June Agathe

#Install libraries  
# install.packages(c("pacman", "tidyverse", "rio"))  
  
#load libraries  
library(tidyverse)

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.4 ✔ readr 2.1.5  
✔ forcats 1.0.0 ✔ stringr 1.5.1  
✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
✔ purrr 1.0.2   
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(sf)

Linking to GEOS 3.12.1, GDAL 3.8.4, PROJ 9.3.1; sf\_use\_s2() is TRUE

library(gstat)  
library(sp)  
  
  
data <- read.table(file = "data/Data\_Example\_1.txt", header = TRUE, sep = "\t")  
data$sampleID <- as.numeric(data$sampleID)  
data$subplot <- as.numeric(data$subplot)  
data$OTU\_all\_taxa <- as.numeric(data$OTU\_all\_taxa)  
  
# Créer des sous-ensembles de données pour June  
june <- data[data$month == "June", ]  
  
# Spécifier les colonnes x et y comme coordonnées  
coordinates(june) <- ~x + y  
  
# Calculer le variogramme empirique pour novembre  
var1 <- variogram((log(OTU\_all\_taxa) \* 10) ~ 1, locations = june, cutoff = 8, width = 1)  
  
# Ajuster le modèle de variogramme  
mod1 <- fit.variogram(var1, vgm(psill = NA, "Exp", range = NA, 1), fit.sills = TRUE, fit.ranges = TRUE, fit.method = 1)  
  
mod1

model psill range  
1 Nug 3.468018 0.000000  
2 Exp 1.775165 2.533755

# Validation croisée pour évaluer la précision du modèle  
cv <- krige.cv((log(OTU\_all\_taxa) \* 10) ~ 1, locations = june, model = mod1, nfold = nrow(june)) # cv : Corss validation  
  
