimos que el personal con diferentes niveles de educacion parecian tener diferencia en horas de ausentismo e incluso encontrabamos mas horas de ausentismos segun la edad del empleado y lo mismo sucedia con el indice de masa corporal.

Dicho eso realicemos algunas pruebas estadisticas, en particular contrastes de hipotesis para determinar si se puede inferir comportamientos a partir de estas variables sobre esta muestra en particular.

Por ejemplo planteemos la siguiente pregunta de investigacion:

- Las horas de ausentismo es mayor en empleados menores de 30 años?

O sea aqui queremos validar:

- Grupo 1: Empleados menores de 30 años
- Grupo 2: Empleados mayores de 30 años

Horas Ausentismo Grupo 1 > Horas Ausentismo Grupo 2

Por lo que las hipotesis nula (H0) y alternativa (H1) seran:

- H0: Horas Ausentismo Grupo 1 <= Horas Ausentismo Grupo 2
- H1: Horas Ausentismo Grupo 1 > Horas Ausentismo Grupo 2

entonces segun lo que comprobemos a continuacion podremos llegar a decir:

- Rechazo la H0 a favor de la H1 y por tanto si que hay evidencias que demuestran que los menores de 30 se ausentan mas que los mayores de 30. Y por tanto la respuesta a la pregunta de investigacion es SI, con el 95 de confianza

o

- No hay evidencia que permita rechazar la hipotesis nula por lo que no puede afirmarse que menores de 30 años se ausenten mas que los mayores de 30 años para la muestra seleccionada.

El contraste que aplicaremos aqui para validar esto esto es un contraste de dos muestras independientes sobre la media, es parametrico porque podemos asumir normalidad de los datos ya sea por el Teorema del Limite Central. Y se tratara de un test parametrico y unilateral por la derecha.

Como se sabe la homocedasticidad y la heterocedasticidad, tiene que ver con la variabiblidad de las muestras, es decir, los menores de 30 años se pueden parecer muchos mientras que los mayores no, o viceversa. Si las varianzas entre las dos muestras son similares se podra aplicar una test, mientras que si son distintas se debe usar otra formula.

Verifiquemos la homoscedasticidad de la variable en cuestion para poder aplicar el estadistico correspondiente a nuestro test.

```
grupo1 = abs_df[abs_df$Age < 30,]$Age
grupo2 = abs_df[abs_df$Age >= 30,]$Age
var.test(grupo1, grupo2, conf.level = 0.95 )
```

```
##
##  F test to compare two variances
##
## data:  grupo1 and grupo2
## F = 0.003334, num df = 130, denom df = 608, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.002578540 0.004413893
## sample estimates:
## ratio of variances
##        0.003333962
```

Como vemos aqui dado que el p-value es mucho menor que el nivel de significancia (0.05) podemos rechazar la hipotesis nula de homogeneidad de las varianzas.

Una vez que podemos asumir que las varianzas son distintas, aplicaremos el metodo correspondiente a la media de dos poblaciones independientes con varianza desconocida distinta.

```
t.test(grupo1,grupo2, alternative="greater", var.equal=FALSE)
```

```
##
##  Welch Two Sample t-test
##
```

```
## data:  grupo1 and grupo2
## t = -44.241, df = 626.29, p-value = 1
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -10.64996       Inf
## sample estimates:
## mean of x mean of y
##  28.00000  38.26765
```

Siendo que el p-value es mayor que el valor de seleccionado, existen evidencias suficientes para no rechazar H0, osea:

