Final Report

Piyush Amitabh

Introduction

Animal pollinators, particularly bees, play a crucial role in the producing global food crops (Klein et al., 2007) and maintaining ecosystem health and biodiversity by facilitating plant reproduction (Biesmeijer et al., 2006). Understanding the dynamics of bee-plant interactions is essential for conservation efforts, agricultural productivity, sustainable development and ecosystem resilience (Patel et al., 2021, Huang et al., 2021).

In this study, we utilize the Oregon Bee Atlas (OBA) dataset which is a citizen science project in which trained volunteers catch bee species around Oregon which is then verified by experts before logging its information and preserving the specimen (Best et al., 2021, 2022). This dataset covers the state of Oregon in the Pacific Northwest of the US, spanning two years (2018-2019) and includes information on bee species, the plants they were foraging on, date caught, and sex. Combined with other open datasets, OBA can provide a valuable opportunity to explore various aspects of bee-plant interactions and their implications for ecosystem functioning.

Background and Research Questions

Part 1: Effect of Sampling Protocols and Urbanization on Bee Sex Ratio

Background: Sampling Protocol and Taxon level biases

Different types of sampling protocols used to collect bees in the OBA are aerial net, blue vane trap, pan trap and trap nest. The aerial net is considered active method while the rest are considered passive. Many studies in the literature have shown that taxon level biases exist between different sampling protocols, some of them are listed below:

1. **Aerial Net** involves using nets to capture insects or small animals while they are flying or moving through the air. Active methods, such as aerial netting, rely heavily on the researcher's observational abilities and netting skill. These techniques involve physically capturing or directly interacting with organisms in their natural environment. Success in these methods often depends on the researcher's ability to accurately observe and predict the behavior of the target organisms, as well as their skill in using the equipment effectively (Roulston et al 2007).

2. **Blue Vane Traps** are effective for capturing larger bees, particularly species like Bombus and Eucerine bees (Geroff et al., 2014; McCravy et al., 2017; Kimoto et al., 2012; Buchanan et al., 2017). These traps typically consist of a blue vane that attracts bees, leading them into a capturing mechanism.

3. **Pan Traps** are useful for catching small bees, especially Halictid bees, but they capture bigger bees less frequently (Droege et al., 2010; Richards et al., 2011; Toler et al., 2005). These traps involve shallow pans filled with soapy water, which attracts and drowns the bees.

4. **Trap Nests** are designed to catch cavity-building and more solitary bees, though only a small subset of potential cavity-nesting bee species actually use them (Campbell et al., 2017). These traps consist of tubes or cavities that mimic natural nesting sites for bees.

Background: Importance of Bee Sex Ratio

Bee Sex Ration is an important factor as it can influence population dynamics of bee communities. It also has strong effects on foraging behavior of the bees. Studies have shown that there are differences in floral preferences (which flowers the bees visit), floral constancy (how likely is a bee to visit the flower of the same species), pollen transfer efficiency (efficiency of pollen transfer between the flowers), and flight distance between foraging bouts among male and female bees (Ritchie et al., 2016; Ne'eman et al., 2006; Ostevik et al., 2010). These differences can have implications for the overall foraging efficiency and reproductive success of bee populations. Moreover, there is evidence to suggest that the sex ratio of bee populations can change with levels of urbanization, with urban areas showing a more male-biased sex ratio. This change is mainly driven by a decline in medium and large-bodied ground-nesting female bees in urban environments (Fitch et al., 2019). Understanding these sex-based differences and their impact on population dynamics is crucial for effective conservation strategies for bee species.

Research Questions

1A. Effect of Sampling Protocols Are different sampling protocols more likely to capture bees of one sex than another?

1B. Effect of Human population (marker for urbanization) How does human population levels affect bee distribution and sex ratio?

Part 2: Effect of Weather on Bee Populations

Background

Weather patterns play a crucial role in influencing the behavior and ecology of bees, both in their nursery and foraging activities. Studies have shown that increasing temperatures can enhance flight activity among bees, while decreasing temperatures have the opposite effect (Riessberger & Crailsheim, 1997; Burrill & Dietz, 1981). Moreover, temperature and precipitation have been identified as more important predictors of bee behavior than landscape composition, quality, or topography (Kammerer et al., 2021). Long-term precipitation patterns can also significantly impact bee communities by affecting the availability of floral resources (Thomson, 2016).

Despite the importance of weather patterns, there is a limited understanding of how short-term weather fluctuations affect wild bee populations in Oregon. Previous studies have primarily focused on long-term trends and the effects of climate change on bees. Understanding the short-term impacts of weather on bee populations is crucial for developing effective conservation strategies, especially in the face of climate change and increasing weather variability (Burrill & Dietz, 1981; Kammerer et al., 2021).

