```
## Warning in fun(libname, pkgname): couldn't connect to display ":0"
```

Analysis part of the project

Import dataset

Let's import the cleaned dataset that we created.

```
Mountain_data_cleaned <- read.csv("../data/Mountain_data_cleaned.csv")

Mountain_data_cleaned$Country <- as.factor(Mountain_data_cleaned$Country)
Mountain_data_cleaned$Mountain_range <- as.factor(Mountain_data_cleaned$Mountain_range)
Mountain_data_cleaned$Locality <- as.factor(Mountain_data_cleaned$Locality)
Mountain_data_cleaned$Plot <- as.factor(Mountain_data_cleaned$Plot)
Mountain_data_cleaned$Subplot <- as.factor(Mountain_data_cleaned$Subplot)

Mountain_data_cleaned$Date <- as.Date(Mountain_data_cleaned$Date)
```

Replace the NAs by the mean of the closest observations

```
mean_for_fill <-colMeans( Mountain_data_cleaned %>%
  select(Plot,Glu_P)%>%
  filter(Plot == c("76", "77"))%>%na.omit()%>% select(Glu_P))

Mountain_data_cleaned[is.na(Mountain_data_cleaned)] <- mean_for_fill
rm(mean_for_fill)
```

Splitting the data into Traning set and Test set

```
set.seed(123) ## for replication purpose

## the index of the rows that will be in the training set
index.tr <- sample(1:nrow(Mountain_data_cleaned), replace=FALSE,
                   size=0.75*nrow(Mountain_data_cleaned))

Mountain_data.tr_notsubs <- Mountain_data_cleaned[index.tr,] ## the training set
Mountain_data.te <- Mountain_data_cleaned[-index.tr,] ## the test set
```

Balancing the Training set

As discussed in the EDA part, we should balance our data because we do not have the same amount of information on each mountains. We have more observations on **Sierra de Guadarrama** and half less on **Central Andes**.

```
no.mountain_1 <- min(table(Mountain_data.tr_notsubs$Mountain_range)) ## 79

## the "Central Andes" cases
data.tr.mountain_1 <- filter(Mountain_data.tr_notsubs, Mountain_range=="Central Andes")

## the "Central Pyrenees" cases
data.tr.mountain_2 <- filter(Mountain_data.tr_notsubs, Mountain_range=="Central Pyrenees")

## The "Sierra de Guadarrama" cases
```

```r
data.tr.mountain_3 <- filter(Mountain_data.tr_notsubs, Mountain_range=="Sierra de Guadarrama")
## sub-sample 79 instances from the number of "Central Pyrenees" cases
index.mountain_2 <- sample(size=no.mountain_1,
                           x=1:nrow(data.tr.mountain_2),
                           replace=FALSE)

## sub-sample 79 instances from the number of "Sierra de Guadarrama" cases
index.mountain_3 <- sample(size=no.mountain_1,
                           x=1:nrow(data.tr.mountain_3),
                           replace=FALSE)

## Bind all the "Central Andes" and the sub-sampled "Central Pyrenees"
## and the sub-sampled "Sierra de Guadarrama"
Mountain_data.tr <- data.frame(rbind(data.tr.mountain_1,
                                     data.tr.mountain_2[index.mountain_2,],
                                     data.tr.mountain_3[index.mountain_3,]))

## The cases are now balanced
table(Mountain_data.tr$Mountain_range)
```

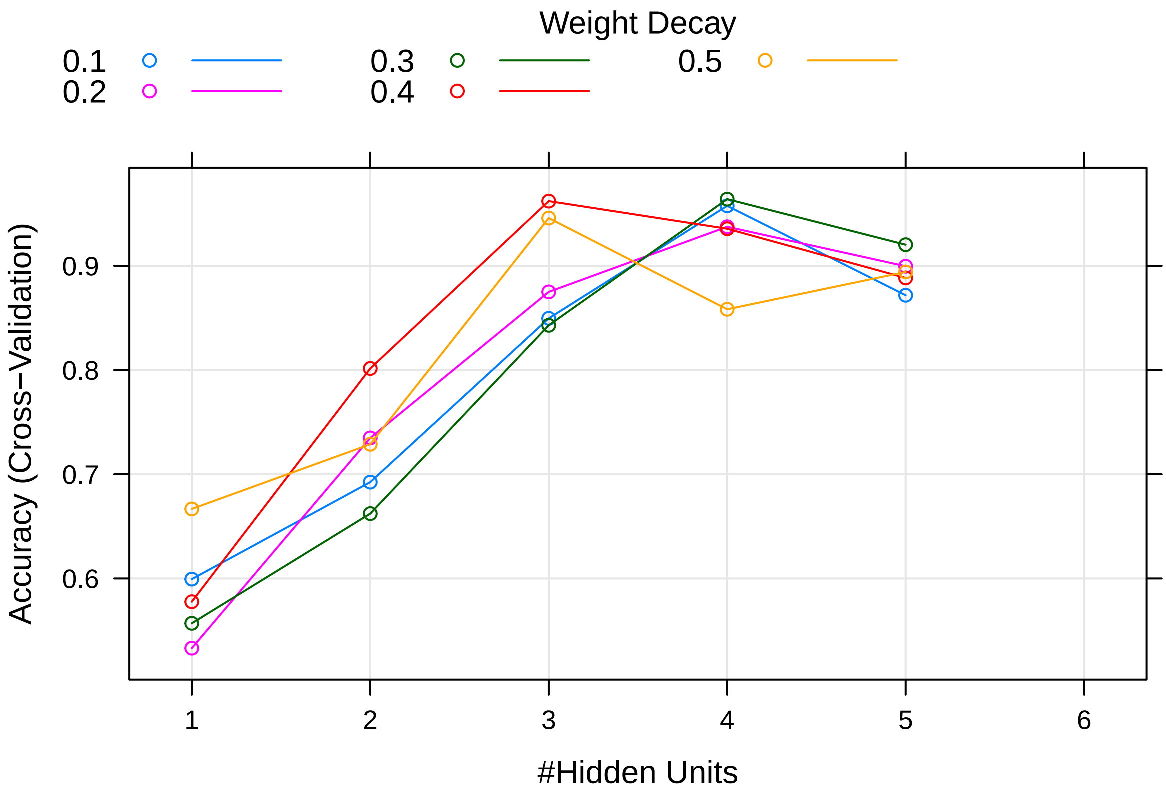
```
##
##        Central Andes     Central Pyrenees Sierra de Guadarrama
##                   79                   79                   79
```

Neural Network Model

Simple hyperparameter tuning, this code takes time to run.

```r
set.seed(1)
fitControl <- trainControl(method = "cv",
                           number = 10)

nnetGrid <-  expand.grid(size = seq(from = 1, to = 6, by = 1),
                        decay = seq(from = 0.1, to = 0.5, by = 0.1))

nnetFit <- train(Mountain_range ~ .,
                 data = Mountain_data.tr,
                 method = "nnet",
                 metric = "Accuracy",
                 tuneGrid = nnetGrid,
                 trControl = fitControl)
```

```r
plot(nnetFit)
```



The best Neural Networks parameters would be to choose 4 hidden layers, with a decay of 0.3.

The manually written Neural Network model

```
set.seed(345)
nn4 <- nnet(Mountain_range ~ ., data=Mountain_data.tr, size=4, decay = 0.3)
```

```
## # weights:  747
## initial  value 368.161785
## iter  10 value 260.374252
## iter  20 value 258.932881
## iter  30 value 247.750026
## iter  40 value 135.640922
## iter  50 value 130.599723
## iter  60 value 127.832084
## iter  70 value 94.604156
## iter  80 value 89.011126
## iter  90 value 79.802935
## iter 100 value 56.287698
## final  value 56.287698
## stopped after 100 iterations
```

```
pred4 <- predict(nn4, type="class")
tab4 <- table(Obs=Mountain_data.tr$Mountain_range, Pred=pred4) # confusion matrix
tab4
```

```
##                         Pred
## Obs                      Central Andes Central Pyrenees Sierra de Guadarrama
##   Central Andes                     71                8                    0
##   Central Pyrenees                   1               78                    0
##   Sierra de Guadarrama               0                0                   79
```

```r
(acc4 <- sum(diag(tab4))/sum(tab4)) # accuracy
```

```
## [1] 0.9620253
```

Here it says that it has almost perfect accuracy (96%).

Visualization of the neural network using neuralnet

Some transformations in order to use the function **neuralnet** in R.

```r
Mountain_data.class <- class.ind(Mountain_data_cleaned$Mountain_range)
colnames(Mountain_data.class) <- c("Central_Andes", "Central_Pyrenees", "Sierra_de_Guadarrama")
Mountain_data.class <- cbind(Mountain_data_cleaned[,10:34], Mountain_data.class)
head(Mountain_data.class)
```

```
##   Radiation   Phos_P    Glu_P    SOC_P     NT_P     PT_P         K_P  pH_P
## 1 0.8088464 4.437515 2.505851 6.315554 4.070239 0.456501 0.008649166 4.925
## 2 0.8088464 4.437515 2.505851 6.315554 4.070239 0.456501 0.008649166 4.925
## 3 0.8088464 4.437515 2.505851 6.315554 4.070239 0.456501 0.008649166 4.925
## 4 0.8088464 4.437515 2.505851 6.315554 4.070239 0.456501 0.008649166 4.925
## 5 0.8088464 4.437515 2.505851 6.315554 4.070239 0.456501 0.008649166 4.925
## 6 0.7879971 5.171000 3.234583 5.092630 4.546233 3.923043 0.013395058 5.232
##   Cond_P   Phos_B    Glu_B    SOC_B     NT_B      PT_B         K_B  pH_B Cond_B
## 1 31.284 2.888396 1.691185 3.762122 3.297689 0.4450694 0.002516518 5.402 22.198
## 2 31.284 2.888396 1.691185 3.762122 3.297689 0.4450694 0.002516518 5.402 22.198
## 3 31.284 2.888396 1.691185 3.762122 3.297689 0.4450694 0.002516518 5.402 22.198
## 4 31.284 2.888396 1.691185 3.762122 3.297689 0.4450694 0.002516518 5.402 22.198
## 5 31.284 2.888396 1.691185 3.762122 3.297689 0.4450694 0.002516518 5.402 22.198
## 6 48.020 3.102327 1.955476 2.993161 3.168254 3.1682544 0.006759548 5.760 28.700
##     Phos_T    Glu_T    SOC_T     NT_T      PT_T         K_T     pH_T   Cond_T
## 1 3.593446 2.203418 5.736881 3.958871 0.4871885 0.005985587 5.214310 27.60922
## 2 3.593446 2.203418 5.736881 3.958871 0.4871885 0.005985587 5.214310 27.60922
## 3 3.593446 2.203418 5.736881 3.958871 0.4871885 0.005985587 5.214310 27.60922
## 4 3.593446 2.203418 5.736881 3.958871 0.4871885 0.005985587 5.214310 27.60922
## 5 3.593446 2.203418 5.736881 3.958871 0.4871885 0.005985587 5.214310 27.60922
## 6 5.607705 3.481617 5.288592 4.912220 4.8853870 0.015316761 5.505655 48.93783
##   Central_Andes Central_Pyrenees Sierra_de_Guadarrama
## 1             0                0                    1
## 2             0                0                    1
## 3             0                0                    1
## 4             0                0                    1
## 5             0                0                    1
## 6             0                0                    1
```

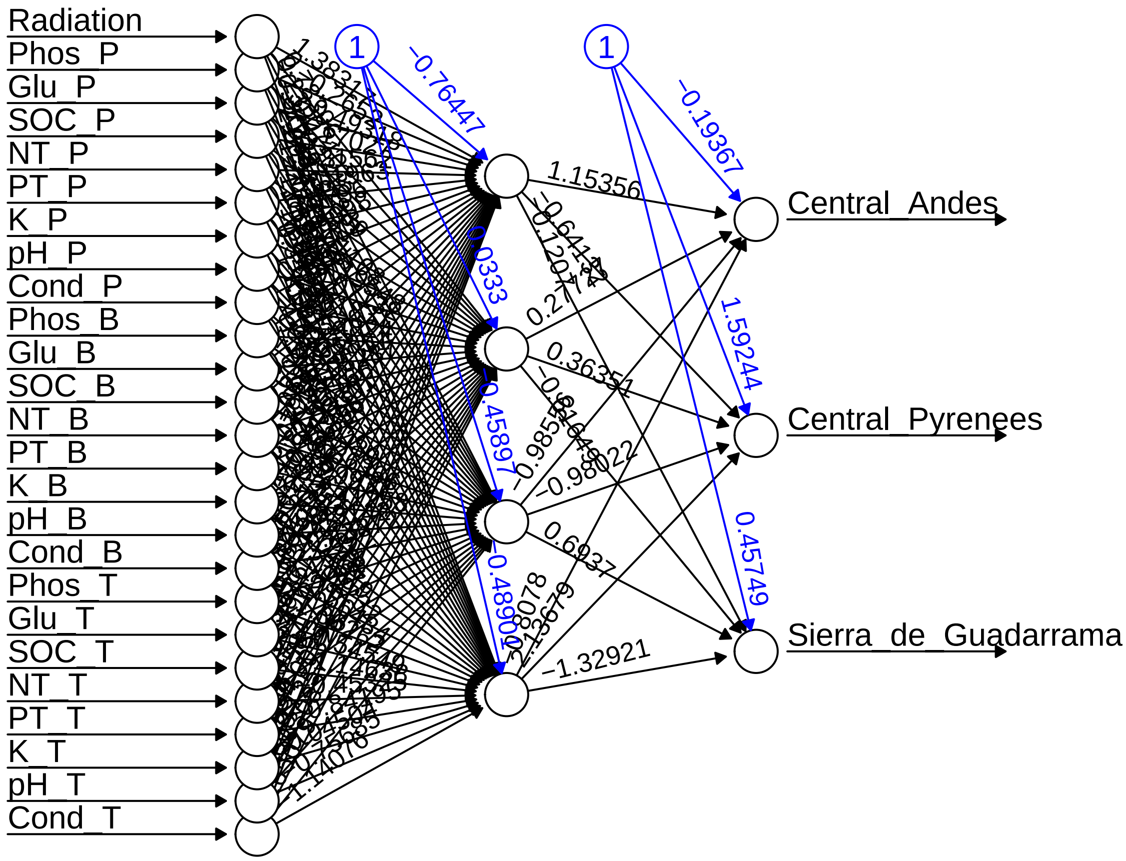
```r
Mountain_data.class.tr <- Mountain_data.class[index.tr,] ## the training set
Mountain_data.class.te <- Mountain_data.class[-index.tr,] ## the test set
```

```r
f <- as.formula(paste(
  "Central_Andes+Central_Pyrenees+Sierra_de_Guadarrama~",
  paste(names(Mountain_data.class.tr[ ,1:25]),
            collapse = " + ")
  ))
f
```

```
## Central_Andes + Central_Pyrenees + Sierra_de_Guadarrama ~ Radiation +
##     Phos_P + Glu_P + SOC_P + NT_P + PT_P + K_P + pH_P + Cond_P +
```

```
##     Phos_B + Glu_B + SOC_B + NT_B + PT_B + K_B + pH_B + Cond_B +
##     Phos_T + Glu_T + SOC_T + NT_T + PT_T + K_T + pH_T + Cond_T
```

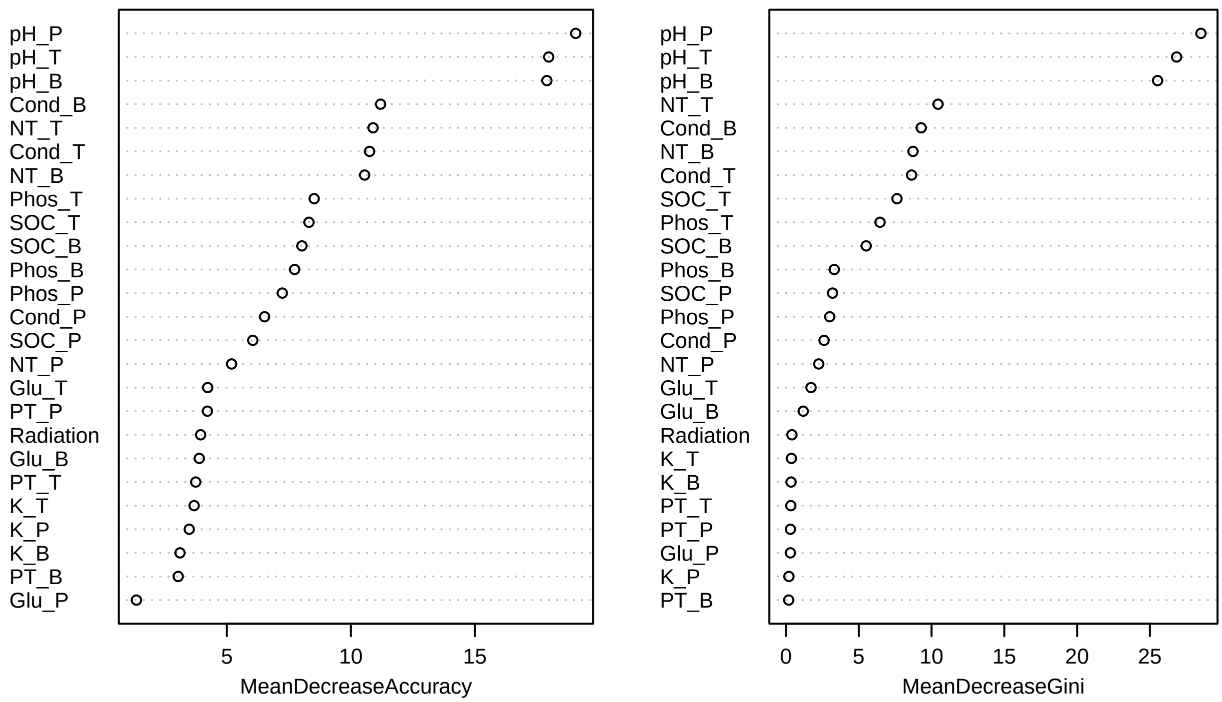
```r
neuralnet4 <- neuralnet(f, data=Mountain_data.class.tr, hidden=4)

plot(neuralnet4, rep="best")
```



Random Forest

```r
train_set_for_RF <- Mountain_data.tr %>%
  select(!c(Plot, Subplot, Date, Day, Month, Year, Locality, Country))
test_set_for_RF <- Mountain_data.te%>%
  select(!c(Plot, Subplot, Date, Day, Month, Year, Locality, Country))

rf <- randomForest( Mountain_range ~ .,data=train_set_for_RF, importance = TRUE)

pred = predict(rf, newdata=test_set_for_RF[-1])

cm = table(test_set_for_RF[,1], pred)

varImpPlot(rf, cex = 0.65)
```

rf




```
importance(rf)
```

```
##           Central Andes Central Pyrenees Sierra de Guadarrama
## Radiation     3.3816116         1.615432            2.1120086
## Phos_P        3.6314751         6.072511            5.9533819
## Glu_P        -0.7930848         1.963743            1.0010015
## SOC_P         4.6672182         2.760467            4.5718335
## NT_P          4.2676355         3.880822            2.7206067
## PT_P          3.4315425         3.530342            1.4027521
## K_P           0.4433611         1.117907            3.4046159
## pH_P         15.1857458        13.634172           18.3183740
## Cond_P        4.8404809         4.962507            2.7583477
## Phos_B        2.8802863         6.932371            5.2339442
## Glu_B         2.0836485         1.639256            3.1723002
## SOC_B         6.6881538         1.520590            7.1270470
## NT_B          9.7712488         6.166361            7.6098218
## PT_B          3.0488432         2.006104            0.1680706
## K_B           2.1467509         2.661264            0.9174051
## pH_B         15.3183832        15.664604           13.2977142
## Cond_B       10.4435675         9.715157            3.2843668
## Phos_T        4.1036155         8.132650            6.9797023
## Glu_T         2.4386839         2.989234            3.4822536
## SOC_T         7.4166140         2.928867            6.9725530
## NT_T         10.3756377         6.510894            8.8162642
## PT_T          3.3954928         2.717599            1.6541175
## K_T           2.3773209         2.563934            2.3582030
## pH_T         15.5842748        14.782024           14.0416379
## Cond_T        9.8729245         9.724750            4.4404246
```

```
##           MeanDecreaseAccuracy MeanDecreaseGini
## Radiation             3.942270        0.4090086
## Phos_P                7.235200        3.0019595
## Glu_P                 1.353404        0.3033256
## SOC_P                 6.043011        3.1984352
## NT_P                  5.191736        2.2508748
## PT_P                  4.216026        0.3064380
## K_P                   3.482336        0.2007332
## pH_P                 19.062961       28.5056588
## Cond_P                6.519301        2.6190379
## Phos_B                7.730592        3.3179881
## Glu_B                 3.887852        1.1867892
## SOC_B                 8.024262        5.5112621
## NT_B                 10.554633        8.7256334
## PT_B                  3.036890        0.1862170
## K_B                   3.109997        0.3495436
## pH_B                 17.900361       25.5272103
## Cond_B               11.197219        9.2885982
## Phos_T                8.519080        6.4598247
## Glu_T                 4.225570        1.7203314
## SOC_T                 8.308735        7.6248962
## NT_T                 10.892500       10.4437612
## PT_T                  3.744260        0.3314549
## K_T                   3.679918        0.3746931
## pH_T                 17.974499       26.8345935
## Cond_T               10.752629        8.6281617
```

```
Acc <- sum(diag(cm))/sum(cm)
```

```
cm
```

```
                  pred
                   Central Andes Central Pyrenees Sierra de Guadarrama
```

