



Universidad Nacional Experimental Del Táchira  
Vicerrectorado Académico  
Decanato de Docencia  
Departamento de Ingeniería en Informática  
Trabajo de aplicación profesional  
Proyecto especial de grado

# Paquete en lenguaje R

# Simulación del modelo MOMOS

Autor: Ghabriel E. Villarreal C.  
Tutor: PhD. Rossana Timaure

# El problema

- En la actualidad el software es una herramienta que permite simplificar tareas que resultan complejas, por ejemplo, dar solución a un sinfín de cálculos estadísticos enfocados en las ciencias agronómicas.
- Se han desarrollado de manera práctica modelos de simulación dinámica para comprender los diferentes procesos y transformaciones que ocurren en el suelo, tales como, el modelo de Materia Orgánica y Micro-Organismos del Suelo (MOMOS).
- Este modelo fue planteado en un software privativo que está limitado al uso de funciones desarrolladas por la empresa de dicho software, lo que no permite ampliar la investigación sin acudir a otros software externos.

# Objetivos

## General:

- Desarrollar un paquete en el lenguaje R implementando el modelo MOMOS.

## Específicos:

- Estudiar la Estructura del modelo MOMOS.
- Crear las funciones primarias en el lenguaje R de la implementación del modelo MOMOS.
- Realizar las pruebas unitarias y funcionales del paquete creado en el lenguaje R.

# Metodología

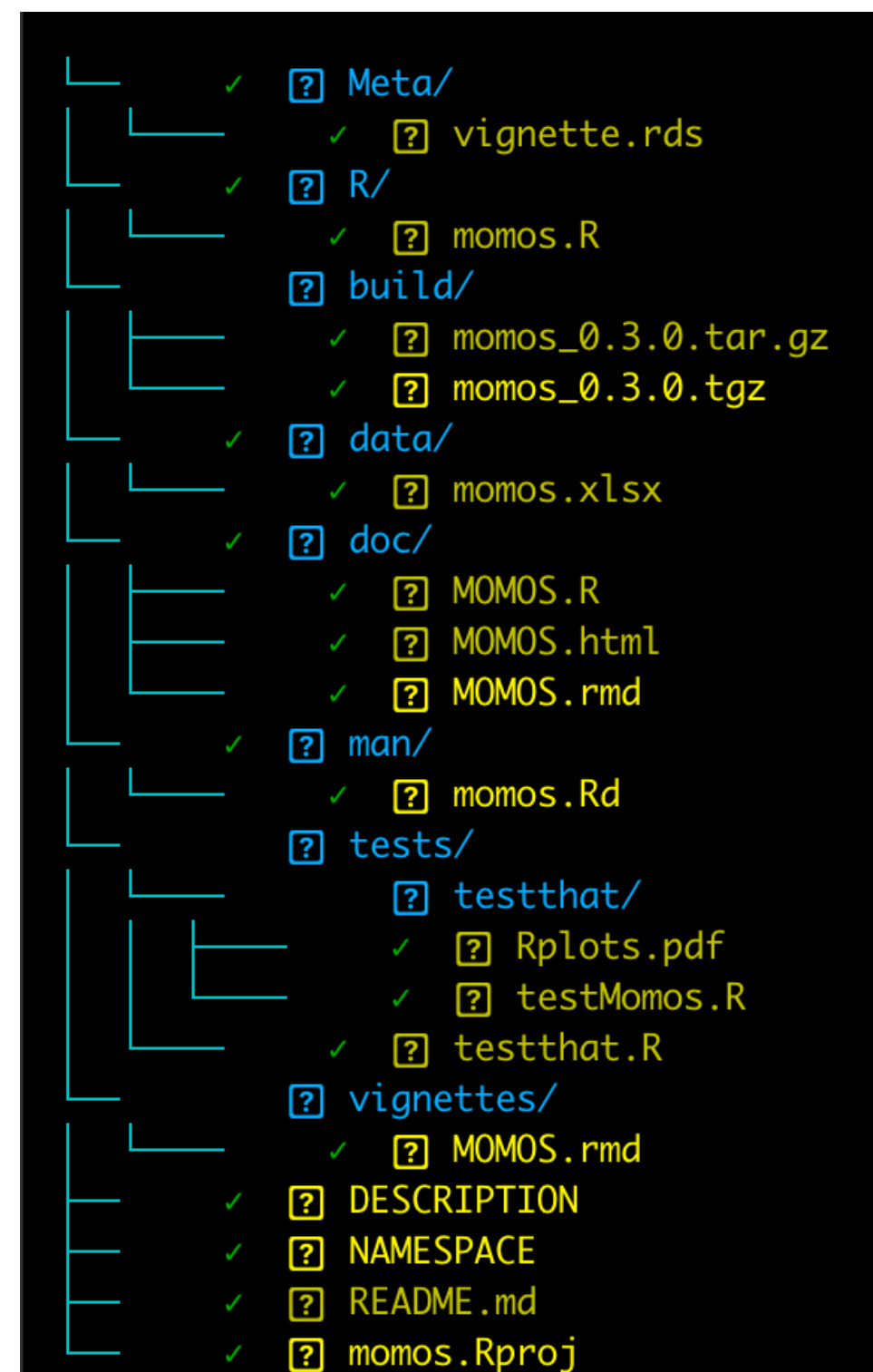
Para el desarrollo del paquete se implementó la metodología de creación de paquetes y extensiones en R:

