Oct 22, 2025 Version 2

Basic cleaning of a CASA dataset with R functions V.2

DOI
https://dx.doi.org/10.17504/protocols.io.8epv5k3jdv1b/v2

Cindy Rivas[1], Claudia Treviño[1], Andrés Aragón Martínez[2]
[1]Institute of Biotechnology, UNAM; [2]FES Iztacala, UNAM

Cindy Rivas

DOI: https://dx.doi.org/10.17504/protocols.io.8epv5k3jdv1b/v2

Protocol Citation: Cindy Rivas, Claudia Treviño, Andrés Aragón Martínez 2025. Basic cleaning of a CASA dataset with R functions . **protocols.io https://dx.doi.org/10.17504/protocols.io.8epv5k3jdv1b/v2**Version created by **Cindy Rivas**

Protocol status: Working
We use this protocol and it's working

Created: October 21, 2025

Last Modified: October 22, 2025

Protocol Integer ID: 230420

Keywords: R software, Data Cleaning, Workflow, essential steps of dataset cleaning, dataset cleaning, na value, dataset, factor levels in specific column, reordering factor level, specific column, data, correct typographical error, reproducible workflow, casa dataset, basic cleaning, column

Funders Acknowledgements:
UNAM-DGAPA-PAPIIT
Grant ID: IN224925
UNAM-DGAPA-PAPIIT
Grant ID: IN215425

Disclaimer

DISCLAIMER – FOR INFORMATIONAL PURPOSES ONLY; USE AT YOUR OWN RISK

The protocol content here is for informational purposes only and does not constitute legal, medical, clinical, or safety advice, or otherwise; content added to **protocols.io** is not peer reviewed and may not have undergone a formal approval of any kind. Information presented in this protocol should not substitute for independent professional judgment, advice, diagnosis, or treatment. Any action you take or refrain from taking using or relying upon the information presented here is strictly at your own risk. You agree that neither the Company nor any of the authors, contributors, administrators, or anyone else associated with **protocols.io**, can be held responsible for your use of the information contained in or linked to this protocol or any of our Sites/Apps and Services.

Abstract

This protocol provides a reproducible R workflow to:
1. Correct typographical errors (typos) in specific columns.
2. Reordering factor levels in specific columns.
3. Remove NA values.

After completing the essential steps of dataset cleaning, we can proceed to perform statistical analyses or apply machine learning algorithms to our data.

Troubleshooting

Cleaning a dataset

1 Set the working directory.

Command

Set the working directory

Set the working directory

```
setwd()
```

Expected result

```
> setwd ("/Users/cindyra/Downloads")
> getwd()
[1] "/Users/cindyra/Downloads"
```

2 Read the CSV file and create an object that contains your data. In this example, we will not use any R packages or libraries; we will work only with base R. We will clean a small dataset containing kinematic parameter values from a Computer-Assisted Sperm Analysis (CASA) system.
The CSV file for this exercise can be downloaded at:

Dataset

CASA data NAME

https://osf.io/cq7pg/overview LINK

Command

Read csv file

Read csv file

```
 casa_data <- read.csv(casa_data.csv, head=TRUE, sep= ,
,stringsAsFactors=TRUE)
```

Review the structure of **casa_data** object.

Command

Object structure

Object structure

```
str(casa_data)
```

Expected result

```
'data.frame':  11 obs. of  8 variables:
 $ date        : Factor w/ 2 levels "01/03/2025","17/05/2025": 1 1 1 2 1 2 2 2 1 1 ...
 $ donor       : Factor w/ 3 levels "p32","p45","p67": 3 3 3 1 3 1 3 1 1 1 ...
 $ Treatment   : Factor w/ 4 levels "5-HT_100nM","5-HT_10nM",..: 4 1 4 4 4 2 1 3 2 3 ...
 $ exposure_time: Factor w/ 3 levels "100sec","10min",..: 3 1 2 1 3 2 2 3 2 1 ...
 $ VCL         : num  110 87.4 117.4 152.1 115.6 ...
 $ VSL         : num  68 77.4 97.7 77 NA ...
 $ LIN         : num  0.62 0.89 0.83 0.51 0.75 0.18 0.68 NA 0.5 1.1 ...
 $ ALH         : num  4.8 3.84 5.24 3.07 NA 4.4 3.11 2.68 4.03 3.57 ...
```