Central Andes 20 1 0 Central Pyrenees 0 29 0 Sierra de Guadarrama 0 0 58

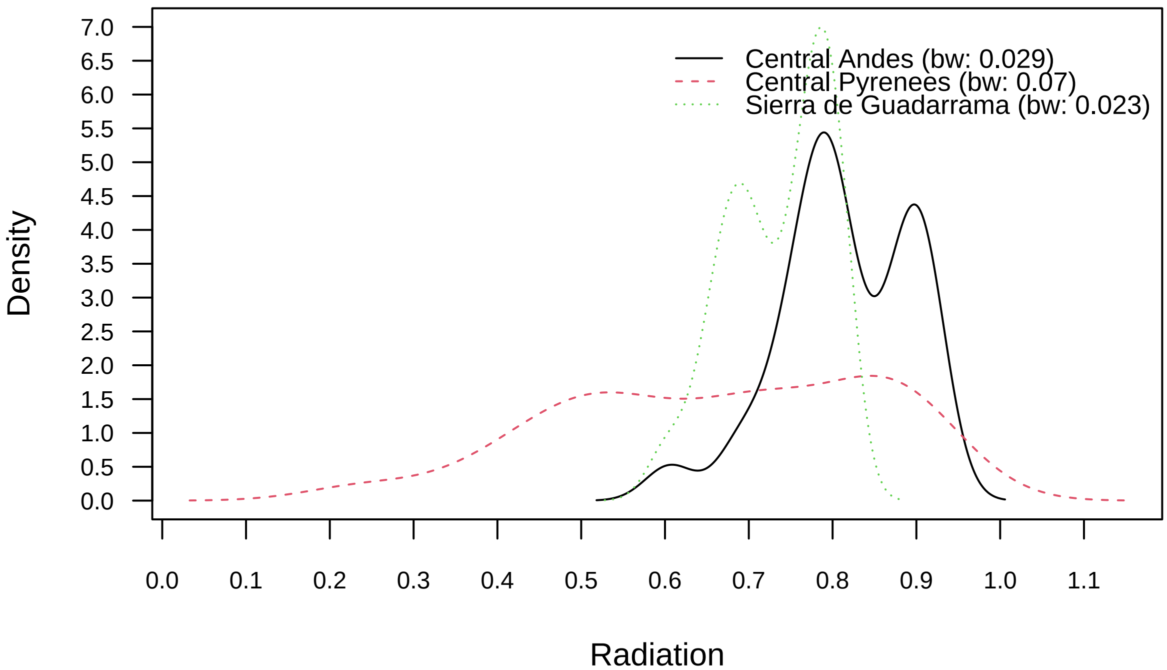
```
Acc
```

[1] 0.9907407

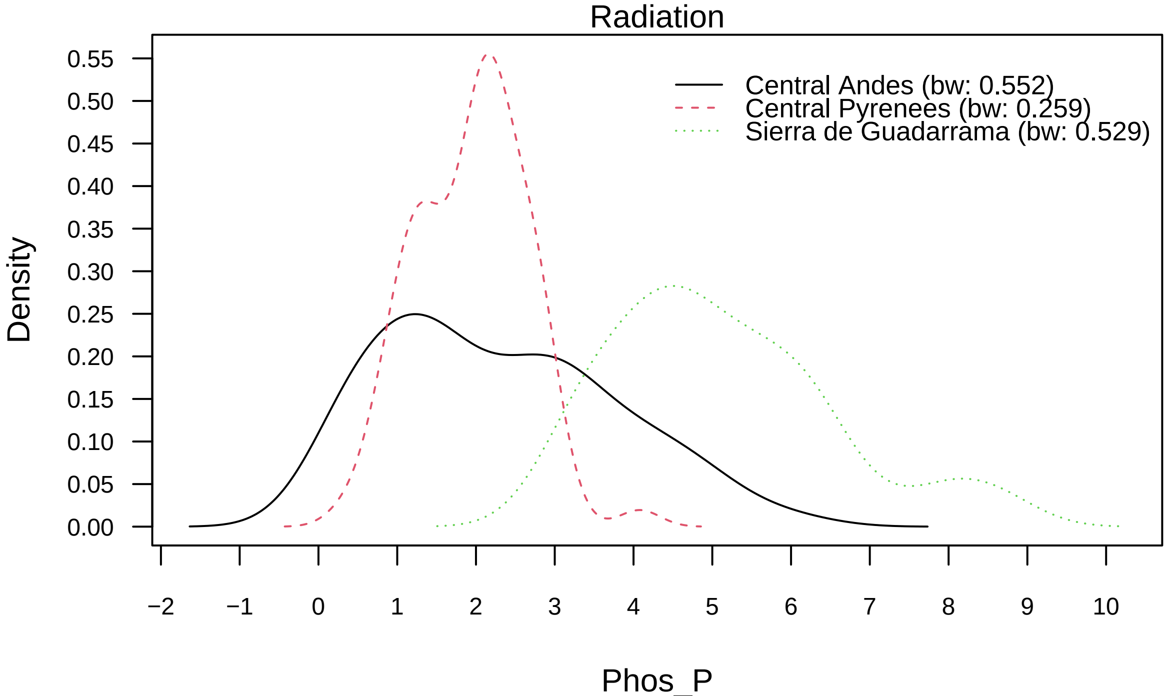
Naive Bayes

Conditional density Plots with 'usekernel=TRUE'

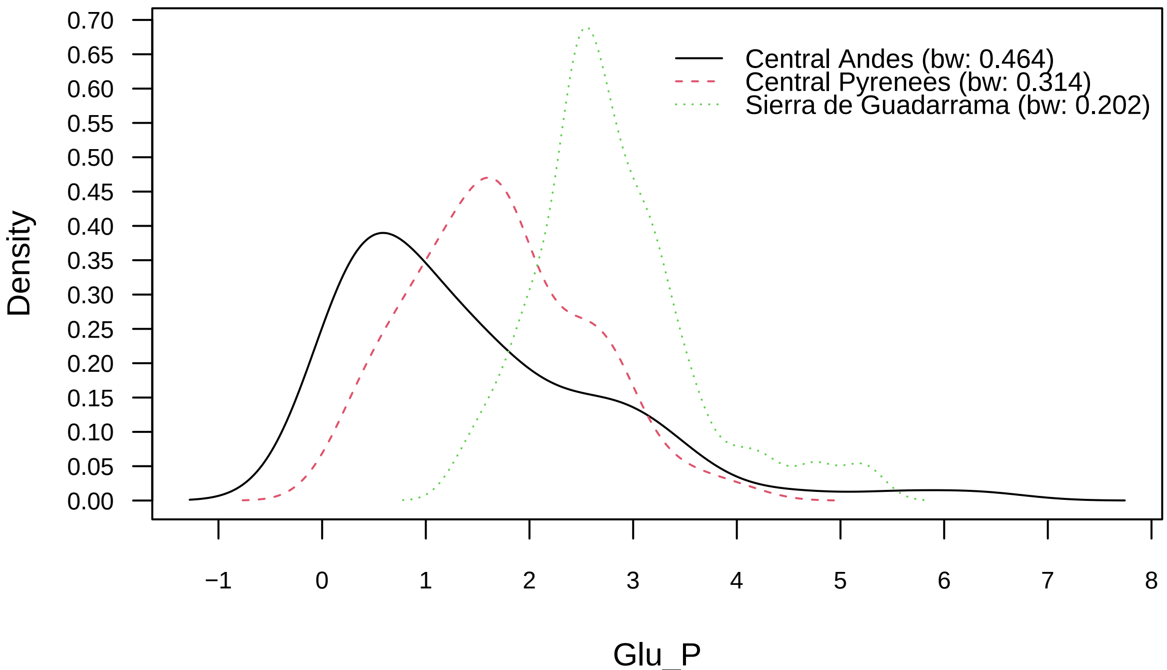
```
mountain.nb <- naive_bayes(Mountain_range ~ .,
                       data = Mountain_data.tr%>%
  select(!c(Plot, Subplot, Date, Day, Month, Year, Locality, Country)), usekernel=TRUE, laplace=1)
#par(mfrow=c(2,2))
plot(mountain.nb, arg.num = list(col = 1:3,
                             legend.position = "topright",
                             legend.cex = 0.8),
     prob="conditional")
```

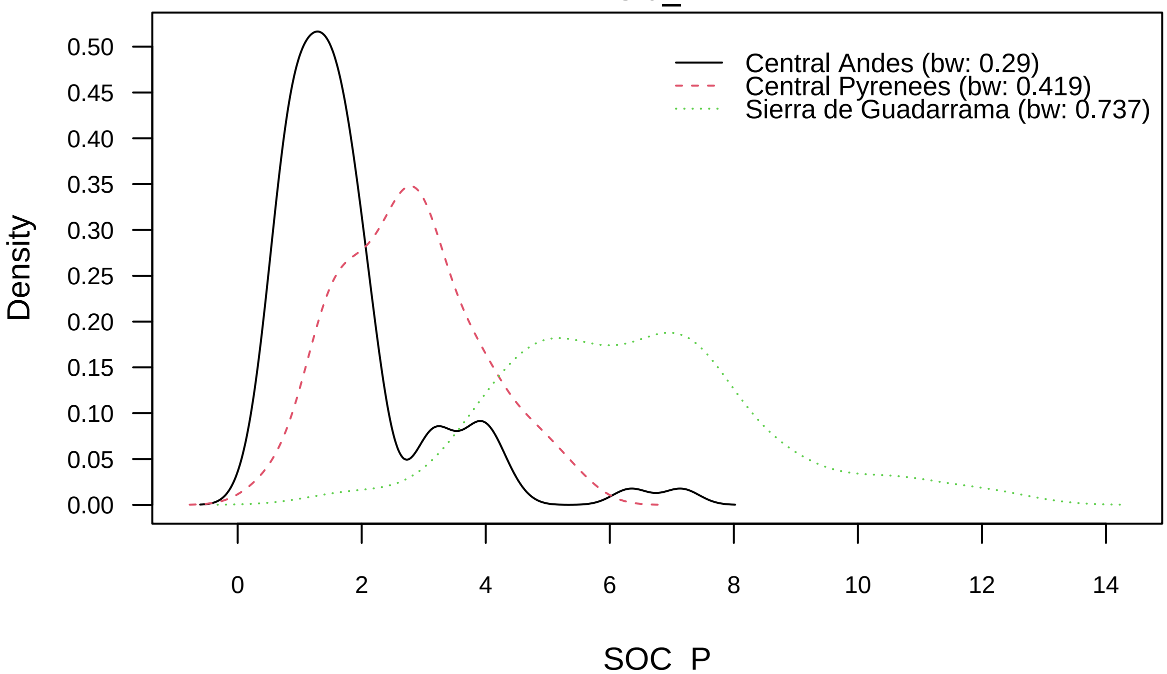


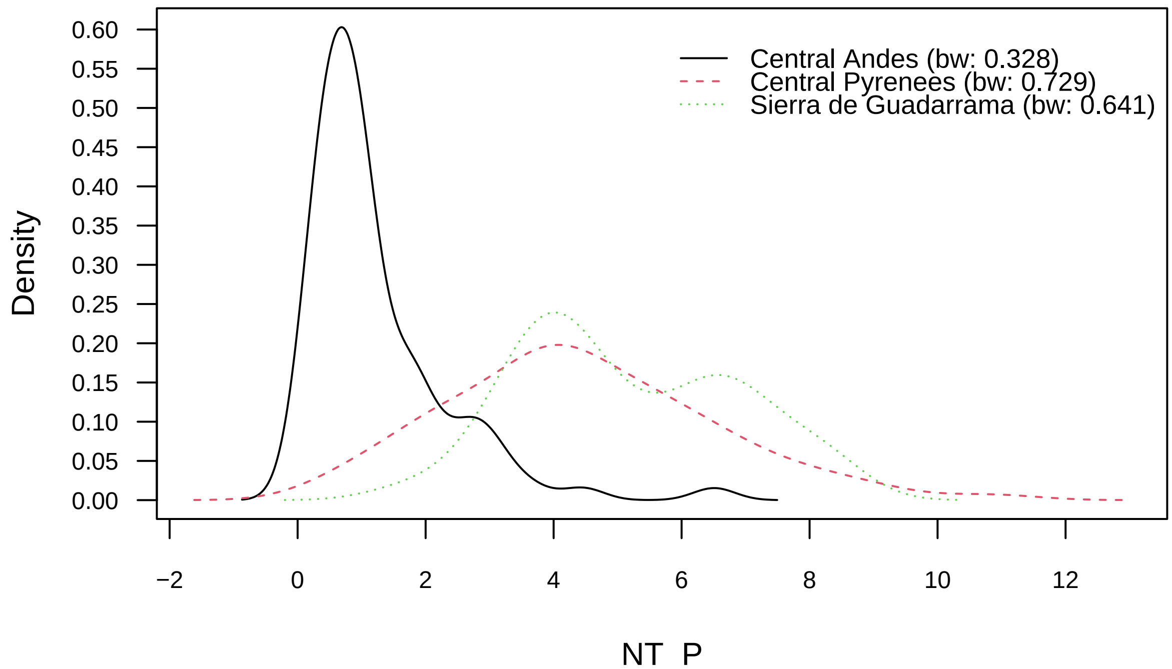
Central Andes (bw: 0.029)
Central Pyrenees (bw: 0.07)
Sierra de Guadarrama (bw: 0.023)
Density
Radiation

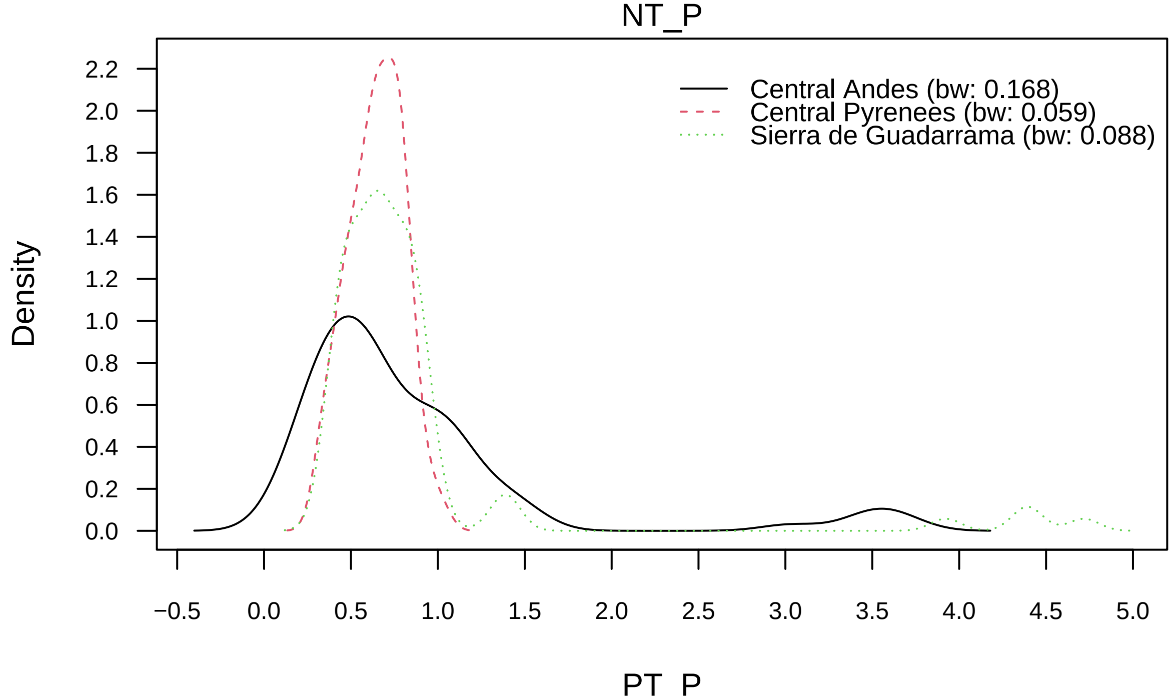


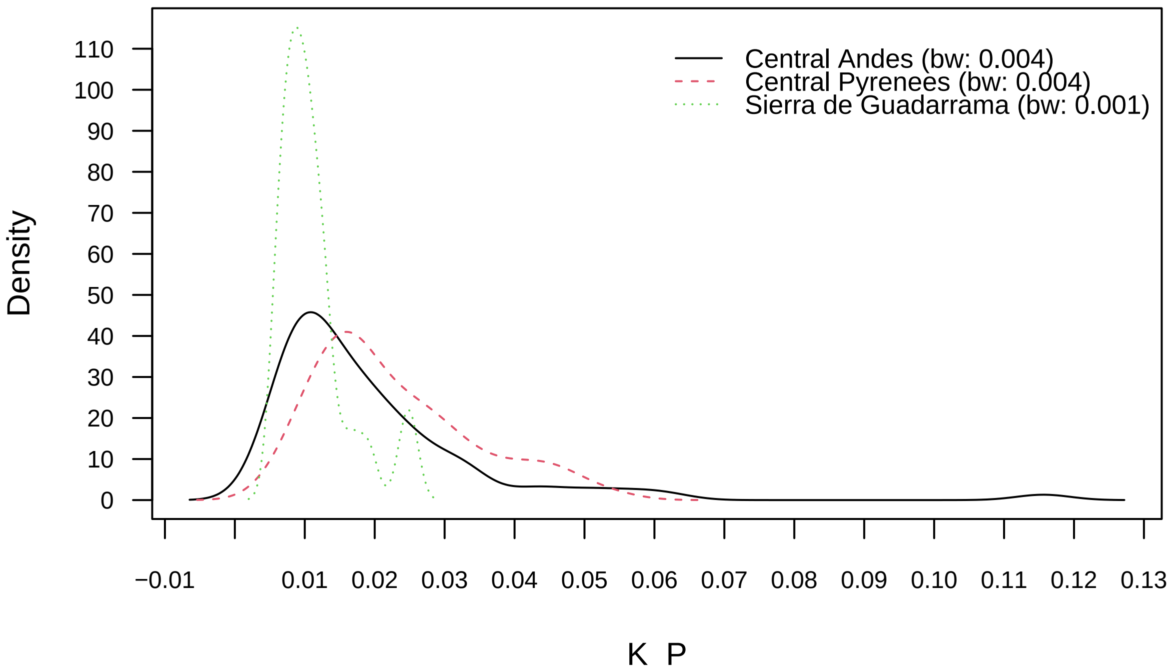
Central Andes (bw: 0.552)
Central Pyrenees (bw: 0.259)
Sierra de Guadarrama (bw: 0.529)
Density
Phos_P

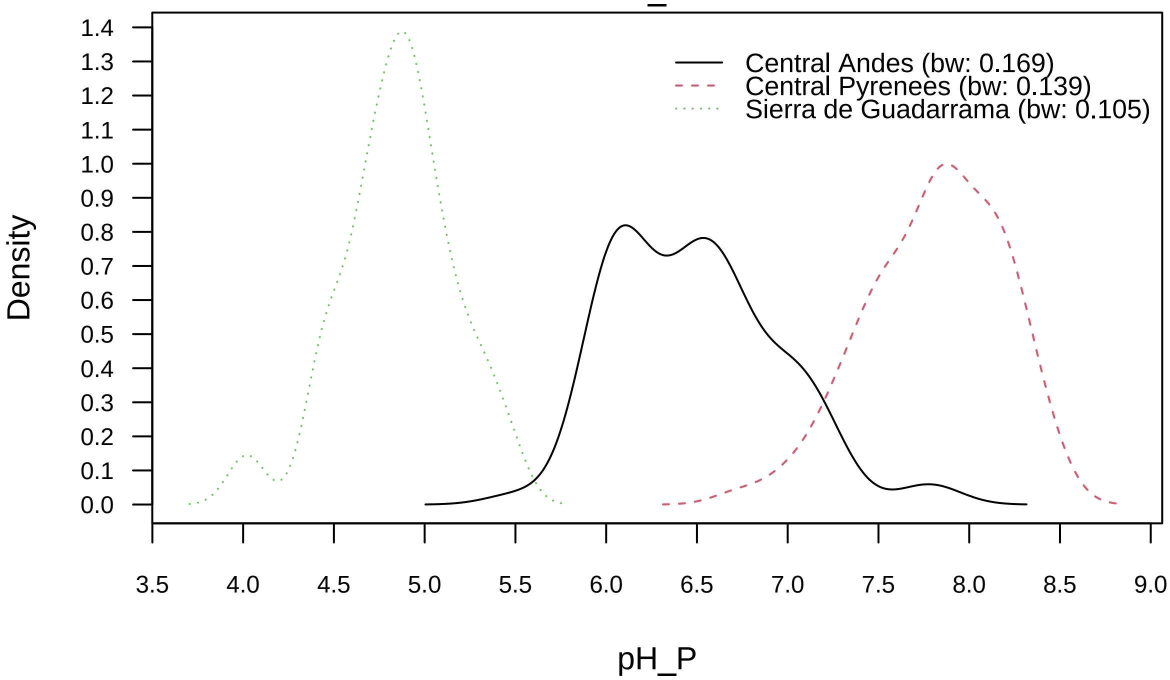

Density
Central Andes (bw: 0.464)
Central Pyrenees (bw: 0.314)
Sierra de Guadarrama (bw: 0.202)
Glu_P