- Grupo 1: Empleados menores de 30 años
- Grupo 2: Empleados mayores de 30 años

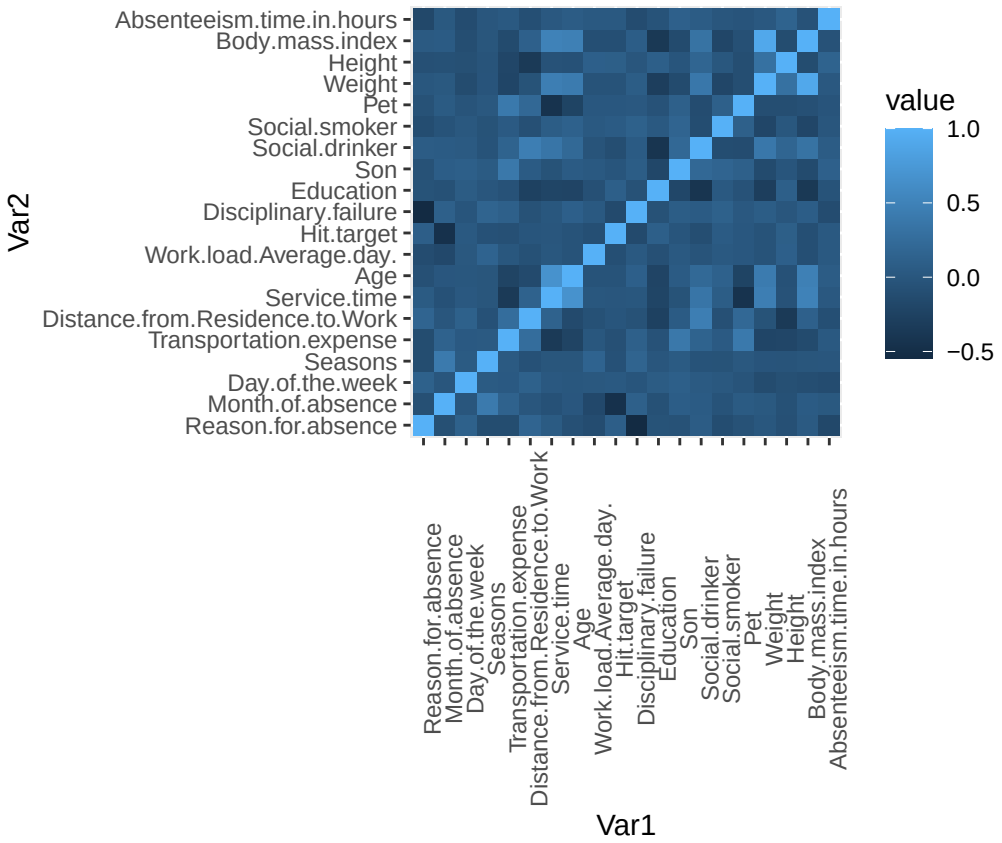
  H0: *Horas Ausentismo Grupo 1 <= Horas Ausentismo Grupo 2* => ACEPTADA

  H1: *Horas Ausentismo Grupo 1 > Horas Ausentismo Grupo 2* => RECHAZADA

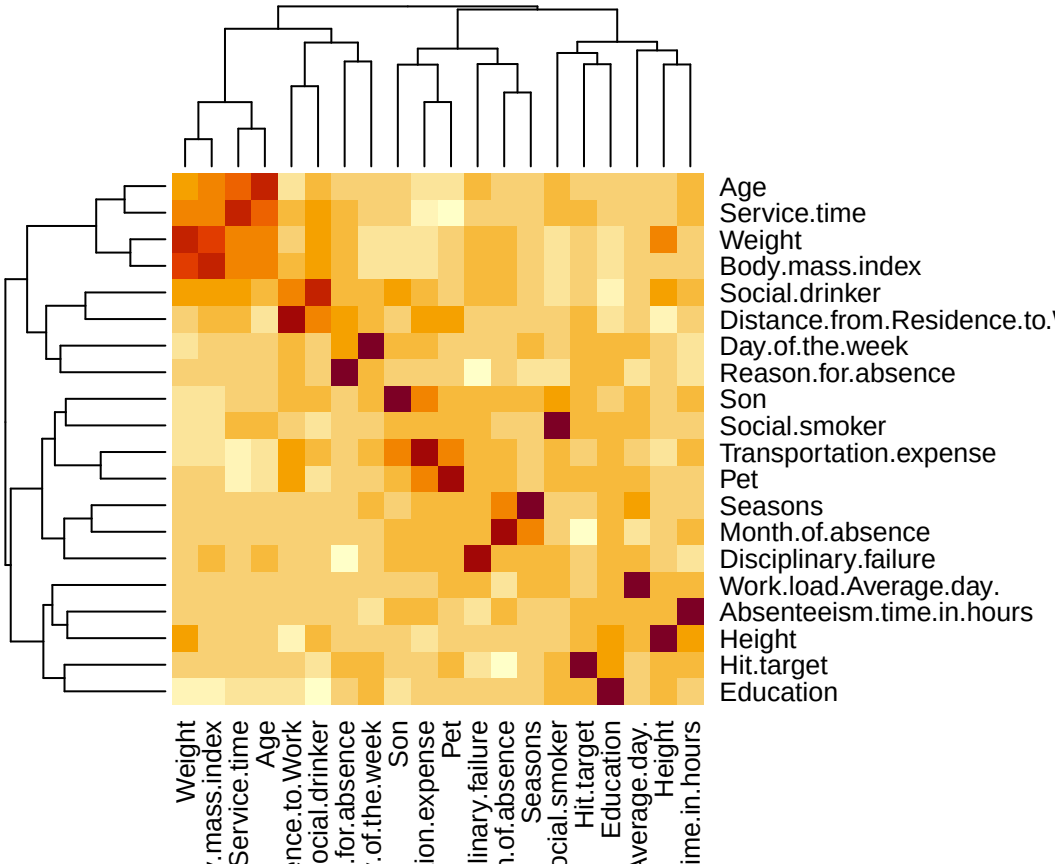
Entonces dado que el p-value es mayor que alpha, se dispone de evidencia suficiente para considerar que los menores de 30 años *NO* se ausentan mas que los mayores de 30 años con una confianza del 95%.

