Wk11 coding exercise answers

Christine Hawkes

3/23/2022

Coding Exercises

1. Use the gls::nlme function to re-run the regression on PC Axis1 (as if homoscedasticity was violated)

- print the model info using the "summary(model)" command
- manual: https://cran.r-project.org/web/packages/nlme/nlme.pdf
- function: https://rdrr.io/cran/nlme/man/gls.html

```
library(nlme); packageVersion("nlme")
model_gls <- nlme::gls(Axis.1 ~ Plant_Height_m + pH + Nmin + WD.F + SPI.F + SLA.F, data = env_new)
summary(model_gls)
# pH and Nmin remain the only significant factors
```

2. Examine alpha diversity as function of soil and plant traits

- calculate observed richness (you know how!)
  - calculate on the untransformed data
  - note that if you use `phyloseq::estimate_richness` it may add an "X" prefix to your row names
  - replace row names before merging richness into the sample_data file
  - merge observed richness into the `env_new`
- define a model that includes plant and soil variables you think will be important
  - if needed, transform richness (for heteroskedastic or non-normal data)
  - if needed, remove highly collinear variables (or combine them into a single variable)
- select a regression approach and explain your choice
- plot and interpret results
- save ps_vst to use next week once you add richness to the sample data
  - add the updated `env_new` to `sample_data(ps_vst)`
  - use grid view in the environment pane, select ps_vst, and hit save icon
  - save as `wk11_data.Rdata`

```
# calculate richness
alpha <- phyloseq::estimate_richness(ps_bac, measures="Observed")
alpha

names_a <- rownames(env_new)
rownames(alpha) <-  names_a

# check that rownames match
all(rownames(alpha) == rownames(env_new))

# merge and replace rownames lost in merge
env_new_a <- merge(env_new, alpha, by="row.names") %>%
          tibble::column_to_rownames(var = "Row.names")
```

```r
env_new_a[1,]

phyloseq::sample_data(ps_vst) <- phyloseq::sample_data(env_new_a)

# save ps_vst to use next week!
# use the grid in the environment pane, select ps_vst, and hit save
# save as "wk11_data.Rdata"

colnames(phyloseq::sample_data(ps_vst))
# example model
model_a <- lm(Observed ~Mean + Lat + SLA.F , data = env_new_a)

# check assumptions - normality questionable, but proceed
olsrr::ols_test_normality(model_a)
olsrr::ols_test_breusch_pagan(model_a, rhs=TRUE, multiple=TRUE)
olsrr::ols_coll_diag(model_a)
# mixed normality results between S-W and A-D, so we'll leave it as is
# other assumptions okay

# run best subsets (or other) regression
a_best <- olsrr::ols_step_best_subset(model_a)
a_best
# not a great fit! (poor hypotheses on my part)

# example plot top results
ggplot(env_new_a, aes(y=Observed, x=Lat)) + geom_point()+geom_smooth(method="lm")
ggplot(env_new_a, aes(y=Observed, x=Mean)) + geom_point()+geom_smooth(method="lm")+ xlab("Mean Water Tab
```

3. Rerun the LASSO regression on a different plant trait

- change the P value cutoff (e.g., 0.05 or 0.0001) and subset the data
- select a new plant trait (see `Wk11_README.txt`)
  - use `tidyr::drop_na` to remove rows with NA in your chosen variable
- rerun `glmnet::lasso`
- plot top results
- explain what you found

```
# use relaxed P value criteria
res05 <- subset(resOrdered, pvalue < 0.05)
res05
names05 <- rownames(res05)

# subset ps_vst by diff exp taxa
ps_vst
allTaxa <- taxa_names(ps_vst)
allTaxa05 <- allTaxa[(allTaxa %in% names05)]
ps_sig05 <- prune_taxa(allTaxa05, ps_vst)
ps_sig05

# prepare data
# env
env_05 <- sample_data(ps_sig05)

# otu - retrieve and autoscale (center by subtracting mean and scale by /sd)
otu_05 <- otu_table(ps_sig05)
otu_05_as <- scale(otu_05, center=TRUE, scale=TRUE) %>% as.data.frame()

# check
all(rownames(env_05) == rownames(otu_05_as))

# merge
ENVOTU05 <- merge(env_05, otu_05_as, by = "row.names", all=TRUE) %>% as.data.frame()

# example with RGR plant trait - remove NAs
ENVOTU05rgr <- drop_na(ENVOTU05, RGR)

# Prep train and test data
set.seed(20220316)
train05 <- ENVOTU05rgr %>% dplyr::sample_frac(0.8) # use 80% of data for training and 20% for testing
test05 <- dplyr::anti_join(ENVOTU05rgr, train05, by = "Row.names")
train05 <- column_to_rownames(train05, "Row.names")
test05 <- column_to_rownames(test05, "Row.names")

x_train05 <- train05 %>% dplyr::select(starts_with("OTU"))
x_train05 <- as.matrix(x_train05)
x_test05 <- test05 %>% dplyr::select(starts_with("OTU"))
x_test05 <- as.matrix(x_test05)
```

```
# Plant height (H)
# assign x and y for train and test data
y_train05 <- train05$RGR
y_test05 <- test05$RGR

anyNA(y_test05)
anyNA(y_train05)

set.seed(20220317)

# use cross-validation (leave-one-out) to select lambda minimum
# can take a few minutes to run
cvfit.RGR <- glmnet::cv.glmnet(x_train05, y_train05,
                        alpha = 1, nfolds = nrow(x_train05), grouped=FALSE)

# plot lambda vs. MSE to examine where error is minimized
plot(cvfit.RGR)
# plot coefficients for each OTU against lambda
plot(cvfit.RGR$glmnet.fit, xvar="lambda", label=TRUE)

# lambda minimum
cvfit.RGR$lambda.min
lam_min <- cvfit.RGR$lambda.min
coef(cvfit.RGR, s=lam_min)

# fit model with lambda minimum
best.RGR <- glmnet::glmnet(x_train05, y_train05, alpha = 1, lambda = lam_min)
best.RGR

# assess model fit on test data
assess.RGR <- glmnet::assess.glmnet(best.RGR, newx = x_test05, newy = y_test05)
assess.RGR

# examine results: plot otu abundance vs. rgr for otus selected by LASSO (only 3 in this case)
p110 <- ggplot2::ggplot(ENVOTU05rgr, aes(y=RGR, x=OTU_110)) +
  geom_point(aes(colour=Treatment)) +
  geom_smooth(method="lm")

p223 <- ggplot2::ggplot(ENVOTU05rgr, aes(y=RGR, x=OTU_223)) +
  geom_point(aes(colour=Treatment)) +
  geom_smooth(method="lm")

p285 <- ggplot2::ggplot(ENVOTU05rgr, aes(y=RGR, x=OTU_285)) +
  geom_point(aes(colour=Treatment)) +
  geom_smooth(method="lm")

ggpubr::ggarrange(p110, p223, p285, ncol=3, nrow=1, common.legend = TRUE)
```