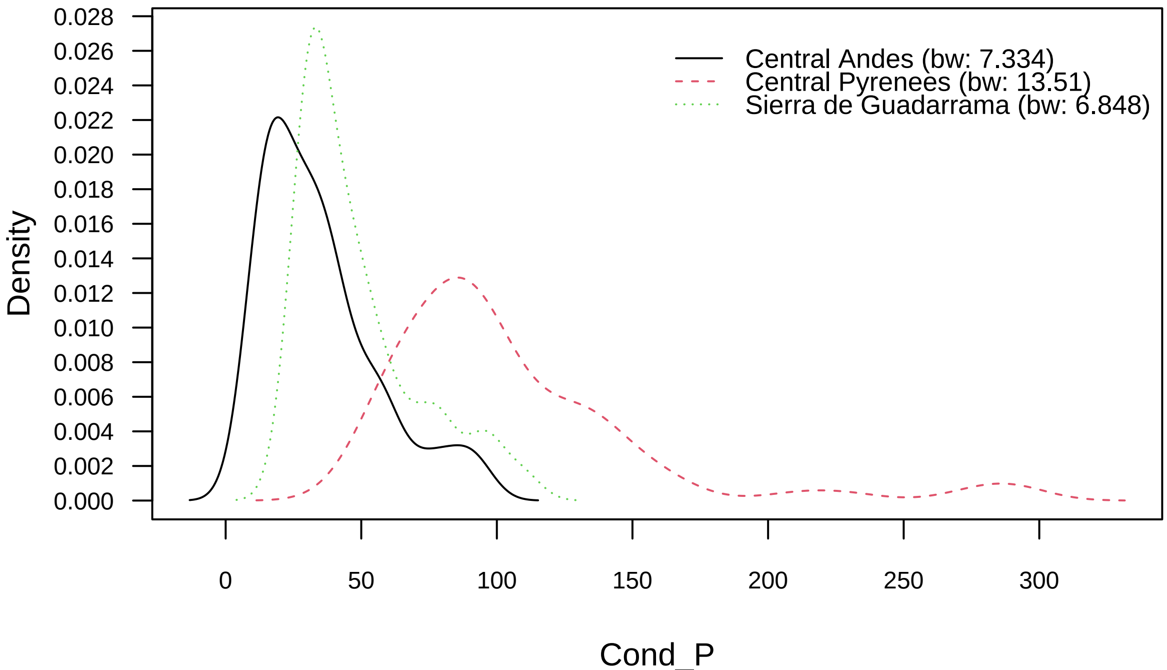

Density
Central Andes (bw: 0.29)
Central Pyrenees (bw: 0.419)
Sierra de Guadarrama (bw: 0.737)
SOC_P

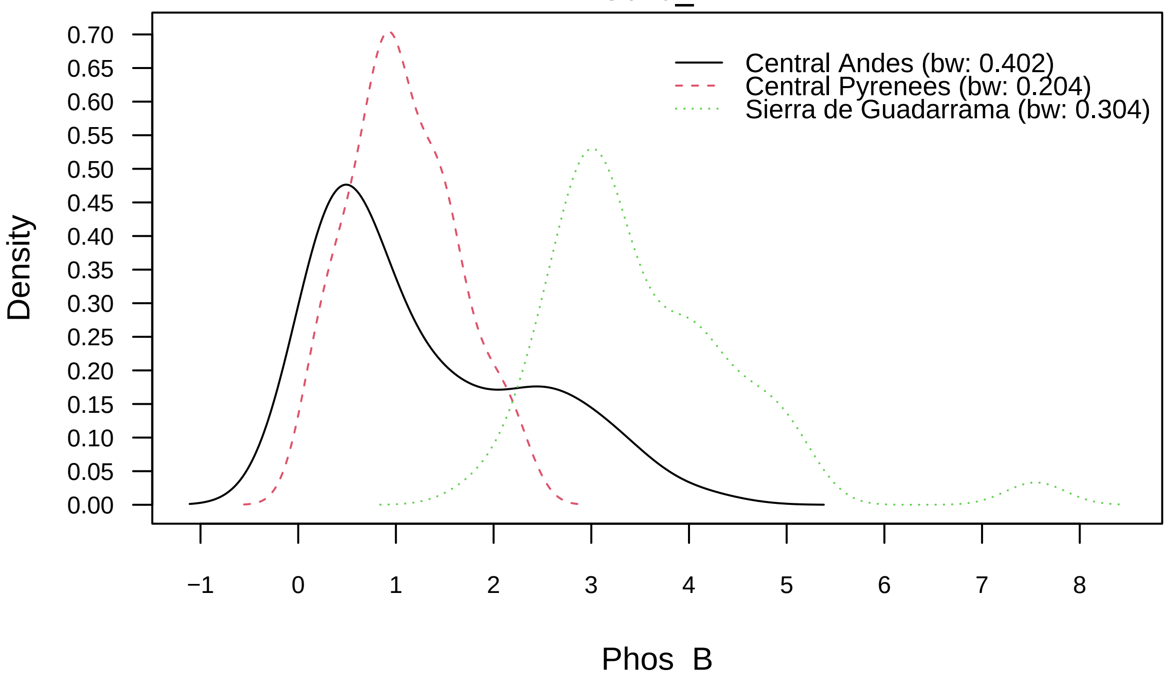

Central Andes (bw: 0.328)
Central Pyrenees (bw: 0.729)
Sierra de Guadarrama (bw: 0.641)
Density
NT_P

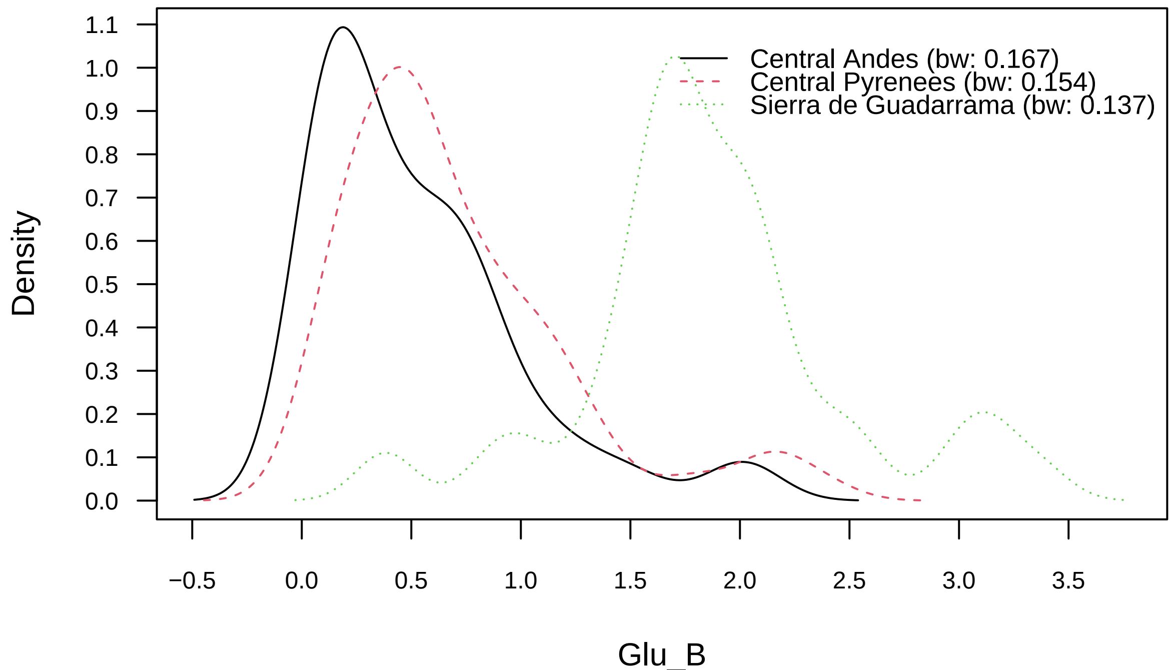

Central Andes (bw: 0.168)
Central Pyrenees (bw: 0.059)
Sierra de Guadarrama (bw: 0.088)
Density
PT_P

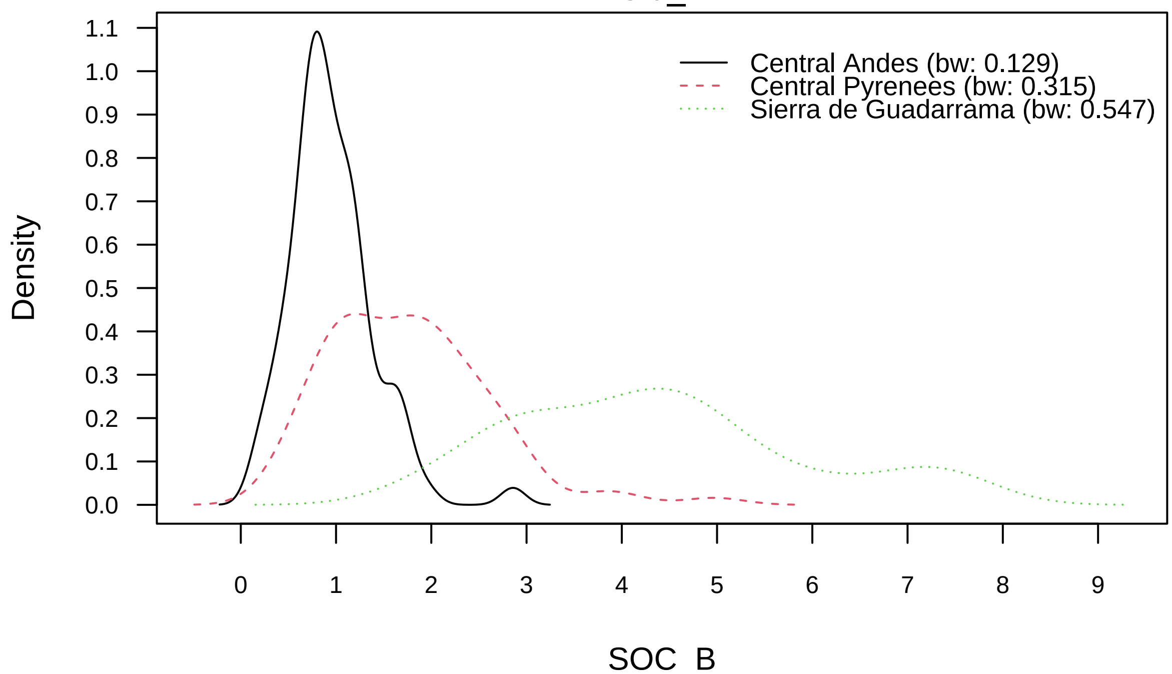

Central Andes (bw: 0.004)
Central Pyrenees (bw: 0.004)
Sierra de Guadarrama (bw: 0.001)
Density
K_P

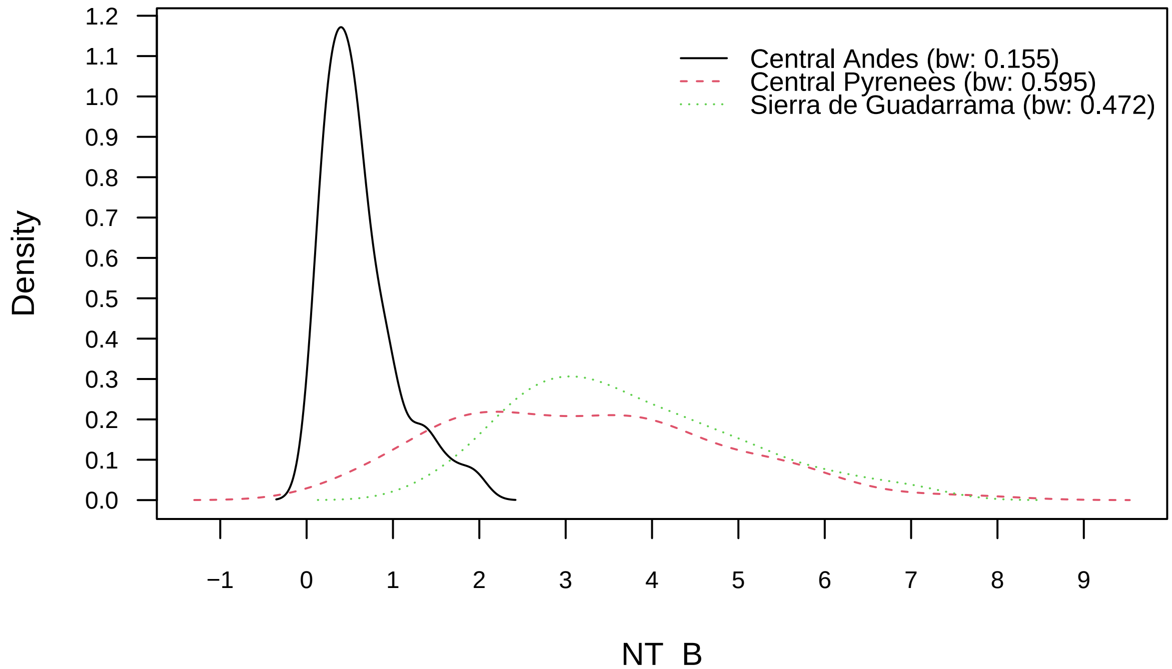

Central Andes (bw: 0.169)
Central Pyrenees (bw: 0.139)
Sierra de Guadarrama (bw: 0.105)
Density
pH_P

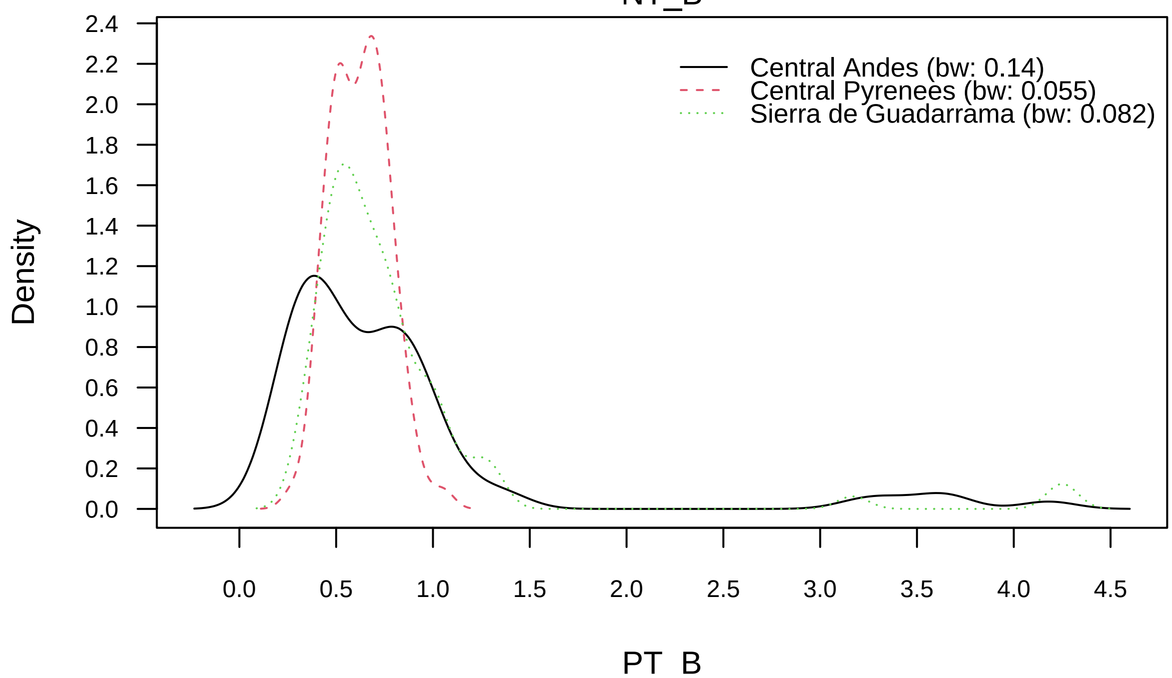

Density
0.028
0.026
0.024
0.022
0.020
0.018
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
Central Andes (bw: 7.334)
Central Pyrenees (bw: 13.51)
Sierra de Guadarrama (bw: 6.848)
0
50
100
150
200
250
300
Cond_P

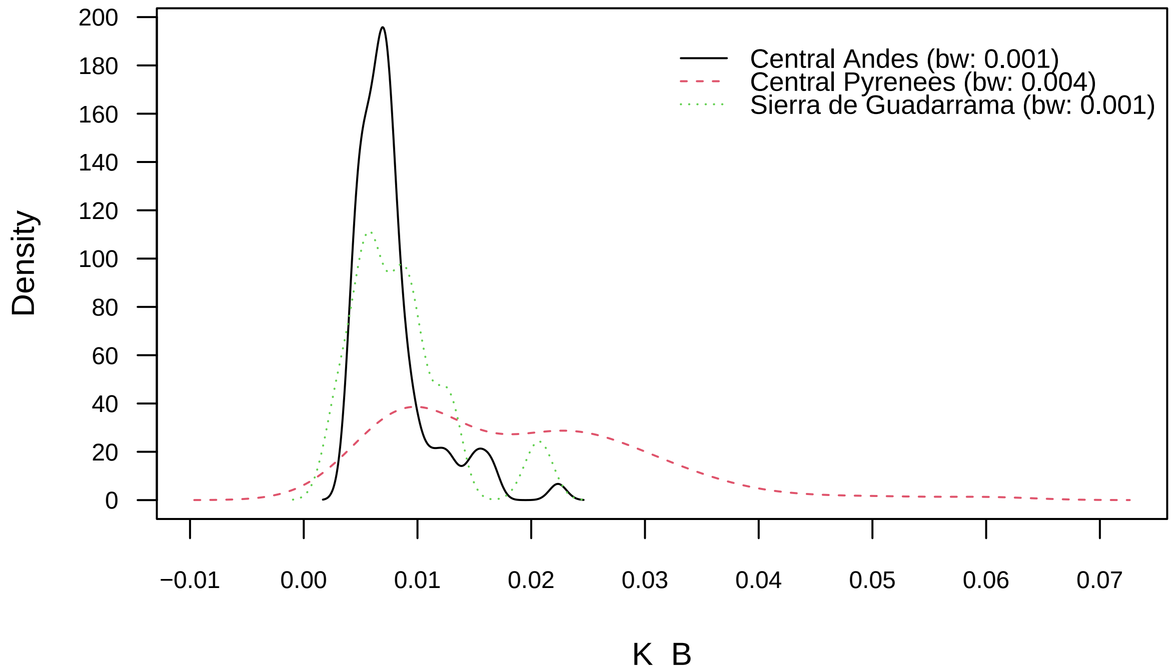

Density
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Central Andes (bw: 0.402)
Central Pyrenees (bw: 0.204)
Sierra de Guadarrama (bw: 0.304)
−1
0
1
2
3
4
5
6
7
8
Phos_B

