

How to Automate Summary Stats in Markdown

Zach Fogel and ChatGPT

2024-04-16

R Markdown

```
# load model
model <- readRDS('Input/wlv-top-model.RData')
# load occupancy data (also includes explanatory variables)
occ <- readRDS('Input/wlv-occ.RData') %>% filter(buffer == 2000)

# Tidy model outputs, add confidence intervals
maketidymod <- function(model) {
  broom::tidy(model) %>%
    mutate(confintlow = estimate - 1.96*std.error,
           confinhigh = estimate + 1.96*std.error) %>%
    rowwise() %>%
    # calculate different measures of significance
    mutate(sigCI = !(0 %btwn% c(confintlow, confinhigh)),
           sig.p = p.value < 0.05)
}

printstats <- function(modelsumm) {
  b <- modelsumm$estimate %>% round(3)
  z <- modelsumm$statistic %>% round(3)
  p <- modelsumm$p.value %>% round(3)
  ci1 <- modelsumm$confintlow %>% round(3)
  ci2 <- modelsumm$confinhigh %>% round(3)
  # \\beta outputs as the letter beta
  # %.3f means to use 3 decimal places
  # you can change the values that are included
  message <- sprintf("\\(\\beta = %.3f\\), \\(z = %.3f\\), \\(p = %.3f\\), 95% CI = [%.3f, %.3f]", b, z, p, ci1, ci2)
  return(message)
}

modelsumm <- maketidymod(model)
stats <- lapply(modelsumm$term, function(x) {
  tmp <- filter(modelsumm, term == x)
  data.frame(term = x, stats = printstats(tmp))
}) %>% bind_rows()
```

For wolverines, the top model included linear disturbance and elevation at a m buffer:

$$\text{Pres} = \beta_0 + \beta_1(\text{lin}) + \beta_2(\text{elev.sd}) + \beta_3(\text{elev.med}) + \beta_4(\text{days})$$

. Linear disturbance ($\beta = -54.622$, $z = -2.513$, $p = 0.013$, 95% CI = [-97.224, -12.019]) and elevation SD ($\beta = -0.003$, $z = -2.722$, $p = 0.007$, 95% CI = [-0.005, -0.001]) had significant effects on wolverine presence.

Chatgpt can probably explain this better, but basically:

- The “`\[[equation] \]`” means that it’s going to be a latex equation block (you probably knew that)
- `r` means “this is going to be an R chunk”
- “`filter(stats, term == ‘lin’)[,2]`” is just referring back to the stats object.