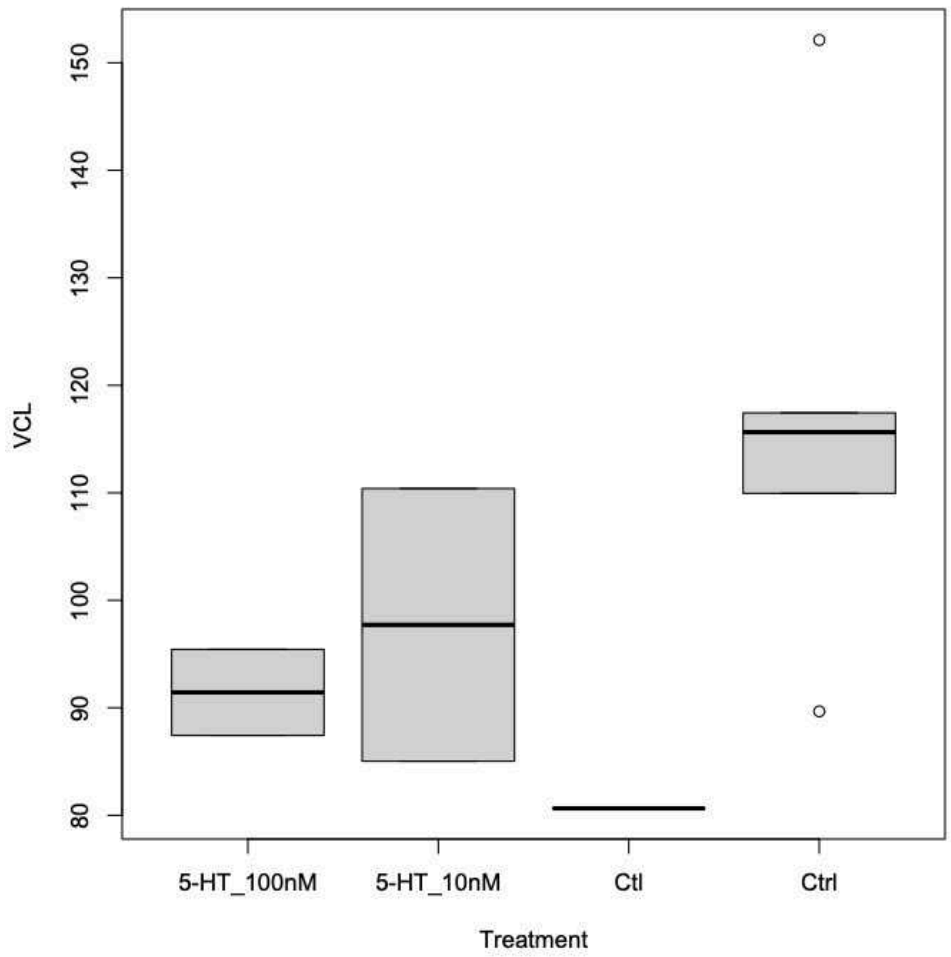
3 Now that we know the structure of the data, we can visualize it.
We are going to plot the **“Treatment”** column against someone of the motility parameter columns, in this example, we use **"VCL"** column.

Command

Plotting

Plotting

```
plot(casa_data_ordered$Treatment, casa_data_ordered$VCL,
xlab=Treatment, ylab=VCL)
```



Data visualization not only allows us to identify patterns or trends within the dataset but also helps detect typographical errors, as illustrated in this example. Cleaning the dataset ensures these errors are corrected, improving the accuracy and reliability of subsequent analyses.

4 **Correct typographical errors (typos) in specific columns.**

Command

Correct typos

Correct typos

```
levels(casa_data$Treatment)[levels(casa_data$Treatment) == Ctl] <-
Ctrl
```

Now, instead of having four factor levels, we have three. After correcting the typographical error, we can clearly visualize the updated and properly labeled factor levels in two ways: first, by printing the factor levels of the column, and second, through a plot.

