Session4_PYCD

2023-10-21

Limpieza de datos

Dado que algunas formulas quimicas no presentan informacion acerca de su formula quimica o especificacion tecnica se decidira eliminar la formulas que no presenten esta informacion:

```
setwd("/cloud/project")
out_final <- read.csv("out_final.csv", header = T)
library(dplyr)
```

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

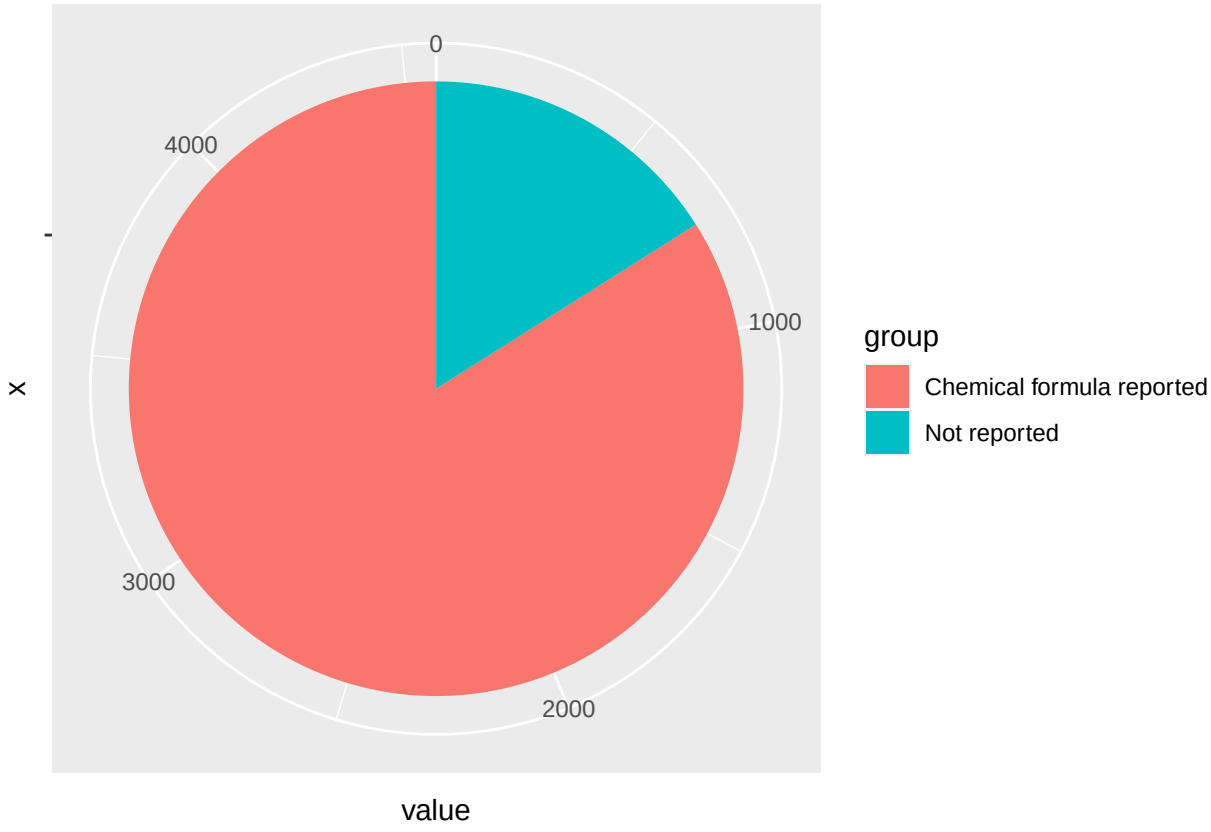
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
out_final[out_final == ''] <- NA
nas <- sum(is.na(out_final$formula))
print(nas)
```

