The Global Journal of Sport and Exercise Science (GJSES)
Vol. 2, No. 1 (2026)
ISSN (Print): 3080-0307 ISSN (Online): 3107-037X
DOI (Crossref): https://doi.org/10.63750/rr27aa25

Received on: 13 December 2025 | Accepted on: 10 January 2026 | Published on: 31 March 2026

Longitudinal clustering in kinesiology 2: the longclust package in R

Keston Lindsay [1]

[1] University of Colorado Colorado Springs, USA klindsay@uccs.edu

ABSTRACT

High variation is common in longitudinal kinesiology studies, as participants may respond differently to interventions. In the previous article, implementation of longitudinal k-means in R is used in tandem with repeated measures (multivariate) analysis of variance to understand overall change, and to discover hidden trajectories. The purpose of this article is to demonstrate how to analyze data using model-based clustering. The same synthetic dataset is analyzed, using the longclust package in R. Comparison with the kml and kml3d packages is discussed.

Keywords: longitudinal clustering, multivariate repeated measures, cluster analysis, longitudinal model-based clustering, R

Corresponding Author:
(Keston Lindsay)

- Affiliation: Helen and Arthur Johnson Beth-El College of Nursing and Health Sciences, University of Colorado Colorado Springs, Colorado Springs, USA
- Email: klindsay@uccs.edu

1. INTRODUCTION:

In the previous article, high variability in longitudinal designs is discussed extensively. Briefly, variability in response patterns is documented in exercise and sports science (Hubal et al. 2005; Ojeda-Aravena, et al. 2021). Such variation in response patterns could lead to situations where the null hypothesis may be rejected overall, but there could be a hidden cohort of non-responders to the intervention. Conversely, a null hypothesis could be retained overall, with hidden cohorts of responders. Methods of longitudinal clustering could complement more traditional general linear model methods in order to better evaluate interventions. In addition to the *kml* and *kml3d* packages (Genolini et

al., 2015), another user-friendly R package for IBM SPSS users exploring R is the *longclust* package (McNicholas et al., 2023).

- ***Model-based clustering***

Model-based clustering (MBC) is a flexible clustering method that uses up to eight potential Gaussian mixture models to cluster longitudinal data (McNicholas & Murphy, 2010; McNicholas & Subedi, 2011). The models assume that potential clusters found in the data may be drawn from different distributions. The Bayesian Information Criterion (BIC; Schwarz, 1978) and the Integrated Completed Likelihood (ICL; Biernacki et al., 2000) are then used to decide the best model for the data. Tables 1 and 2 describe a glossary of terms and brief description of the models respectively.

Table 1 - Glossary of Terms

Term	Definition
Volume	Overall variation in the cluster.
Shape	How variances vary between time points.
Isotropic	Variance at each time point is the same
Anisotropic	Variance at each time point differs
Orientation	If and how earlier timepoints influence later ones (i.e., autoregressive structure)

Table 2 - Model Type

Model	Volume	Shape	Orientation	Complexity
EEI	Equal	Equal	Isotropic	Simplest
EEA	Equal	Equal	Anisotropic	Moderate
VEI	Variable	Equal	Isotropic	Moderate
EVI	Equal	Variable	Isotropic	Moderate
VVI	Variable	Variable	Isotropic	Moderate
VEA	Variable	Equal	Anisotropic	Complex
EVA	Equal	Variable	Anisotropic	Complex
VVA	Variable	Variable	Anisotropic	Most complex

- ***The longclust package***

The package used to implement MBC in R is *longclust* (McNicholas et al., 2023). Similar to IBM SPSS and *kml/kml3d* packages, it uses the wide format for analysis. Also similar to *kml/kml3d*, it visualizes trajectories, is capable of both univariate and multivariate analyses, and cluster membership may be appended to the original data. These cluster memberships may be used for necessary follow-up analyses (Genolini et al., 2015). Any follow-up analysis is

to be treated as exploratory, and not hypothesis-driven. The code for *longclust* is easy to write, and is friendly to R beginners. While *kml* and *kml3d* are non-parametric, MBC is parametric, and uses BIC and ICL as model criteria. Although there is more interpretation, *longclust* automatically chooses the best cluster solution based upon these criteria. Unlike *kml/kml3d,* the minimum number of clusters that *longclust* will compare is 2. Hence it will not compare a homogenous trajectory (i.e., a single cluster) to others as the *kml/kml3d* packages do. While more technical than *kml/kml3d*, *longclust* is still approachable for IBM SPSS users with some R experience. This article is meant to demonstrate the use of MBC with RMM and RMA to understand variation in longitudinal data.