Command

Print levels

Print levels

```
print(levels(casa_data$Treatment))
```

Expected result

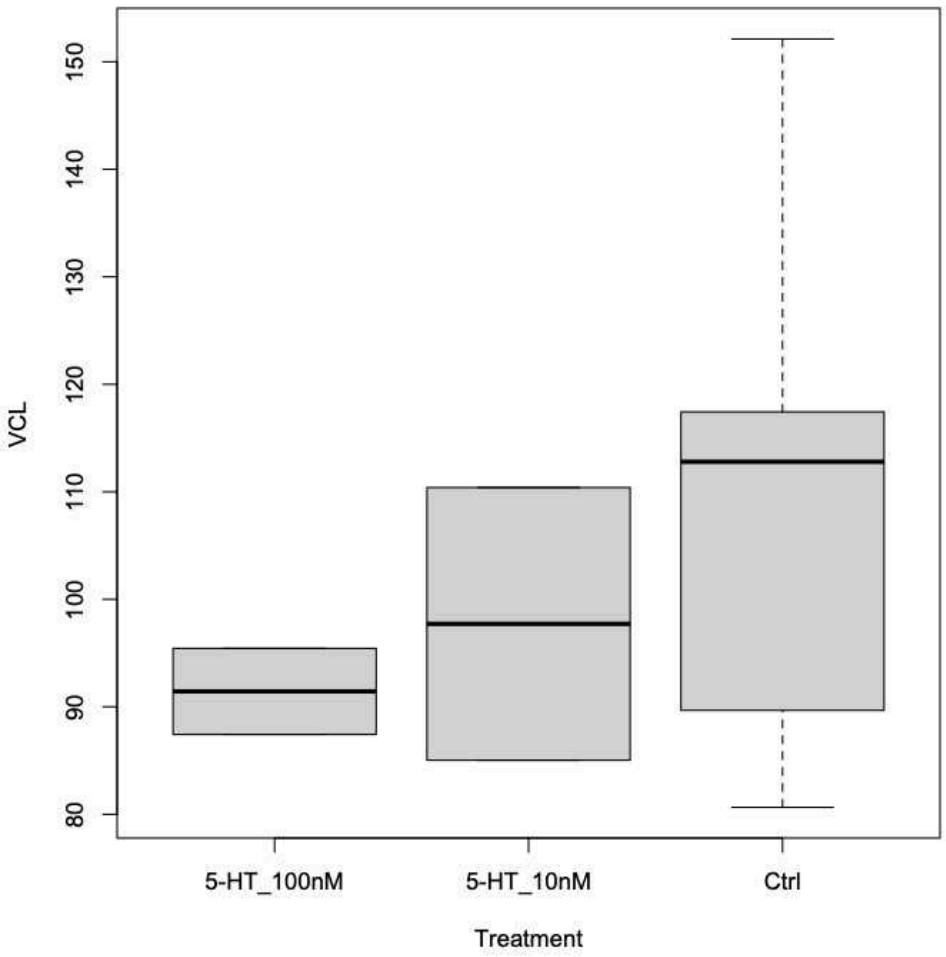
[1] "5-HT_100nM" "5-HT_10nM" "Ctrl"

Command

Plotting

Plotting

```
plot(casa_data_ordered$Treatment, casa_data_ordered$VCL,
xlab=Treatment, ylab=VCL)
```



5 **Reordering factor levels in specific columns.**

As you can see in the previous plot, the factor levels of the ***Treatment*** column are not ordered. It is common practice to plot the control data first, followed by the treatments. Reordering factor levels within a column is an important step in the dataset cleaning process.

Command

Reordering the factor levels

Reordering the factor levels

```
> casa_data_ordered<-casa_data[order(casa_data$Treatment,
decreasing=FALSE),]
```