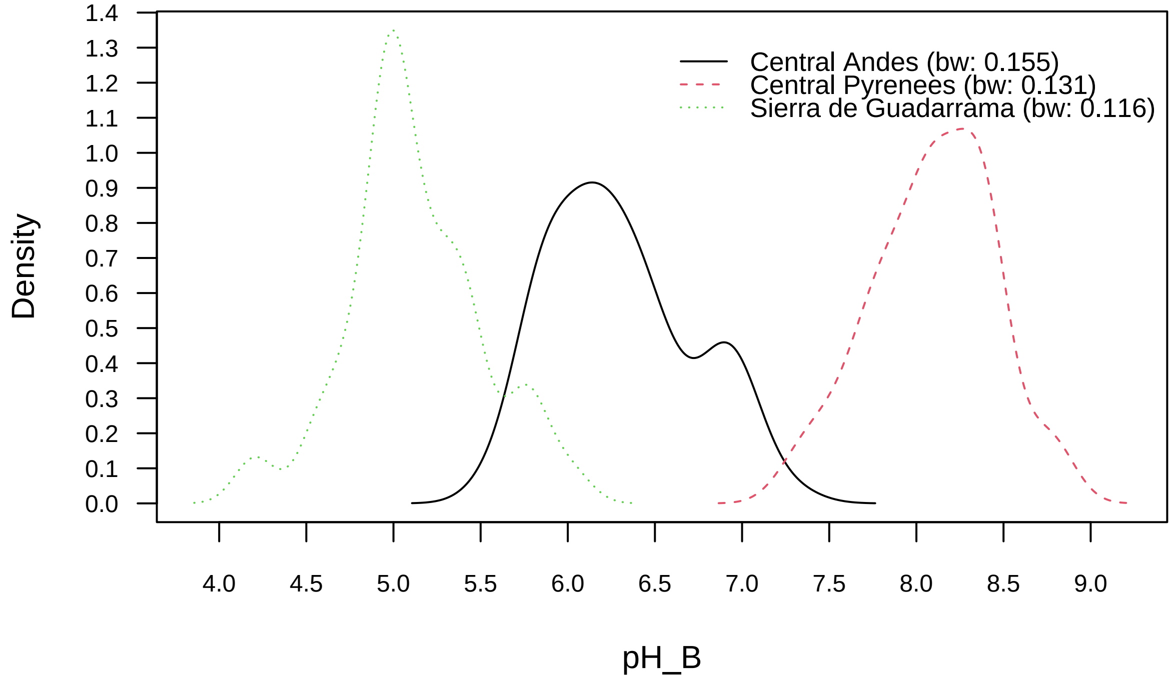

Central Andes (bw: 0.167)
Central Pyrenees (bw: 0.154)
Sierra de Guadarrama (bw: 0.137)
Density
Glu_B

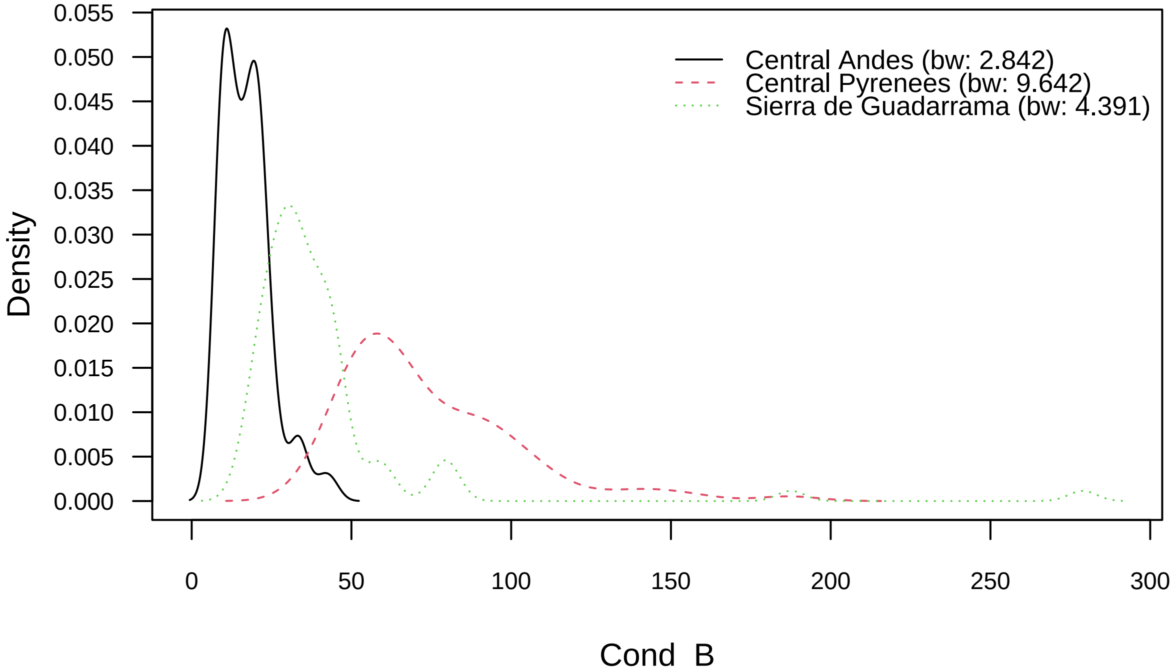

Central Andes (bw: 0.129)
Central Pyrenees (bw: 0.315)
Sierra de Guadarrama (bw: 0.547)
Density
SOC_B

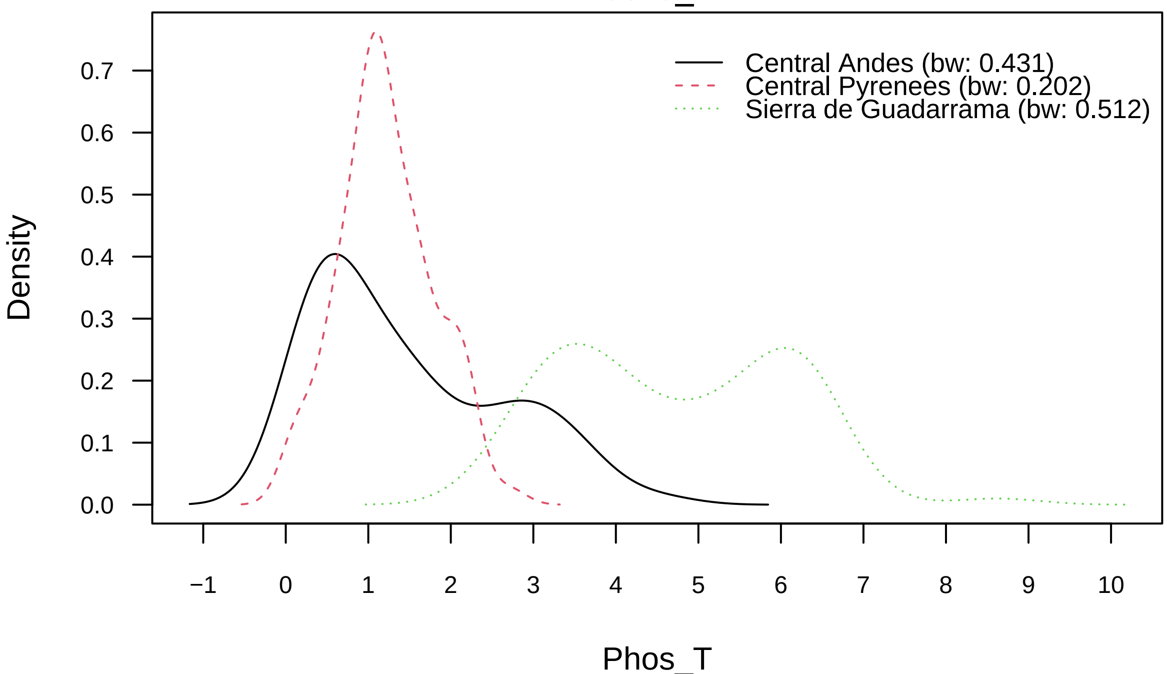

Central Andes (bw: 0.155)
Central Pyrenees (bw: 0.595)
Sierra de Guadarrama (bw: 0.472)
Density
NT_B

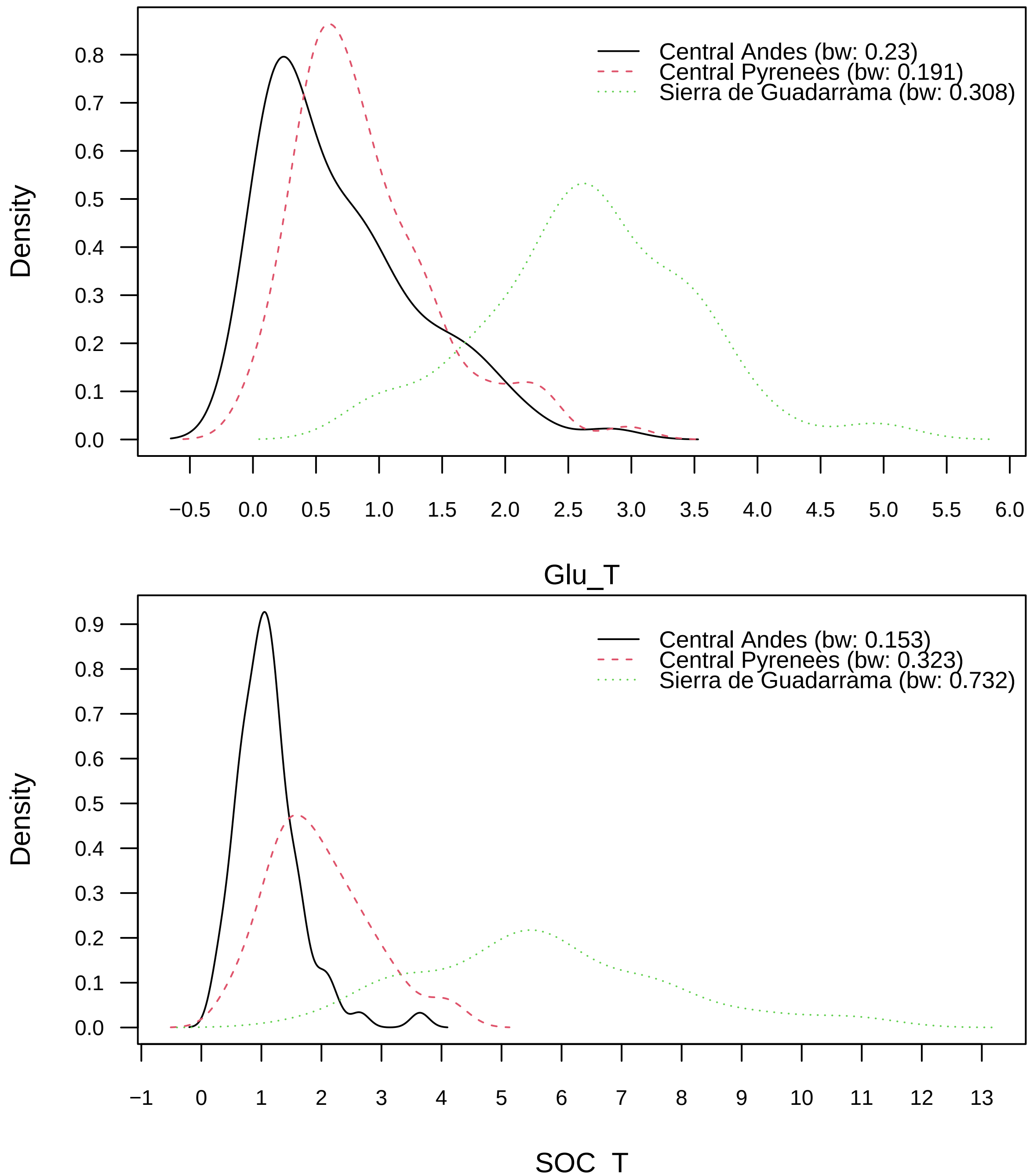

Central Andes (bw: 0.14)
Central Pyrenees (bw: 0.055)
Sierra de Guadarrama (bw: 0.082)
Density
PT_B

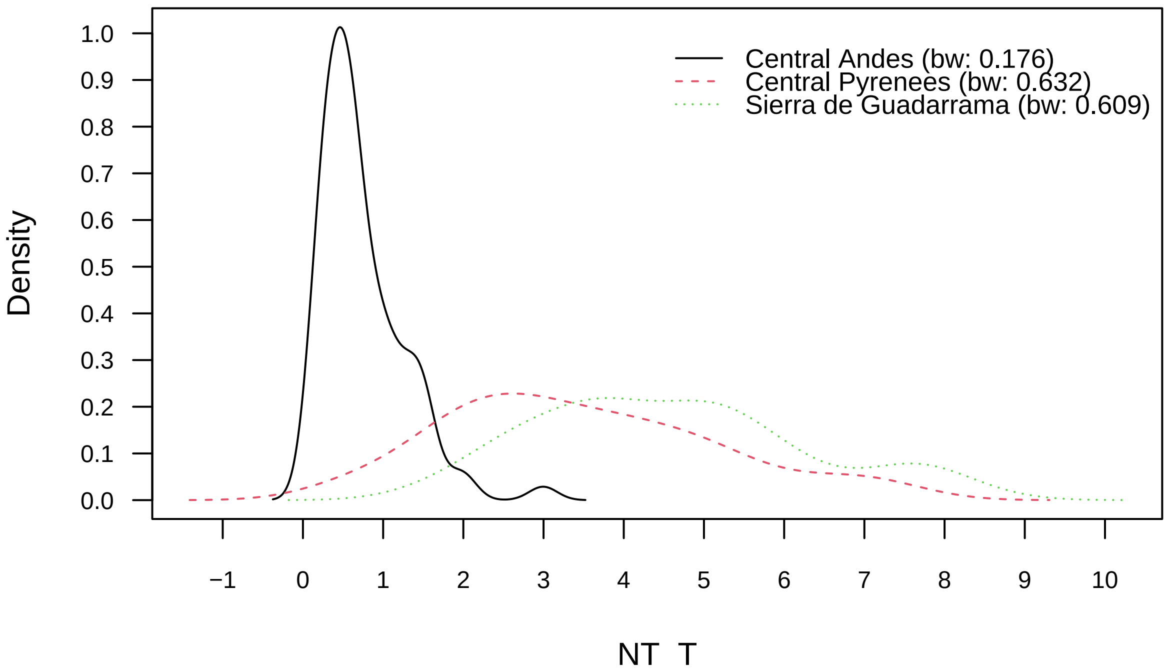

Density
K_B
Central Andes (bw: 0.001)
Central Pyrenees (bw: 0.004)
Sierra de Guadarrama (bw: 0.001)


Density
pH_B
Central Andes (bw: 0.155)
Central Pyrenees (bw: 0.131)
Sierra de Guadarrama (bw: 0.116)

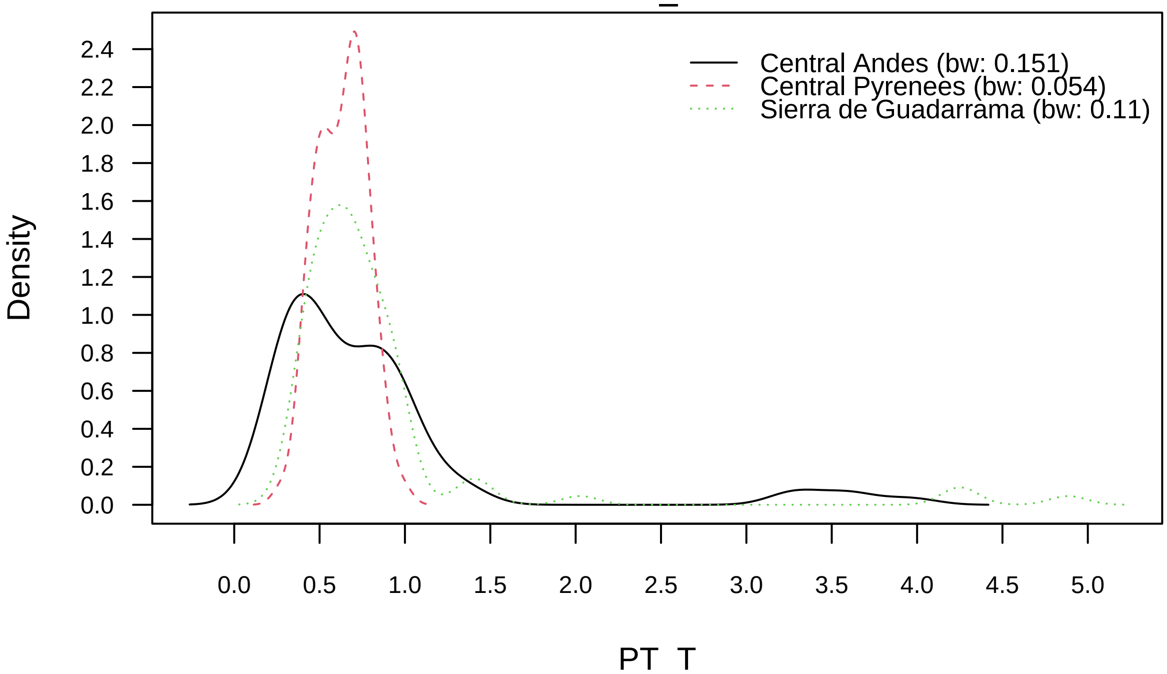

Central Andes (bw: 2.842)
Central Pyrenees (bw: 9.642)
Sierra de Guadarrama (bw: 4.391)
Density
0.055
0.050
0.045
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
50
100
150
200
250
300
Cond_B


Central Andes (bw: 0.431)
Central Pyrenees (bw: 0.202)
Sierra de Guadarrama (bw: 0.512)
Density
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−1
0
1
2
3
4
5
6
7
8
9
10
Phos_T

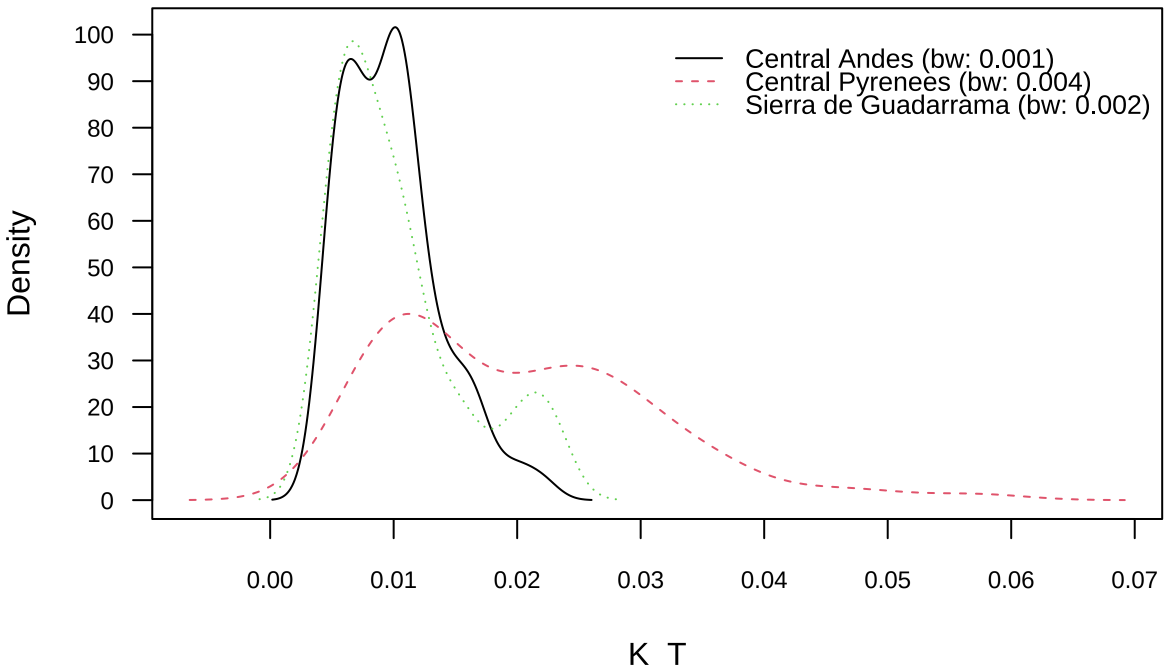

Central Andes (bw: 0.23)
Central Pyrenees (bw: 0.191)
Sierra de Guadarrama (bw: 0.308)
Density
Glu_T
Central Andes (bw: 0.153)
Central Pyrenees (bw: 0.323)
Sierra de Guadarrama (bw: 0.732)
Density
SOC_T