- ***Dataset and data management***

The synthetic dataset ("DII_VB+SC_3clusters_WC_RM") and premise used are described in article 1, and is available here: https://osf.io/mrw5v/files/osfstorage. It was created using ChatGPT (OpenAI, 2025) to simulate an applied sports sciences question: how does a training program affect lower body strength (3-repetition max [3RM] in kg) and peak power in female volleyball players over five equal intervals over the course of an off-season?

It is re-used in this article to provide consistency of data across tutorials. Readers will note that the data renamed "MKS" is the exact same dataset that was named "sc3c" in the previous article. It was renamed as a form of data management, as the "sc3c" copy had the cluster membership appended to it. That said, the reader is encouraged to manage their data in the way that best suits them. Table 3 is a summary of the *longclust* commands. Similar to the previous article, the univariate strength and power outcomes will be analyzed. The multivariate analysis will comprise of analyzing both outcomes simultaneously, and will be called "performance".

Table 3 - Summary of commands for clustering in longclust

Command	Purpose
library(longclust)	Enables operation of the longclust package
MKS<-DII_VB_SC_3clusters_WC	Renames the original dataset as the primary copy to work with on this project.
strengthdata<-as.matrix(MKS[, 2:6])	Creation of cluster object for strength data. The five time points for strength are columns 2 through 6.
powerdata<-as.matrix(MKS[, 7:11])	Creation of cluster object for power data

performancedata<-as.matrix(MKS[, 2:11])	Creation of cluster object for multivariate strength and power data Ii.e., "performance") data.
View(performancedata) **View(powerdata)** **View(strengthdata)**	Visually check the cluster objects.
sc<-longclustEM(x = strengthdata, Gmin = 2, Gmax = 6, gaussian = FALSE, criteria = "BIC", initWithKMeans = TRUE)	Performs strength clustering using the *longclustEM* command. "initWithKMeans = TRUE" initializes the algorithm with k-means clustering
pc<-longclustEM(x = powerdata, Gmin = 2, Gmax = 6, gaussian = FALSE, criteria = "BIC", initWithKMeans = TRUE)	Performs power clustering.
pfc<-longclustEM(x = performancedata, Gmin = 2, Gmax = 6, gaussian = FALSE, criteria = "BIC", initWithKMeans = TRUE)	Performance clustering.
summary(sc) **summary(pc)** **summary(pfc)**	Obtains output for all three clustering solutions.
plot(sc, data = strengthdata) **plot(pc, data = powerdata)** **plot(pfc, data = performancedata)**	Visualizes clusters