cv

coordinates var1.pred var1.var observed residual zscore fold  
1 (0.835, 9.75) 32.35271 4.781706 33.32205 0.96933260 0.443283306 1  
2 (0.835, 9.25) 32.07693 4.720920 34.33987 2.26293944 1.041500830 2  
3 (3, 9.75) 31.95989 4.655785 31.35494 -0.60495221 -0.280365533 3  
4 (3, 9.25) 31.73952 4.558201 32.18876 0.44923638 0.210415729 4  
5 (4.67, 8.75) 31.87721 4.500186 28.90372 -2.97349173 -1.401688424 5  
6 (4.67, 8.25) 31.30981 4.447956 32.18876 0.87894793 0.416756992 6  
7 (6.34, 9.75) 31.28219 4.633743 32.58097 1.29877574 0.603348459 7  
8 (6.34, 9.25) 32.24692 4.505686 28.33213 -3.91478511 -1.844282639 8  
9 (7.515, 8.75) 31.91314 4.520799 34.01197 2.09883219 0.987120384 9  
10 (7.515, 8.25) 32.08732 4.499322 33.67296 1.58563744 0.747532977 10  
11 (9.68, 9.75) 32.10943 4.816092 31.35494 -0.75448991 -0.343800152 11  
12 (9.68, 9.25) 32.08867 4.750311 31.78054 -0.30813446 -0.141377152 12  
13 (1.33, 6.75) 32.23264 4.550461 29.44439 -2.78824861 -1.307084831 13  
14 (1.33, 6.25) 31.79202 4.544358 31.78054 -0.01147949 -0.005385008 14  
15 (2.505, 7.75) 31.63679 4.493226 32.95837 1.32158108 0.623468739 15  
16 (2.505, 7.25) 31.88997 4.472782 31.35494 -0.53502960 -0.252981596 16  
17 (4.175, 6.75) 32.24608 4.420558 28.90372 -3.34235775 -1.589697069 17  
18 (4.175, 6.25) 31.79726 4.376703 32.58097 0.78370363 0.374609107 18  
19 (5.845, 7.75) 31.96153 4.408810 32.18876 0.22722908 0.108218929 19  
20 (5.845, 7.25) 31.71723 4.406019 34.33987 2.62263940 1.249439924 20  
21 (7.02, 6.75) 32.29022 4.408811 30.44522 -1.84499903 -0.878689422 21  
22 (7.02, 6.25) 32.57417 4.343509 29.44439 -3.12977684 -1.501733979 22  
23 (9.68, 6.75) 32.64167 4.658884 32.95837 0.31670304 0.146727426 23  
24 (9.68, 6.25) 32.51639 4.615357 34.01197 1.49558371 0.696158391 24  
25 (0.34, 4.75) 32.46203 4.609096 32.58097 0.11893521 0.055399083 25  
26 (0.34, 4.25) 32.35534 4.571477 33.67296 1.31762133 0.616257691 26  
27 (2.505, 4.75) 32.49285 4.462378 31.78054 -0.71231164 -0.337199446 27  
28 (2.505, 4.25) 32.25564 4.397724 33.67296 1.41731962 0.675855431 28  
29 (4.67, 5.75) 32.41617 4.351863 32.18876 -0.22741521 -0.109013907 29  
30 (4.67, 5.25) 32.50868 4.375041 33.32205 0.81336047 0.388858850 30  
31 (5.35, 4.75) 32.46519 4.393325 35.55348 3.08829145 1.473403164 31  
32 (5.35, 4.25) 33.31398 4.444281 31.35494 -1.95903490 -0.929268910 32  
33 (7.02, 5.75) 32.50780 4.348754 32.18876 -0.31903729 -0.152988593 33  
34 (7.02, 5.25) 32.57615 4.427833 34.33987 1.76372575 0.838176382 34  
35 (9.68, 4.75) 33.05431 4.536408 32.58097 -0.47334907 -0.222241668 35  
36 (9.68, 4.25) 33.09967 4.481405 32.95837 -0.14130552 -0.066750118 36  
37 (0.34, 2.75) 31.41421 4.577411 36.37586 4.96165130 2.319082891 37  
38 (0.34, 2.25) 32.46852 4.618190 29.95732 -2.51119252 -1.168541435 38  
39 (2.01, 3.75) 32.69018 4.382312 31.78054 -0.90963831 -0.434527318 39  
40 (2.01, 3.25) 31.96020 4.432279 34.65736 2.69716223 1.281130892 40  
41 (4.175, 2.75) 32.42860 4.421872 32.18876 -0.23984327 -0.114057660 41  
42 (4.175, 2.25) 31.40521 4.318406 35.55348 4.14827032 1.996205918 42  
43 (6.34, 2.75) 32.97553 4.429113 34.65736 1.68183364 0.799143295 43  
44 (6.34, 2.25) 32.36135 4.421899 35.83519 3.47384168 1.651983170 44  
45 (7.515, 3.75) 33.16835 4.459111 34.96508 1.79672820 0.850860303 45  
46 (7.515, 3.25) 33.43689 4.461026 33.32205 -0.11484915 -0.054376393 46  
47 (9.185, 3.75) 33.32148 4.434835 33.67296 0.35148067 0.166902448 47  
48 (9.185, 3.25) 32.72946 4.535677 36.10918 3.37972043 1.586937163 48  
49 (1.33, 0.75) 30.69983 4.588768 28.90372 -1.79611348 -0.838466533 49  
50 (1.33, 0.25) 30.92687 4.716387 27.72589 -3.20098658 -1.473937660 50  
51 (2.505, 1.75) 31.31822 4.457222 29.44439 -1.87382650 -0.887559004 51  
52 (2.505, 1.25) 30.76730 4.489029 29.95732 -0.80997874 -0.382293948 52  
53 (4.175, 1.75) 31.61529 4.314196 30.44522 -1.17006366 -0.563325717 53  
54 (4.175, 1.25) 31.44929 4.441197 28.33213 -3.11715957 -1.479139153 54  
55 (5.35, 1.75) 31.87595 4.373899 31.78054 -0.09540858 -0.045619766 55  
56 (5.35, 1.25) 31.64385 4.447483 29.95732 -1.68652425 -0.799715329 56  
57 (7.02, 0.75) 31.03347 4.567930 31.78054 0.74706978 0.349543681 57  
58 (7.02, 0.25) 31.95688 4.664073 26.39057 -5.56630784 -2.577416119 58  
59 (9.185, 0.75) 32.06224 4.725295 30.44522 -1.61701339 -0.743873584 59  
60 (9.185, 0.25) 31.46920 4.784392 31.78054 0.31133858 0.142337573 60

cv\_df <- as.data.frame(cv)  
  
rio::export(cv\_df, "docs/cv\_agathe", format = "csv")  
  