4.3.2. Correlaciones A partir de aqui graficos de correlacion y heatmaps es posible ver como se correlacionan las variables de un dataset

```
# Una de las herramientas más útiles es calcular la matriz de correlación entre
# las variables. Con la función qplot y la correlación de variables, calculada con
# la función cor, podemos visualizar de manera fácil aquellas variables más
# correlacionadas, que corresponden a una intensidad mayor de color.

heat <- abs_df[,2:21]

qplot(x=Var1, y=Var2, data=melt(cor(heat, use="p")), fill=value, geom="tile") +
   theme(axis.text.x = element_text(angle = 90)) +
   coord_fixed()
```

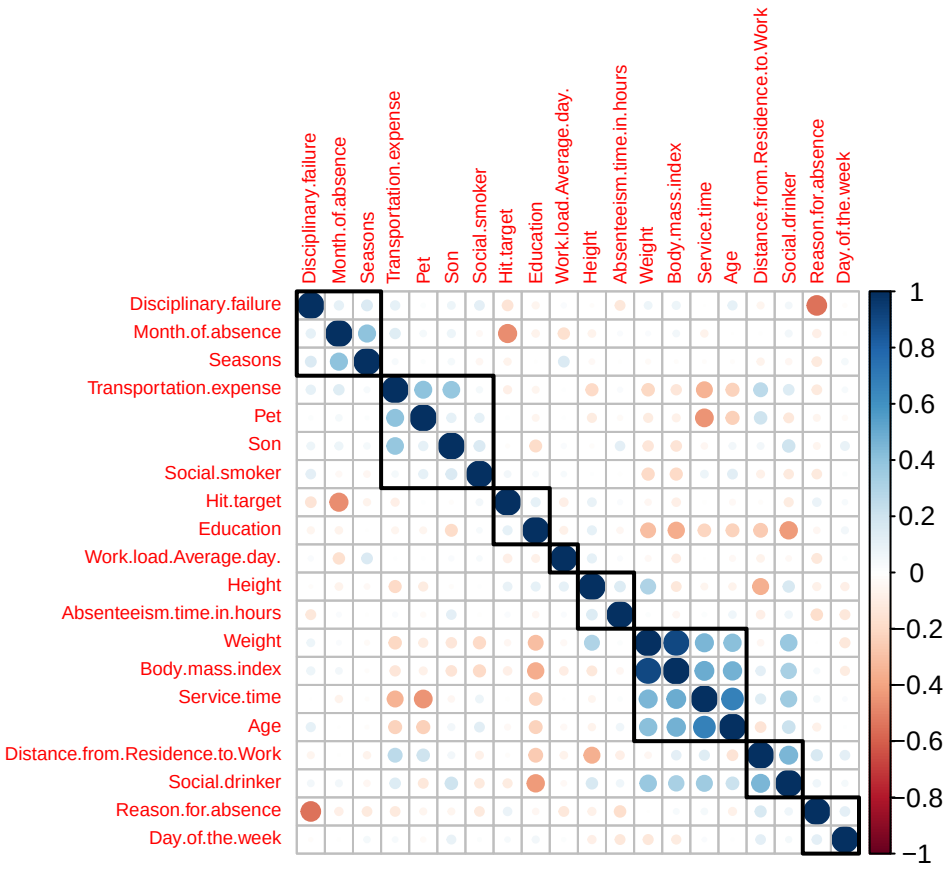



```
# El heatmap nos permite ir un paso más allá y agrupar las variables que tienen
# más relación entre ellas, a partir de un algoritmo de clustering a partir de la
# información de la correlación
abs_matrix <- as.matrix(scale(cor(abs_df[,2:21], use="p")))
heatmap(abs_matrix, Colv=F, scale='none')
```