2. METHOD

Univariate analyses: clustering and visualization for strength and power

When the summary command is run for strength clustering, the first output is the posterior probability (z-) matrix, as shown below in Figure 1:

```
> summary(sc)
Number of components: 3
z:
             [,1]         [,2]         [,3]
 [1,] 1.214933e-09 1.049849e-21 1.000000e+00
 [2,] 1.353315e-09 2.577036e-21 1.000000e+00
 [3,] 6.645359e-08 7.530640e-26 9.999999e-01
 [4,] 3.482635e-09 2.484143e-28 1.000000e+00
 [5,] 6.878135e-08 1.059991e-20 9.999999e-01
 [6,] 5.833578e-10 3.676042e-27 1.000000e+00
 [7,] 3.662561e-09 2.409916e-22 1.000000e+00
 [8,] 1.057637e-08 1.613264e-29 1.000000e+00
 [9,] 1.947944e-08 1.429458e-20 1.000000e+00
[10,] 3.248912e-09 5.041700e-25 1.000000e+00
[11,] 4.559933e-10 1.978967e-25 1.000000e+00
[12,] 7.025578e-10 2.915908e-24 1.000000e+00
[13,] 5.677307e-10 1.000000e+00 5.320845e-23
[14,] 9.732107e-10 1.000000e+00 2.477645e-28
[15,] 1.083296e-08 1.000000e+00 7.929768e-19
[16,] 5.078376e-10 1.000000e+00 1.607995e-28
[17,] 4.747394e-10 1.000000e+00 2.540336e-27
[18,] 5.421683e-10 1.000000e+00 1.419996e-21
[19,] 4.501190e-11 1.000000e+00 7.964239e-26
[20,] 2.695376e-10 1.000000e+00 1.828663e-26
[21,] 1.290127e-10 1.000000e+00 1.825180e-22
[22,] 2.298261e-10 1.000000e+00 1.250698e-28
[23,] 4.138109e-10 1.000000e+00 6.707741e-19
[24,] 1.799455e-10 1.000000e+00 1.649120e-24
[25,] 1.000000e+00 1.278718e-27 1.545401e-22
[26,] 1.000000e+00 7.838772e-25 1.697783e-17
[27,] 1.000000e+00 1.785348e-24 1.596361e-21
[28,] 1.000000e+00 9.710732e-25 3.010263e-21
[29,] 1.000000e+00 1.615392e-34 4.471423e-25
[30,] 1.000000e+00 2.029135e-29 1.038531e-23
[31,] 1.000000e+00 2.078061e-26 1.915967e-23
[32,] 1.000000e+00 1.594311e-27 3.204570e-24
[33,] 1.000000e+00 6.462929e-22 5.044698e-17
[34,] 1.000000e+00 2.320999e-29 3.224867e-19
[35,] 1.000000e+00 5.122900e-27 2.661245e-19
[36,] 1.000000e+00 1.774834e-23 9.467075e-19
```

Figure 1. Z-matrix

The columns correspond to cluster membership (called “components”), with the rows being designated as participants. Participants 1-12 have posterior probabilities of 1 in cluster 3, and values close to 0 for the other two clusters. Hence they have a probability at or close to 1, of belonging to cluster 3. Similarly, participants 13-24 have a probability of 1 of belonging to cluster 2, while participants 25-36 have a probability of 1, of belonging to cluster 1. Generally, the closer the posterior probability is to 1, is the more confident the algorithm is about the participant’s cluster membership. Following the z-matrix, are several lines of output and matrices for each cluster:

```
cluster:  1
v:  14.72262
mean: 98.45974 100.041 98.77317 99.68406 97.7148
T:
             [,1]        [,2]       [,3]       [,4] [,5]
[1,]  1.00000000  0.00000000 0.00000000 0.00000000    0
[2,]  0.38431829  1.00000000 0.00000000 0.00000000    0
[3,] -0.04431934 -0.02534261 1.00000000 0.00000000    0
[4,] -0.04356032  0.20172333 0.03269395 1.00000000    0
[5,] -0.15517826 -0.35446276 0.04851952 0.09445408    1
D:
         [,1]     [,2]     [,3]     [,4]     [,5]
[1,] 8.053221 0.000000 0.000000 0.000000 0.000000
[2,] 0.000000 8.053221 0.000000 0.000000 0.000000
[3,] 0.000000 0.000000 8.053221 0.000000 0.000000
[4,] 0.000000 0.000000 0.000000 8.053221 0.000000
[5,] 0.000000 0.000000 0.000000 0.000000 8.053221

cluster:
v:  221.3801
mean: 110.6605 112.0853 114.0775 115.6688 118.6639
T:
             [,1]        [,2]       [,3]       [,4] [,5]
[1,]  1.00000000  0.00000000 0.00000000 0.00000000    0
[2,]  0.38431829  1.00000000 0.00000000 0.00000000    0
[3,] -0.04431934 -0.02534261 1.00000000 0.00000000    0
[4,] -0.04356032  0.20172333 0.03269395 1.00000000    0
[5,] -0.15517826 -0.35446276 0.04851952 0.09445408    1
D:
         [,1]     [,2]     [,3]     [,4]     [,5]
[1,] 8.053221 0.000000 0.000000 0.000000 0.000000
[2,] 0.000000 8.053221 0.000000 0.000000 0.000000
[3,] 0.000000 0.000000 8.053221 0.000000 0.000000
[4,] 0.000000 0.000000 0.000000 8.053221 0.000000
[5,] 0.000000 0.000000 0.000000 0.000000 8.053221

cluster:  3
v:  236.2914
mean: 89.92118 97.16413 105.0907 111.2526 119.0812
T:
             [,1]        [,2]       [,3]       [,4] [,5]
[1,]  1.00000000  0.00000000 0.00000000 0.00000000    0
[2,]  0.38431829  1.00000000 0.00000000 0.00000000    0
[3,] -0.04431934 -0.02534261 1.00000000 0.00000000    0
[4,] -0.04356032  0.20172333 0.03269395 1.00000000    0
[5,] -0.15517826 -0.35446276 0.04851952 0.09445408    1
D:
         [,1]     [,2]     [,3]     [,4]     [,5]
[1,] 8.053221 0.000000 0.000000 0.000000 0.000000
[2,] 0.000000 8.053221 0.000000 0.000000 0.000000
[3,] 0.000000 0.000000 8.053221 0.000000 0.000000
[4,] 0.000000 0.000000 0.000000 8.053221 0.000000
[5,] 0.000000 0.000000 0.000000 0.000000 8.053221
```

