Fall 2024 Practical Computing Final Project

Mic Schulte

2024-12-05

Petrolisthes armatus data analysis

Loading libraries

```
library(maps)
library(ggplot2)
library(sf)
```

```
## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```
library(rnaturalearth)
library(rnaturalearthdata)
```

```
##
## Attaching package: 'rnaturalearthdata'

## The following object is masked from 'package:rnaturalearth':
##
##     countries110
```

```
library(rgbif)
library(terra)
```

```
## terra 1.7.78
```

```
library(geodata)
library(sdmpredictors)
library(stringr)
library(raster)
```

```
## Loading required package: sp
```

```
library(ggspatial)
library(stars)
```

```
## Loading required package: abind
```

```r
library(elevatr)
```

```
## elevatr v0.99.0 NOTE: Version 0.99.0 of 'elevatr' uses 'sf' and 'terra'.  Use
## of the 'sp', 'raster', and underlying 'rgdal' packages by 'elevatr' is being
## deprecated; however, get_elev_raster continues to return a RasterLayer.  This
## will be dropped in future versions, so please plan accordingly.
```

```r
library(tigris)
```

```
## To enable caching of data, set `options(tigris_use_cache = TRUE)`
## in your R script or .Rprofile.
```

```
##
## Attaching package: 'tigris'
```

```
## The following object is masked from 'package:terra':
##
##     blocks
```

```r
library(ggsflabel)
```

```
##
## Attaching package: 'ggsflabel'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     geom_sf_label, geom_sf_text, StatSfCoordinates
```

```r
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following object is masked from 'package:raster':
##
##     shift
```

```
## The following object is masked from 'package:terra':
##
##     shift
```

```r
library(readxl)
library(ggrepel)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:raster':
##
##     intersect, select, union

## The following objects are masked from 'package:terra':
##
##     intersect, union

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

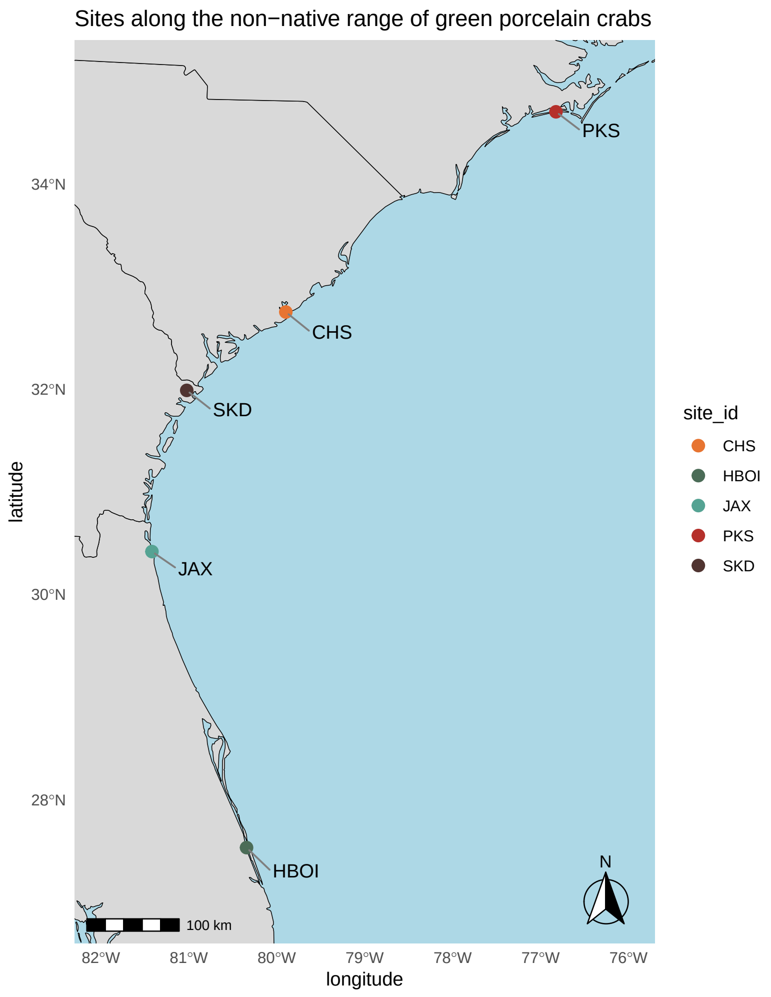
Read in data