Research Questions

How does the number of foraging bees change with temperature and precipitation?

Additional Data Sources Used

1. Oregon County level Population from Oregon Demographics by Cubit (https://www.oregon-demographics.com)
2. Oregon Climate Data from National Oceanic and Atmospheric Administration (https://graphical.weather.gov/sectors/oregon.php)

Code

Import OBA datasets

```r
getwd()
```

```
## [1] "/home/piyush/Dropbox/Academics/BI510L Data Sci in R/BI510L Data Sci in R_code/bi510l_project"
```

```r
setwd("/home/piyush/Dropbox/Academics/BI510L Data Sci in R/BI510L Data Sci in R_code/bi510l_project")
oba_2018 <- read.csv("data/2018_occurrence.csv")
oba_2019 <- read.csv("data/2019_occurrence.csv")
cat("The colnames are identical:", identical(colnames(oba_2018), colnames(oba_2019)))
```

```
## The colnames are identical: TRUE
```

```r
oba_2018_2019 <- rbind(oba_2018, oba_2019)
# head(oba_2018_2019, 3)
```

Clean dataset using regex

I will clean and update the associatedTaxa column which is supposed to have the plant name

1. Only keep the plant names in the string

```r
sample(oba_2018_2019$associatedTaxa, size = 10)
```

```
##  [1] ""
##  [2] ""
##  [3] "\"foraging on\" : \"Rubus parviflorus\""
##  [4] "\"foraging on\" : \"Penstemon\""
##  [5] "foraging on : \"Achillea millefolium\""
##  [6] "foraging on : \"Lomatium\""
##  [7] "foraging on : \"Lupinus\""
##  [8] ""
##  [9] "foraging on : \"Symphyotrichum foliaceum\""
## [10] "foraging on : \"Ericameria nauseosa\""
```

```r
# using regex to select for the pattern to keep rather than the pattern to drop
# noticed that the plant name is always bounded by the second double quotes ""

# Define the regex pattern
pattern <- '(.*)"(?<plant>.*)"$'

# Use str_match() to extract the match
match <- str_match(oba_2018_2019$associatedTaxa, pattern)
plant_name_list <- match[, "plant"]
plant_name_list[is.na(plant_name_list)] <- ""
oba_2018_2019$associatedTaxa <- plant_name_list
sample(oba_2018_2019$associatedTaxa, size = 10)
```

```
##  [1] ""                           "Cirsium neomexicanum"
##  [3] "Calochortus tolmiei"        ""
##  [5] "Holodiscus discolor"        "Eriophyllum confertiflorum"
##  [7] ""                           "Plagiobothrys"
##  [9] "Rosaceae"                   "Mentha pulegium"
```

2. Remove all weed names and keep only one plant name

```
cat("We have these many weeds:", sum(grepl("Weedy", oba_2018_2019$associatedTaxa)), "\n")
```

```
## We have these many weeds: 3
```

```
oba_2018_2019$associatedTaxa <- gsub("Weedy yellow comp.", "", oba_2018_2019$associatedTaxa)
oba_2018_2019$associatedTaxa <- gsub("Weedy", "", oba_2018_2019$associatedTaxa)
cat("Now we have these many weeds:", sum(grepl("Weedy", oba_2018_2019$associatedTaxa)), "\n")
```

```
## Now we have these many weeds: 0
```

```
# get rid of ;
cat("We have these many ';'", sum(grepl(";", oba_2018_2019$associatedTaxa)), "\n")
```

```
## We have these many ';' 11
```

```
oba_2018_2019$associatedTaxa[grepl(";", oba_2018_2019$associatedTaxa)] <- sapply(
    strsplit(
        oba_2018_2019$associatedTaxa[grepl(";", oba_2018_2019$associatedTaxa)], ";"
    ),
    function(x) x[1]
)
cat("We have these many ';'", sum(grepl(";", oba_2018_2019$associatedTaxa)), "\n")
```

```
## We have these many ';' 0
```