Figure 2. T and D matrices

The most important for general interpretation are the means for each time point for each cluster. The means from cluster 1's timepoints go from around 98.5 to 97.7, suggesting this cluster's trajectory is relatively flat. The means for cluster 2's timepoints show an increase from around 111 to 119. The means for cluster 3 show a sharper increase from around 90 to 119. Hence, cluster 1 is a group of non-responders, cluster 2 is a group of moderate responders, and cluster 3 is a group of high responders.

The v-estimate represents the overall variability of the cluster. Cluster 1 has a variance of 14.7, while clusters 2 and 3 have relatively larger variances (221.4, and 236.3 respectively). The D-Matrix and T-matrix are produced by Cholesky decomposition of the covariance matrix. The T-Matrix shows how earlier timepoints predict later ones. The D-matrix shows the variability at each timepoint. Plainly stated, Cholesky decomposition separates time-related variability from the variability at each time point. It is consistent (~8.1) across both timepoints and clusters, and there is no correlation between timepoints (variances are 0 off-diagonal). More detail about Cholesky decomposition used in model-based clustering is beyond the scope of this paper, and may be obtained elsewhere (McNicholas & Murphy, 2010; McNicholas & Subedi, 2011; Pourahmadi, 1999). Figure 3 contains the BIC and ICL tables.

```
bicres:
      VVI     VVA     EEI     EEA     VEI     VEA     EVA     EVI
2 -1135.4 -1143.2 -1127.9 -1131.6 -1140.4 -1140.9 -1147.2 -1129.6
3 -1130.4 -1154.6 -1068.0 -1078.5 -1124.5 -1132.4 -1104.7 -1073.7
4 -1167.9    -Inf -1076.5 -1088.4 -1164.2    -Inf -1117.6 -1087.2
5    -Inf    -Inf -1096.9 -1109.2    -Inf    -Inf -1147.2 -1105.9
6    -Inf    -Inf -1115.4 -1125.0    -Inf    -Inf    -Inf -1133.8
iclres:
      VVI     VVA     EEI     EEA     VEI     VEA     EVA     EVI
2 -1135.4 -1143.2 -1127.9 -1131.6 -1140.4 -1140.9 -1147.2 -1129.6
3 -1130.4 -1154.6 -1068.0 -1078.5 -1124.5 -1132.4 -1104.7 -1073.7
4 -1168.0    -Inf -1076.8 -1088.6 -1164.2    -Inf -1117.6 -1087.4
5    -Inf    -Inf -1098.3 -1111.2    -Inf    -Inf -1147.2 -1106.4
6    -Inf    -Inf -1119.8 -1128.1    -Inf    -Inf    -Inf -1136.5
```

Figure 3. BIC and ICL tables

The best clustering solution should have the lowest log likelihood for all criteria across all clusters. The EEI of -1068 fits this criterion, which is why 3 clusters were chosen as the optimal solution. Note that running the summary command for power will yield a similar result, and interpretation is virtually identical. The clusters may be plotted using the aforementioned command:

>plot (sc, data = strengthdata)

Two figures will be displayed: one showing the relative trajectories of all clusters, and another showing trajectories for each individual cluster.