```
petro <- read_excel("~/Desktop/R/in_progress/mock_petro_dataset.xlsx")
```

```
## New names:
## * `date` -> `date...4`
## * `date` -> `date...7`
```

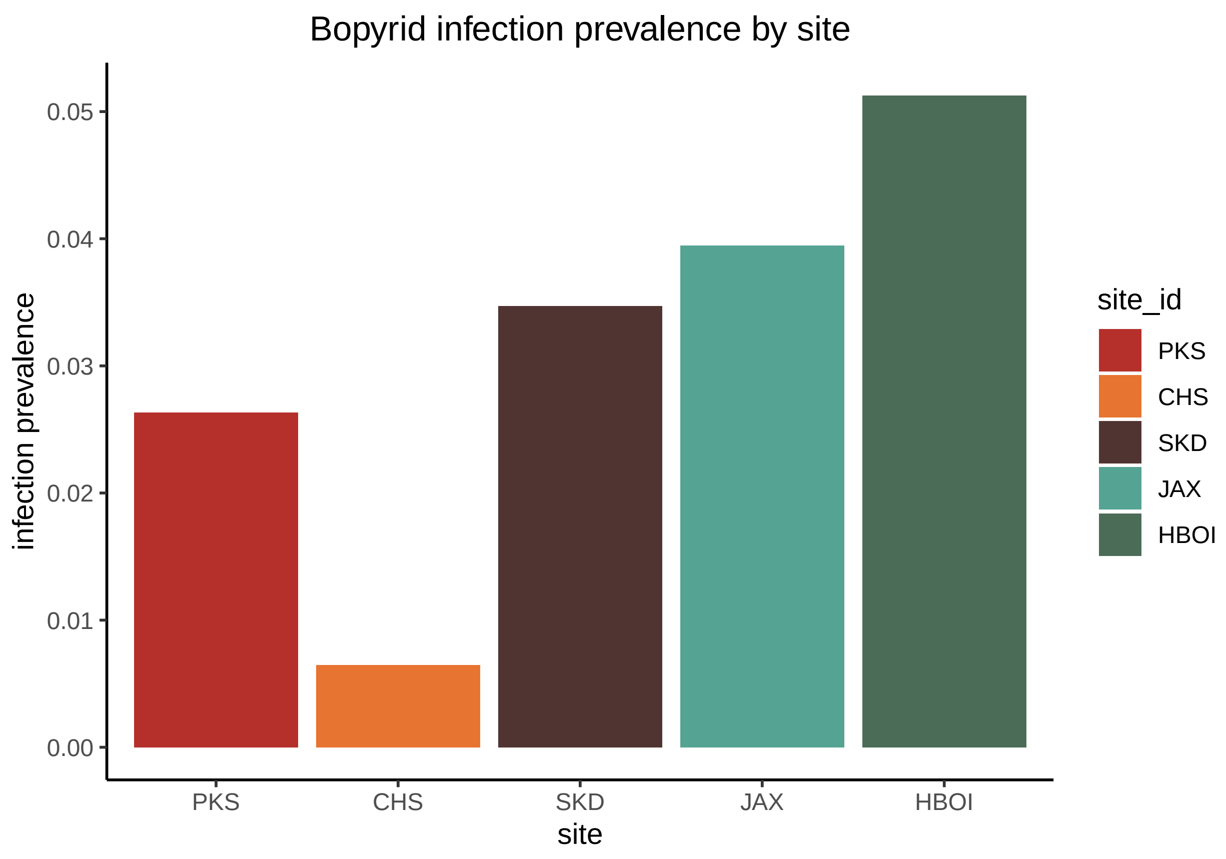
Creating a map of sites from south Florida through mid-coast North Carolina (geospatial mapping)

```
# Get state boundaries from rnaturalearth
petro_sites_states <- ne_states(country = "united states of america", returnclass = "sf")

# Filter data to include site locations
site_locations <- petro %>%
  dplyr::select(site_id, latitude, longitude) %>%
  distinct()