```
## [1] 733
```

```
notna <- sum(!is.na(out_final$formula))
df <- data.frame(
  group = c("Chemical formula reported", "Not reported"),
  value = c(notna, nas)
  )
library(ggplot2)
# Barplot
bp<- ggplot(df, aes(x="", y=value, fill=group))+geom_bar(width = 1, stat = "identity")
pie <- bp + coord_polar("y", start=0)
pie
```

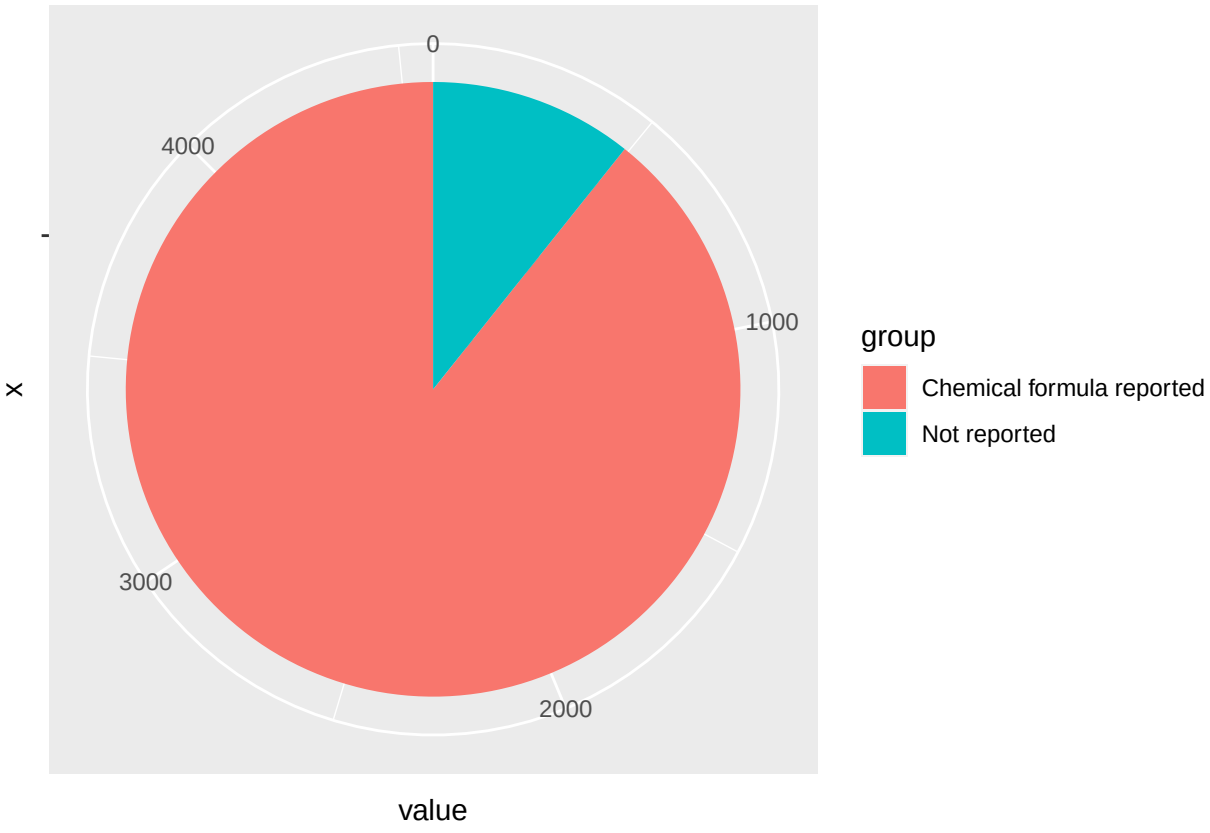



```
knitr::kable(df)
```

Tabla de frecuencias de formulas quimicas reportadas

group	value
Chemical formula reported	3840
Not reported	733

Realizar el mismo procedimiento para la columna peso molar:




```
knitr::kable(df)
```

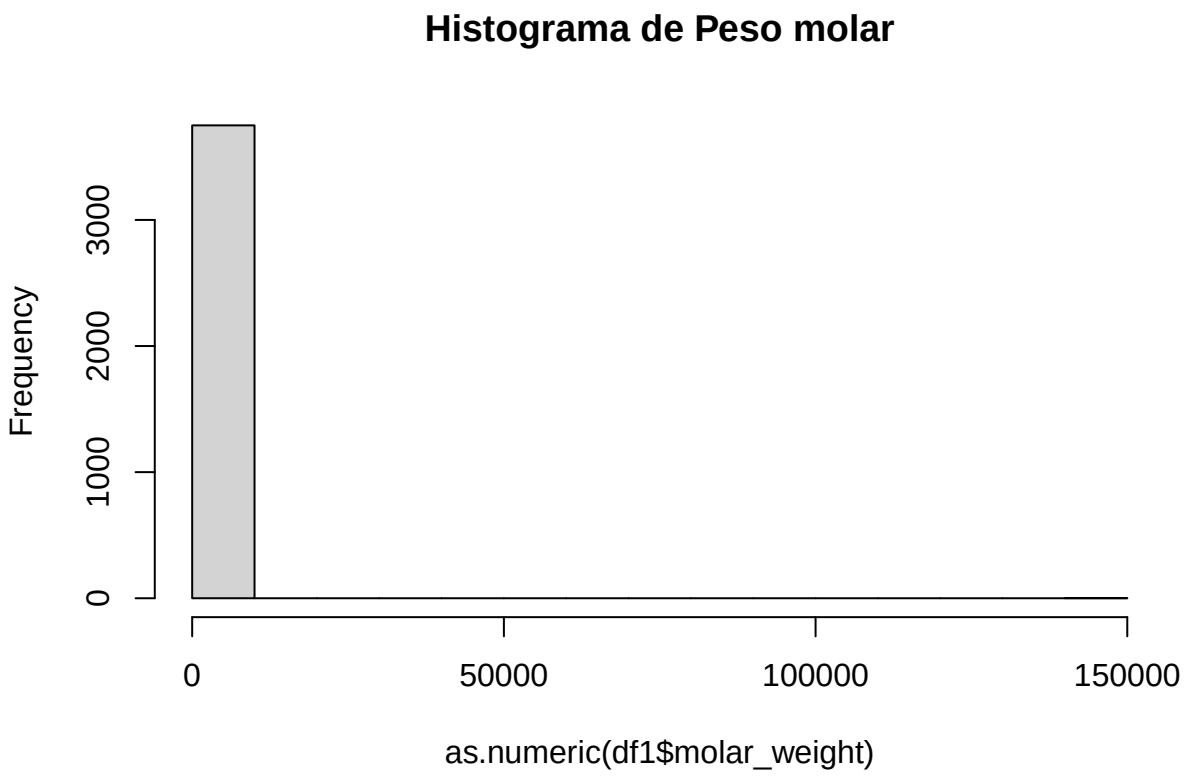
Tabla de frecuencias de Peso Molar