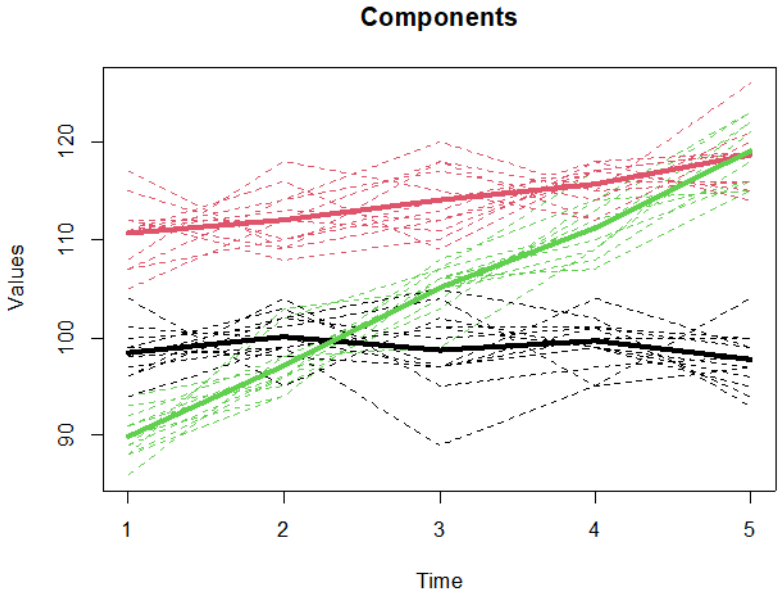

Figure 4. Comparative trajectories of all clusters

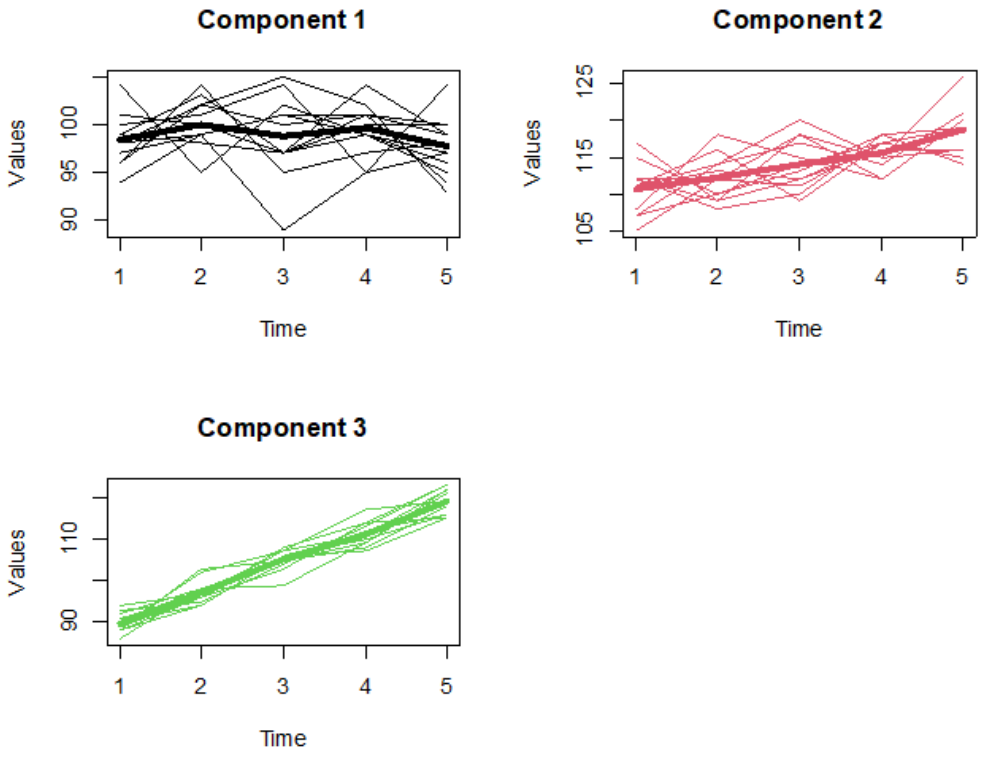

Figure 5. Single cluster trajectories

- **Multivariate analysis: clustering and visualization for performance**

The interpretation for performance is also identical to strength and power in this tutorial. Once again, the z-table shows an identical clustering pattern to the univariate strength and power clustering solutions. The major difference is the BIC and ICL models, where the EEA (anisotropic variance) is chosen instead of the EEI (equal variance). Both BIC and ICL values are -3260.8, confirming the 3-cluster solution.

```
bicres:
      VVI      VVA      EEI      EEA      VEI      VEA      EVA      EVI
2 -3807.2 -3339.6 -4035.1 -3318.1     -Inf -3391.6 -3324.4 -4036.0
3    -Inf -3394.3 -4086.4 -3260.8     -Inf -3433.5 -3306.9 -4089.6
4    -Inf     -Inf -4106.6 -3285.5     -Inf     -Inf     -Inf -4114.2
5    -Inf     -Inf -4109.7 -3306.5     -Inf     -Inf     -Inf -4123.0
6    -Inf     -Inf     -Inf -3320.9     -Inf     -Inf     -Inf     -Inf
iclres:
      VVI      VVA      EEI      EEA      VEI      VEA      EVA      EVI
2 -3807.5 -3339.6 -4035.1 -3318.1     -Inf -3391.6 -3324.4 -4036.0
3    -Inf -3394.3 -4086.4 -3260.8     -Inf -3433.5 -3306.9 -4089.6
4    -Inf     -Inf -4106.6 -3285.7     -Inf     -Inf     -Inf -4114.2
5    -Inf     -Inf -4109.7 -3306.7     -Inf     -Inf     -Inf -4123.0
6    -Inf     -Inf     -Inf -3321.0     -Inf     -Inf     -Inf     -Inf
```