- Creación del esqueleto del paquete.
- Registrar el método para el envío y uso de funciones.
- Diseño y codificación de las funciones.
- Pruebas unitarias de las funciones.
- Chequear la carga del paquete.
- Construcción del método de distribución del paquete.

# Desarrollo

## Creación del esqueleto del paquete

En esta etapa se creó y diseñó la estructura del paquete que se utilizó en la implementación.



# Desarrollo

## Registrar el método para el envío y uso de funciones

Se determina que el paquete reciba una lista de parámetros opcionales de entrada, los cuales sirven para calcular los valores de la simulación del modelo, también se implementaron las siguientes funciones:

- `calculate_momos()`: Realiza los cálculos del modelo dinámico y da como salida de la simulación las variables de estudio CM (carbono inicial de la biomasa microbiana) y RA (respiración inicial de la biomasa microbiana).
- `calibrate_momos()`: Ajusta los valores de salida de la función anterior para obtener un margen de error aceptable entre los valores experimentales y simulados.
- `graph_momos()`: Permite visualizar las curvas de las salidas de las funciones anteriores.
- `momos()`: Permite invocar en una sola función la salida de todas las funciones creadas.

# Desarrollo

## Diseño y codificación de las funciones

- Calcular los diferentes compartimientos necesarios para obtener la función inicial

```
y <- c(  
  VL = Necromasa * (1 - fs),  
  VS = Necromasa * fs,  
  CM = Ci,  
  HL = HLo,  
  HS = HSo,  
  RA = 0  
)
```

- Calcular las N derivadas las funciones de los compartimientos, donde N es rango del tiempo de ejecución

# Desarrollo

## Diseño y codificación de las funciones

```
# Derivative
derivs <- function(times, y, params) {
  with (as.list(c(params, y)), {
    Fv1 <- VL * Kv1
    Fvs <- VS * Kvs
    Fmor <- CM * Kmb
    Resp <- CM * CM * Kresp / Co
    Fh1 <- HL * Kh1
    Fh1s <- HL * Kh1s
    Fhs <- HS * Khs

    dVL <- -Fv1
    dVS <- -Fvs
    dCM <- Fv1 + Fvs - Fmor - Resp + Fh1 + Fhs
    dHL <- Fmor - Fh1 - Fh1s
    dHS <- Fh1s - Fhs
    dRA <- Resp

    return (list(c(dVL, dVS, dCM, dHL, dHS, dRA)))
  })
}
```



# Desarrollo

Diseño y codificación de las funciones

- Resolver el sistema de ecuaciones diferenciales ordinarias obtenidas de las derivadas usando la librería deSolve.

```
# Differential Equations  
out <- ode(  
  func = derivs,  
  y = y,  
  parms = pars,  
  times = times,  
  method = "rk4"  
)
```

# Desarrollo

## Diseño y codificación de las funciones

- Para calibrar el modelo de simulación dinámica se usa el algoritmo de levenberg-marquart para obtener el valor ideal del parámetro Kresp que permita el acercamiento entre los datos simulados y experimentales.

```
# parameter fitting using levenberg marquart algorithm  
# initial guess for parameters  
parms=c(Kresp=Kresp)  
# fitting  
fitval=nls.lm(par=parms,fn=ssq)  
print(summary(fitval))
```

- Para visualizar las curvas en cada paso del desarrollo de funciones se hace uso de la librería ggplot2 obteniendo una representación gráfica de los datos.