Add more information

1. Add plant resolution

We want to make a new column called plantResolution and assign it to “Family”, “Genus” or “Species” depending on the level of resolution associated taxa is resolved to.

```
oba_2018_2019$plantTaxaWordCount <- str_count(oba_2018_2019$associatedTaxa, "\\w+")
head(oba_2018_2019$associatedTaxa, 10)
```

```
##  [1] "Vaccinium ovatum" ""                 ""                 "Whipplea modesta"
##  [5] "Whipplea modesta" ""                 ""                 ""
##  [9] ""                 "Horkelia fusca"
```

```
head(oba_2018_2019$plantTaxaWordCount, 10)
```

```
##  [1] 2 0 0 2 2 0 0 0 0 2
```

```
summary(oba_2018_2019$plantTaxaWordCount)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   1.000   1.084   2.000   4.000
```

```
plant_resolution_list <- rep(NA, nrow(oba_2018_2019))

for (i in seq_along(plant_resolution_list)) {
    word_count <- oba_2018_2019$plantTaxaWordCount[i]
    plant_name <- oba_2018_2019$associatedTaxa[i]

    if (word_count == 1) { # can be genus or family
        if (grepl("aceae$", plant_name)) { # family
            plant_resolution_list[i] <- "family"
```

```r
        } else {
            plant_resolution_list[i] <- "genus"
        }
    } else if (word_count > 1) { # it is species/subspecies
        plant_resolution_list[i] <- "species"
    }
}
oba_2018_2019$plantResolution <- plant_resolution_list
head(oba_2018_2019$associatedTaxa, 10)
```

```
##  [1] "Vaccinium ovatum" ""                 ""                 "Whipplea modesta"
##  [5] "Whipplea modesta" ""                 ""                 ""
##  [9] ""                 "Horkelia fusca"
```

```r
head(oba_2018_2019$plantTaxaWordCount, 10)
```

```
##  [1] 2 0 0 2 2 0 0 0 0 2
```

```r
head(oba_2018_2019$plantResolution, 10)
```

```
##  [1] "species" NA        NA        "species" "species" NA        NA
##  [8] NA        NA        "species"
```

2. Add plant genus

Create a new column called plantGenus that is the genus if the associated taxa was resolved to species or genus, and NA if it was resolved to family.

```r
plant_genus_list <- rep(NA, nrow(oba_2018_2019))

for (i in seq_along(plant_genus_list)) {
    plant_res <- oba_2018_2019$plantResolution[i]
    plant_name <- oba_2018_2019$associatedTaxa[i]

    if (!is.na(plant_res) && (plant_res == "genus")) {
        plant_genus_list[i] <- plant_name
    } else if (!is.na(plant_res) && (plant_res == "species")) {
        plant_genus_list[i] <- strsplit(plant_name, " ")[[1]][1]
    }
}

oba_2018_2019$plantGenus <- plant_genus_list

head(oba_2018_2019$associatedTaxa, 10)
```

```
##  [1] "Vaccinium ovatum" ""                 ""                 "Whipplea modesta"
##  [5] "Whipplea modesta" ""                 ""                 ""
##  [9] ""                 "Horkelia fusca"
```

```r
head(oba_2018_2019$plantTaxaWordCount, 10)
```

```
##  [1] 2 0 0 2 2 0 0 0 0 2
```

```r
head(oba_2018_2019$plantResolution, 10)
```

```
##  [1] "species" NA        NA        "species" "species" NA        NA
##  [8] NA        NA        "species"
```

```
head(oba_2018_2019$plantGenus, 10)
```

```
##  [1] "Vaccinium" NA          NA          "Whipplea"  "Whipplea"  NA
##  [7] NA          NA          NA          "Horkelia"
```

Save/Read the new dataframe as csv

Drop the columns that we are not going to use

```
col_to_save <- c(
    "gbifID", "genericName", "specificEpithet", "species", "sex",
    "taxonRank", "eventDate", "year", "month", "samplingProtocol",
    "associatedTaxa", "county", "decimalLatitude", "decimalLongitude",
    "plantResolution", "plantGenus"
)
oba_2018_2019 <- oba_2018_2019[, col_to_save]
colnames(oba_2018_2019)
```

```
##  [1] "gbifID"           "genericName"      "specificEpithet"  "species"
##  [5] "sex"              "taxonRank"        "eventDate"        "year"
##  [9] "month"            "samplingProtocol" "associatedTaxa"   "county"
## [13] "decimalLatitude"  "decimalLongitude" "plantResolution"  "plantGenus"
```

Save

```
write.csv(oba_2018_2019, "data/cleaned_oba_2018_2019.csv", row.names = FALSE)
```

Read

```
oba_2018_2019 <- read.csv("data/cleaned_oba_2018_2019.csv")
```

```
# check if all values of column gbifID are unique
unique_gbifID <- length(unique(oba_2018_2019$gbifID))
total_gbifID <- nrow(oba_2018_2019)

if (unique_gbifID == total_gbifID) {
    print("All values of column 'gbifID' are unique.")
} else {
    print("Not all values of column 'gbifID' are unique.")
}
```

```
## [1] "All values of column 'gbifID' are unique."
```