While the code is provided for visualizing the multivariate trajectories, they do not give any more information in this scenario. However, multivariate clustering is useful when multiple outcomes are measured together. For example, multivariate clustering could find participants who may improve more

in strength than power, or show the opposite pattern, thus giving a more complete picture of how individuals respond to training. The complexity increases with more outcome variables.

Table 4 - Cluster membership and export commands

Command	Purpose
strengthclusters<-apply(sc$zbest, 1, which.max)	Creates a column for strength cluster membership. The *apply* command assigns clusters (referred to as "components" in the output) based on most likely membership.
MKS1<-cbind(MKS, strengthclusters) **View(MKS1)**	Creates a copy of the MKS data matrix with the strength cluster membership column appended. The "View" command is separate, and shows the new, appended dataset.
powerclusters<-apply(pc$zbest, 1, which.max)	Creates a column for power cluster membership
MKS2<-cbind(MKS1, powerclusters) **View(MKS2)**	Creates a copy of the MKS1 data matrix with the power cluster membership column appended.
performanceclusters<-apply(pfc$zbest, 1, which.max)	Creates a column for performance cluster membership.
MKS3<-cbind(MKS2, performanceclusters)	Creates a copy of the MKS2 data matrix with the power cluster membership column appended.
library(openxlsx)	Loads openxlsx package to export data to Excel spreadsheet
write.xlsx(MKS3, "MKS3.xlsx.", rownames = FALSE)	Exports Microsoft Excel spreadsheet to the folder you are using.
library(haven)	Loads the haven package to export to an IBM SPSS spreadsheet
write_sav(MKS3, "MKS3.sav")	Saves spreadsheet as stated above.

