Datasets and assignment description

Dataset 1: Lactase in piglets

**Description**

As part of a larger study of the intestinal health in newborn piglets, the gut enzyme

lactase was measured in 20 piglets taken from 5 different litters. For each of the 20 piglets the lactase level was measured in three different regions. At the time the measurementwas taken the piglet was either unborn (status=1) or newborn (status=2). These data are kindly provided by Charlotte Reinhard Bjørnvad, Department of Animal Science and Animal Health, Division of Animal Nutrition, Royal Veterinary and Agricultural University.

**The data**

* Number of observations: 60

Variable Description:

* Litter: Litter identification, numbered 1–5
* Pig: Piglet identification, numbered 1–22
* Status: Status identification, numbered 1,2
* loclact :Log transformed lactase measurement

Source :Charlotte Reinhard Bjørnvad, Department of Animal Science and Animal Health, Divisionof Animal Nutrition, Royal Veterinary and Agricultural University.

R-code data encoding:

dat <- read\_csv("alldata/lactase.txt") # read in data

# set variables correctly as factors and numerical for R to read

dat$litter=as.factor(dat$litter)

dat$pig=as.factor(dat$pig)

dat$reg=as.factor(dat$reg)

dat$status=as.factor(dat$status)

dat$loglact=as.numeric(dat$loglact)

# remove missing obs

dat=na.omit(dat)

Note: that the pig variable should not be included, as it is intended for a random effect model. However you can make a random effect model if you are up for the challenge.

Also it is not possible to do a test/training splitting to do predictions on this data set as it is too small (and messes up the balance design of the experiment) – I’ve included it anyway since it may be akin to your own type of data. Instead you should focus on the interpretation of the model. Hint anova might be a good type of model here! (Argue why ☺) – see <https://02429.compute.dtu.dk/eBook> , e-note 3 for a great example of one way to do this ☺

Dataset 2: Dunes with Dunes environment

The dune meadow vegetation data, dune, has cover class values of 30 species on 20 sites. That is 30 columns with 30 species names and 20 rows with each row being a sample from a new site within the dune meadow. The values of the rows represents abundance of a given specie, the higher the number, the higher abundance.

The corresponding environmental data frame dune.env has following entries:

* A1: A numeric vector of thickness of soil A1 horizon. (numerical value)
* Moisture: an ordered factor with levels: 1 < 2 < 4 < 5.
* Management: a factor with levels: BF (Biological farming), HF (Hobby farming), NM (Nature Conservation Management), and SF (Standard Farming).
* Use: an ordered factor of land-use with levels: Hayfield < Haypastu < Pasture.
* Manure: an ordered factor with levels: 0 < 1 < 2 < 3 < 4.

**Source**

Jongman, R.H.G, ter Braak, C.J.F & van Tongeren, O.F.R. (1987). *Data Analysis in Community and Landscape Ecology*. Pudoc, Wageningen.

R-data encoding:

# load the data

data("dune")

data("dune.env")

# combine data

data=cbind(dune,dune.env)

data$A1=as.numeric(data$A1)

data$Moisture=as.factor(data$Moisture)

data$Management=as.factor(data$Management)

data$Use=as.factor(data$Use)

data$Manure=as.factor(data$Manure)

**Supervised data analysis code**

Note for this data some aid with the supervised analysis will be given:

1. Modeling the MDS space: (vegan package) - link environment variables to dunes data

* CCA of data in ordinate MDS space code:

ord <- cca(dune ~., data=dune.env) # canonical correspondence analysis – it’s a way to link two datasets with one another

Take the CCA and preform ANOVA:

* anova(ord, by="term", permutations=199) # reduce like normal ANOVA

1. Alternative approach, modelling the Principle component space – link soil thickness to bacterial species

PLS model of data in PC space (caret and pls package), remember to split data accordingly

model <- train(

A1~., data = train[,1:31], method = "pls",

scale = FALSE, #think about why this is false

trControl = trainControl("cv", number = 10 ),

tuneLength = X # insert number of PCs

)

plot(model) # plot the model

PC\_opt=model$bestTune # get best PC number

# traning data

pred=predict(model,train[,1:30]) # training data prediction

data.frame(

RMSE = caret::RMSE(pred, train$A1),

Rsquare = caret::R2(pred, train$A1)

)

# testing data

pred=predict(model,test[,1:30])

data.frame(

RMSE = caret::RMSE(pred, test$A1),

Rsquare = caret::R2(pred,test$A1 )

)

# jack-knifing back to real space via pls package

mod <- plsr(A1 ~ . , ncomp = PC\_opt$ncomp, data = train[,1:31],validation="LOO", scale = FALSE, jackknife = TRUE)

obsfit <- predplot(mod, labels = rownames(train), which = "validation")

abline(lm(obsfit[,2] ~ obsfit[,1]))

plot(mod, "validation", estimate = c("train", "CV"), val.type = "R2",legendpos = "bottomright")

coefplot(mod, se.whiskers = TRUE, labels = prednames(mod), cex.axis = 0.5)

biplot(mod)

# get significant numbers

jack.test(mod, ncomp = PC\_opt$ncomp)

See E-note <https://27411.compute.dtu.dk/enotes/07_-_Partial_Least_Squares_Regression> for more help and/or the pls script on github.

Note you are not obliged to use PLS or CCA to model the data these are just suggestions