Owen et al. - Figure 2

This script will reproduce Figure 2 from *Owen et al. - Sample size planning for insect critical thermal limits studies.*

Warning: This script takes 5 - 10 minutes to run on my PC.

Load required packages

```
if (!require("pacman")) install.packages("pacman")
pacman::p_load(tidyverse,
               tidyr,
               readr,
               data.table,
               viridis,
               here)
```

Set ggplot theme (makes nice plots)

```
theme_set(theme_classic() +
            theme(panel.border = element_rect(colour = "black", fill = NA),
                  axis.text = element_text(colour = "black"),
                  axis.title.x = element_text(margin = unit(c(2, 0, 0, 0), "mm")),
                  axis.title.y = element_text(margin = unit(c(0, 4, 0, 0), "mm")),
                  legend.position = "none"))
```

Load bootstrap data

We have already drawn our bootstrap samples, load that data. This is the output from script-01.

```
# Load raw data (with CI's)
boot_tci <- readr::read_csv(here::here("./data_clean/ct_min_bootstrap_with_ci.csv"))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   insect_sp = col_character(),
##   sample_size = col_double(),
##   iter = col_double(),
##   mean = col_double(),
##   sd = col_double(),
##   median_pop_val = col_double(),
##   std_error = col_double(),
```

```
##   error = col_double(),
##   lower_ci = col_double(),
##   upper_ci = col_double()
## )
```

```
# Make insect_sp column into a factor
boot_tci <- boot_tci %>%
  dplyr::mutate(insect_sp = as.factor(insect_sp))
```

Perform analysis

We have already calculated median CTmin values per species (median for n = 30; max. sample size in our study) in script 01.Following Pearson and Groves (2013), we assumed that our maximum sample size (n = 30) is a reasonable approximation of the actual critical thermal limit value (population parameter) i.e. if we hypothetically had sampled the entire population. Obviously, we must interpret these data with caution, as there is no possible way to 100% accurately determine the population parameter, so we must derive an *ESTIMATE*.

'Coverage' refers to the proportion of bootstrap resamples for which the estimated population parameter falls within the bounds of a 95% confidence interval. Coverage provides an estimate of parameter accuracy.

```
# Calculate the proportion of times the median_pop_val falls
# within the lower_ci and upper_ci bounds (95%  CI)
bootstrap_cover <- boot_tci %>%
  dplyr::group_by(insect_sp, sample_size, iter) %>%
  dplyr::mutate(ci_falls = dplyr::case_when(
    median_pop_val < lower_ci ~ 0,
    median_pop_val > upper_ci ~ 0,
    median_pop_val > lower_ci & median_pop_val < upper_ci ~ 1,
    median_pop_val < upper_ci & median_pop_val > lower_ci ~ 1
  )) %>%
  dplyr::group_by(insect_sp, sample_size) %>%
  dplyr::mutate(n_correct = sum(ci_falls),
                max_n = max(iter),
                prop_correct = n_correct/max_n) %>%
  dplyr::slice(1)
```