cv\_df

var1.pred var1.var observed residual zscore fold x y  
1 32.35271 4.781706 33.32205 0.96933260 0.443283306 1 0.835 9.75  
2 32.07693 4.720920 34.33987 2.26293944 1.041500830 2 0.835 9.25  
3 31.95989 4.655785 31.35494 -0.60495221 -0.280365533 3 3.000 9.75  
4 31.73952 4.558201 32.18876 0.44923638 0.210415729 4 3.000 9.25  
5 31.87721 4.500186 28.90372 -2.97349173 -1.401688424 5 4.670 8.75  
6 31.30981 4.447956 32.18876 0.87894793 0.416756992 6 4.670 8.25  
7 31.28219 4.633743 32.58097 1.29877574 0.603348459 7 6.340 9.75  
8 32.24692 4.505686 28.33213 -3.91478511 -1.844282639 8 6.340 9.25  
9 31.91314 4.520799 34.01197 2.09883219 0.987120384 9 7.515 8.75  
10 32.08732 4.499322 33.67296 1.58563744 0.747532977 10 7.515 8.25  
11 32.10943 4.816092 31.35494 -0.75448991 -0.343800152 11 9.680 9.75  
12 32.08867 4.750311 31.78054 -0.30813446 -0.141377152 12 9.680 9.25  
13 32.23264 4.550461 29.44439 -2.78824861 -1.307084831 13 1.330 6.75  
14 31.79202 4.544358 31.78054 -0.01147949 -0.005385008 14 1.330 6.25  
15 31.63679 4.493226 32.95837 1.32158108 0.623468739 15 2.505 7.75  
16 31.88997 4.472782 31.35494 -0.53502960 -0.252981596 16 2.505 7.25  
17 32.24608 4.420558 28.90372 -3.34235775 -1.589697069 17 4.175 6.75  
18 31.79726 4.376703 32.58097 0.78370363 0.374609107 18 4.175 6.25  
19 31.96153 4.408810 32.18876 0.22722908 0.108218929 19 5.845 7.75  
20 31.71723 4.406019 34.33987 2.62263940 1.249439924 20 5.845 7.25  
21 32.29022 4.408811 30.44522 -1.84499903 -0.878689422 21 7.020 6.75  
22 32.57417 4.343509 29.44439 -3.12977684 -1.501733979 22 7.020 6.25  
23 32.64167 4.658884 32.95837 0.31670304 0.146727426 23 9.680 6.75  
24 32.51639 4.615357 34.01197 1.49558371 0.696158391 24 9.680 6.25  
25 32.46203 4.609096 32.58097 0.11893521 0.055399083 25 0.340 4.75  
26 32.35534 4.571477 33.67296 1.31762133 0.616257691 26 0.340 4.25  
27 32.49285 4.462378 31.78054 -0.71231164 -0.337199446 27 2.505 4.75  
28 32.25564 4.397724 33.67296 1.41731962 0.675855431 28 2.505 4.25  
29 32.41617 4.351863 32.18876 -0.22741521 -0.109013907 29 4.670 5.75  
30 32.50868 4.375041 33.32205 0.81336047 0.388858850 30 4.670 5.25  
31 32.46519 4.393325 35.55348 3.08829145 1.473403164 31 5.350 4.75  
32 33.31398 4.444281 31.35494 -1.95903490 -0.929268910 32 5.350 4.25  
33 32.50780 4.348754 32.18876 -0.31903729 -0.152988593 33 7.020 5.75  
34 32.57615 4.427833 34.33987 1.76372575 0.838176382 34 7.020 5.25  
35 33.05431 4.536408 32.58097 -0.47334907 -0.222241668 35 9.680 4.75  
36 33.09967 4.481405 32.95837 -0.14130552 -0.066750118 36 9.680 4.25  
37 31.41421 4.577411 36.37586 4.96165130 2.319082891 37 0.340 2.75  
38 32.46852 4.618190 29.95732 -2.51119252 -1.168541435 38 0.340 2.25  
39 32.69018 4.382312 31.78054 -0.90963831 -0.434527318 39 2.010 3.75  
40 31.96020 4.432279 34.65736 2.69716223 1.281130892 40 2.010 3.25  
41 32.42860 4.421872 32.18876 -0.23984327 -0.114057660 41 4.175 2.75  
42 31.40521 4.318406 35.55348 4.14827032 1.996205918 42 4.175 2.25  
43 32.97553 4.429113 34.65736 1.68183364 0.799143295 43 6.340 2.75  
44 32.36135 4.421899 35.83519 3.47384168 1.651983170 44 6.340 2.25  
45 33.16835 4.459111 34.96508 1.79672820 0.850860303 45 7.515 3.75  
46 33.43689 4.461026 33.32205 -0.11484915 -0.054376393 46 7.515 3.25  
47 33.32148 4.434835 33.67296 0.35148067 0.166902448 47 9.185 3.75  
48 32.72946 4.535677 36.10918 3.37972043 1.586937163 48 9.185 3.25  
49 30.69983 4.588768 28.90372 -1.79611348 -0.838466533 49 1.330 0.75  
50 30.92687 4.716387 27.72589 -3.20098658 -1.473937660 50 1.330 0.25  
51 31.31822 4.457222 29.44439 -1.87382650 -0.887559004 51 2.505 1.75  
52 30.76730 4.489029 29.95732 -0.80997874 -0.382293948 52 2.505 1.25  
53 31.61529 4.314196 30.44522 -1.17006366 -0.563325717 53 4.175 1.75  
54 31.44929 4.441197 28.33213 -3.11715957 -1.479139153 54 4.175 1.25  
55 31.87595 4.373899 31.78054 -0.09540858 -0.045619766 55 5.350 1.75  
56 31.64385 4.447483 29.95732 -1.68652425 -0.799715329 56 5.350 1.25  
57 31.03347 4.567930 31.78054 0.74706978 0.349543681 57 7.020 0.75  
58 31.95688 4.664073 26.39057 -5.56630784 -2.577416119 58 7.020 0.25  
59 32.06224 4.725295 30.44522 -1.61701339 -0.743873584 59 9.185 0.75  
60 31.46920 4.784392 31.78054 0.31133858 0.142337573 60 9.185 0.25

#Creons une fonction pour calculer l'erreur moyenne  
  
EM <- function(effectif, zcore){  
 sum(zcore)/effectif  
}  
  
EM(nrow(cv), zcore = cv$zscore)

[1] -0.001445649