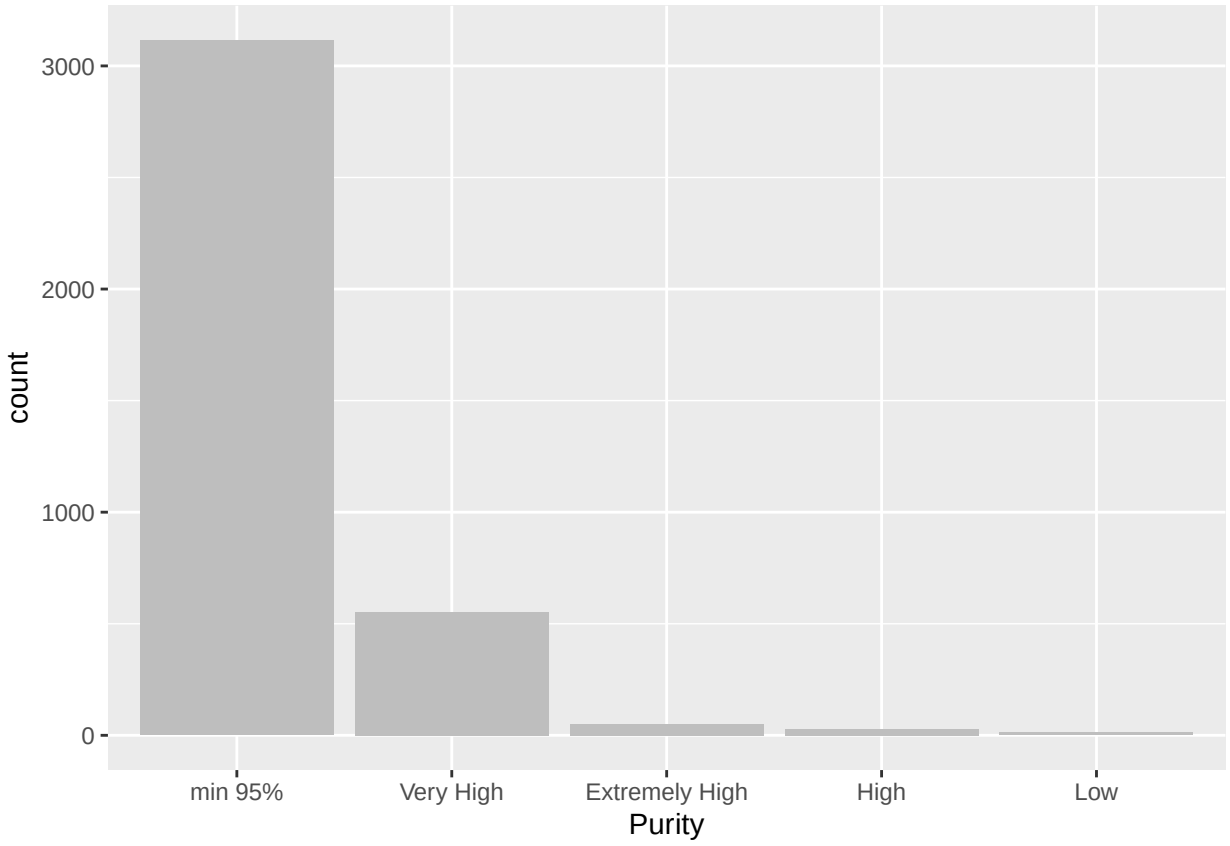
group	value
Chemical formula reported	4083
Not reported	490