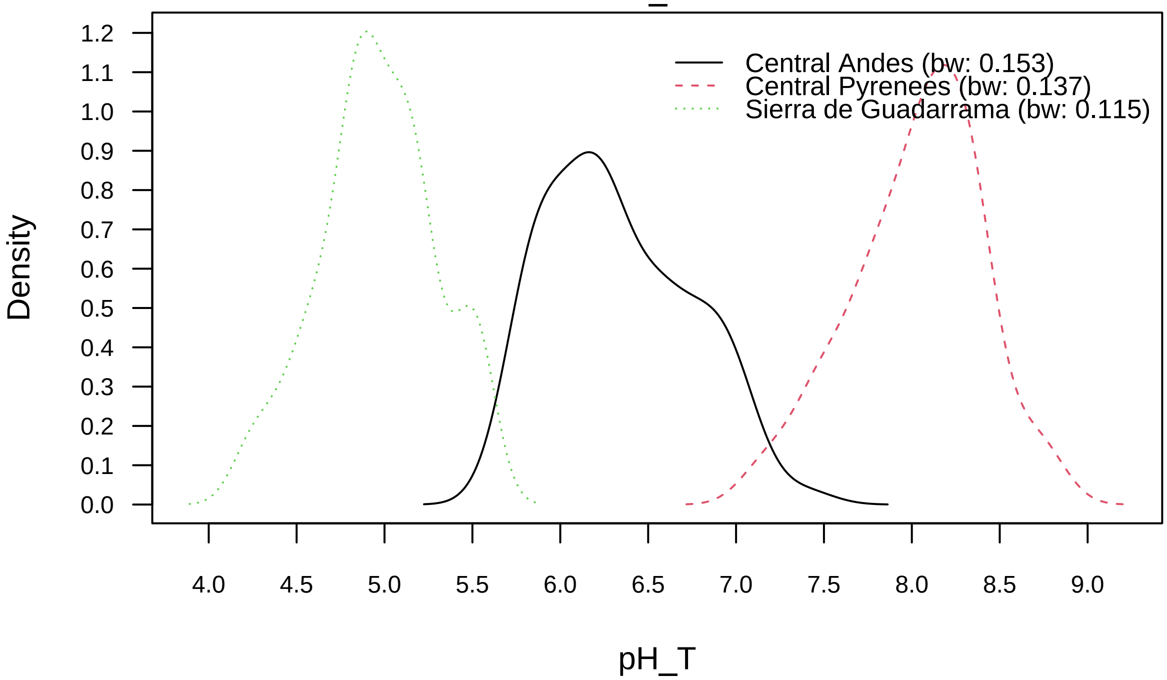


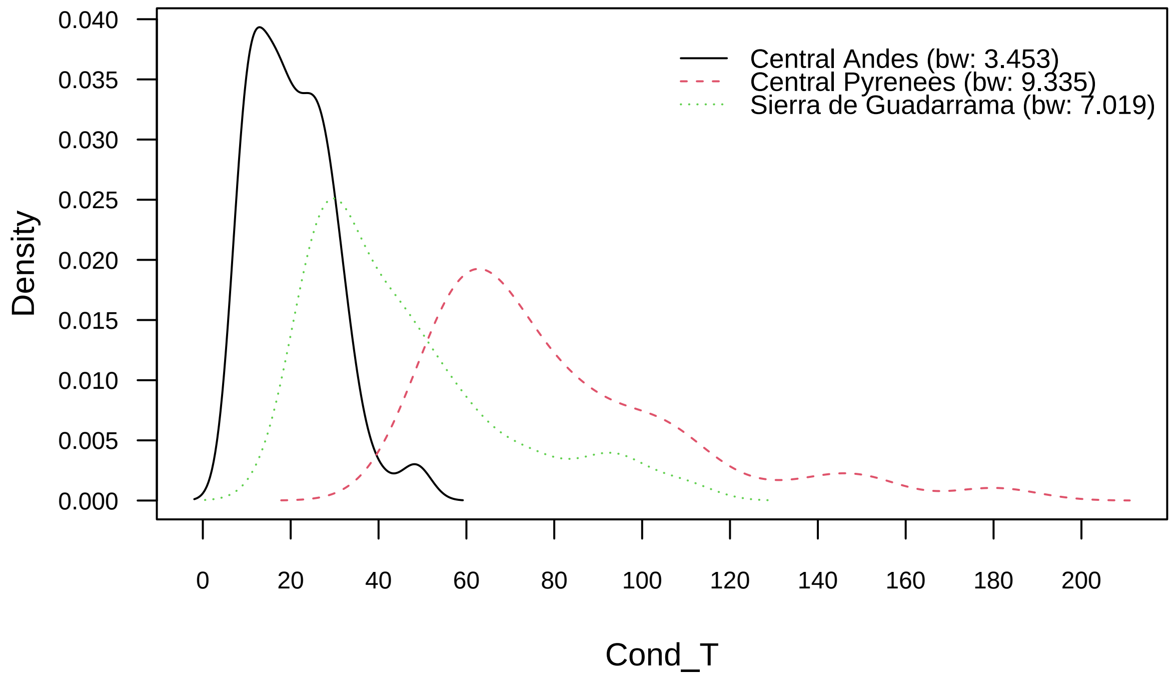
Central Andes (bw: 0.176)
Central Pyrenees (bw: 0.632)
Sierra de Guadarrama (bw: 0.609)
Density
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−1
0
1
2
3
4
5
6
7
8
9
10
NT_T



Central Andes (bw: 0.151)
Central Pyrenees (bw: 0.054)
Sierra de Guadarrama (bw: 0.11)
Density
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
PT_T


Central Andes (bw: 0.001)
Central Pyrenees (bw: 0.004)
Sierra de Guadarrama (bw: 0.002)
Density
100
90
80
70
60
50
40
30
20
10
0
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
K_T


Central Andes (bw: 0.153)
Central Pyrenees (bw: 0.137)
Sierra de Guadarrama (bw: 0.115)
Density
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0
pH_T




```r
par(mfrow=c(1,1))
mountain.nb
```

```
##
## ================================== Naive Bayes ==================================
##
##  Call:
## naive_bayes.formula(formula = Mountain_range ~ ., data = Mountain_data.tr %>%
##     select(!c(Plot, Subplot, Date, Day, Month, Year, Locality,
##         Country)), laplace = 1, usekernel = TRUE)
##
## ---------------------------------------------------------------------------------
##
## Laplace smoothing: 1
##
## ---------------------------------------------------------------------------------
##
##  A priori probabilities:
##
##          Central Andes     Central Pyrenees Sierra de Guadarrama
##              0.3333333            0.3333333            0.3333333
##
## ---------------------------------------------------------------------------------
##
##  Tables:
##
## ---------------------------------------------------------------------------------
##  ::: Radiation::Central Andes (KDE)
## ---------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
```

```
## Data: x (79 obs.);   Bandwidth 'bw' = 0.02926
##
##        x                y
##  Min.   :0.5183   Min.   :0.00581
##  1st Qu.:0.6402   1st Qu.:0.45242
##  Median :0.7620   Median :1.44455
##  Mean   :0.7620   Mean   :2.04876
##  3rd Qu.:0.8839   3rd Qu.:3.67152
##  Max.   :1.0058   Max.   :5.44067
##
## --------------------------------------------------------------------------------
##  ::: Radiation::Central Pyrenees (KDE)
## --------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (79 obs.);   Bandwidth 'bw' = 0.06972
##
##        x                 y
##  Min.   :0.03275   Min.   :0.00244
##  1st Qu.:0.31174   1st Qu.:0.19066
##  Median :0.59073   Median :0.90156
##  Mean   :0.59073   Mean   :0.89512
##  3rd Qu.:0.86972   3rd Qu.:1.57317
##  Max.   :1.14871   Max.   :1.84414
##
## --------------------------------------------------------------------------------
##  ::: Radiation::Sierra de Guadarrama (KDE)
## --------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (79 obs.);   Bandwidth 'bw' = 0.02259
##
##        x                y
##  Min.   :0.5281   Min.   :0.005388
##  1st Qu.:0.6165   1st Qu.:0.610106
##  Median :0.7049   Median :2.861213
##  Mean   :0.7049   Mean   :2.824994
##  3rd Qu.:0.7933   3rd Qu.:4.536993
##  Max.   :0.8817   Max.   :6.995343
##
## --------------------------------------------------------------------------------
##  ::: Phos_P::Central Andes (KDE)
## --------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (79 obs.);   Bandwidth 'bw' = 0.5516
##
##        x                 y
```

```
##  Min.   :-1.6350   Min.   :0.0001048
##  1st Qu.: 0.7071   1st Qu.:0.0146593
##  Median : 3.0491   Median :0.0968547
##  Mean   : 3.0491   Mean   :0.1066326
##  3rd Qu.: 5.3912   3rd Qu.:0.2008034
##  Max.   : 7.7333   Max.   :0.2496053
##
## ------------------------------------------------------------------------------
##  ::: Phos_P::Central Pyrenees (KDE)
## ------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (79 obs.);   Bandwidth 'bw' = 0.2593
##
##        x                y
##  Min.   :-0.428   Min.   :0.0002185
##  1st Qu.: 0.893   1st Qu.:0.0131761
##  Median : 2.214   Median :0.0973505
##  Mean   : 2.214   Mean   :0.1890609
##  3rd Qu.: 3.535   3rd Qu.:0.3797546
##  Max.   : 4.856   Max.   :0.5555300
##
## ------------------------------------------------------------------------------
##  ::: Phos_P::Sierra de Guadarrama (KDE)
## ------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (79 obs.);   Bandwidth 'bw' = 0.5291
##
##        x                y
##  Min.   : 1.508   Min.   :0.0003259
##  1st Qu.: 3.690   1st Qu.:0.0311437
##  Median : 5.873   Median :0.0655336
##  Mean   : 5.873   Mean   :0.1144329
##  3rd Qu.: 8.055   3rd Qu.:0.2143203
##  Max.   :10.237   Max.   :0.2826972
##
## ------------------------------------------------------------------------------
##  ::: Glu_P::Central Andes (KDE)
## ------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (79 obs.);   Bandwidth 'bw' = 0.4639
##
##        x                 y
##  Min.   :-1.2777   Min.   :0.0001228
##  1st Qu.: 0.9773   1st Qu.:0.0134185
##  Median : 3.2323   Median :0.0427511
```

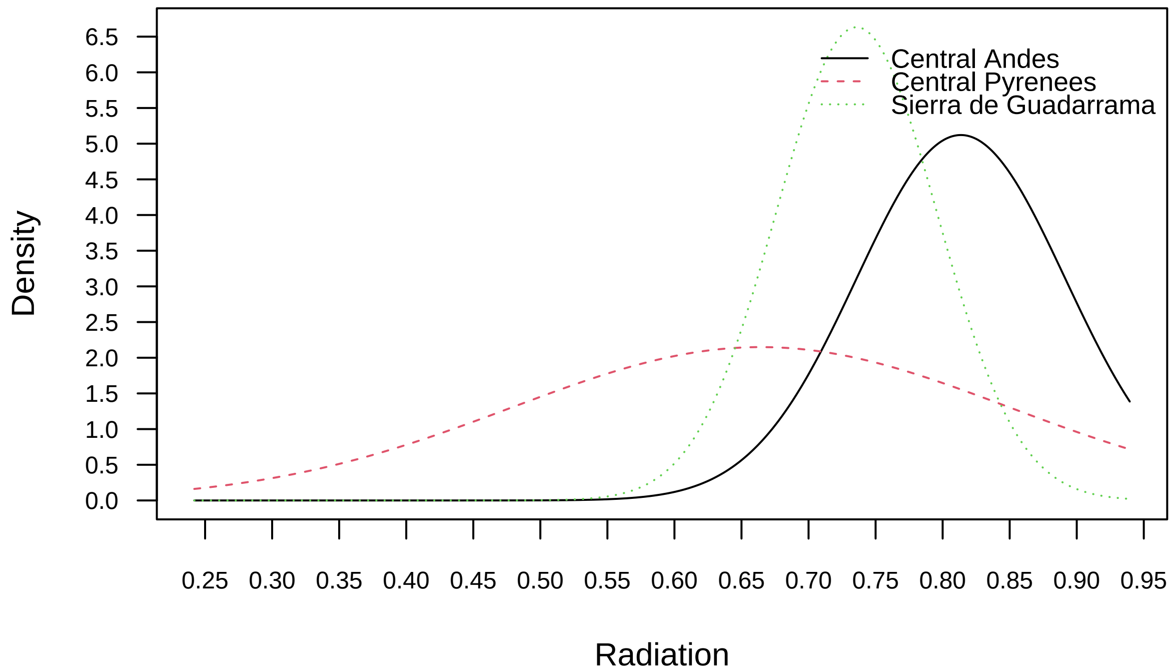
```
##  Mean   : 3.2323   Mean   :0.1107377
##  3rd Qu.: 5.4873   3rd Qu.:0.1821924
##  Max.   : 7.7423   Max.   :0.3897858
##
## ------------------------------------------------------------------------------------
##  ::: Glu_P::Central Pyrenees (KDE)
## ------------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (79 obs.);   Bandwidth 'bw' = 0.3141
##
##        x                 y
##  Min.   :-0.7663   Min.   :0.0001823
##  1st Qu.: 0.6648   1st Qu.:0.0237255
##  Median : 2.0960   Median :0.1363899
##  Mean   : 2.0960   Mean   :0.1745036
##  3rd Qu.: 3.5272   3rd Qu.:0.2977656
##  Max.   : 4.9583   Max.   :0.4703023
##
## ------------------------------------------------------------------------------------
##  ::: Glu_P::Sierra de Guadarrama (KDE)
## ------------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (79 obs.);   Bandwidth 'bw' = 0.2021
##
##        x                y
##  Min.   :0.7808   Min.   :0.0005627
##  1st Qu.:2.0427   1st Qu.:0.0515247
##  Median :3.3047   Median :0.0848366
##  Mean   :3.3047   Mean   :0.1978990
##  3rd Qu.:4.5666   3rd Qu.:0.3300094
##  Max.   :5.8286   Max.   :0.6894132
##
## ------------------------------------------------------------------------------------
##  ::: SOC_P::Central Andes (KDE)
## ------------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (79 obs.);   Bandwidth 'bw' = 0.2899
##
##        x                 y
##  Min.   :-0.6056   Min.   :0.0000947
##  1st Qu.: 1.5510   1st Qu.:0.0086619
##  Median : 3.7076   Median :0.0466798
##  Mean   : 3.7076   Mean   :0.1158063
##  3rd Qu.: 5.8642   3rd Qu.:0.1111669
##  Max.   : 8.0208   Max.   :0.5164940
```

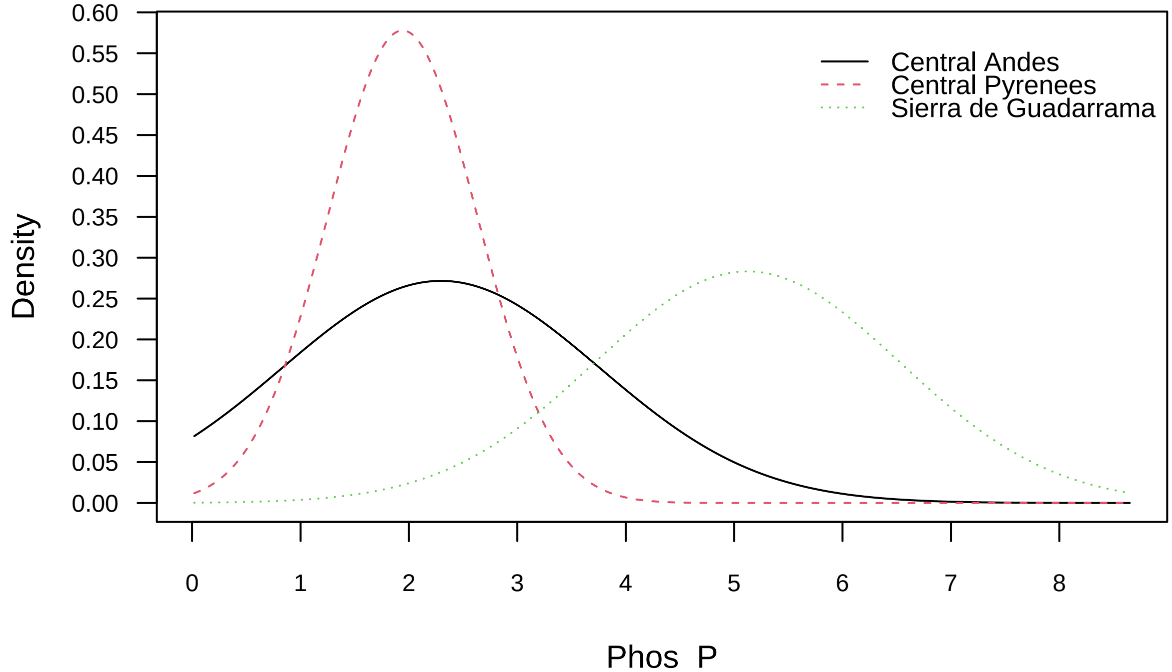
```
##
## ------------------------------------------------------------------------------
##  ::: SOC_P::Central Pyrenees (KDE)
## ------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (79 obs.);   Bandwidth 'bw' = 0.4193
##
##        x                 y
##  Min.   :-0.7739   Min.   :0.0001561
##  1st Qu.: 1.1170   1st Qu.:0.0183161
##  Median : 3.0080   Median :0.0984330
##  Mean   : 3.0080   Mean   :0.1320730
##  3rd Qu.: 4.8990   3rd Qu.:0.2464198
##  Max.   : 6.7899   Max.   :0.3478511
##
## ------------------------------------------------------------------------------
##  ::: SOC_P::Sierra de Guadarrama (KDE)
## ------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (79 obs.);   Bandwidth 'bw' = 0.7374
##
##        x                 y
##  Min.   :-0.3238   Min.   :0.0001544
##  1st Qu.: 3.3329   1st Qu.:0.0127008
##  Median : 6.9896   Median :0.0332595
##  Mean   : 6.9896   Mean   :0.0682960
##  3rd Qu.:10.6463   3rd Qu.:0.1391295
##  Max.   :14.3030   Max.   :0.1878687
##
## ------------------------------------------------------------------------------
##  ::: NT_P::Central Andes (KDE)
## ------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (79 obs.);   Bandwidth 'bw' = 0.3283
##
##        x                 y
##  Min.   :-0.8691   Min.   :0.000173
##  1st Qu.: 1.2216   1st Qu.:0.007242
##  Median : 3.3124   Median :0.023837
##  Mean   : 3.3124   Mean   :0.119451
##  3rd Qu.: 5.4031   3rd Qu.:0.152429
##  Max.   : 7.4938   Max.   :0.602964
##
## ------------------------------------------------------------------------------
##  ::: NT_P::Central Pyrenees (KDE)
```

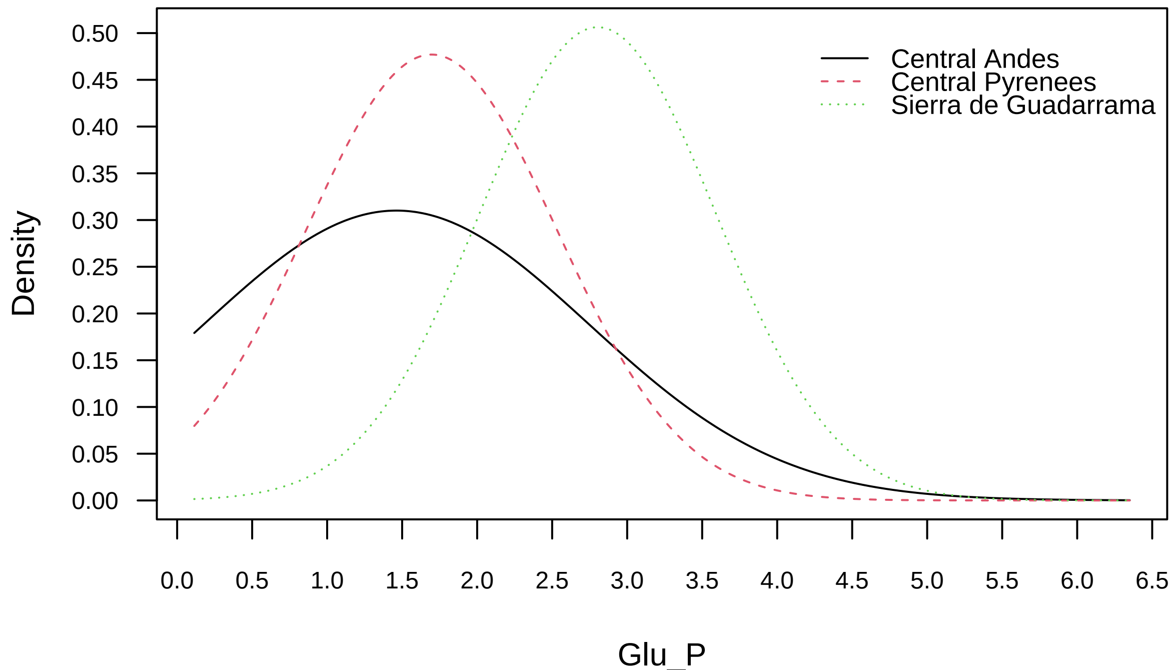
```
## --------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (79 obs.);   Bandwidth 'bw' = 0.7291
##
##        x                 y
##  Min.   :-1.615   Min.   :7.796e-05
##  1st Qu.: 2.039   1st Qu.:7.878e-03
##  Median : 5.694   Median :4.450e-02
##  Mean   : 5.694   Mean   :6.834e-02
##  3rd Qu.: 9.349   3rd Qu.:1.246e-01
##  Max.   :13.003   Max.   :1.979e-01
##
## --------------------------------------------------------------------------------
##  ::: NT_P::Sierra de Guadarrama (KDE)
## --------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (79 obs.);   Bandwidth 'bw' = 0.6406
##
##        x                  y
##  Min.   :-0.1991   Min.   :0.0001779
##  1st Qu.: 2.4441   1st Qu.:0.0157512
##  Median : 5.0872   Median :0.0917607
##  Mean   : 5.0872   Mean   :0.0944846
##  3rd Qu.: 7.7304   3rd Qu.:0.1537367
##  Max.   :10.3735   Max.   :0.2392512
##
## --------------------------------------------------------------------------------
##
## # ... and 20 more tables
##
## --------------------------------------------------------------------------------
```

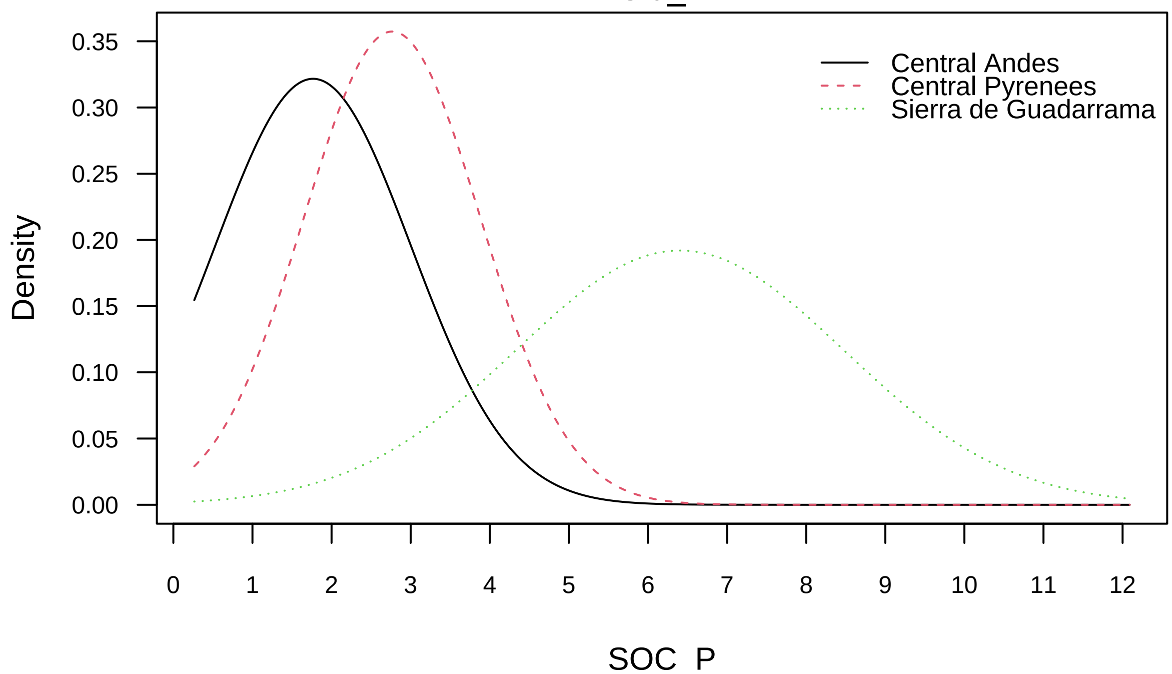
Conditional density plots with 'uskernel=FALSE'. It means that Gaussian distribution is applied to 'numeric' vairable.