Mientras que este otro grafico tambien nos permite encontrar variables correlacionadas y removerlas si asi quisieramos:

```
data_corr<- cor(abs_df[,2:21])
corrplot::corrplot(data_corr, order = "hclust", tl.cex = 0.6, addrect = 8)
```



Aqui nos encontramos con algo que no habiamos analizado anteriormente de manera manual, como por ej, variables como: service time, hit target, disciplinary failure, todas ellas muestran correlacion ademas de las que comentamos antes, como obvias como podian ser education, BMI, Social Drinker, Distancia, etc…

Pero veamos todo esto de manera mas programatica y analicemos especificamente por las variables mas correlacionadas con la variable dependiente que buscamos analizar (Absenteeism.time.in.hours)

```
corr_matrix <- matrix(nc = 2, nr = 0)
colnames(corr_matrix) <- c("estimate", "p-value")

# Calcular el coeficiente de correlación para cada variable cuantitativa
# con respecto al campo "Absenteeism.time.in.hours"
for (i in 3:20) {
  if ( is.integer(unlist(abs_df[,i])) |
       is.numeric(unlist(abs_df[,i]))  )
  {
    # utilizamos spearman ya que no todas las variables pasaron el test
    # homocedasticidad
    spearman_test = cor.test( as.vector(unlist(abs_df[,i])),
                              as.vector(unlist(abs_df$Absenteeism.time.in.hours)),
                                method = "spearman", exact=FALSE)
    corr_coef = spearman_test$estimate
    p_val = spearman_test$p.value
    # Add row to matrix
    pair = matrix(ncol = 2, nrow = 1)

    pair[1][1] = corr_coef
```

```
    pair[2][1] = p_val
    corr_matrix <- rbind(corr_matrix, pair)
    rownames(corr_matrix)[nrow(corr_matrix)] <- colnames(abs_df)[i]
  }
}
```

```
corr_matrix[order(corr_matrix[,1]),]
```

```
##                                    estimate      p-value
## Disciplinary.failure            -0.396971284 2.425617e-29
## Day.of.the.week                 -0.094547231 1.007101e-02
## Seasons                         -0.070518454 5.518028e-02
## Age                             -0.070306847 5.591649e-02
## Body.mass.index                 -0.062574181 8.894461e-02
## Service.time                    -0.029951375 4.158909e-01
## Weight                          -0.009216048 8.023669e-01
## Education                        0.001159684 9.748760e-01
## Social.smoker                    0.002404981 9.479255e-01
## Pet                              0.002407285 9.478757e-01
## Month.of.absence                 0.009767736 7.908028e-01
## Distance.from.Residence.to.Work  0.009913308 7.877589e-01
## Work.load.Average.day.           0.011757661 7.494889e-01
## Hit.target                       0.046821928 2.032883e-01
## Height                           0.071249478 5.269872e-02
## Social.drinker                   0.105317718 4.129602e-03
## Son                              0.150153715 4.113873e-05
## Transportation.expense           0.166173569 5.509197e-06
```

Aqui vemos que las mas correlacionadas positivamente son Transportation.expense, Son, Social.drinker, mientras que las negativas son Disciplinary.failure, Day.of.the.week y Seasons.