# Desarrollo

## Pruebas unitarias de las funciones

- Consiste en validar las funciones desarrolladas haciendo uso de la librería testthat asegurando que cualquier modificación del código no afecte la salida de cualquier función.
- Se valida la cobertura de pruebas del código escrito usando el paquete devtools disponible en R.

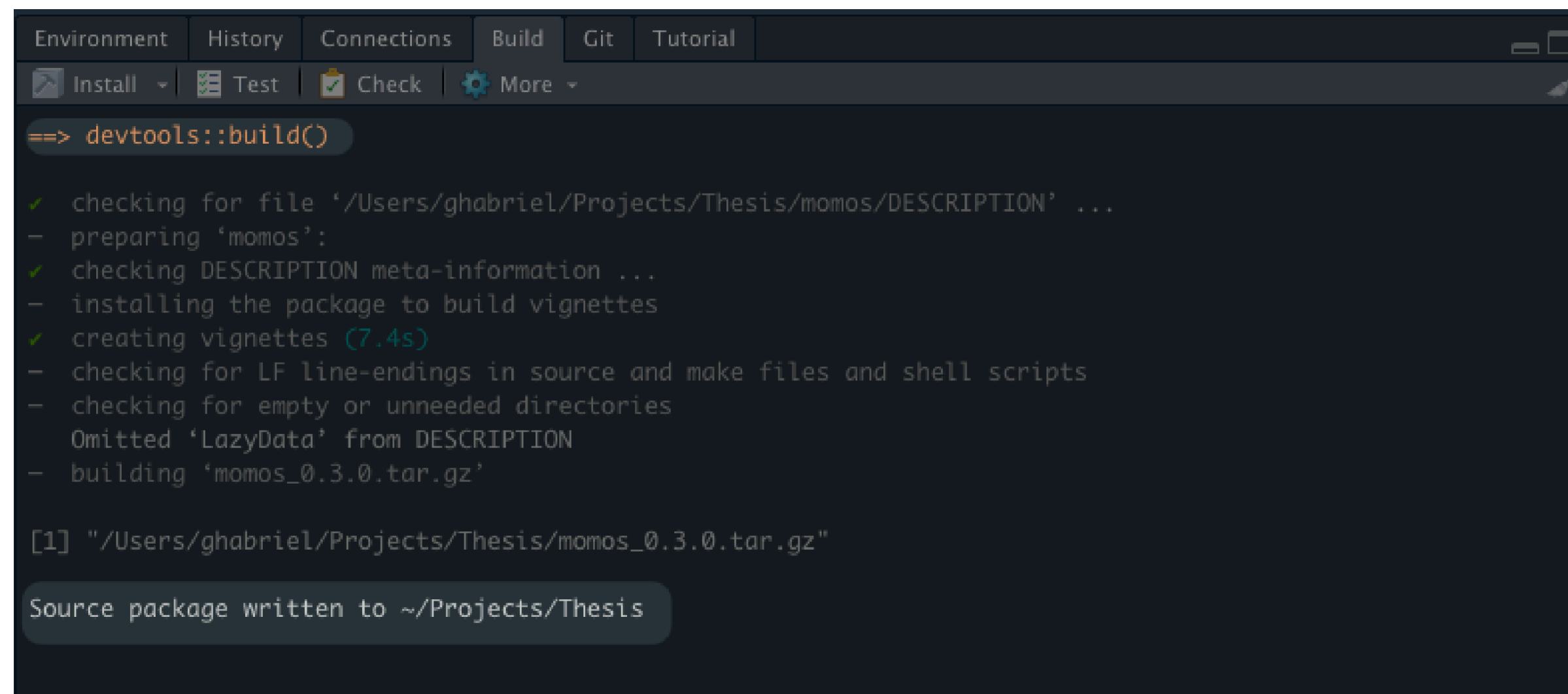
## Chequear la carga del paquete

- Se valida a través del método check de la librería devtools que el paquete cumple con la estructura de un paquete en R, y también generar la documentación respectiva.