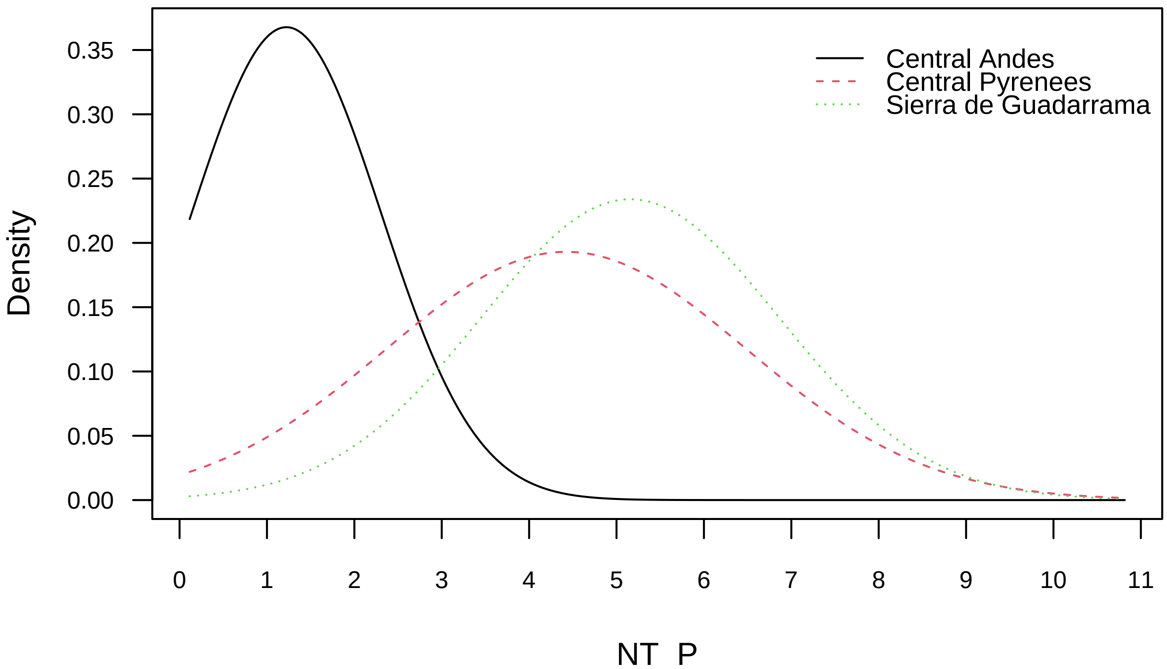
```
mountain.nb <- naive_bayes(Mountain_range ~ .,
                       data = Mountain_data.tr%>%
  select(!c(Plot, Subplot, Date, Day, Month, Year, Locality, Country)), usekernel=FALSE, laplace=1)
#par(mfrow=c(2,2))
plot(mountain.nb, arg.num = list(col = 1:3,
                             legend.position = "topright",
                             legend.cex = 0.8), prob="conditional")
```


Density
6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95
Central Andes
Central Pyrenees
Sierra de Guadarrama
Radiation

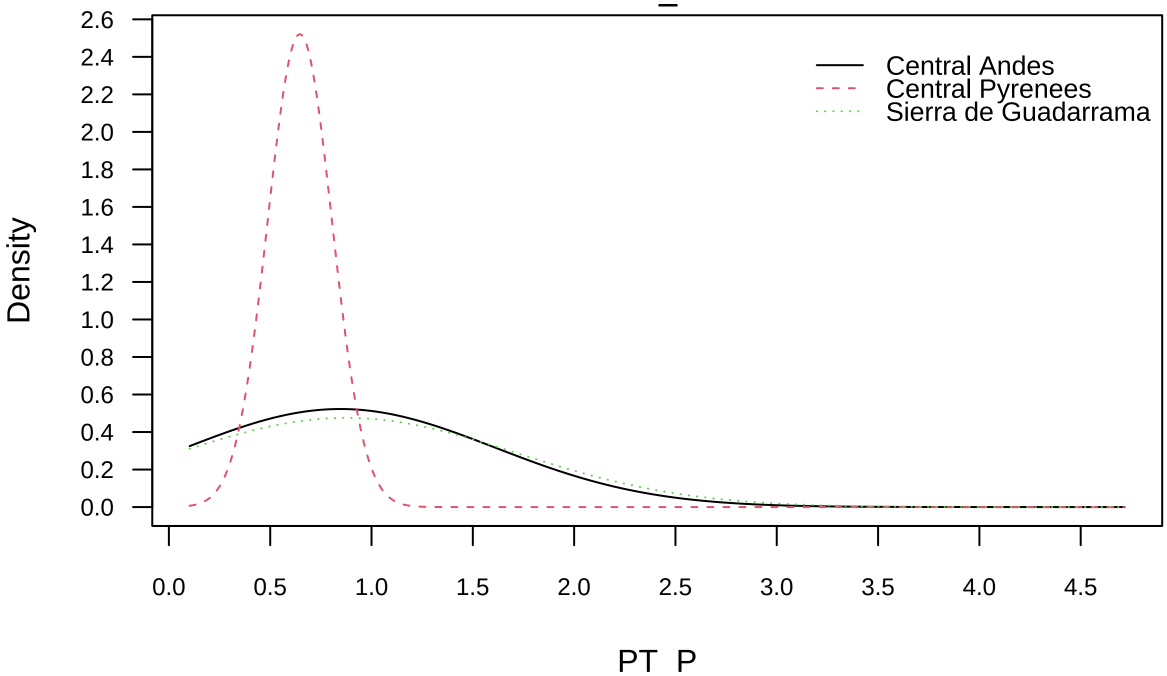

Density
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0 1 2 3 4 5 6 7 8
Central Andes
Central Pyrenees
Sierra de Guadarrama
Phos_P


Density
Glu_P
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
Central Andes
Central Pyrenees
Sierra de Guadarrama

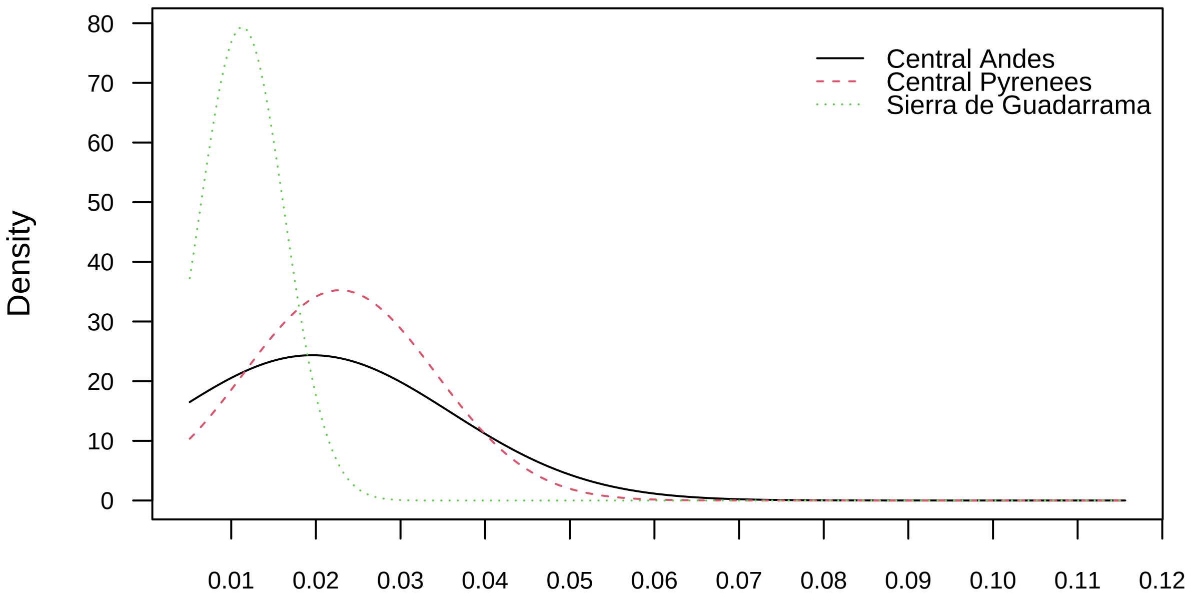

Density
SOC_P
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0
1
2
3
4
5
6
7
8
9
10
11
12
Central Andes
Central Pyrenees
Sierra de Guadarrama

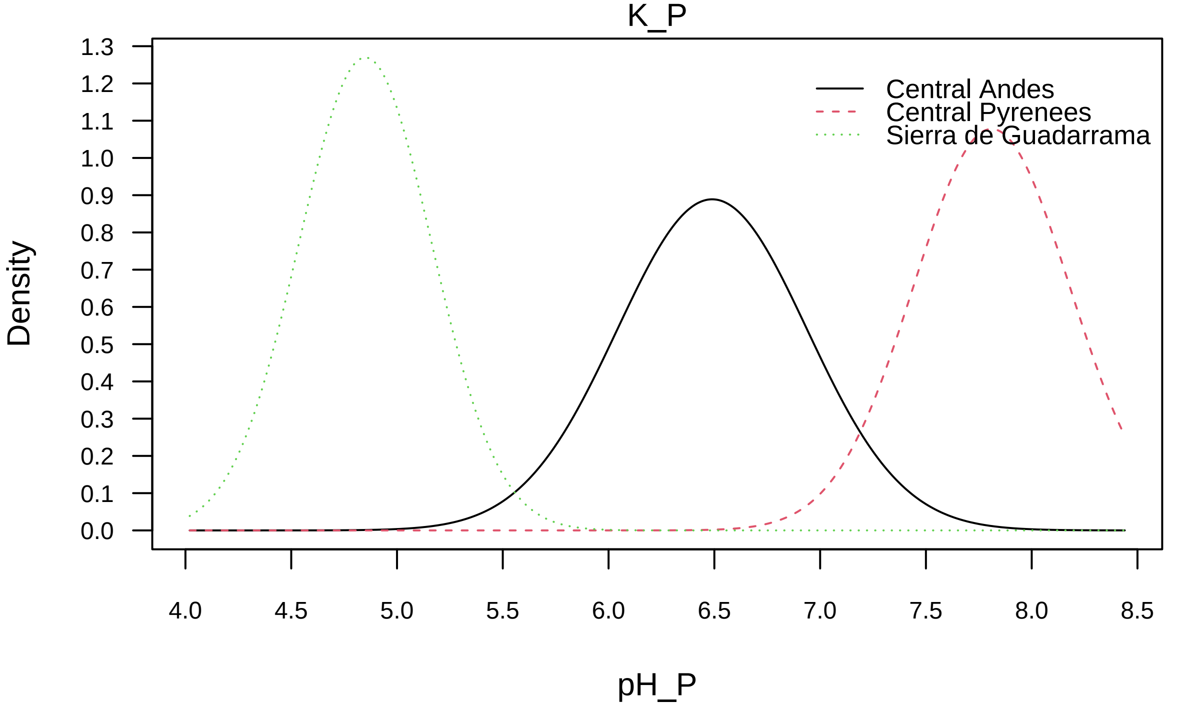


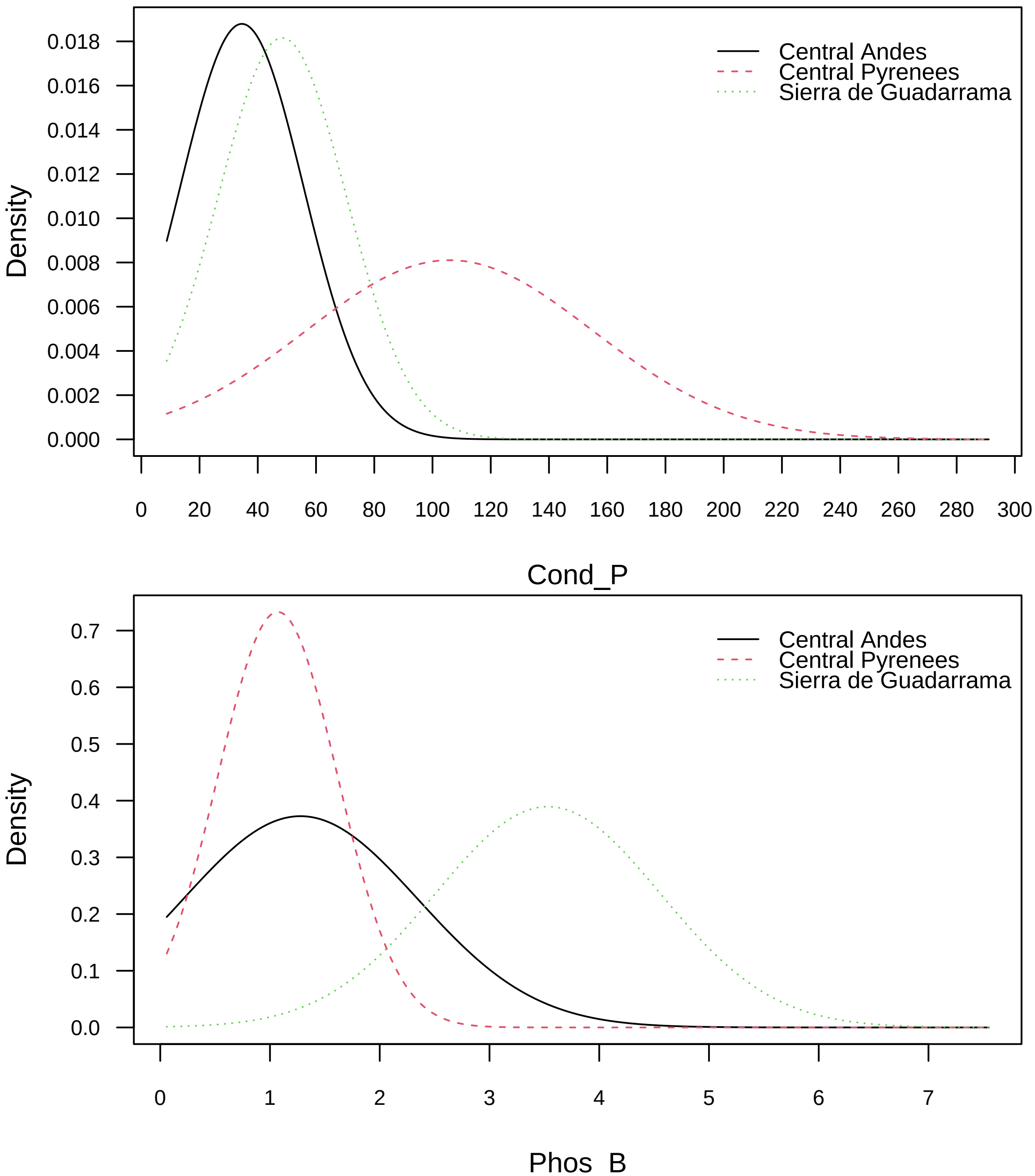
Central Andes
Central Pyrenees
Sierra de Guadarrama
Density
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
1
2
3
4
5
6
7
8
9
10
11
NT_P

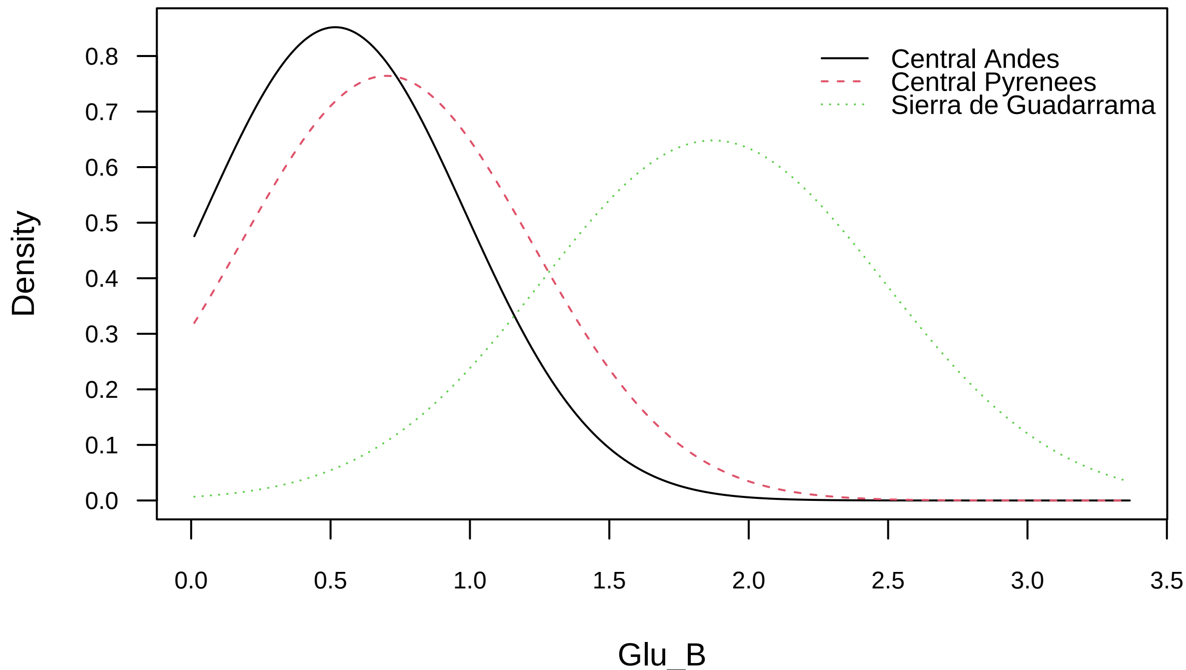


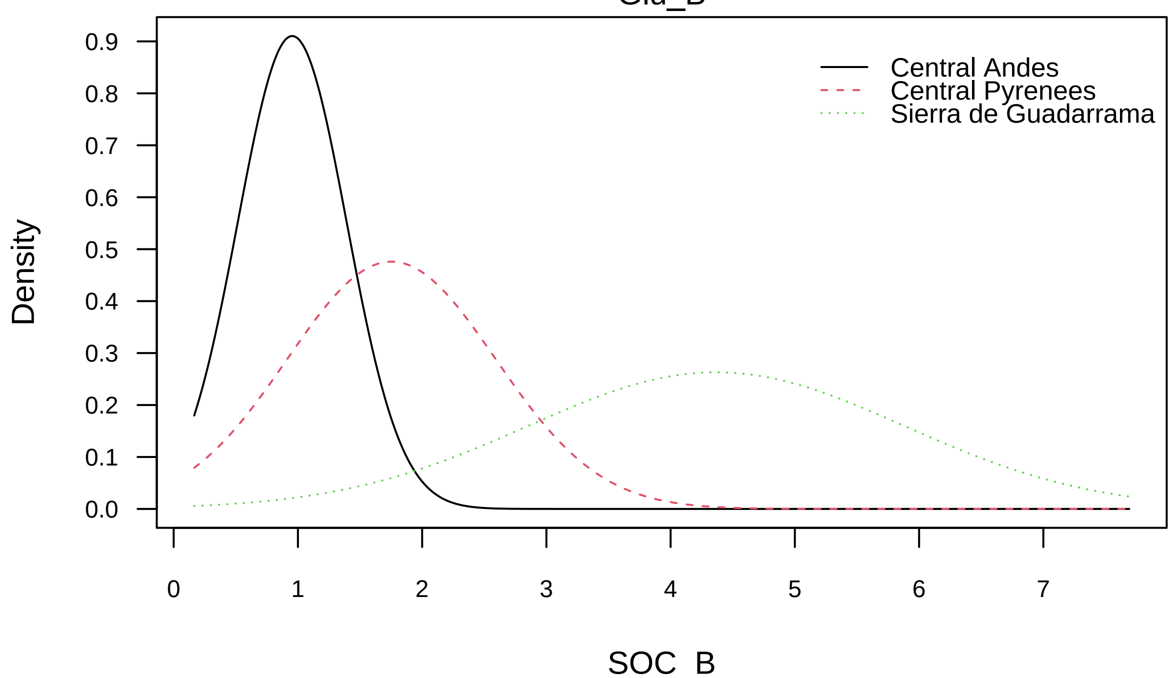
Central Andes
Central Pyrenees
Sierra de Guadarrama
Density
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
PT_P