# Incorporate coordinates from data set
site_locations$site_id <- as.factor(site_locations$site_id)

# Plot non-native range map & sites as points
ggplot() +
  geom_rect(aes(xmin = -86, xmax = -74, ymin = 22, ymax = 38), fill = "lightblue",
            color = NA) +
  geom_sf(data = petro_sites_states, fill = "gray85", color = "black", size = 0.3) +
  coord_sf(xlim = c(-82, -76), ylim = c(27, 35)) +
  geom_point(data = site_locations, aes(x = longitude, y = latitude, color = site_id),
             size = 3) +
  scale_color_manual(values = c("PKS" = "#b5302b", "CHS" = "#e77431",
                               "SKD" = "#503431", "JAX" = "#55a393",
                               "HBOI" = "#4b6c57")) +
  geom_text_repel(data = site_locations, aes(x = longitude, y = latitude,
                                             label = site_id),
                  box.padding = 1, point.padding = 0.15,
                  nudge_y = -0.10, nudge_x = 0.4,
                  direction = "both",
                  segment.color = 'grey50') +
  labs(title = ("Sites along the non-native range of green porcelain crabs"),
       x = "longitude", y = "latitude") +
  annotation_scale(location = "bl", width_hint = 0.3) +
  annotation_north_arrow(location = "br", which_north = "true",
                         style = north_arrow_fancy_orienteering) +
  theme_minimal() +
  theme()
```



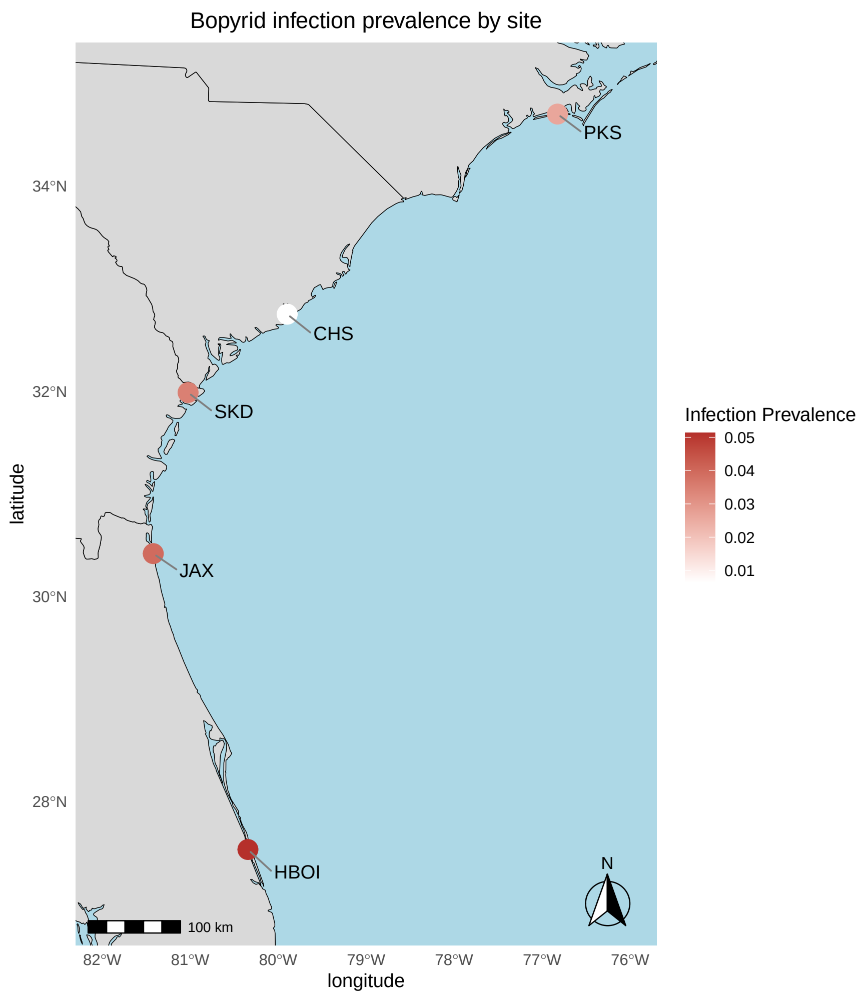
Determining parasite infection prevalence along non-native range (bar plot)

```r
petro$site_id <- factor(petro$site_id, levels = c("PKS", "CHS", "SKD", "JAX", "HBOI"))

# Calculate bopyrid infection prevalence by site
prevalence_by_site <- petro %>%
  group_by(site_id) %>%
  summarise(prevalence = sum(inf_status == 1) / n())

# Plot of infection prevalence by site
ggplot(prevalence_by_site, aes(x = site_id, y = prevalence, fill = site_id)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("PKS" = "#b5302b", "CHS" = "#e77431",
                               "SKD" = "#503431", "JAX" = "#55a393",
                               "HBOI" = "#4b6c57")) +
  labs(title = "Bopyrid infection prevalence by site",
       x = "site",
       y = "infection prevalence") +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))
```



Plotting parasite infection prevalence by site on the original map (geospatial mapping)

```r
petro$site_id <- factor(petro$site_id, levels = c("PKS", "CHS", "SKD", "JAX", "HBOI"))

# Calculate bopyrid infection prevalence by site
prevalence_by_site <- petro %>%
  group_by(site_id) %>%
  summarise(prevalence = sum(inf_status == 1) / n())

# Join prevalence data with site data
site_locations <- site_locations %>%
  left_join(prevalence_by_site, by = "site_id")