Once cluster memberships have been appended to the data, they may be exported to Microsoft Excel and other packages like IBM SPSS using the openxlsx (Schauberger et al., 2025) or haven (Wickham et al., 2025) packages respectively. Further evaluation and comparison of clusters can be done using ANOVA methods, but it should again be emphasized that this step is exploratory only, and not hypothesis driven. This step is discussed extensively in Article 1.

3. RESULTS

Sample Analysis and Results Sections. This section is dedicated to providing a template for comprehensive and transparent reporting of analysis and findings.

- **Analysis**

Descriptive statistics were performed using IBM SPSS version 30 (Armonk, NY). Repeated measures (RM) MANOVA for the effect of time on performance, and follow up RM ANOVA analyses for both strength and power were also performed using IBM SPSS. The interaction between cluster membership and time on performance was analyzed using a 3 x 5 MANOVA in IBM SPSS, followed by 3 x 5 ANOVA on strength and power.

Model based clustering was performed in R (R Core Team, 2025) using the *longclust* package (McNicholas et al., 2023). Exploratory comparisons between clusters were also completed using RM MANOVA and RM ANOVA in IBM SPSS.

- **Results.**

The null hypothesis for repeated measures MANOVA was rejected ($\Lambda = .59$, $F(8, 278) = 10.4$, $p < .001$, $\eta 2p = .23$). Similarly, the null hypothesis for strength [$F(1.6, 56.3) = 23.6$, $p < .001$, $\eta 2p = .40$)] and power [$F(1.6, 56.6) = 14.8$, $p < .001$, $\eta 2p = .30$] were rejected. As sphericity was violated, Greenhouse-Geisser adjustments were applied to both the strength and power analyses.

Model-based clustering revealed three clusters for strength, power, and performance. For strength and power, EEI (see Table 2) was interpreted for both the Bayesian Information Criterion (-1068), and the Integrated Completed Likelihood (-1068). For performance, EEA was interpreted (-3260.8).

- **Cluster Interpretation**

The previous article contains an extensive discussion of the cluster differences. It may be used interchangeably here, as the outcome is exactly the same between both packages. Cluster 1 in this analysis is equivalent to Cluster C in the previous article, Cluster 2 (current) is Cluster B (previous), and Cluster 3 (current) is Cluster A (previous). Generally speaking, Cluster 1 is a group of non-responders, Cluster 2 is a relatively higher performing group of moderate responders, and Cluster 3 is a group is high responders. Overall, clusters differed meaningfully across timepoints, with Cluster 3 showing the largest gains.

Table 5 - Cluster descriptive statistics

Variable/Timepoint	Cluster 1	Cluster 2	Cluster 3
Strength/ Time 1	98.4 ± 2.6	110.7 ± 3.4	89.9 ± 2.4
Strength/ Time 2	100.0 ± 2.4	112.1 ± 3.1	97.2 ± 2.8
Strength/ Time 3	98.8 ± 4.4	114.1 ± 3.5	105.1 ± 2.4
Strength/ Time 4	99.4 ± 2.7	115.7 ± 2.1	111.3 ± 2.9
Strength/ Time 5	97.7 ± 3.0	118.7 ± 3.2	119.1 ± 3.1
Power/ Time 1	2776.9 ± 42.4	3035.6 ± 69.3	2582.3 ± 78.8
Power/ Time 2	2806.0 ± 78.1	3059.2 ± 54.7	2785.4 ± 58,1
Power/ Time 3	2773.7 ± 58.8	3052.0 ± 66.1	2897.7 ± 62.4
Power/ Time 4	2787.1 ± 62.0	3052.0 ± 66.1	3066.7 ± 84.4
Power/ Time 5	2746.4 ± 55.9	3118.3 ± 122.8	3166.4 ± 81.9

- **Discussion**

Although *longclust* and the *kml/kml3d* packages use different algorithms, the results were identical using this dataset. However, clustering similarity between cluster algorithms is not guaranteed. Regardless of which clustering algorithm is used, there is no "correct" number of clusters, but the trajectories should make sense based upon theory and context. Both packages utilize "wide" format, with each column representing a timepoint, or context, in longitudinal designs. This is friendly to IBM SPSS users, as longitudinal data are organized in this way.