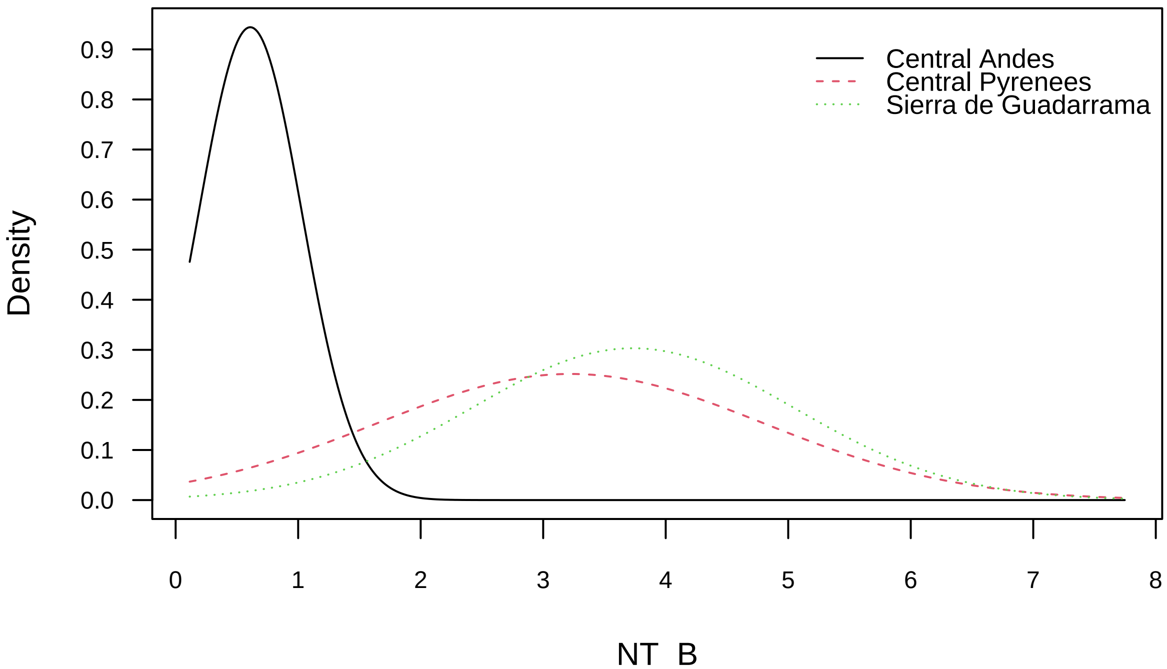

Density
0
10
20
30
40
50
60
70
80
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
0.11
0.12
K_P
Central Andes
Central Pyrenees
Sierra de Guadarrama

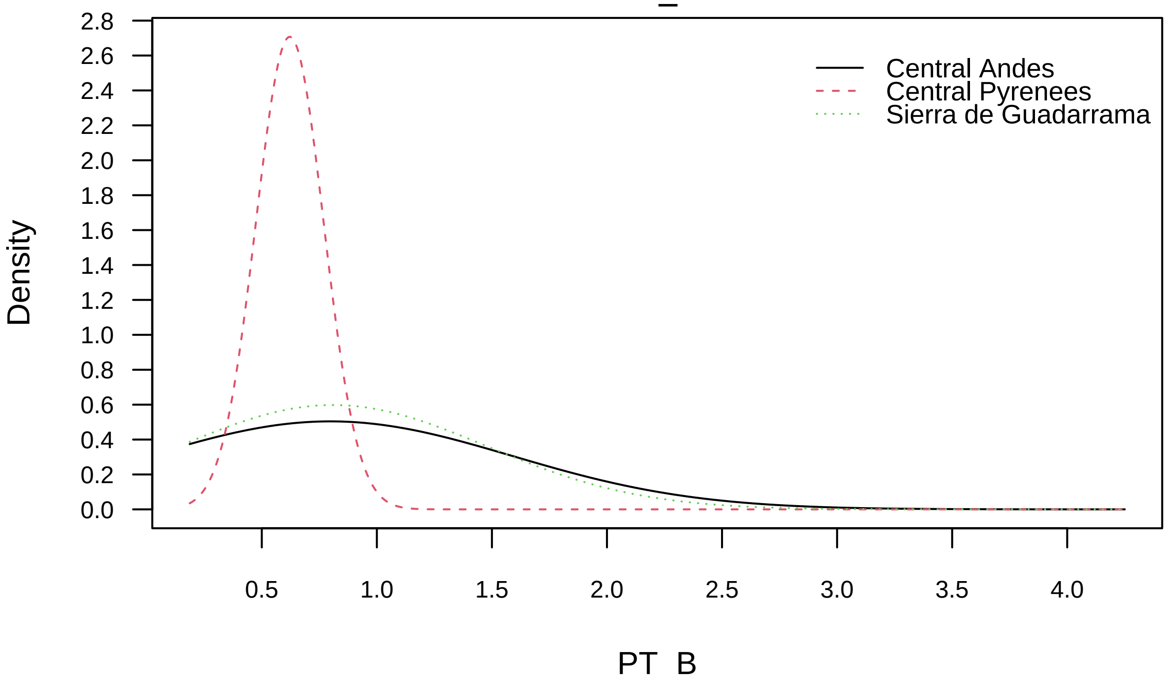

Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
1.3
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
pH_P
Central Andes
Central Pyrenees
Sierra de Guadarrama

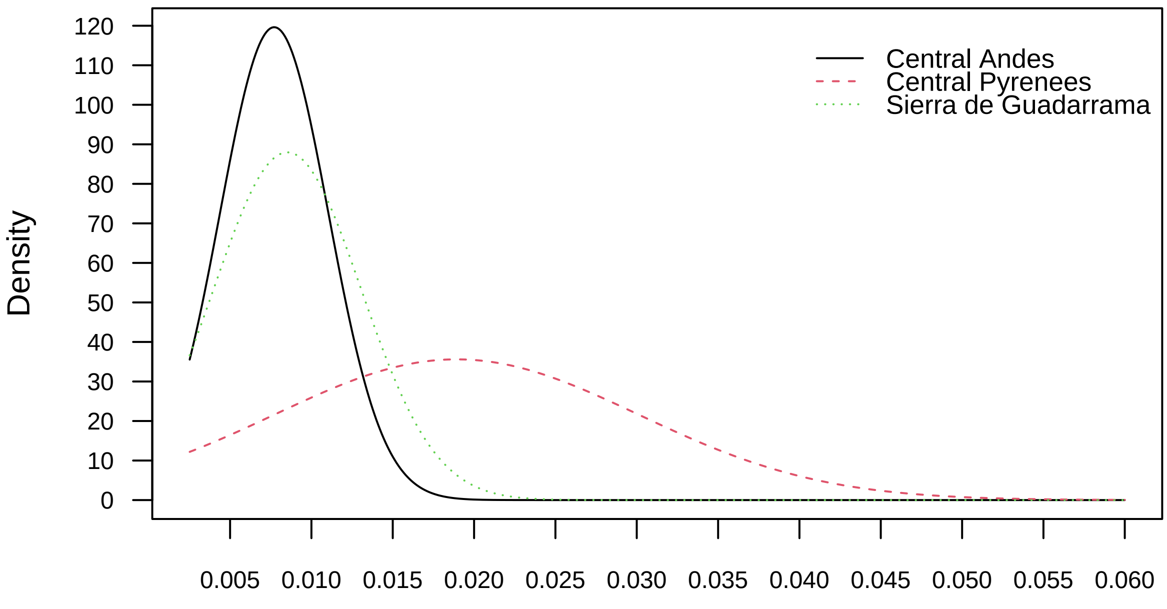

Central Andes
Central Pyrenees
Sierra de Guadarrama
Density
0.018
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0 20 40 60 80 100 120 140 160 180 200 220 240 260 280 300
Cond_P
Central Andes
Central Pyrenees
Sierra de Guadarrama
Density
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0 1 2 3 4 5 6 7
Phos_B

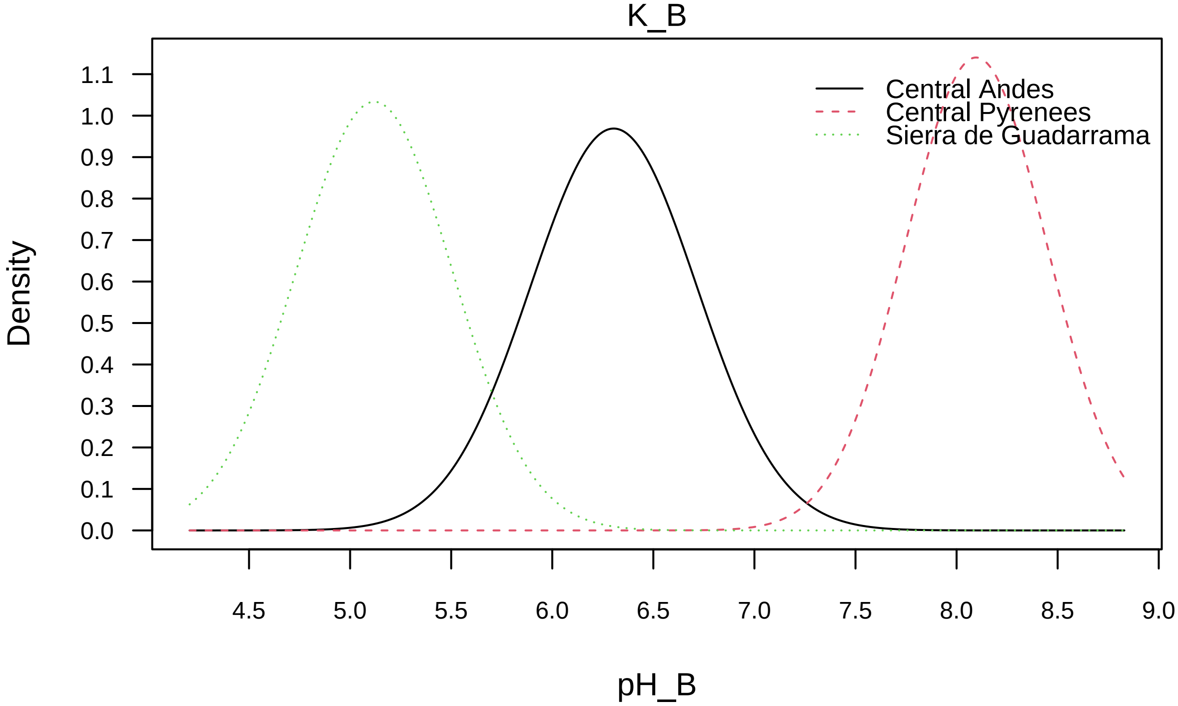

Density
Glu_B
Central Andes
Central Pyrenees
Sierra de Guadarrama
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

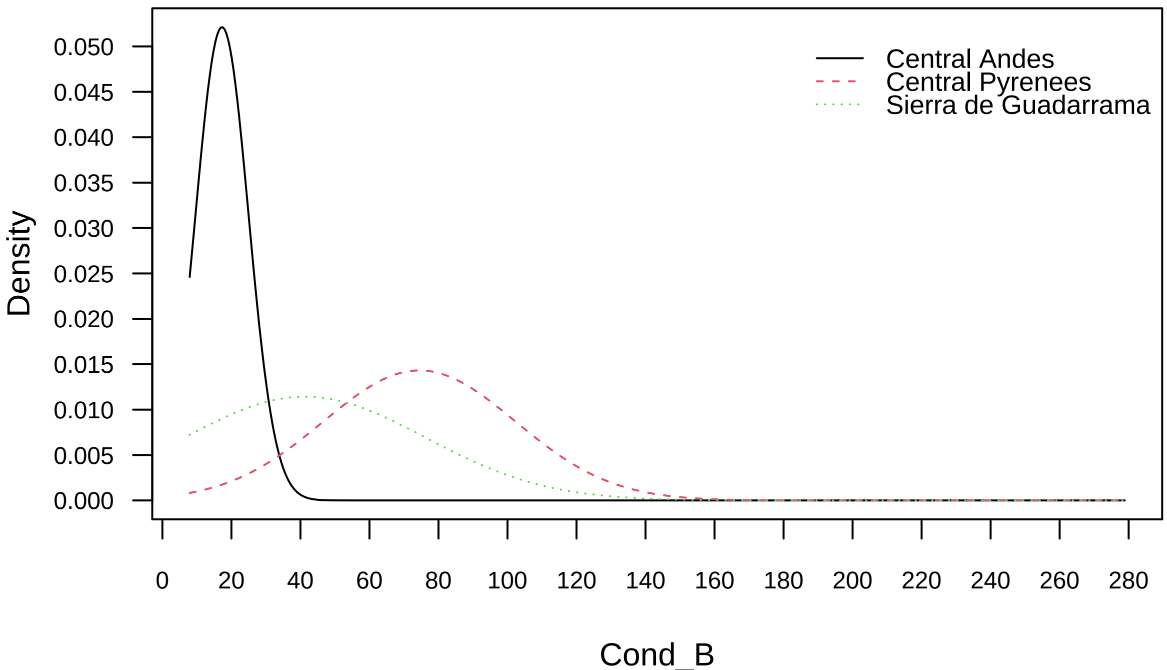

Density
SOC_B
Central Andes
Central Pyrenees
Sierra de Guadarrama
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
1
2
3
4
5
6
7

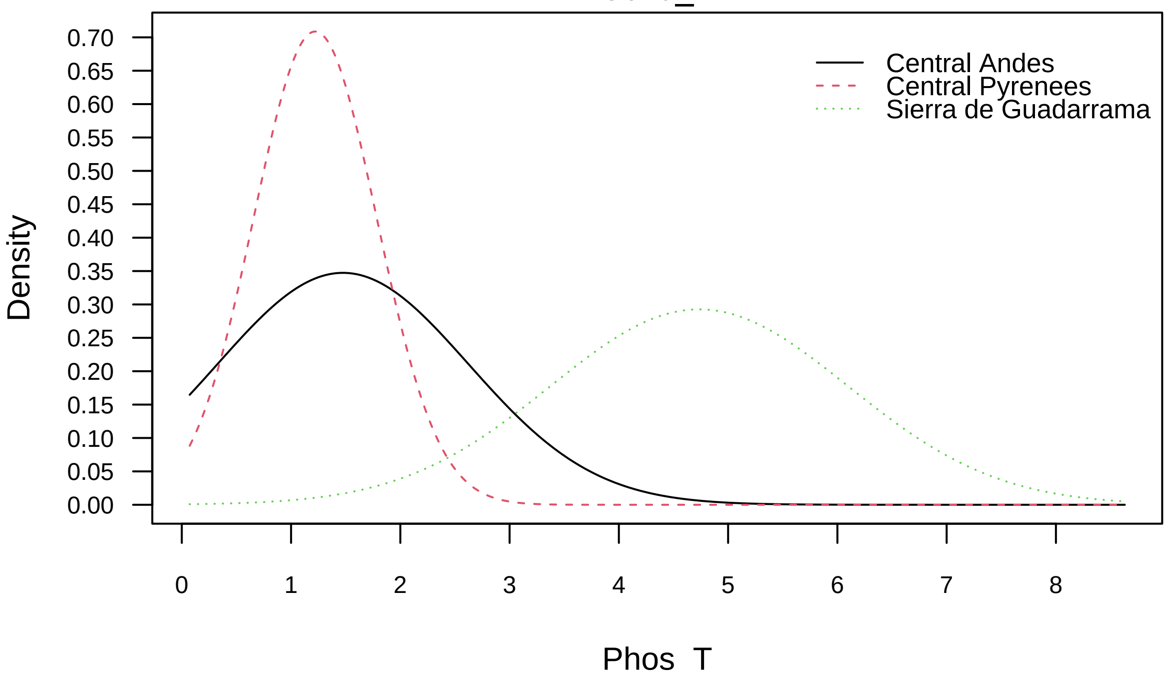

Density
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5
6
7
8
NT_B
Central Andes
Central Pyrenees
Sierra de Guadarrama

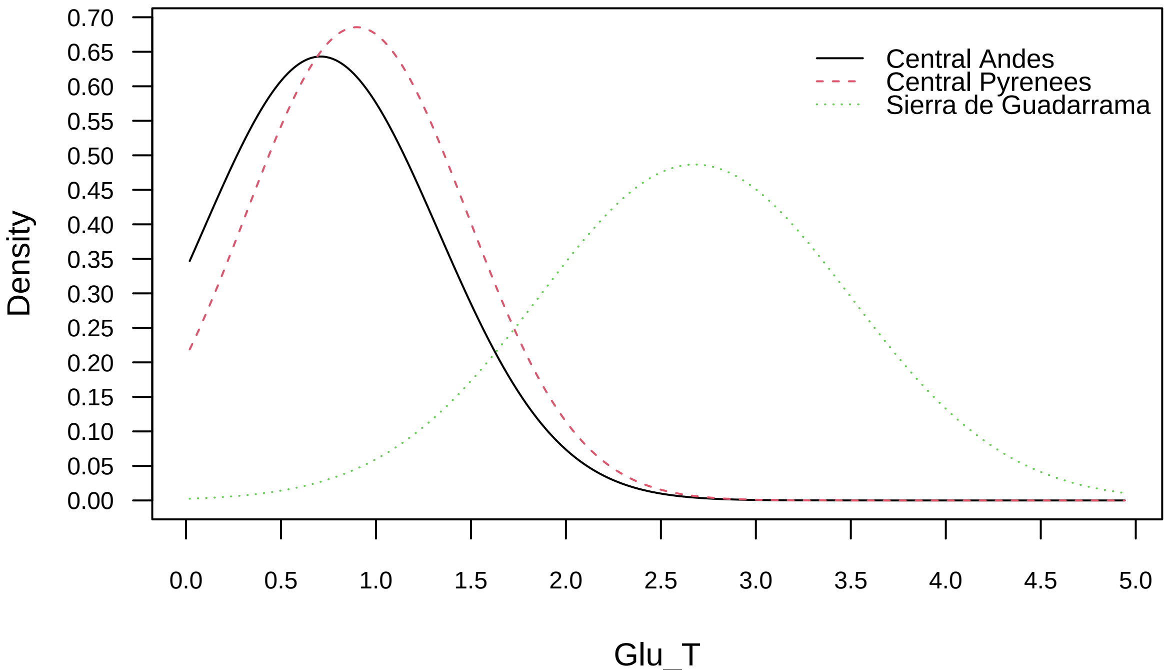

Density
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
PT_B
Central Andes
Central Pyrenees
Sierra de Guadarrama