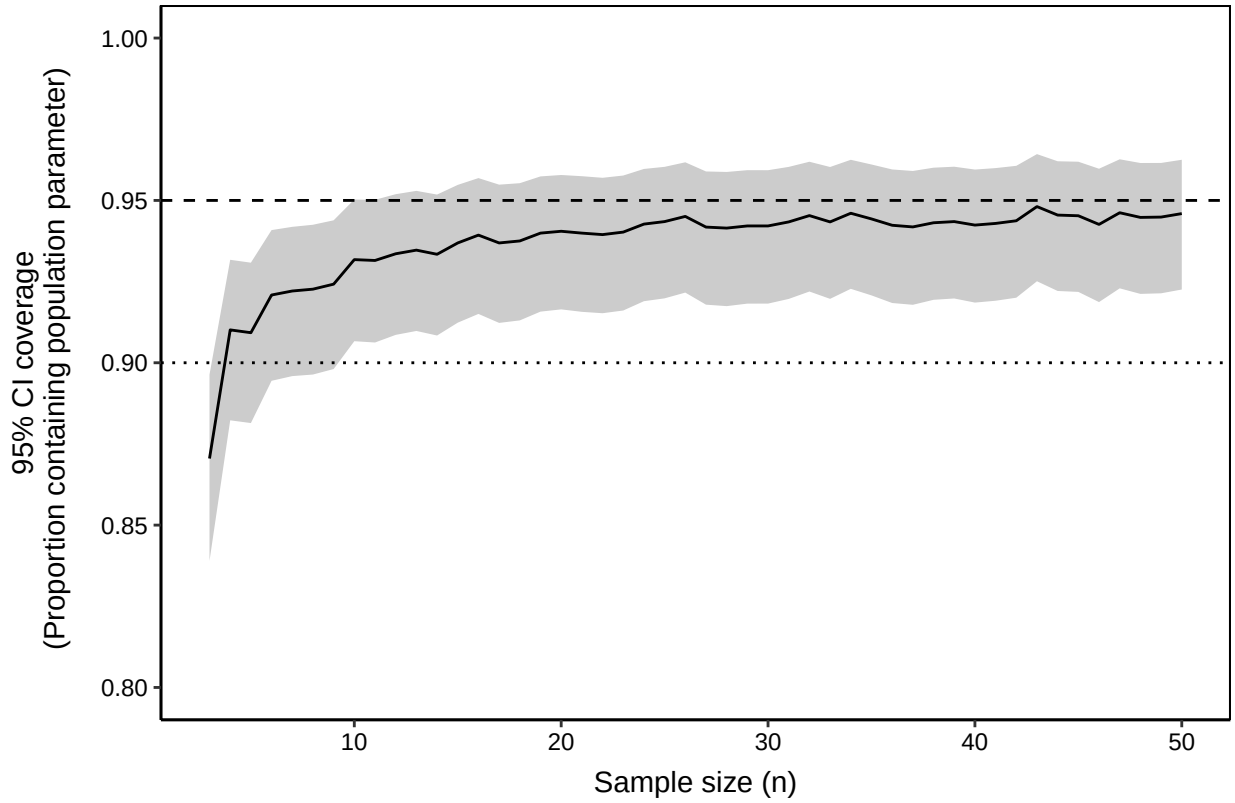
```
# Manually calculate binomial confidence intervals
bin_prop_ci <- as_tibble(cbind(bootstrap_cover,
                               Hmisc::binconf(bootstrap_cover$n_correct,
                                              bootstrap_cover$max_n)))

# Extract binconf output
aa <- as_tibble(bin_prop_ci$...15)

# Add binconf output back to bootstrap samples
bin_prop_ci <- dplyr::bind_cols(bin_prop_ci, aa) %>%
  dplyr::select(insect_sp,
                sample_size,
                prop_correct = PointEst,
                lower_ci = Lower,
                upper_ci = Upper)
```

Make figure

```
# Plot relationship - averaged across species (focused y axis)
bin_prop_ci %>%
  dplyr::group_by(sample_size) %>%
  dplyr::summarise(prop_correct = mean(prop_correct),
                   lower_ci = mean(lower_ci),
                   upper_ci = mean(upper_ci)) %>%
  ggplot(data = ., aes(x = sample_size,
                       y = prop_correct)) +
  geom_ribbon(aes(ymin = lower_ci,
                  ymax = upper_ci),
              fill = "gray80") +
  geom_line() +
  labs(x = "Sample size (n)",
       y = "95% CI coverage\n (Proportion containing population parameter)",
       subtitle = " ") +
  scale_y_continuous(breaks = c(0.8, 0.85, 0.9, 0.95, 1.00),
                     limits = c(0.8, 1)) +
  geom_hline(yintercept = 0.95, linetype = "dashed") +
  geom_hline(yintercept = 0.90, linetype = "dotted")
```




```
# Save figure to file
ggplot2::ggsave(here::here("./figures/figure_2.png"),
```

```
                dpi = 600,
                width = 6,
                height = 6)
```