- **Limitations**

The *longclust* package assumes complete data, and does not support imputation at the time of this writing. It also does not allow for the inclusion of covariates in clustering. This may limit its use in studies where contextual variables could be important for trajectory interpretation. Researchers should ensure complete (or already imputed) data, and consider other methods if covariate adjustment is needed. The minimum number of clusters that this version of the *longclust* package compares is two. Hence it cannot compare cluster solutions with a homogenous trajectory (i.e., 1 cluster).

The dataset used in this tutorial is synthetic and is meant for demonstration only. As it may not emulate the complexity of real data, its generalizability is limited. Researchers are strongly encouraged to test these longitudinal clustering methods using actual study data.

- **Conclusion**

Like *kml/kml3d*, the *longclust* package is easy to use for beginner R programmers. Like *kml/kml3d,* they may be a useful tool who are more comfortable with GUI-driven packages. It may be used in tandem with classic general linear models to better understand longitudinal data. Users are encouraged to try both the non-parametric (*kml/kml3d*) and model-based (*longclust*) approaches to compare solutions. By comparing different packages, students may appreciate how different algorithms may converge on cluster solutions.

- ***Contributions***

The sole author bears the responsibility for all categories of this item.

- ***Acknowledgements***

None.

- ***Funding information***

None

- ***Data and Supplementary Material Accessibility***

The dataset is available here: https://osf.io/mrw5v/files/osfstorage

References

- Biernacki, C., Celeux, G., & Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(7), 719–725. https://doi.org/10.1109/34.865189
- Genolini, C., Alacoque, X., Sentenac, M., & Arnaud, C. (2015). kml and kml3d: R packages to cluster longitudinal data. *Journal of Statistical Software, 65*, 1–34. https://doi.org/10.18637/jss.v065.i04
- Hubal, M. J., Gordish-Dressman, H., Thompson, P. D., Price, T. B., Hoffman, E. P., Angelopoulos, T. J., Gordon, P. M., Moyna, N. M., Pescatello, L. S., Visich, P. S., Zoeller, R. F., Seip, R. L., & Clarkson, P. M. (2005). Variability in muscle size and strength gain after unilateral resistance training. *Medicine & Science in Sports & Exercise, 37*(6), 964–972. https://doi.org/10.1249/01.mss.0000170469.83910.20
- McNicholas, P. D., & Murphy, T. B. (2010). Model-based clustering of longitudinal data. *Canadian Journal of Statistics, 38*(1), 153–168. https://doi.org/10.1002/cjs.10047

- McNicholas, P. D., & Subedi, S. (2011). Clustering gene expression time course data using mixtures of multivariate t-distributions. *Journal of Statistical Planning and Inference, 142*(5), 1114–1127. https://doi.org/10.1016/j.jspi.2011.11.026
- McNicholas, P. D., Jampani, K. R., & Subedi, S. (2023). *longclust: Model-based clustering and classification for longitudinal data* (Version 1.5) [Computer software]. CRAN. https://CRAN.R-project.org/package=longclust
- OpenAI. (2025). *ChatGPT* [Large language model]. https://chat.openai.com/
- Pourahmadi, M. (1999). Joint mean-covariance models with applications to longitudinal data: Unconstrained parameterisation. *Biometrika, 86*(3), 677–690. https://doi.org/10.1093/biomet/86.3.677
- R Core Team. (2025). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.r-project.org/
- Schauberger, P., Walker, A., Braglia, L., Sturm, J., Garbuszus, J. M., Barbone, J. M., Zimmermann, D., & Kainhofer, R. (2025). *openxlsx: Read, write and edit xlsx files* (Version 4.2.8) [Computer software]. CRAN. https://CRAN.R-project.org/package=openxlsx
- Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics, 6*(2), 461–464. https://doi.org/10.1214/aos/1176344136
- Wickham, H., Miller, E., & Smith, D. (2025). *haven: Import and export 'SPSS', 'Stata' and 'SAS' files* (Version 2.5.5.9000) [Computer software]. GitHub. https://github.com/tidyverse/haven