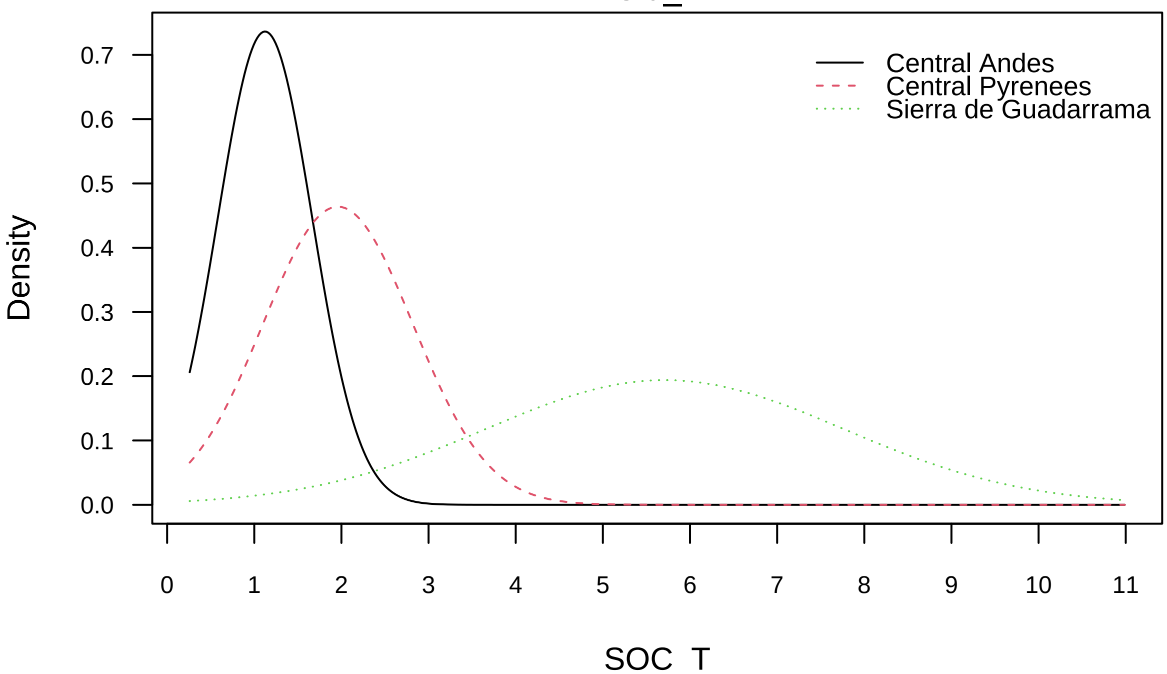

Density
K_B
Central Andes
Central Pyrenees
Sierra de Guadarrama
0
10
20
30
40
50
60
70
80
90
100
110
120
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040
0.045
0.050
0.055
0.060

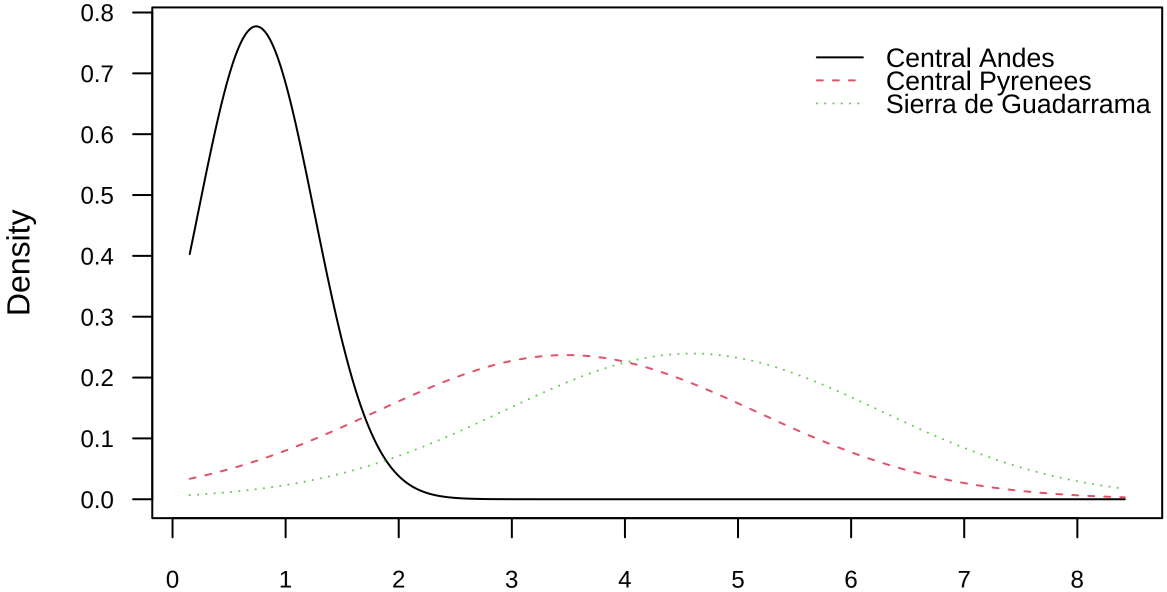

Density
pH_B
Central Andes
Central Pyrenees
Sierra de Guadarrama
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0


Central Andes
Central Pyrenees
Sierra de Guadarrama
Density
0.050
0.045
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
Cond_B

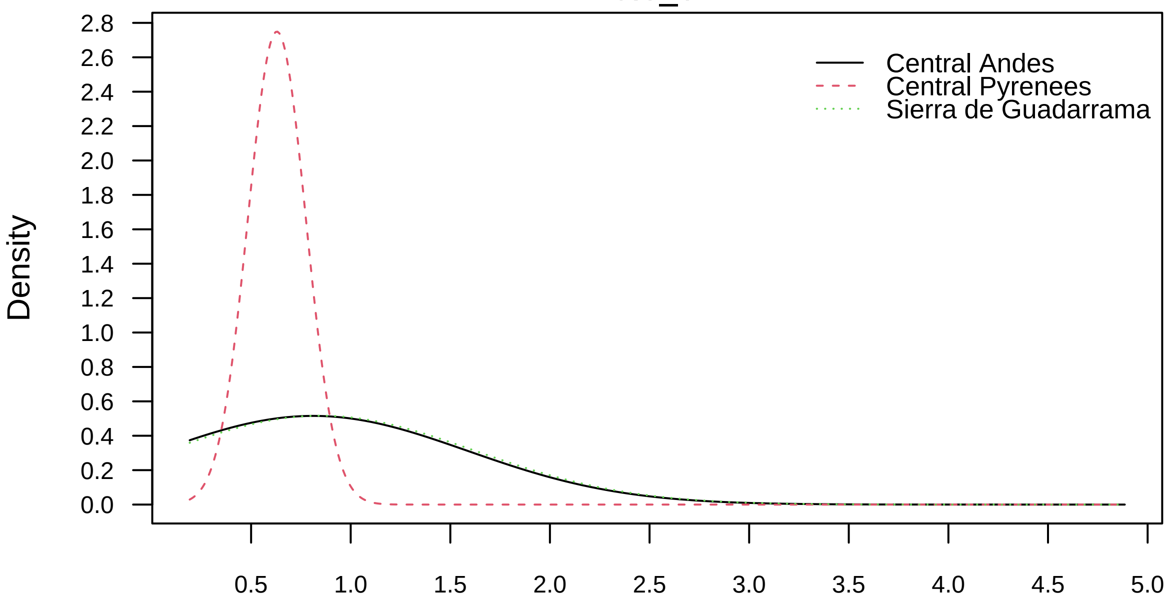

Central Andes
Central Pyrenees
Sierra de Guadarrama
Density
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
1
2
3
4
5
6
7
8
Phos_T


Central Andes
Central Pyrenees
Sierra de Guadarrama
Density
0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65 0.70
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Glu_T

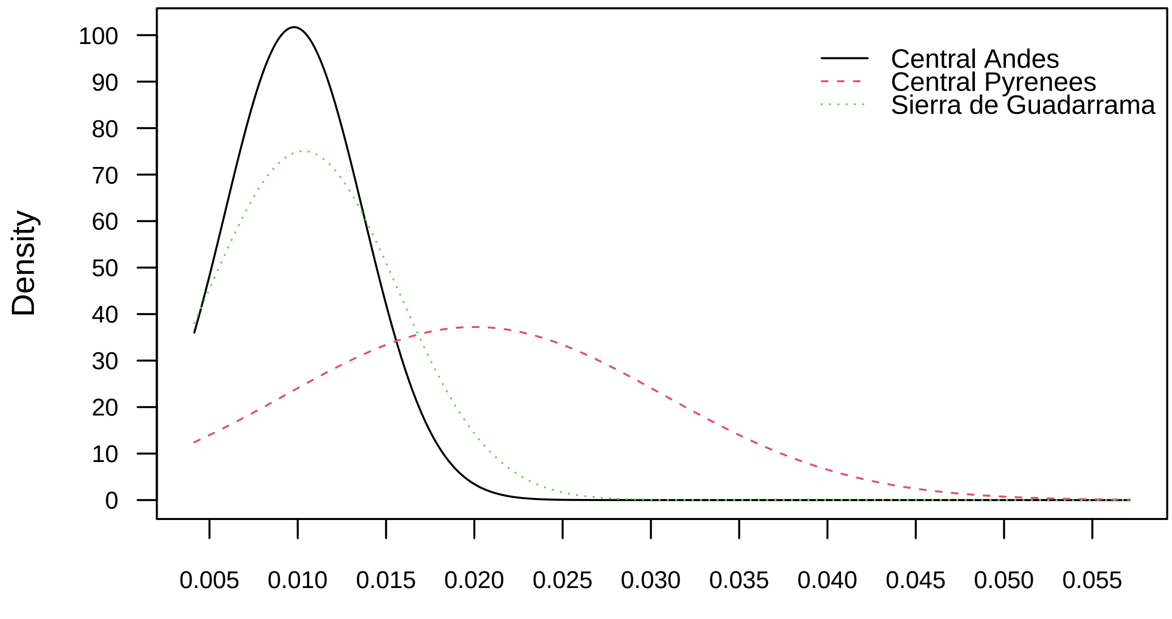

Central Andes
Central Pyrenees
Sierra de Guadarrama
Density
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7
0 1 2 3 4 5 6 7 8 9 10 11
SOC_T

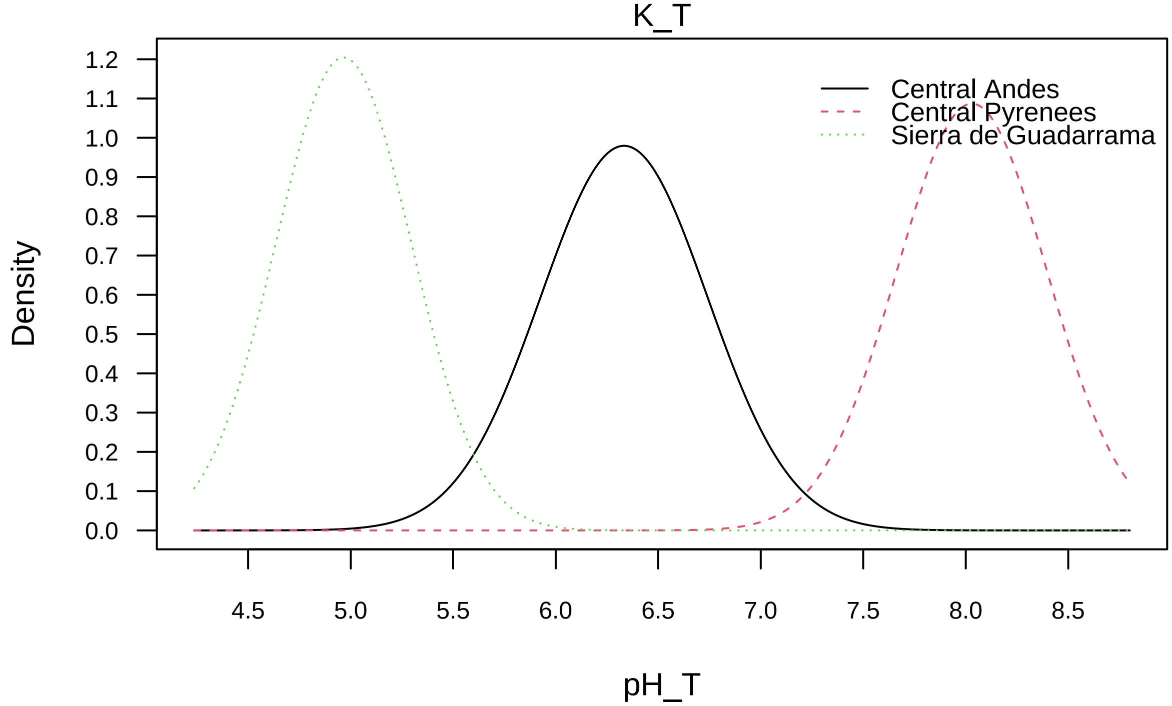

Central Andes
Central Pyrenees
Sierra de Guadarrama
Density
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5
6
7
8

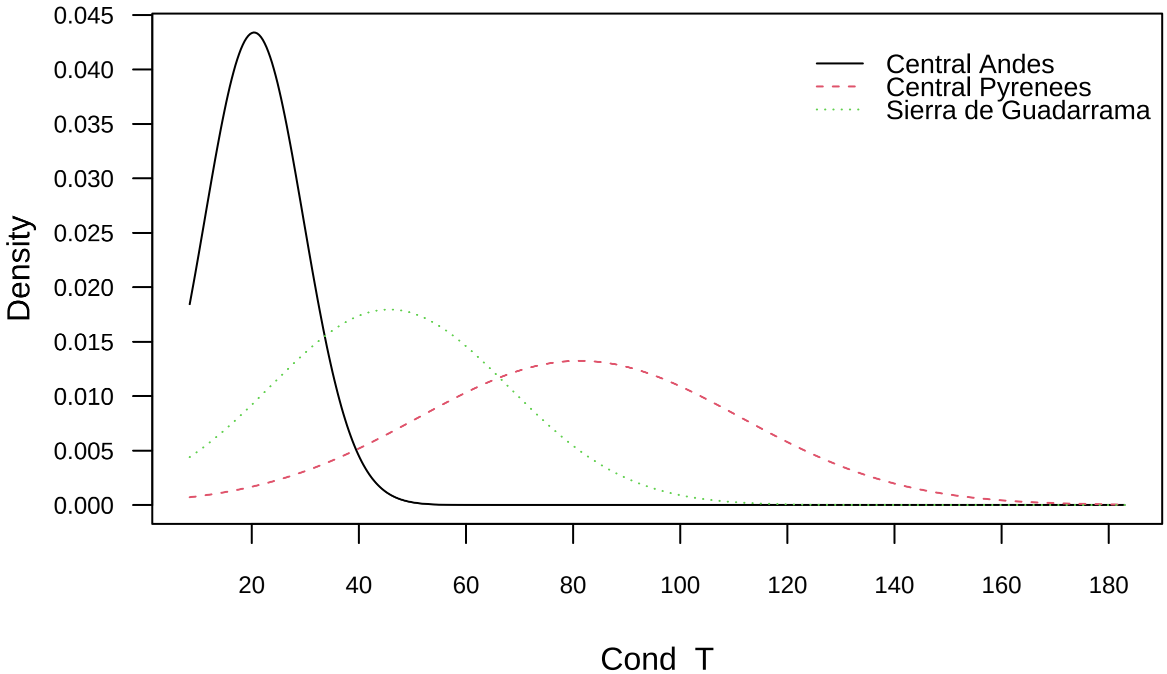
NT_T


Central Andes
Central Pyrenees
Sierra de Guadarrama
Density
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

PT_T


Central Andes
Central Pyrenees
Sierra de Guadarrama
Density
K_T
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040
0.045
0.050
0.055
0
10
20
30
40
50
60
70
80
90
100


Central Andes
Central Pyrenees
Sierra de Guadarrama
Density
pH_T
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2




```r
par(mfrow=c(1,1))
mountain.nb
```

```
##
## ================================== Naive Bayes ==================================
##
##  Call:
## naive_bayes.formula(formula = Mountain_range ~ ., data = Mountain_data.tr %>%
##     select(!c(Plot, Subplot, Date, Day, Month, Year, Locality,
##         Country)), laplace = 1, usekernel = FALSE)
##
## ---------------------------------------------------------------------------------
##
## Laplace smoothing: 1
##
## ---------------------------------------------------------------------------------
##
##  A priori probabilities:
##
##        Central Andes     Central Pyrenees Sierra de Guadarrama
##            0.3333333            0.3333333            0.3333333
##
## ---------------------------------------------------------------------------------
##
##  Tables:
##
## ---------------------------------------------------------------------------------
##  ::: Radiation (Gaussian)
## ---------------------------------------------------------------------------------
##
## Radiation Central Andes Central Pyrenees Sierra de Guadarrama
##      mean    0.81360704       0.66437606           0.73577776
##      sd      0.07790181       0.18563689           0.06015075
```

```
##
## -------------------------------------------------------------------------------
##  ::: Phos_P (Gaussian)
## -------------------------------------------------------------------------------
##
## Phos_P Central Andes Central Pyrenees Sierra de Guadarrama
##   mean     2.2952973        1.9415222            5.1226074
##   sd       1.4686357        0.6904481            1.4087032
##
## -------------------------------------------------------------------------------
##  ::: Glu_P (Gaussian)
## -------------------------------------------------------------------------------
##
## Glu_P  Central Andes Central Pyrenees Sierra de Guadarrama
##   mean     1.4616516        1.6961709            2.8037052
##   sd       1.2864604        0.8363111            0.7879578
##
## -------------------------------------------------------------------------------
##  ::: SOC_P (Gaussian)
## -------------------------------------------------------------------------------
##
## SOC_P  Central Andes Central Pyrenees Sierra de Guadarrama
##   mean      1.766067         2.765690             6.403696
##   sd        1.240214         1.116397             2.077923
##
## -------------------------------------------------------------------------------
##  ::: NT_P (Gaussian)
## -------------------------------------------------------------------------------
##
## NT_P   Central Andes Central Pyrenees Sierra de Guadarrama
##   mean      1.222499         4.425051             5.155693
##   sd        1.084850         2.066501             1.705442
##
## -------------------------------------------------------------------------------
##
## # ... and 20 more tables
##
## -------------------------------------------------------------------------------
```

pH_T is medium in Central Andes, lower in Sierra de Guadarrama and higher in Central Pyrenees.

Predictions

```
head(predict(mountain.nb, newdata = Mountain_data.tr[-c(1:9)]),20)
```

```
##  [1] Central Andes Central Andes Central Andes Central Andes Central Andes
##  [6] Central Andes Central Andes Central Andes Central Andes Central Andes
## [11] Central Andes Central Andes Central Andes Central Andes Central Andes
## [16] Central Andes Central Andes Central Andes Central Andes Central Andes
## Levels: Central Andes Central Pyrenees Sierra de Guadarrama
```

```
head(predict(mountain.nb, newdata = Mountain_data.tr[-c(1:9)], type="prob"),10)
```

```
##       Central Andes Central Pyrenees Sierra de Guadarrama
##  [1,]             1     1.261897e-21         1.753060e-31
##  [2,]             1     1.295786e-31         3.223123e-26
```

```
##  [3,]              1     1.810897e-25        1.712095e-27
##  [4,]              1     2.129147e-25        8.880046e-30
##  [5,]              1     4.134568e-12        5.379184e-38
##  [6,]              1     7.661957e-30        5.203245e-28
##  [7,]              1     2.363779e-24        2.070797e-29
##  [8,]              1     2.143505e-14        7.570771e-54
##  [9,]              1     5.808141e-34        1.691998e-27
## [10,]              1     3.203829e-29        1.365213e-31
```

Confusion matrix train data set

```r
p1 <- predict(mountain.nb, Mountain_data.tr[-c(1:9)])
(tab1 <- table(p1, Mountain_data.tr$Mountain_range))
```

```
##
## p1                     Central Andes Central Pyrenees Sierra de Guadarrama
##   Central Andes                   78                0                    2
##   Central Pyrenees                 1               79                    0
##   Sierra de Guadarrama             0                0                   77
```

```r
sum(diag(tab1)) / sum(tab1)
```

```
## [1] 0.9873418
```

Confusion Matrix test data set

```r
p2 <- predict(mountain.nb, Mountain_data.te[-c(1:9)])
(tab2 <- table(p2, Mountain_data.te$Mountain_range))
```

```
##
## p2                     Central Andes Central Pyrenees Sierra de Guadarrama
##   Central Andes                   19                0                    1
##   Central Pyrenees                 1               29                    0
##   Sierra de Guadarrama             1                0                   57
```

```r
sum(diag(tab2)) / sum(tab2)
```

```
## [1] 0.9722222
```

K-NN Model

We use a 2-NN to predict the test set using the training set

```r
library(caret)
set.seed(123)
KNN <- knn3(data=Mountain_data.tr,Mountain_data.tr$Mountain_range ~ ., k=2)
MR.te.pred <- predict(KNN, newdata = Mountain_data.tr,type ="class")

TAB <- table(Obs=Mountain_data.tr$Mountain_range, Pred=MR.te.pred) # confusion matrix
TAB
```

```
##                         Pred
## Obs                      Central Andes Central Pyrenees Sierra de Guadarrama
##   Central Andes                     79                0                    0
##   Central Pyrenees                   0               79                    0
##   Sierra de Guadarrama               0                0                   79
```

```r
(ACC <- sum(diag(TAB))/sum(TAB)) # accuracy
```

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