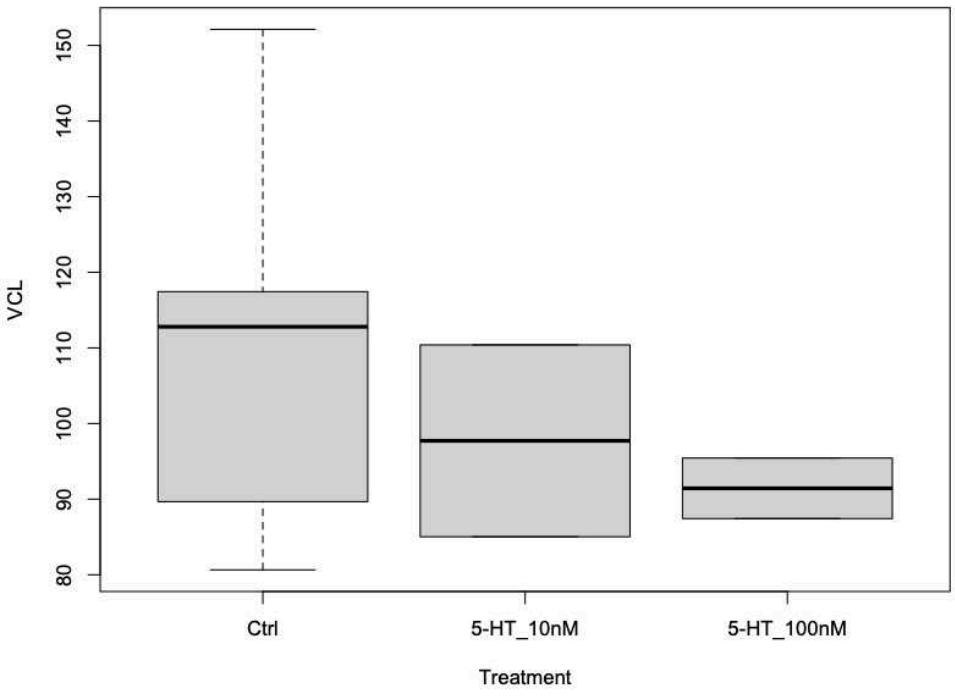
Let’s visualize the data again to confirm that the factor levels are properly ordered.

Command

Plotting

Plotting

```
plot(casa_data_ordered$Treatment, casa_data_ordered$VCL,
xlab=Treatment, ylab=VCL)
```



6 **Remove NA values**

A common issue when analyzing data is the presence of NA values, which can bias the results of statistical analyses. To ensure the quality, accuracy, and reliability of the analysis, it is necessary to remove NA values.

One of the functions that allows us to check for NA values in our dataset is

```
is.na()
```

However, this function returns a matrix of the same size as our dataframe.

Expected result

```
date donor Treatment exposure_time   VCL   VSL   LIN   ALH
1  FALSE FALSE     FALSE         FALSE FALSE FALSE FALSE FALSE
3  FALSE FALSE     FALSE         FALSE FALSE FALSE FALSE FALSE
4  FALSE FALSE     FALSE         FALSE FALSE FALSE FALSE FALSE
5  FALSE FALSE     FALSE         FALSE FALSE  TRUE FALSE  TRUE
8  FALSE FALSE     FALSE         FALSE  TRUE FALSE  TRUE FALSE
10 FALSE FALSE     FALSE         FALSE FALSE FALSE FALSE FALSE
11 FALSE FALSE     FALSE         FALSE FALSE FALSE FALSE FALSE
6  FALSE FALSE     FALSE         FALSE FALSE FALSE FALSE FALSE
9  FALSE FALSE     FALSE         FALSE FALSE FALSE FALSE FALSE
2  FALSE FALSE     FALSE         FALSE FALSE FALSE FALSE FALSE
7  FALSE FALSE     FALSE         FALSE FALSE FALSE FALSE FALSE
```

In our example, we are working with a small dataset. However, when dealing with kinematic parameter data, datasets typically contain a large amount of information, making it more difficult to visualize and identify NA values using this method.

On the other hand, we can simply count the number of NA values per column using

```
colSums(is.na())
```

Command

Count the number of NA values

Count the number of NA values

```
colSums(is.na(casa_data))
```

Expected result

```
date       donor   Treatment exposure_time         VCL         VSL         LIN         ALH
   0           0           0             0           1           1           1           1
```

Now that we know there are NA values, we can remove them.

Command

Remove NA values

Remove NA values

```
> casa_data<- na.omit(casa_data_ordered)
```

Expected result

date	donor	Treatment	exposure_time	VCL	VSL	LIN	ALH
0	0	0	0	0	0	0	0

At this point, we have a clean dataset: we have corrected typographical errors, reordered factor levels, and removed NA values.
We can use this dataset for statistical analyses or machine learning algorithms.

7 Save cleaned dataset
Finally, it is important to save the CSV file with the cleaned data so that it can be used for future analyses.

Command

Save csv

Save csv

```
> write.csv(casa_data, file = casa_data.csv, row.names = FALSE)
```

The file will be saved in your computer's working directory.