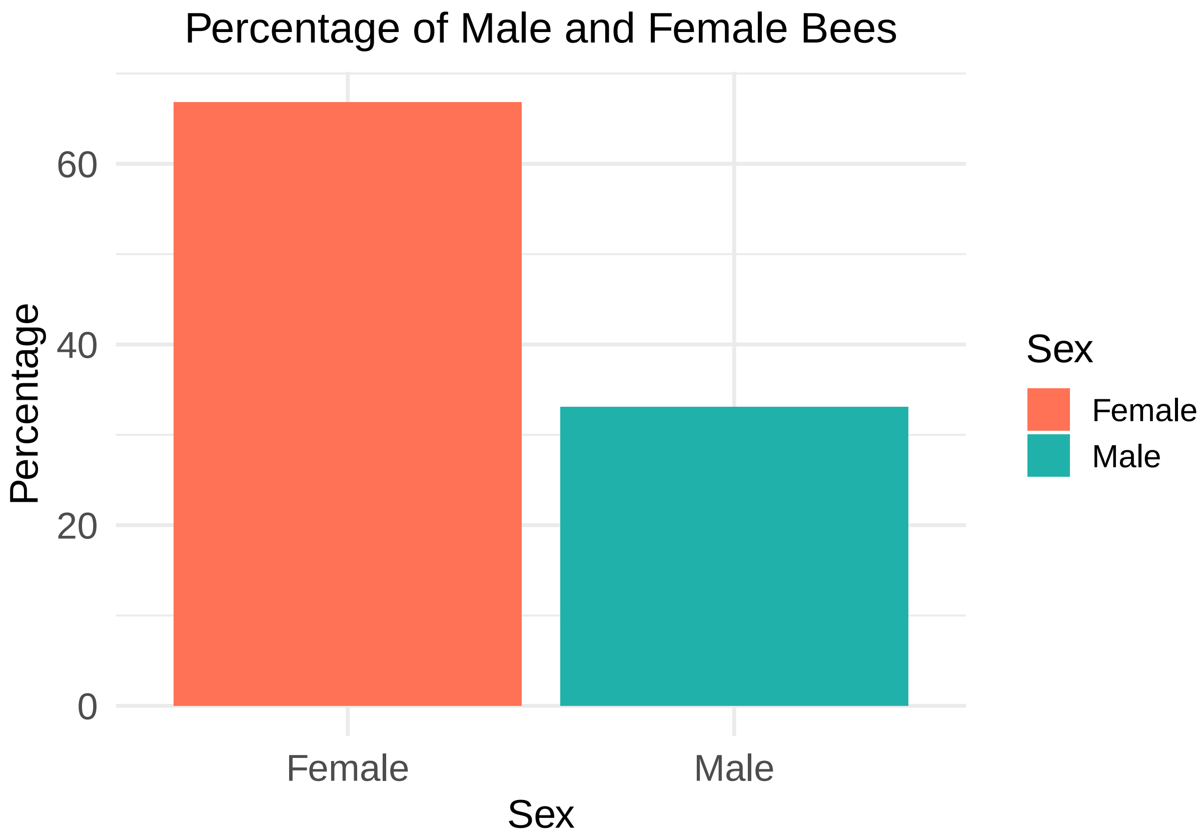
This means we can use gbifID as a unique identifier