# Plot sites as points with prevalence indicated by color brightness
ggplot() +
  geom_rect(aes(xmin = -86, xmax = -74, ymin = 22, ymax = 38), fill = "lightblue",
            color = NA) +
  geom_sf(data = petro_sites_states, fill = "gray85", color = "black", size = 0.3) +
  coord_sf(xlim = c(-82, -76), ylim = c(27, 35)) +
  geom_point(data = site_locations, aes(x = longitude, y = latitude, color = prevalence),
             size = 5) +
  scale_color_gradient(low = "white", high = "#b5302b", name = "Infection Prevalence") +
  geom_text_repel(data = site_locations, aes(x = longitude, y = latitude,
                                             label = site_id),
                  box.padding = 1, point.padding = 0.15,
                  nudge_y = -0.10, nudge_x = 0.4,
                  direction = "both",
                  segment.color = 'grey50') +
  labs(title = "Bopyrid infection prevalence by site",
       x = "longitude", y = "latitude", ) +
  annotation_scale(location = "bl", width_hint = 0.3) +
  annotation_north_arrow(location = "br", which_north = "true",
                         style = north_arrow_fancy_orienteering) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```



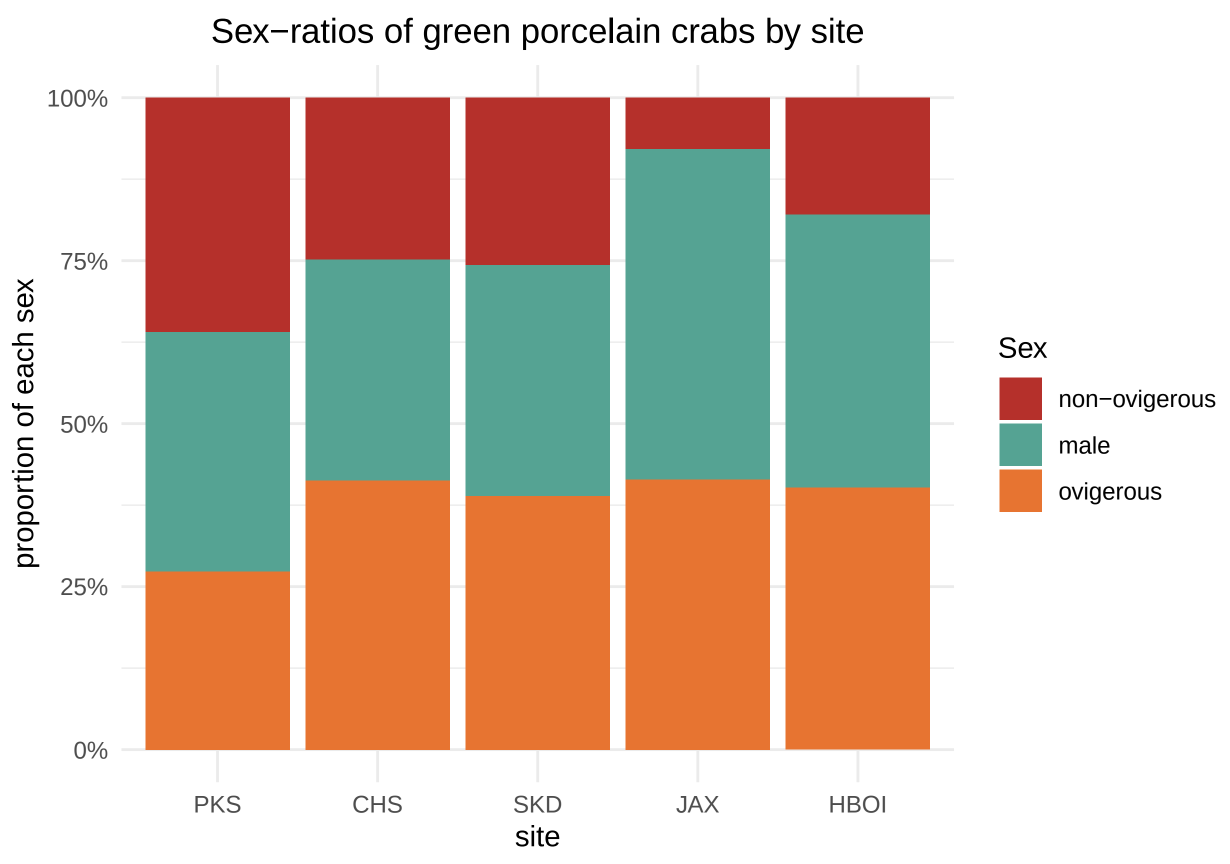
Calculating sex-ratios of porcelain crabs by site (stacked bar plot)

```
# Calcualte sex-ratio by site
sex_ratio <- petro %>%
  filter(sex != "J") %>% # Exclude juveniles
  group_by(site_id, sex) %>% # Group by site and sex
  summarise(count = n()) %>%
  mutate(proportion = count / sum(count)) # Calculate proportion of M, F, and O crabs
```

```
## ‘summarise()‘ has grouped output by ’site_id’. You can override using the
## ‘.groups‘ argument.
```

```
# Plot the sex-ratio by site as a stacked bar plot
ggplot(sex_ratio, aes(x = site_id, y = proportion, fill = sex)) +
  geom_bar(stat = "identity", position = "fill") +
  scale_y_continuous(labels = scales::percent_format()) +
  scale_fill_manual(
    values = c("F" = "#b5302b", "M" = "#55a393", "O" = "#e77431"),
    labels = c("non-ovigerous", "male", "ovigerous")
```

```r
  ) +
  labs(title = "Sex-ratios of green porcelain crabs by site",
       x = "site",
       y = "proportion of each sex",
       fill = "Sex") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

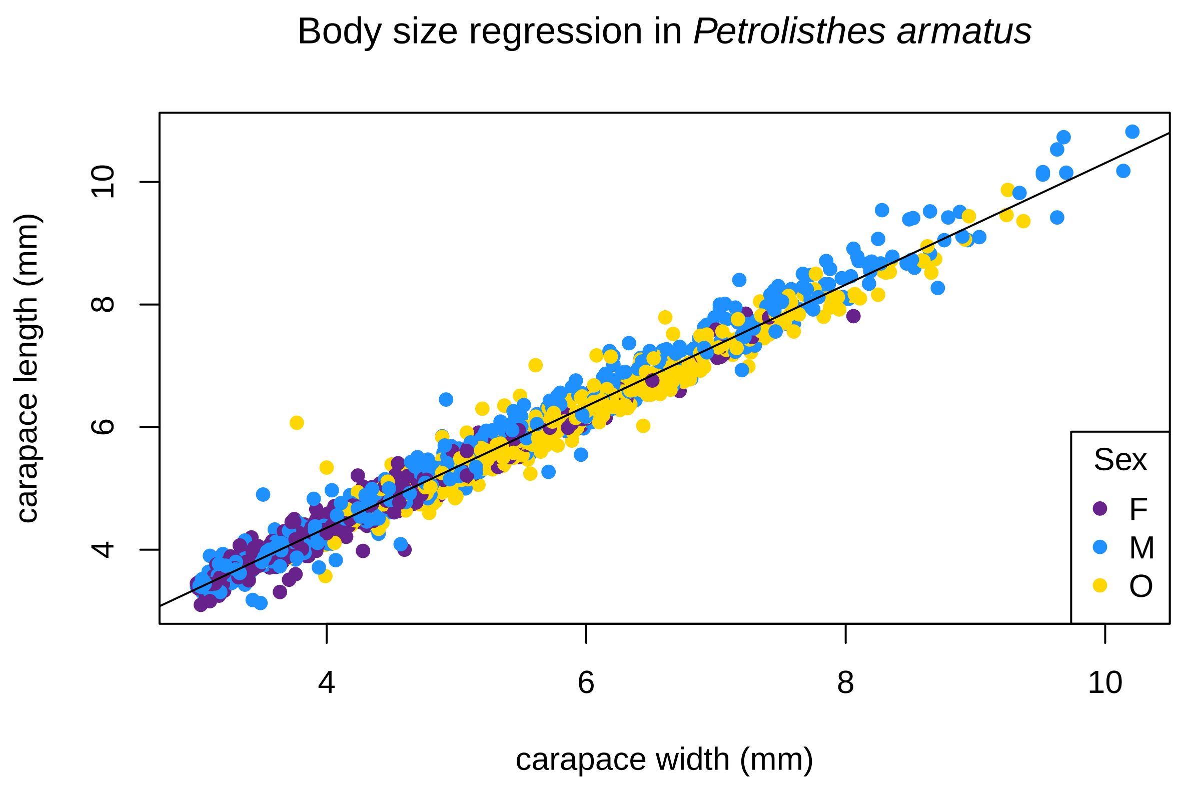


Plotting a regression model of body size measurements by sex (linear regression)