# Desarrollo

## Construcción del método de distribución del paquete

- Se construye este paquete para la distribución del mismo en formato tgz para el código fuente, y en formato tar.gz para el binario de la librería.



```
Environment History Connections Build Git Tutorial
Install Test Check More
==> devtools::build()
✓ checking for file '/Users/ghabriel/Projects/Thesis/momos/DESCRIPTION' ...
- preparing 'momos':
✓ checking DESCRIPTION meta-information ...
- installing the package to build vignettes
✓ creating vignettes (7.4s)
- checking for LF line-endings in source and make files and shell scripts
- checking for empty or unneeded directories
  Omitted 'LazyData' from DESCRIPTION
- building 'momos_0.3.0.tar.gz'

[1] "/Users/ghabriel/Projects/Thesis/momos_0.3.0.tar.gz"

Source package written to ~/Projects/Thesis
```

# Resultados

Salida de la función `calculate_momos()` donde se muestra los datos simulados

```
> library(momos)
> momos()
      CM      RA
1  50.04000  0.0000
2 347.89934 245.6057
3 368.49484 668.1389
4 360.32194 1038.2973
5 337.31676 1357.5673
6 310.19164 1627.7234
7 283.82091 1854.0023
8 259.64030 2043.1562
9 237.86573 2201.6005
10 218.37927 2334.8244
11 200.99087 2447.3685
12 185.50634 2542.9515
13 171.74363 2624.6102
14 159.53579 2694.8246
15 148.73058 2755.6206
16 139.18931 2808.6549
17 130.78552 2855.2839
18 123.40378 2896.6196
19 116.93858 2933.5750
20 111.29344 2966.9007
21 106.38005 2997.2153
22 102.11767 3025.0290
23 98.43248 3050.7640
24 95.25716 3074.7696
25 92.53046 3097.3357
26 90.19685 3118.7033
27 88.20615 3139.0727
28 86.51329 3158.6107
29 85.07791 3177.4564
30 83.86417 3195.7257
> |
```

# Resultados

Salida de la función `calibrate_momos()` donde se muestra los datos calibrados

```
> print(summary(fitval))
```

Parameters:

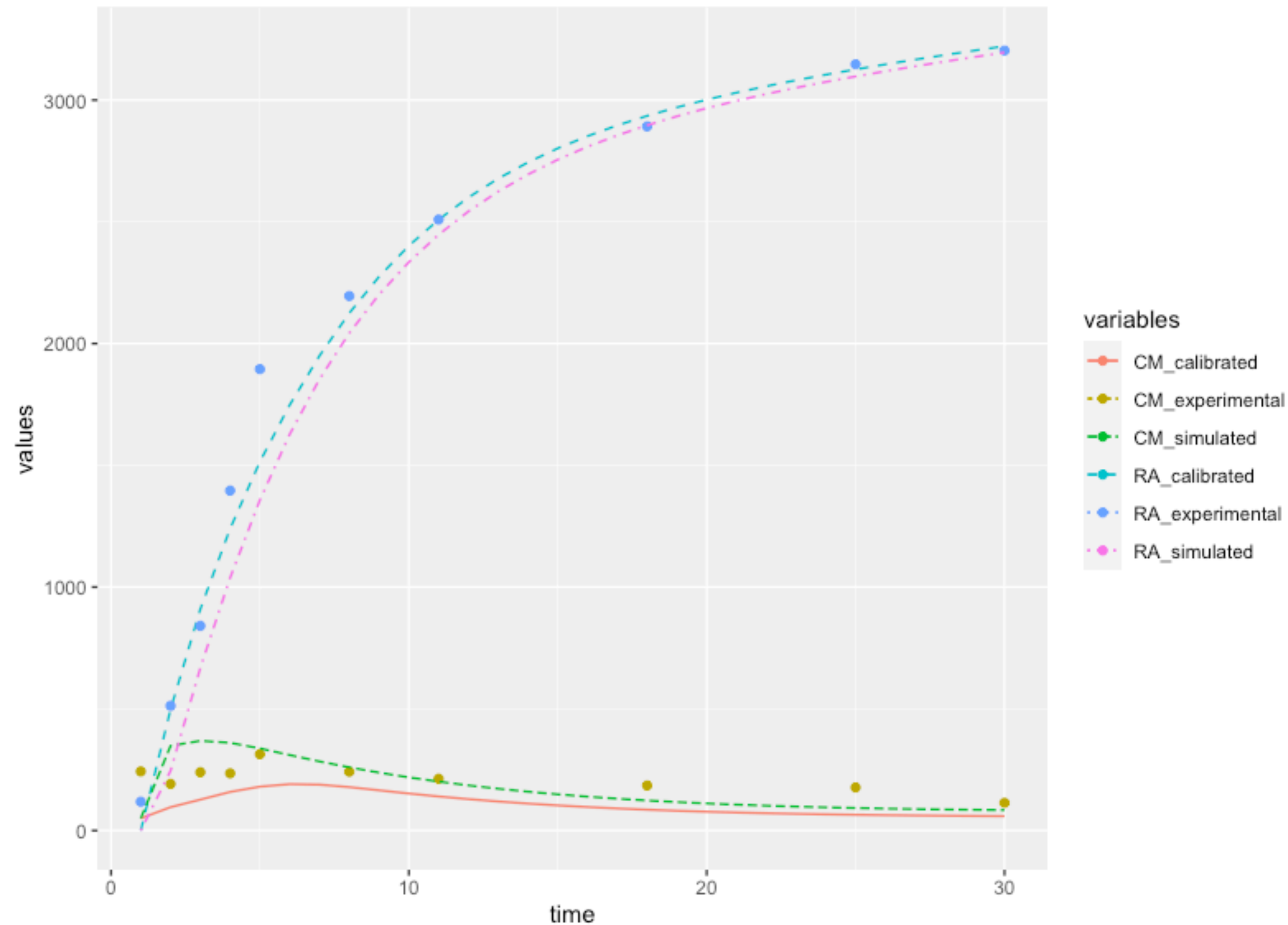
	Estimate	Std. Error	t value	Pr(> t )
Kresp	0.27879	0.03619	7.704	2.92e-07 ***

```
> calibrate_momos()
      CM      RA
1  50.04000  0.0000
2  95.99952 497.5417
3 127.02085 909.8104
4 158.61259 1240.3441
5 180.25281 1515.0821
6 190.50106 1747.9557
7 188.52944 1949.9086
8 178.35583 2125.1159
9 165.21227 2274.9809
10 152.07010 2401.9059
11 139.95188 2509.2205
12 129.06790 2600.2396
13 119.38400 2677.8530
14 110.80708 2744.4670
15 103.23571 2802.0574
16  96.57274 2852.2396
17  90.72794 2896.3341
18  85.61812 2935.4209
19  81.16674 2970.3841
20  77.30339 3001.9486
21  73.96328 3030.7095
22  71.08690 3057.1562
23  68.61969 3081.6914
24  66.51178 3104.6471
25  64.71781 3126.2977
26  63.19669 3146.8695
27  61.91146 3166.5498
28  60.82908 3185.4934
29  59.92022 3203.8284
30  59.15902 3221.6608
>
```