1. Bee Sex Ratio

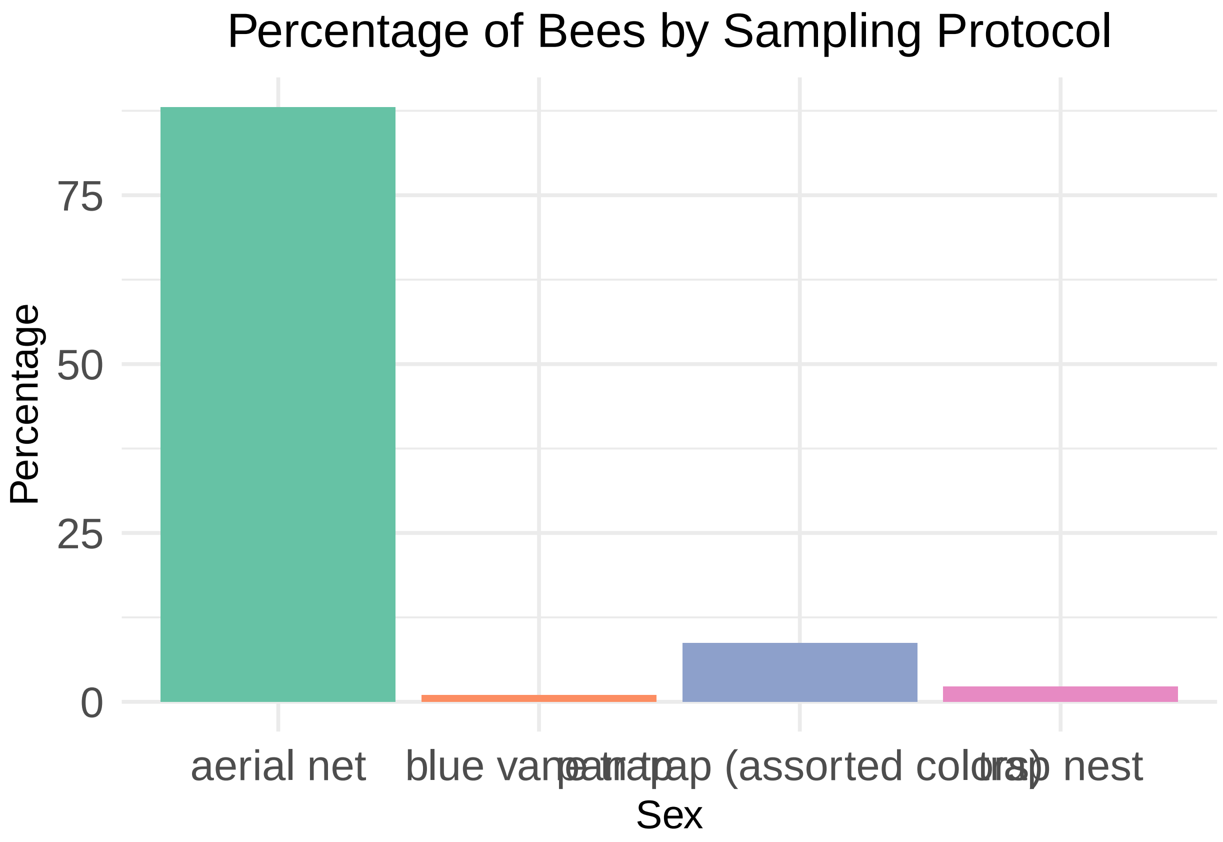
A. Effect of Sampling Protocols

```
# check the TOTAL PERCENTAGE of MALE vs FEMALE bees caught
total_male <- sum(oba_2018_2019$sex == "MALE", na.rm = TRUE)
total_female <- sum(oba_2018_2019$sex == "FEMALE", na.rm = TRUE)
total_bees <- nrow(oba_2018_2019)
```

```r
percentage_male <- (total_male / total_bees) * 100
percentage_female <- (total_female / total_bees) * 100

# Create a data frame for the plot
sex_per_df <- data.frame(
    Sex = c("Female", "Male"),
    Percentage = c(percentage_female, percentage_male)
)

# Create a bar plot with ggplot
p_title <- "Percentage of Male and Female Bees"
p <- ggplot(sex_per_df, aes(x = Sex, y = Percentage, fill = Sex)) +
    geom_bar(stat = "identity") +
    scale_fill_manual(values = c("coral1", "lightseagreen")) +
    labs(x = "Sex", y = "Percentage") +
    ggtitle(p_title) +
    theme_minimal(15) +
    theme(
        axis.text = element_text(size = 14),
        plot.title = element_text(hjust = 0.5, size = 16)
    )
p
```



```r
# ggsave(glue("plots/{p_title}.png"), dpi = 300, plot = p)
# Calculate the percentage of bees caught using different sampling patterns
sampling_patterns <- unique(oba_2018_2019$samplingProtocol)
percentage_per_pattern <- sapply(sampling_patterns, function(pattern) {
    total_bees <- sum(oba_2018_2019$samplingProtocol == pattern, na.rm = TRUE)
    percentage <- (total_bees / nrow(oba_2018_2019)) * 100
    return(percentage)
})

sampling_per_df <- data.frame(
    patterns = sampling_patterns,
    percentage = percentage_per_pattern
)

# Create a stacked bar plot with percentages
p_title <- "Percentage of Bees by Sampling Protocol"
p <- ggplot(sampling_per_df, aes(x = patterns, y = percentage, fill = patterns)) +
    geom_bar(stat = "identity", position = "stack", show.legend = F) +
    labs(x = "Sex", y = "Percentage", fill = "Sampling Protocol") +
    ggtitle(p_title) +
    theme_minimal(15) +
    theme(
        axis.text = element_text(size = 16),
        plot.title = element_text(hjust = 0.5)
    ) +
    scale_fill_brewer(palette = "Set2")
p
```