```r
# Set colors for each sex
sex_colors <- c("F" = "darkorchid4", "M" = "dodgerblue", "O" = "gold")

# Plot carapace length (cl_mm) ~ carapace width (cw_mm), colored by sex
plot(petro$cl_mm ~ petro$cw_mm, col = sex_colors[petro$sex], pch = 16,
     main = expression("Body size regression in " * italic("Petrolisthes armatus")),
     xlab = "carapace width (mm)",
     ylab = "carapace length (mm)")

# Add abline to lm
petro_size_lm <- lm(petro$cl_mm ~ petro$cw_mm)
coef(petro_size_lm)
```

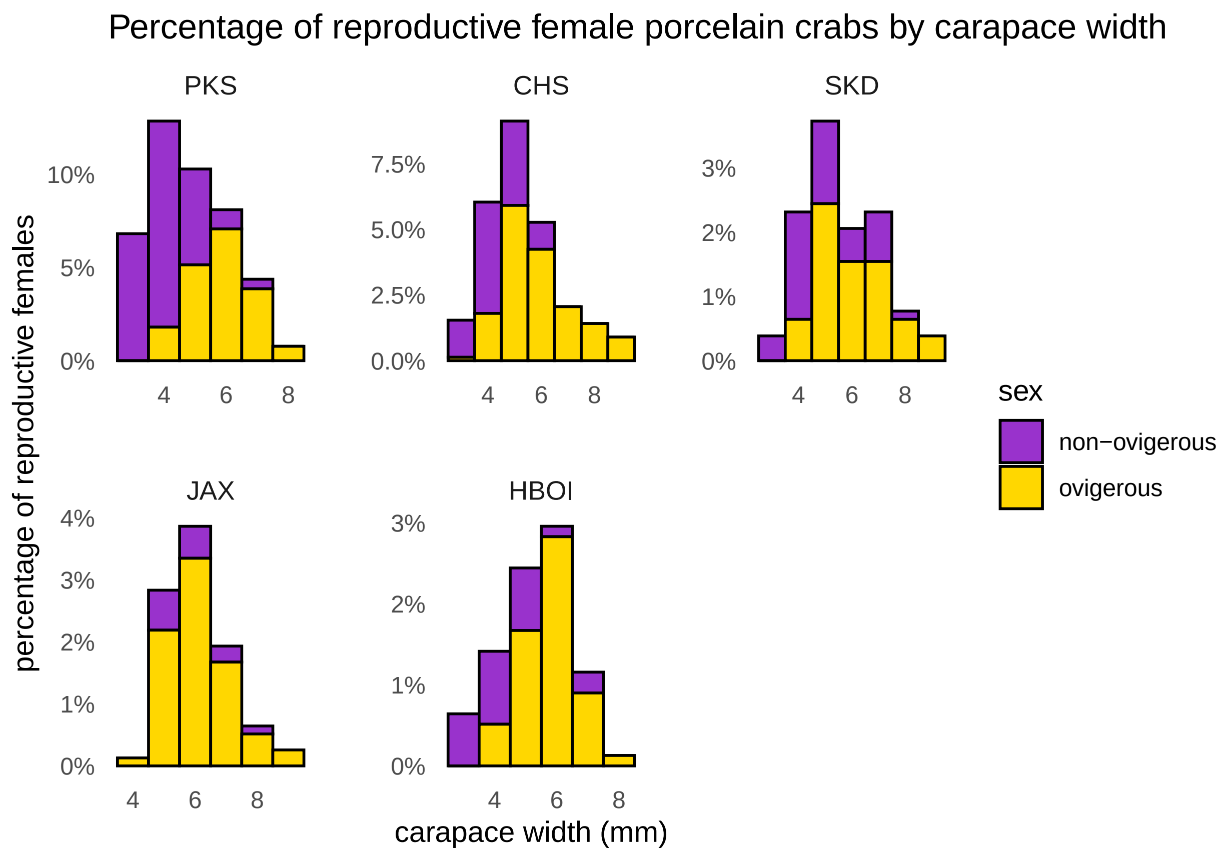
```
## (Intercept) petro$cw_mm
##   0.3877578   0.9920104
```

```
abline(petro_size_lm)

# Add a legend
legend("bottomright", legend = names(sex_colors), col = sex_colors,
       pch = 16, title = "Sex")
```



Calculating the proportion of ovigerous & non-ovigerous female crabs by body size (histogram)

```
# Define crabs sex as "O" or "F" are both female
female_crabs <- petro %>%
  filter(sex == "F" | sex == "O") %>%
   group_by(sex)