# Resultados

Salida de la función graph\_momos() donde se muestra las curvas de los datos



# Resultados

Ejecución de las pruebas unitarias y la cobertura

EnvironmentHistoryConnectionsBuildGitTutorial

InstallTestCheckMore

```
26 26 63.19669 3146.8695
27 27 61.91146 3166.5498
28 28 60.82908 3185.4934
29 29 59.92022 3203.8284
30 30 59.15902 3221.6608
[1] "===== DATOS DE MOMOS EXPERIMENTALES ====="
time CM RA
1 1 242.7826 118.520
2 2 191.5656 512.245
3 3 239.5755 840.725
4 4 234.9729 1396.190
5 5 313.7010 1895.146
6 8 241.4439 2195.296
7 11 212.7132 2509.596
8 18 184.8797 2892.021
9 25 176.9822 3147.396
10 30 113.9122 3203.661
[1] "===== GRAFICANDO EL MODELO ====="
✓ | 9 | Momos [1.2s]
```

Results

Duration: 1.2 s

[ FAIL 0 | WARN 0 | SKIP 0 | PASS 9 ]

FilesPlotsPackagesHelpViewerPresentation

momos coverage - 100.00%

FilesSource

File	Lines	Relevant	Covered	Missed	Hits / Line	Coverage
R/momos.R	258	126	126	0	1067	100.00%

