Example of Function Use

Peter Boyd

11/26/2018

First, load in a few data files and packages.

```
library(tidyverse)
library(here)
library(MASS)
library(broom)
library(rsample)
whale <- read_csv(here("data", "sperm_whale_pop.csv"))
addax <- read_csv(here("data", "addax.csv"))
grizzly_ukraine <- read_csv(here("data", "grizzly_ukraine.csv"))
ringed_seal <- read_csv(here("data", "ringed_seal_finland.csv"))
```

Next, we write our function.

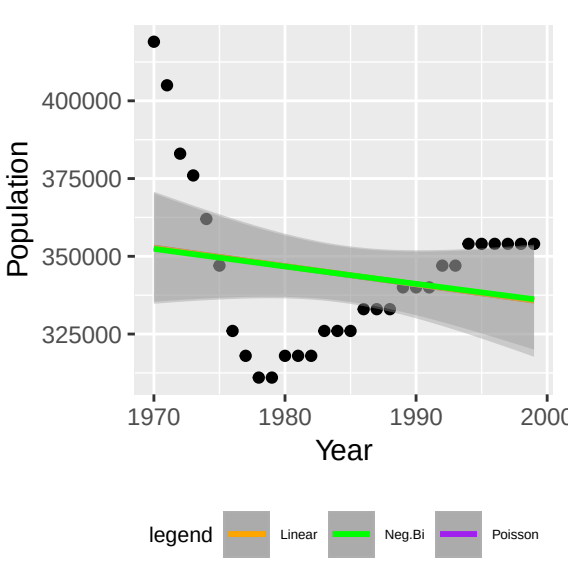
```
pop.decline <- function(df, ntimes){
  #Fit several models
  df <- na.omit(df)
  fit.glm <- glm(df$Population ~ df$Year, data = df)
  fit.poi <- glm(df$Population ~ df$Year, data = df, family = "poisson")
  fit.nb <- glm.nb(df$Population ~ df$Year, data = df)
  fits <- list((fit.glm), (fit.poi), (fit.nb))
  #####

  #Compare AIC
  mods <- c("Linear Model", "Poisson", "Negative Binomial")
  aics <- c(summary(fit.glm)$aic, summary(fit.poi)$aic, summary(fit.nb)$aic)
  best.model <- fits[[order(aics)[1]]]
  aic.summary <- cat("AIC for Each Model", "\n",
                     mods[1], ": ", aics[1], "\n",
                     mods[2], ": ", aics[2], "\n",
                     mods[3], ": ", aics[3], "\n")
  #####

  # Do simulation for lowest aic
  # First, select best model and simulate accordingly
  # Then, fit models to each simulation
  # Finally, average the models to obtain an approximate
  # Estimate of extinction date
  n <- max(df$Year) - min(df$Year) +1
  y.hat <- best.model$fitted.values

  if(order(aics)[1] == 2){
    y.rep <- matrix(ncol = ntimes, data = c(rpois(n*ntimes, y.hat)))
  } else if(order(aics)[1] == 3){
    y.rep <- matrix(ncol = ntimes, data = c(rnegbin(n = n*ntimes, mu = y.hat,
                                                   theta = summary(fit.nb)$theta)))
  } else {
    new_data <- data.frame("Year" = seq(max(df$Year)+1, max(df$Year) + n*ntimes, 1))
```

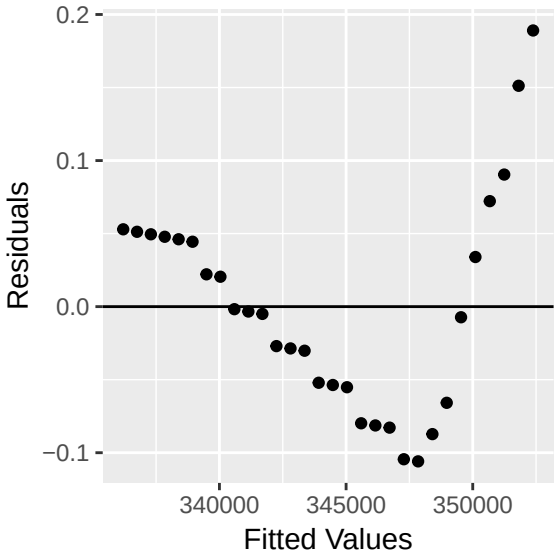
```
    preds <- predict(fit.glm, newdata = new_data, type = 'response')
    preds <- preds + runif(n = n*ntimes, min = min(fit.glm$residuals), max =
                             max(fit.glm$residuals))
    y.rep <- matrix(ncol = ntimes, data = preds)
  }

  years <- c(min(df$Year):max(df$Year))
  y.rep.sort <- -apply(-y.rep, 2, sort)
  y.coefs <- t(apply(y.rep.sort, 2, function(y.col) lm(y.col~years)$coef))
  end.dates <- -y.coefs[,1] / y.coefs[,2]
  end.date <- cat("Predicted Extinction: ", mean(end.dates))
  #####

  #Create plot with each model fit
  plot1 <- ggplot(data = df, aes(x = Year, y = Population)) +
    geom_point() +
    geom_smooth(method='glm',formula=y~x, method.args = list(family = "poisson"),
                aes(color = "Poisson")) +
    geom_smooth(method='glm',formula=y~x, aes(color = "Linear")) +
    geom_smooth(method='glm.nb',formula=y~x, aes(color = "Neg.Bi")) +
    scale_colour_manual(name="legend", values=c("orange", "green", "purple")) +
    labs(x = "Year", y = "Population") +
    theme(legend.position="bottom") +
    theme(legend.text = element_text(size = 5)) +
    theme(legend.title = element_text(size = 8))
  #####

  #Create residual plot for lowest aic model
  resid.plot <- qplot(best.model$fitted.values, best.model$residuals) +
    geom_hline(yintercept=0) +
    labs(x = "Fitted Values", y = "Residuals")
  #####

  #Create function to neatly display plots
  print_plots <- function(x, y){
    par(mfrow = c(2,2))
      print(x)
      print(y)
  }
  #####

  #Return aics, extinction, plots
  invisible(list(aic.summary, end.date, print_plots(plot1, resid.plot)))
}
```

Finally, we can play with our function.

```
pop.decline(df = whale, ntimes = 100)
```

```
## AIC for Each Model
##  Linear Model :  698.5794
##  Poisson :  53225.4
##  Negative Binomial :  697.0227
## Predicted Extinction:  2110.536
```






```
# pop.decline(df = addax, ntimes = 100)
# pop.decline(df = grizzly_ukraine, ntimes = 100)
# pop.decline(df = ringed_seal, ntimes = 100)
```