Dicho todo esto pasemos a la construccion de un modelo que nos permita predecir todo esto que fuimos analizando hasta ahora a partir de la exploracion de los datos y la inferencia estadistica.

4.3.3. Regresiones Creamos en esta seccion 3 modelos de regresion lineal utilizando:

1- Todas las variables del dataset

2- Las primeras n componentes principales que expliquen minimo un 70% de la varianza total

3- Las variables mas correlacionadas a la variable dependiente identificadas con el test de spearman.

Veamoslo a continuacion.

4.3.3.1. Generacion de dataset de training y testing Con la función sample_frac de dplyr generamos una muestra aleatoria de nuestro dataset, donde crearemos el dataset de train especificando la proporción del dataset original que queremos, y finalmente asignaremos al dataset de test el resto de las filas que no han estado seleccionadas con la función anti_join. Esta division es necesaria para la posterior validacion de nuestro modelo. Es por eso que necesitamos tener dos conjuntos, el de entrenamiento y el de prueba.

Es por eso que arranquemos generando los dos set de datos para ese primer objetivo:

```r
set.seed(123)

rows <- 1:nrow(abs_df)
abs_df <- abs_df %>% mutate(rowID = rows)

train <- sample_frac(abs_df[rows,], .75)
test  <-   anti_join(abs_df[rows,], train, by='rowID')

# Datasets de entrenamiento
training_x <- train[,c(3:21)]
training_y <- train$Absenteeism.time.in.hours

# Datasets de validacion
testing_x <- test[,c(3:21)]
testing_y <- test$Absenteeism.time.in.hours
```

4.3.3.2. Creacion de Modelos Comencemos con el modelo con todas las variables

```r
modelo_all <- lm(Absenteeism.time.in.hours ~ ., data = training_x)

summary(modelo_all)
```

```
##
## Call:
## lm(formula = Absenteeism.time.in.hours ~ ., data = training_x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.070  -4.955  -1.826   1.157 106.811
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     41.31710   91.11186   0.453 0.650389
## Month.of.absence                 0.24053    0.21873   1.100 0.271971
## Day.of.the.week                 -1.58410    0.41001  -3.864 0.000125 ***
## Seasons                         -0.43434    0.60980  -0.712 0.476608
## Transportation.expense           0.01812    0.01118   1.621 0.105645
## Distance.from.Residence.to.Work -0.09440    0.06166  -1.531 0.126339
## Service.time                     0.03232    0.23600   0.137 0.891110
## Age                              0.39252    0.14330   2.739 0.006367 **
## Work.load.Average.day.          -0.00269    0.01569  -0.171 0.863994
## Hit.target                       0.20625    0.18236   1.131 0.258560
## Disciplinary.failure            -9.10779    2.59600  -3.508 0.000489 ***
## Education                       -1.50582    1.04323  -1.443 0.149487
## Son                              0.54053    0.59427   0.910 0.363458
## Social.drinker                   1.40325    1.79919   0.780 0.435775
## Social.smoker                   -3.25524    2.54004  -1.282 0.200547
## Pet                             -0.02380    0.60121  -0.040 0.968439
## Weight                           0.43681    0.57455   0.760 0.447424
## Height                          -0.29376    0.51801  -0.567 0.570890
## Body.mass.index                 -1.71275    1.64922  -1.039 0.299496
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.12 on 536 degrees of freedom
## Multiple R-squared:  0.104,  Adjusted R-squared:  0.07387
## F-statistic: 3.455 on 18 and 536 DF,  p-value: 2.286e-06
```

```
# MSE empleando las observaciones de entrenamiento
training_mse <- mean((modelo_all$fitted.values - training_y)^2)
training_mse
```

```
## [1] 166.3601
```

```
# MSE empleando observaciones de testo
predicciones <- predict(modelo_all, newdata = testing_x)
testing_mse <- mean((predicciones - testing_y)^2)
testing_mse
```

```
## [1] 156.3399
```

```
# Tabla con los coeficientes de determinación de cada modelo
# Donde iremos guardando todas las variables resultado para su
# posterior analisis
resultados = data.frame("modelo_all", summary(modelo_all)$r.squared,
                        training_mse, testing_mse)
colnames(resultados) <- c("Modelo", "R^2", "training_MSE", "testing_MSE")
```

Arriba vemos que solo 3 de todas las variables combinadas son significativas para el modelo, de ahi el bajo R^2 obtenido.