# Plot proportion of reproductive crabs in histogram
female_crabs %>%
  ggplot(aes(x = cw_mm, fill = sex)) +
  geom_histogram(aes(y = after_stat(count)/sum(after_stat(count))), binwidth = 1,
                 color = "black") +
  scale_y_continuous(labels = scales::percent) +
  facet_wrap(~ site_id, ncol = 3, scales = "free") +
  scale_fill_manual(
    values = c("F" = "darkorchid", "O" = "gold"),
    labels = c("non-ovigerous", "ovigerous")
  ) +
  labs(
    x = "carapace width (mm)",
    y = "percentage of reproductive females",
    title = "Percentage of reproductive female porcelain crabs by carapace width"
  ) +
  theme_minimal() +
  theme(
    strip.text = element_text(size = 10),
    panel.grid = element_blank(),
    panel.spacing = unit(1.5, "lines")
  )
```



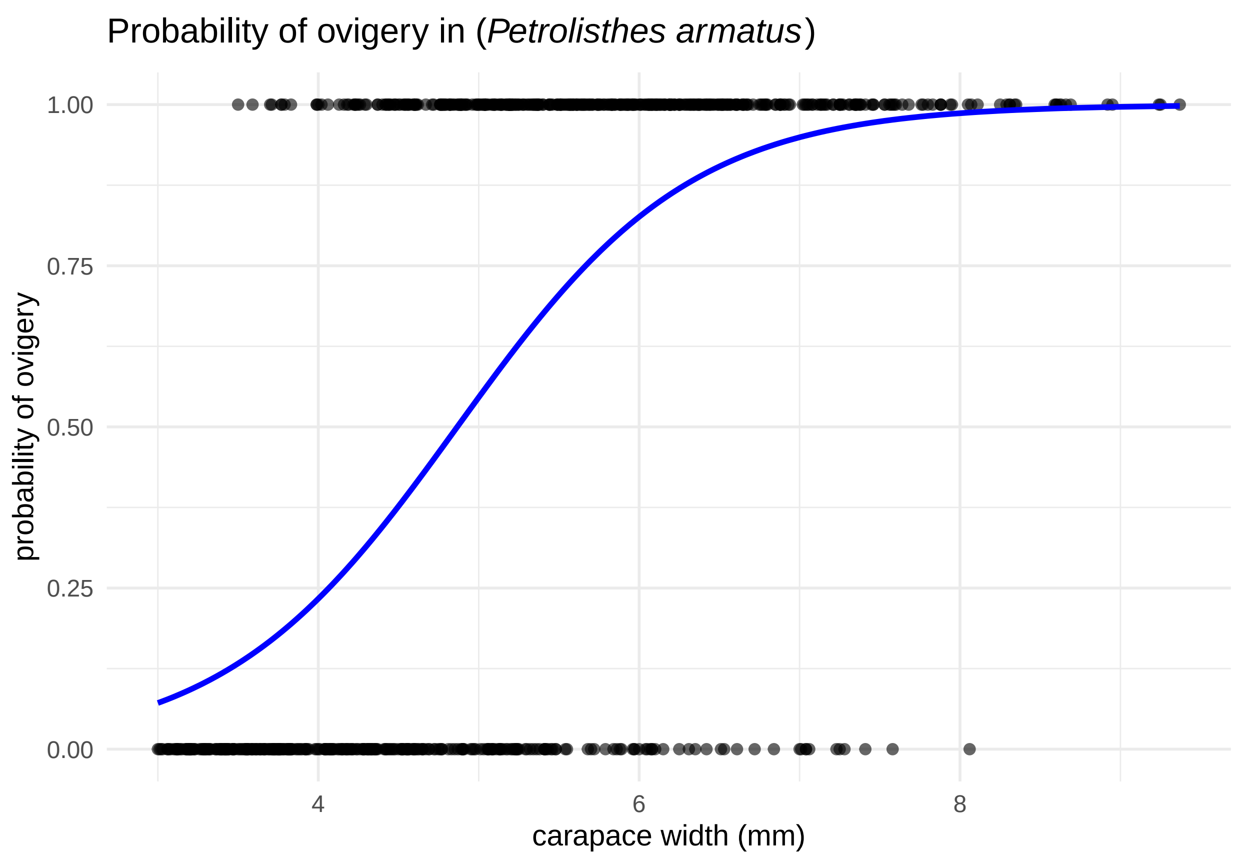
Determining the probability of ovigery by body size (glm)

```
# Filter dataset to include only "O" and "F" and make O or F binary
binary_female_crabs <- petro %>%
  filter(sex %in% c("O", "F")) %>%
  mutate(sex_binary = ifelse(sex == "O", 1, 0))

prob_of_ovig <- glm(sex_binary ~ cw_mm, data = binary_female_crabs, family = binomial)
summary(prob_of_ovig)
```

```
##
## Call:
## glm(formula = sex_binary ~ cw_mm, family = binomial, data = binary_female_crabs)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -6.6801     0.5056  -13.21   <2e-16 ***
## cw_mm         1.3728     0.1007   13.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1063.98  on 776  degrees of freedom
## Residual deviance:  742.47  on 775  degrees of freedom
## AIC: 746.47
##
## Number of Fisher Scoring iterations: 5
```

```r
# Combine data frame including female crab binary and probability of ovigery
crabs_filtered <- binary_female_crabs %>%
  mutate(predicted_prob = predict(prob_of_ovig, type = "response"))

# Plot data points and logistic curve
ggplot(crabs_filtered, aes(x = cw_mm, y = sex_binary)) +
  geom_point(alpha = 0.6) +
  geom_line(aes(y = predicted_prob), color = "blue", size = 1) +
  labs(
    x = "carapace width (mm)",
    y = "probability of ovigery",
    title =
      expression("Probability of ovigery in (" * italic("Petrolisthes armatus") * ")")
  ) +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