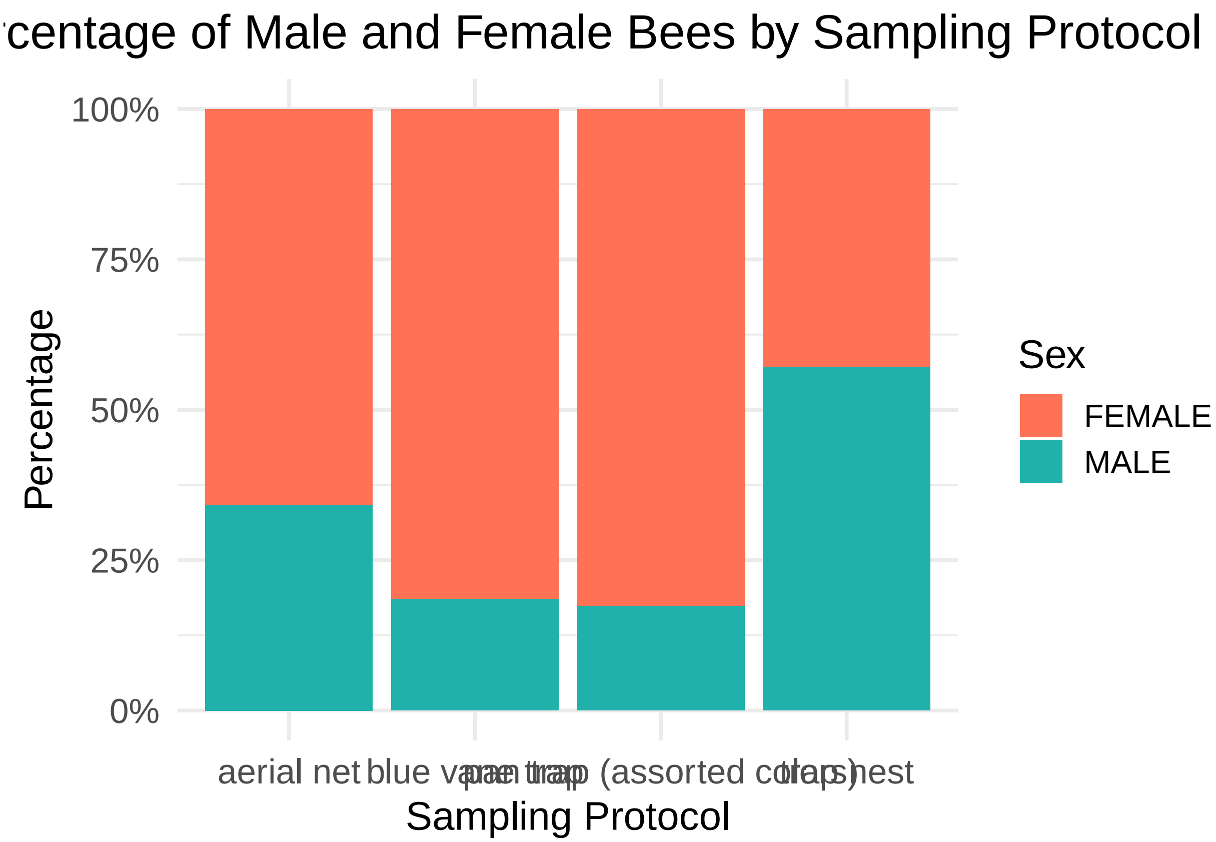


```
# ggsave(glue("plots/{p_title}.png"), dpi = 300, plot = p)
```

```
p_title <- "Percentage of Male and Female Bees by Sampling Protocol"
p <- oba_2018_2019 %>%
    filter(sex != "") %>%
    group_by(sex, samplingProtocol) %>%
    summarise(n_group = n()) %>%
    ungroup() %>%
    group_by(samplingProtocol) %>%
    mutate(
        n_sampling = sum(n_group),
        percentage = n_group / n_sampling
    ) %>%
    ggplot(aes(x = samplingProtocol, y = percentage, fill = sex)) +
    geom_bar(stat = "identity", position = "stack") +
    labs(x = "Sampling Protocol", y = "Percentage", fill = "Sex") +
    ggtitle(p_title) +
    scale_fill_manual(values = c("coral1", "lightseagreen")) +
    scale_y_continuous(labels = scales::percent_format()) +
    theme_minimal(15) +
    theme(
        axis.text = element_text(size = 13),
        plot.title = element_text(hjust = 0.5)
    )
```

```
## `summarise()` has grouped output by 'sex'. You can override using the `.groups`
## argument.
```