Ahora creemos el modelo de regresion a partir de las componentes principales. En este caso utilizaremos PCR, el método Principal Components Regression consiste en ajustar un modelo de regresión lineal por mínimos cuadrados empleando como predictores las componentes generadas a partir del analisis de componentes principales (PCA) anteriormente realizado. De esta forma, con un número reducido de componentes se puede explicar la mayor parte de la varianza de los datos.

Sabiamos que si queriamos explicar al menos el 70% de la varianza debiamos minimo usar 8 componentes principales como vemos aqui:

```
summary(pca)$importance[, 1:8]
```

```
##                             PC1      PC2      PC3      PC4      PC5      PC6
## Standard deviation     1.827512 1.502469 1.392644 1.228002 1.178021 1.110794
## Proportion of Variance 0.175780 0.118810 0.102080 0.079370 0.073040 0.064940
## Cumulative Proportion  0.175780 0.294590 0.396670 0.476030 0.549070 0.614010
##                             PC7       PC8
## Standard deviation     1.039648 0.9999536
## Proportion of Variance 0.056890 0.0526300
## Cumulative Proportion  0.670900 0.7235300
```

Entonces apliquemos pcr para ello. Con la función pcr() del paquete pls se evita tener que codificar cada uno de los pasos intermedios.

```
modelo_pcr <- pcr(formula = Absenteeism.time.in.hours ~ ., data = training_x,
                  scale. = TRUE, ncomp = 8)
summary(modelo_pcr)
```

```
## Data:    X dimension: 555 18
##  Y dimension: 555 1
## Fit method: svdpc
## Number of components considered: 8
## TRAINING: % variance explained
##                             1 comps   2 comps  3 comps  4 comps  5 comps
## X                           68.0892  92.37165   95.513   98.385   99.088
## Absenteeism.time.in.hours    0.0918   0.09447    1.172    1.584    1.802
##                             6 comps  7 comps  8 comps
## X                             99.45   99.715   99.818
## Absenteeism.time.in.hours      3.65    3.743    3.934
```

```
training_mse_pcr <- mean((modelo_pcr$fitted.values - training_y)^2)
training_mse_pcr
```

```
## [1] 181.931
```

```
predicciones_pcr <- predict(modelo_pcr, newdata = testing_x, ncomp = 8)
testing_mse_pcr <- mean((predicciones_pcr - testing_y)^2)
testing_mse_pcr
```

```
## [1] 151.8021
```

```
resultados = rbind(resultados, c("modelo_pcr", NA, training_mse_pcr, testing_mse_pcr))
```

Vemos que con pcr el modelo predice mejor dado que notamos una disminucion del MSE con los datos de testing. Y de hecho es un valor menor al del modelo con todas las variables.

Por ultimo usemos las 8 variables mas correlacionadas obtenidas en el paso “4.2.2” para contruir el 3er modelo.

```
modelo_cor <- lm(Absenteeism.time.in.hours ~
                   Disciplinary.failure+Day.of.the.week+Seasons+Age+
                   Height+Social.drinker+Son+Transportation.expense,
                 data = training_x)
summary(modelo_cor)
```

```
##
## Call:
## lm(formula = Absenteeism.time.in.hours ~ Disciplinary.failure +
##     Day.of.the.week + Seasons + Age + Height + Social.drinker +
##     Son + Transportation.expense, data = training_x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.654  -4.730  -2.108   1.026 108.577
##
```

```
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -39.73507   18.72776  -2.122 0.034311 *
## Disciplinary.failure    -9.38171    2.55298  -3.675 0.000262 ***
## Day.of.the.week         -1.52480    0.39980  -3.814 0.000152 ***
## Seasons                 -0.09844    0.52346  -0.188 0.850909
## Age                      0.32795    0.09850   3.329 0.000929 ***
## Height                   0.21555    0.09703   2.221 0.026729 *
## Social.drinker          -0.17173    1.24615  -0.138 0.890441
## Son                      1.01795    0.55879   1.822 0.069044 .
## Transportation.expense   0.01645    0.01010   1.629 0.103830
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.18 on 546 degrees of freedom
## Multiple R-squared:  0.07945,    Adjusted R-squared:  0.06596
## F-statistic: 5.891 on 8 and 546 DF,  p-value: 3.028e-07
```

```
# MSE empleando las observaciones de entrenamiento
training_mse_cor <- mean((modelo_cor$fitted.values - training_y)^2)
training_mse_cor
```

```
## [1] 170.9099
```

```
# MSE empleando observaciones de testo
predicciones_cor <- predict(modelo_cor, newdata = testing_x)
testing_mse_cor <- mean((predicciones_cor - testing_y)^2)
testing_mse_cor
```

```
## [1] 149.1686
```

```
resultados = rbind(resultados, c("modelo_cor", summary(modelo_cor)$r.squared,
                                 training_mse_cor, testing_mse_cor))
```

