# Spatial Diet Data Analysis

Sam James

January 23, 2019

## Spatial Analysis of Juvenile Sockeye Diets in 2015

The following is a collection of the scripts run to generate and analyse the data for spatial analyses of the juvenile sockeye diet data collected in 2015.

Research Questions: 1. How do diets vary across different ocean conditions experienced during the early outmigration? 2. Are they food-limited during their residence in the 'trophic gauntlet'?

#### Stomach Fullness

What percent of their body weight are they consuming (%BW) are they consuming in this region that has been hypothesized to be a trophic gauntlet?

### Importing and Wrangling the Data

```
library(tidyverse)
## -- Attaching packages -
## v ggplot2 3.0.0
                                 0.2.5
                       v purrr
## v tibble 1.4.2
                       v dplyr
                                 0.7.7
## v tidvr
             0.8.1
                       v stringr 1.3.1
## v readr
             1.1.1
                       v forcats 0.3.0
## -- Conflicts -----
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()
                    masks stats::lag()
library(here)
## here() starts at C:/Users/Sam/Documents/Academic/UBC/Grad/My Research/R/sockeye_diets
spatial <- read_csv(here("raw", "Spatial_2019.csv"))</pre>
## Warning: Duplicated column names deduplicated: 'species' =>
## 'species_1' [29]
## Parsed with column specification:
## cols(
##
     .default = col_character(),
     sam.id = col_integer(),
##
##
     sam.proc.date = col_date(format = ""),
##
     lab.proc.date = col_date(format = ""),
##
     year = col_integer(),
##
     month = col_integer(),
     day = col_integer(),
##
##
     species.no = col_integer(),
     stomtray = col_integer(),
##
##
     forklength = col_integer(),
##
     fishweight = col_double(),
```

```
##
    dry.content.w = col_double(),
##
    wet.content.w = col_double(),
##
    corrected.ww = col_double(),
    fullness.mean = col_double(),
##
##
    DI = col_integer(),
    count = col_integer(),
##
    length.min = col_double(),
##
    length.max = col_double(),
##
##
    group.weight = col_double(),
##
    corrected.weight = col_double()
    # ... with 2 more columns
## )
## See spec(...) for full column specifications.
## Warning in rbind(names(probs), probs_f): number of columns of result is not
## a multiple of vector length (arg 1)
## Warning: 6 parsing failures.
## row # A tibble: 5 x 5 col
                               row col
                                           expected actual file
## ... ......
## See problems(...) for more details.
# Calculate the percent body weight of stomach contents for each fish: select desired columns, remove N
percent.bw <- spatial %>%
       select(ufn, location, site, fishweight, corrected.ww) %>%
       filter(!is.na(corrected.ww), !is.na(fishweight)) %>%
       mutate(site = factor(site, levels = c("D06", "D07", "D08", "D09", "D10", "D11", "J06", "J07", "
       group by(ufn) %>%
       mutate(perc.bw = (((corrected.ww)/1000)/(fishweight-(corrected.ww/1000)))*100) %>%
       distinct()
write.csv(percent.bw, "GFI.csv")
#to get a list of %BW values with means, standard deviations and standard error averaged for each site:
gfi.site <- percent.bw %>%
       #filter(site != 'D06', site != 'J02') %>%
       group_by(site) %>%
       summarise(meangfi = mean(perc.bw), sd.gfi = sd(perc.bw), se.gfi = sd(perc.bw)/sqrt(length(perc.
#to get a list of %BW values with means, standard deviations and standard error averaged for each locat
gfi.location <- percent.bw %>%
       filter(site != 'D06', site != 'J02') %>%
       group_by(location) %>%
       summarise(meangfi = mean(perc.bw), sd.gfi = sd(perc.bw), se.gfi = sd(perc.bw)/sqrt(length(perc.
```

#### Visualizing the Data

```
axis.text.x = element_text(size = 14),
    axis.text.y = element_text(size = 14),
    axis.title.x = element_blank(),
    axis.title.y = element_text(size = 14),
    panel.background = element_rect(fill = "white", colour = "grey50"),
    strip.background = element_blank(),
    strip.text = element_text(size = 16)
) +
scale_y_continuous("Gut Fullness Index", breaks = c(0,1,2,3,4,5,6)) +
labs(y = "Gut Fullness Index")
```

