STAT 113: Regression "Mini-Project"

Last Revised November 21, 2017

Overview

The goal of this lab is to carry out the process of choosing, fitting, assessing, and using a regression model from beginning to end, documenting the process and writing about the results in a reproducible format (in our case, RMarkdown). I have not provided step-by-step instructions on what lines of code to use, though I have included a quick-reference guide at the end for the most important R commands.

What to Turn In

You should turn in a Markdown report documenting your exploration and conclusions by **Friday, December 1st**. Your report should not just be a bunch of code: *explain* what you are doing at each step in text, and *interpret* the output and results. You should aim for the report to look more like a (short) research paper, and less like an electronic lab notebook.

I have provided reference material on the following pages for essential R commands.

SLOs for this Mini-Project

This mini-project assesses the following SLOs: D1, D2, E2, E7, E8, F1, G1, G4, H1, H2, H4, H5, J6

Instructions

There is a dataset on my website with several macro-level variables on 100 randomly selected world countries, stored in the file `CountryData.csv`, and a description of what the variables mean in a companion file called the "code book". You can view the code book in your browser. *Make sure you understand what variables mean before using them in a regression model!*

Data URL: `http://colinreimerdawson.com/data/CountryData.csv`

Code book URL: `http://colinreimerdawson.com/data/CountryData-CodeBook.txt`

Your goal is to try to model the relationship between the life expectancy of a country and (some of) the other variables in the data. We have not yet discussed models with multiple predictors, so for now your models will be limited to one predictor at a time. However, you may want to consider transformation of the response variable and/or the explanatory variable, as well as "composite" variables that are computed using more than one of the other variables (for example, a ratio).

What to Include (at a Minimum)

1. Plots of the data for the predictors under consideration (start with four or five predictors that you want to investigate), together with the accompanying linear model.
2. Diagnostic plots that allow you to assess modeling conditions, detect outliers, etc., along with a verbal interpretation of what they show.
3. After applying transformations as needed to meet regression conditions and possibly considering composite variables (e.g., ratios of two other variables), select a handful (about three or so) of "finalist" models, and give mathematical statements in the $Y = \beta_0 + \beta_1 \cdot X + \varepsilon$ form (where X may be a variable name, a transformation of a variable name, a composite expression, etc., and the βs are filled in with numbers). Interpret the coefficients in text.
4. Confidence interval and hypothesis test of the slopes of your "finalist" models (describe in text; not just by referring to computer output, and interpret what the slope tells us).
5. R^2 for your finalists (describe in text; not just by referring to computer output)
6. Plot of confidence and prediction bands for finalists with a text interpretation of what they say (for example, pick an X value and describe what the corresponding intervals are and what they mean).
7. An overall interpretation of the results.

Essential R commands

(Most of the following require `mosaic`)

Case selection / defining new variables (e.g., ratio of two others)

```
FilteredData <- filter(DataName, CONDITION)
NewDataName <-
    mutate(DataName, NewVariableName = <some function of another variable>)
##   Note: CONDITION is an expression like
##     `Meltdown == 0`, or `Height >= 60`, or
##     `Color == "red" & Size < 100` (without the quotes)
```

Scatterplots

```
xyplot(Y ~ X, data = DataName, type = c("p","r"))
##   Note: the groups= argument will create different plotting
##     symbols for different levels of the categorical variable Z.
##     Can also create a binary (TRUE/FALSE) variable on the fly by
##     doing something like: groups = (Z <= 100).
##     auto.key = TRUE displays a legend showing the symbol meanings
```

Fitting, describing and assessing a regression model

```
## Fit (can also apply transformations, e.g., log(Y) ~ X)
my.model <- lm(Y ~ X, data = DataName)
## T-tests of slope, and other miscellaneous output
summary(my.model)
## To get LEVEL confidence intervals for the intercept
## and slope
confint(my.model, level = LEVEL)
## Key Residual plots
plot(my.model, which = 1)  ## fitted by residuals
plot(my.model, which = 2)  ## quantile-quantile plot
histogram(~residuals(my.model), data = DataName)
## Getting parts of the model as variables
## (Note: can also use these in plots, e.g., histogram())
coef(my.model); residuals(my.model); fitted.values(my.model)
```

Using the model (after fitting):

```
## Create a prediction function
f.hat <- makeFun(my.model)

## Plot the line over the data, or, if the predictor
## is transformed, the curve on the original scale
plotModel(my.model)

## Get the predicted Y value when X = 3
# replace X with the variable name and 3 with a value
f.hat(X = 3)
## with LEVEL confidence/prediction intervals
f.hat(X = 3, interval = "confidence", level = LEVEL)
f.hat(X = 3, interval = "prediction", level = LEVEL)

## Plot the data with LEVEL confidence and prediction bands
xyplot(Y ~ X, data = DataName, panel = panel.lmbands, level = LEVEL)
```