Por ultimo con este modelo vemos que el R^2 es aun mas bajo, y esta combinacion de variables a pesar de ser las mas correlacionadas (aunque ya sabiamos de antemano que eran valores muy bajos, mas cercanos a la no correlacion "< 0.5" que a estar fuertemente correlacionadas "> 0.8") solo algunas de ellas son significativas para el modelo. De ahi tambien el valor bajo de R^2 y de ahi tambien que un modelo de regresion lineal no aplicaba a este conjunto de datos.

5. Representacion de los Resultados

Representemos aqui el resultado de haber generado 3 modelos para predecir el ausentismo:

```
resultados
```

```
##       Modelo                R^2     training_MSE      testing_MSE
## 1 modelo_all  0.103956497500517 166.360147923584 156.339906852564
## 2 modelo_pcr               <NA> 181.931003042726 151.802146933424
## 3 modelo_cor 0.0794510020440571 170.909857661677 149.168571403076
```

Como vemos el R^2 es bastante malo para los 3 modelos creados. Por lo que que podemos decir que ninguno de estos modelos de regresion lineal ajustan bien a los datos que tenemos. Incluso, por mas que para cada uno de ellos el MSE sea menor para el dataset de testing, lo cual suele indicar una buena capacidad de prediccion del modelo cuando este MSE es menor al MSE del dataset de training,....el R^2 sigue siendo demasiado bajo.

Ni siquiera aplicando PCR logramos una gran mejora, porque si bien logramos bajar el mean squared error respecto al modelo con todas las variables. No es un disminucion muy significativa.

Tal vez lo mejor es utilizar regresion logistica y buscar predecir la variable dependiente de manera dicotomica tal como habiamos analizado en el EDA al dicotomizar la variable. O incluso en lugar de predecir n rangos, podriamos buscar con regresion logistica predecir si el empleado se ausentara poco o se ausentara demasiado. Por ejemplo creando una variable que represente los que se ausentan menos de 16 hs o los que se ausentan mas de eso. O incluso se podria utilizar arboles de decision para generar reglas que nos ayuden a generar modelos que ajusten mejor a los datos.

6. Resolucion del Problema

Si bien no hemos logrado llegar a un modelo de regresion lineal que ajuste bien a los datos, hemos podido identificar varios puntos que nos servirian para una segunda iteracion de este trabajo como ser:

- Con el EDA hemos podido exponer ciertos comportamientos de los datos que son relevantes a la hora de entender que empleados se ausentan mas o menos segun sus caracteristicas.
- Tanto con PCA como los analisis de correlacion nos permitieron identificar variables mas influyentes sobre la variable a clasificar.
- Con los contrastes de hipotesis pudimos revalidar o inferir situaciones de la poblacion que no fueron tan visibles con un simple analisis visual.

7. Codigo

El codigo para generar este PDF fue realizado en R Markdown, el cual puede ser encontrado en el siguiente repositorio github:

Absenteeism at work cleaning code

Recursos

- Calvo M, Subirats L, Pérez D (2019). Introducción a la limpieza y análisis de los datos. Editorial UOC.
- Espacio de recursos UOC para ciencia de datos
- Buscador de código R

- Colección de cheatsheets en R
- Sitio Web: Ciencia de Datos