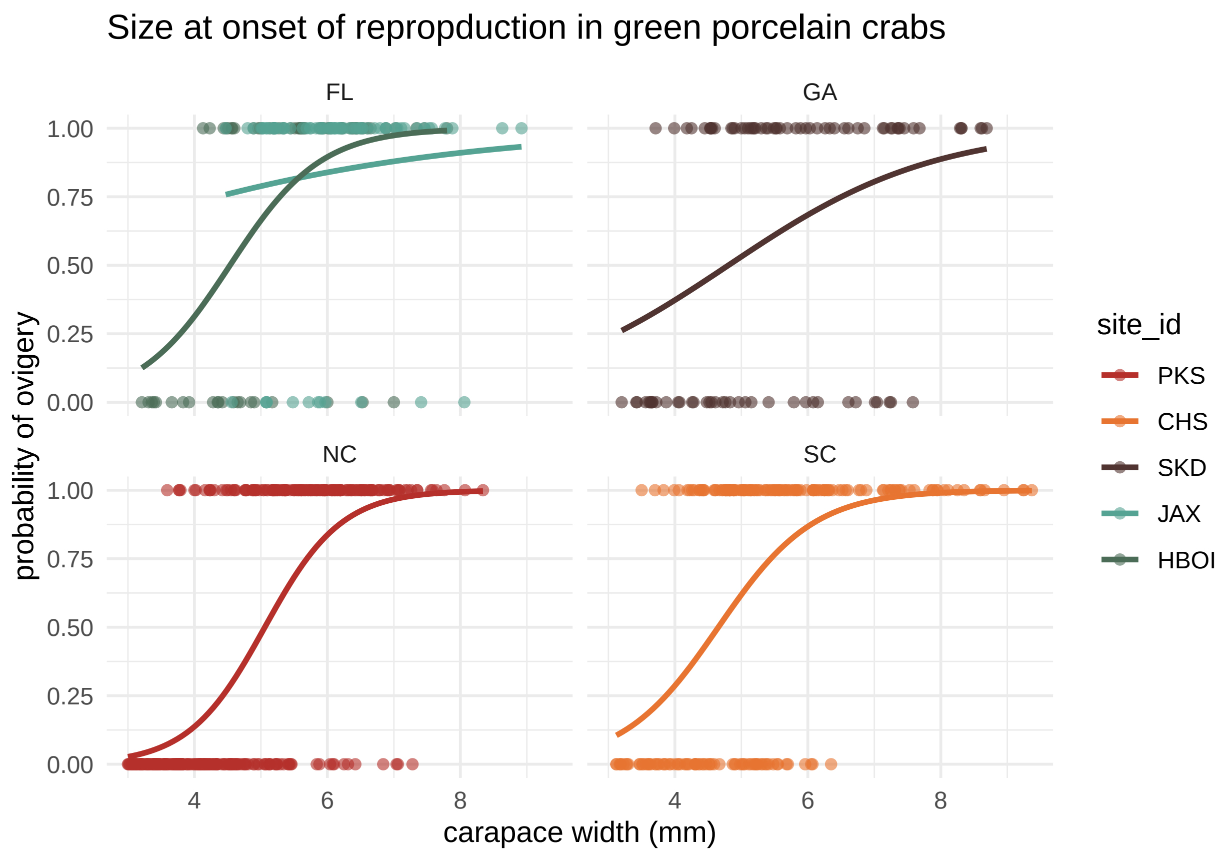
```r
size_at_onset_reproduction <- -coef(prob_of_ovig)[1] / coef(prob_of_ovig)[2]
print(size_at_onset_reproduction)
```

```
## (Intercept)
##    4.866072
```

Plot probability of ovigery, facet wrapped by site (glm)

```r
ggplot(crabs_filtered, aes(x = cw_mm, y = sex_binary, color = site_id)) +
  geom_point(alpha = 0.6) +
  geom_smooth(
    method = "glm",
    method.args = list(family = "binomial"),
    se = FALSE,
    aes(group = site_id)) +  # Logistic regression lines by site
  scale_color_manual(values = c("PKS" = "#b5302b", "CHS" = "#e77431",
                                "SKD" = "#503431", "JAX" = "#55a393",
                                "HBOI" = "#4b6c57")) +
  facet_wrap(~ state) +
  labs(
    x = "carapace width (mm)",
    y = "probability of ovigery",
    title = expression("Size at onset of repropduction in green porcelain crabs")
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```


Size at onset of repropduction in green porcelain crabs
FL
GA
NC
SC
probability of ovigery
carapace width (mm)
site_id
PKS
CHS
SKD
JAX
HBOI