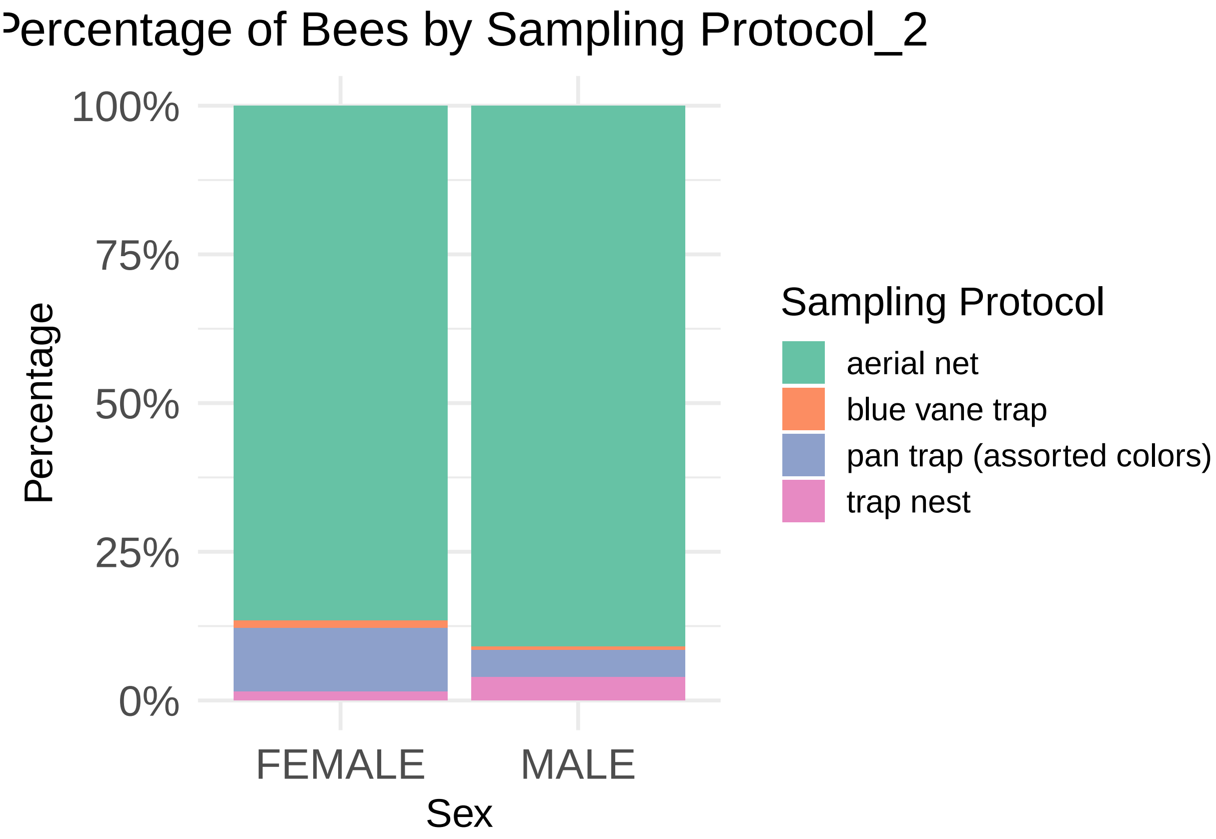
```
p
```



```
# ggsave(glue("plots/{p_title}.png"), dpi = 300, plot = p)

# Generate the plot
p_title <- "Percentage of Bees by Sampling Protocol_2"
p <- oba_2018_2019 %>%
    filter(sex != "") %>%
    group_by(sex, samplingProtocol) %>%
    summarise(n_group = n()) %>%
    ungroup() %>%
    group_by(sex) %>%
    mutate(
        n_sampling = sum(n_group),
        percentage = n_group / n_sampling
    ) %>%
    ggplot(aes(x = sex, y = percentage, fill = samplingProtocol)) +
    geom_bar(stat = "identity", position = "stack") +
    labs(x = "Sex", y = "Percentage", fill = "Sampling Protocol") +
    ggtitle(p_title) +
    scale_fill_brewer(palette = "Set2") +
    scale_y_continuous(labels = scales::percent_format()) +
    theme_minimal(15) +
    theme(
        axis.text = element_text(size = 16),
        plot.title = element_text(hjust = 0.5)
    )
```

```
## `summarise()` has grouped output by 'sex'. You can override using the `.groups`
## argument.
```

```
p
```



```
# ggsave(glue("plots/{p_title}.png"), dpi = 300, plot = p)
```