```
library("tidyr")
df1= out_final %>% drop_na(molar_weight) %>% drop_na(formula)
#Replace NA in purity category
df1 = df1 %>% replace_na(list( purity_category = "min 95%"))
# Histogram with density plot
hist(as.numeric( df1$molar_weight), main = "Histograma de Peso molar")
```

Histgrama de Peso molecular

```
## Warning in hist(as.numeric(df1$molar_weight), main = "Histograma de Peso
## molar"): NAs introduced by coercion
```





Grafica de Barras por Categoria de Pureza

Boxplot masa molar reportada por cada proveedor

kruskal-wallis-test Dado el boxplot anterior se tiene sospecha de que que los componentes reportados con categoria de pureza presentan diferente tipo de varianza. para estos se usara un metodo no parametrico para comprobar si la diferencia de varianzas existe en diferentes categorias.

```
kruskal.test(molar_weight ~ purity_category, data = df1)
```

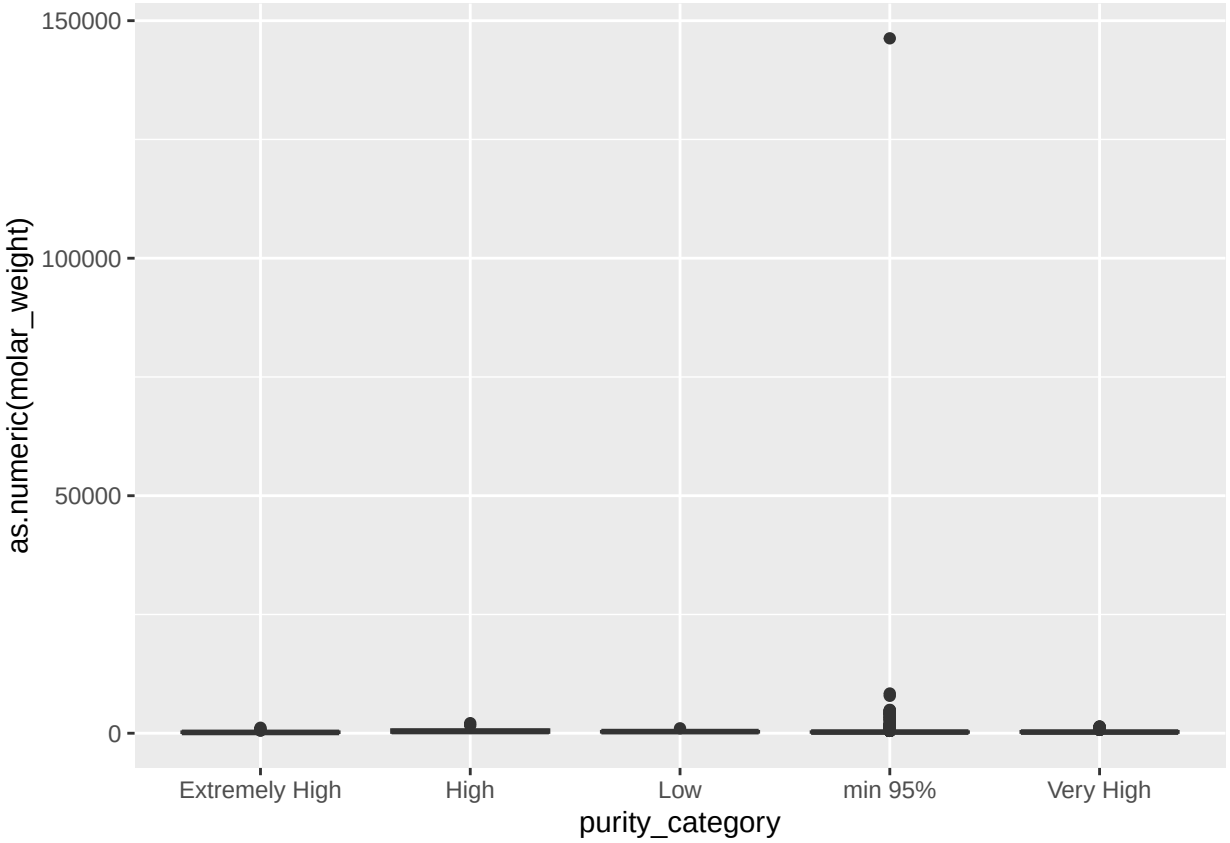
```
##
##  Kruskal-Wallis rank sum test
##
## data:  molar_weight by purity_category
## Kruskal-Wallis chi-squared = 10.495, df = 4, p-value = 0.03287
```

```
kruskal.test(molar_weight ~ substance_category, data = df1)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  molar_weight by substance_category
## Kruskal-Wallis chi-squared = 1.537, df = 2, p-value = 0.4637
```

Conclusiones:

Al final, la imputacion se realizo con respecto a las columnas con presencia de datos nulos, se obtuvieron despues de la imputacion **3759** registros, los cuales podriamos trabajar con un tipo de clustering o algun tipo de **regresion no parametrica**


150000
100000
50000
0
as.numeric(molar_weight)
Extremely High
High
Low
min 95%
Very High
purity_category