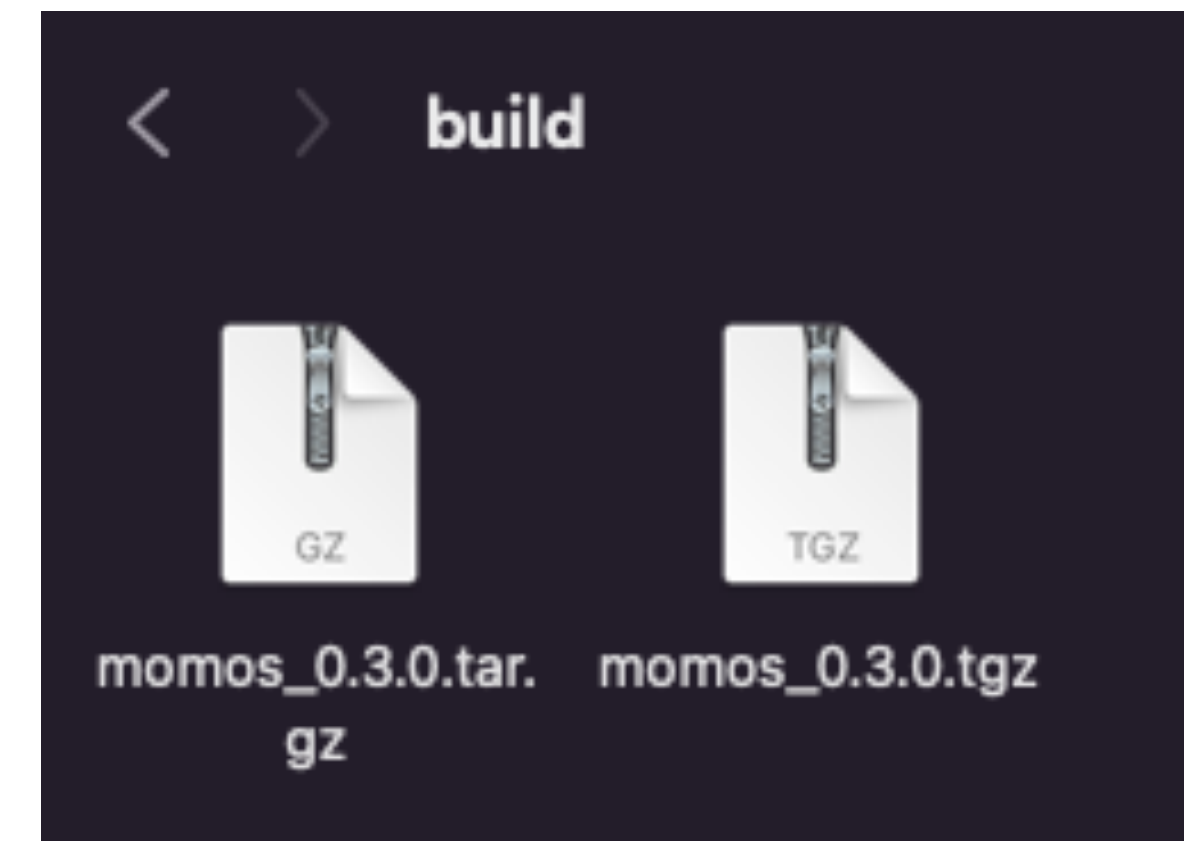


# Resultados

## Chequeo de estructura y compilación del paquete

```
Environment History Connections Build Git Tutorial
Install Test Check More
graph_momos: no visible binding for global variable 'values'
graph_momos: no visible binding for global variable 'variables'
graph_momos: no visible global function definition for 'geom_line'
graph_momos: no visible global function definition for 'geom_point'
momos: no visible binding for '<<-' assignment to 'out_simulated'
momos: no visible binding for global variable 'out_simulated'
momos: no visible binding for '<<-' assignment to 'out_calibrated'
momos: no visible binding for global variable 'experimental_data'
ssq: no visible binding for global variable 'from'
ssq: no visible binding for global variable 'to'
ssq: no visible binding for global variable 'at'
ssq: no visible binding for global variable 'experimental_data'
ssq: no visible global function definition for 'melt'
Undefined global functions or variables:
  Kresp aes at experimental_data from geom_line geom_point ggplot melt
  nls.lm ode out_calibrated out_simulated read.xlsx time to values
  variables
Consider adding
  importFrom("stats", "time")
to your NAMESPACE file.

0 errors ✓ | 0 warnings ✓ | 4 notes ✖
R CMD check succeeded
```



# Conclusiones y recomendaciones

- Se desarrolló el paquete en lenguaje R para la simulación del modelo MOMOS.
- Se implementaron las funciones matemáticas como: derivadas, ecuaciones diferenciales, suma de cuadrados, entre otros en dicho lenguaje.
- Se escribieron pruebas unitarias para cubrir el desarrollo realizado, y se ejecutaron pruebas funcionales con datos simulados y experimentales.
- Se calibró el parámetro  $K_{resp}$  para ajustar el modelo de simulación.
- Se recomienda calibrar otros parámetros del modelo.
- Capturar más datos experimentales para mejorar la precisión del algoritmo de ajuste utilizado.

**Gracias por su atención...**