B. Effect of Human Population

Download basemaps into R using the map_data function in ggplot

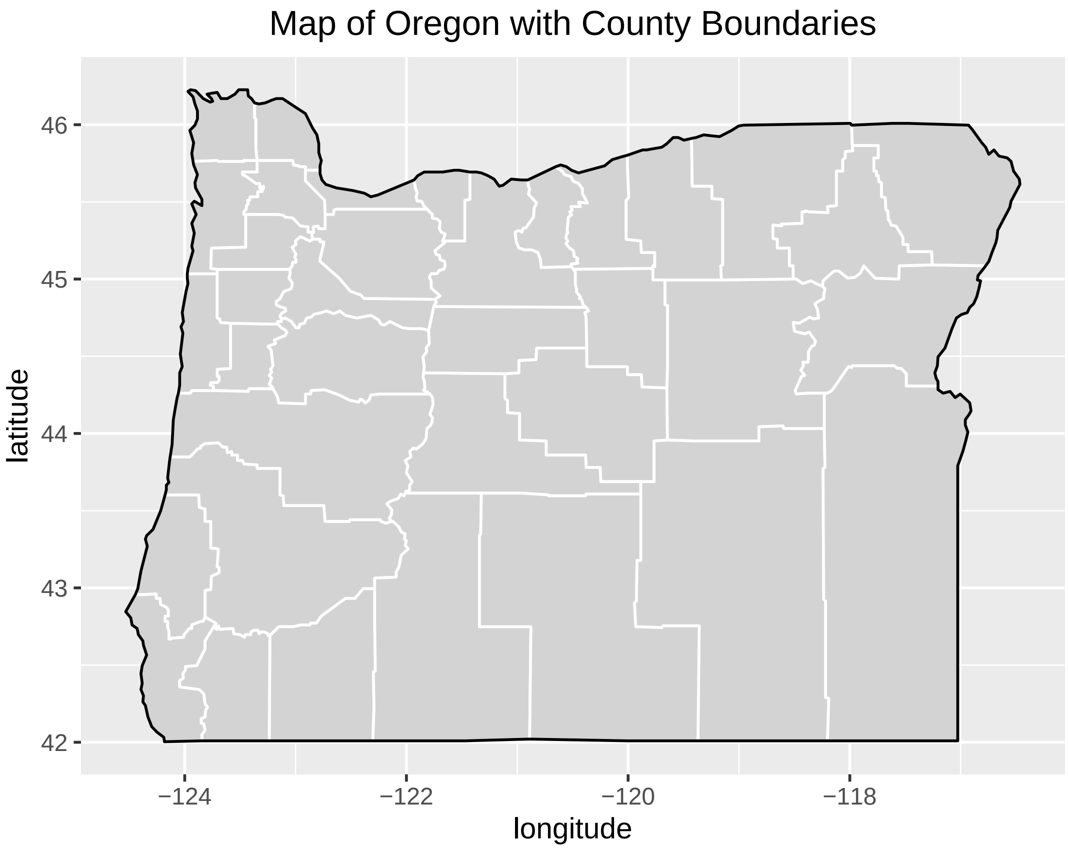
```
# get Oregon map
or_map_base_df <- map_data("state") %>%
  filter(region == "oregon") %>%
  select(longitude = long, latitude = lat, group_id = group, state_name = region)
str(or_map_base_df)
```

```
## 'data.frame':    236 obs. of  4 variables:
##  $ longitude : num  -117 -117 -117 -117 -117 ...
##  $ latitude  : num  46 46 45.9 45.9 45.9 ...
##  $ group_id  : num  44 44 44 44 44 44 44 44 44 44 ...
##  $ state_name: chr  "oregon" "oregon" "oregon" "oregon" ...
```

```
or_map_county_df <- map_data("county", "oregon") %>%
  select(longitude = long, latitude = lat, group_id = group, county_name = subregion)
str(or_map_county_df)
```

```
## 'data.frame':    1633 obs. of  4 variables:
##  $ longitude  : num  -117 -117 -117 -118 -118 ...
##  $ latitude   : num  44.3 44.3 44.4 44.4 44.4 ...
##  $ group_id   : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ county_name: chr  "baker" "baker" "baker" "baker" ...
```

```
# plot Oregon map with bee location overlaid
plot_or_w_county <- ggplot() +
  geom_polygon(
    data = or_map_county_df, fill = "lightgray", color = "white",
    aes(x = longitude, y = latitude, group = county_name)
  ) +
  geom_polygon(
    data = or_map_base_df, fill = NA, color = "black",
    aes(x = longitude, y = latitude, group = state_name)
  ) +
  theme(plot.title = element_text(hjust = 0.5))
plot_or_w_county +
  ggtitle(glue("Map of Oregon with County Boundaries")) +
  coord_quickmap()
```



```
# data taken from https://www.oregon-demographics.com/counties_by_population

# make a data frame
or_dem <- readxl::read_xlsx("./data/oregon_demographics_by_county.xlsx", skip = 4, col_names = TRUE)
head(or_dem)
```

```
## # A tibble: 6 x 67
##   name  state county city  most_current_pop_year population pop_dens_sq_mi  mhhi
##   <chr> <chr> <lgl> <lgl>                 <dbl>      <dbl>          <dbl> <dbl>
## 1 Oreg~ Oreg~ NA    NA                     2022    4229374           44.1 76632
## 2 Bake~ Oreg~ NA    NA                     2022      16685            5.4 51657
## 3 Bent~ Oreg~ NA    NA                     2022      95615          142.  72882
## 4 Clac~ Oreg~ NA    NA                     2022     420925          225   95740
## 5 Clat~ Oreg~ NA    NA                     2022      41190           NA      NA
## 6 Colu~ Oreg~ NA    NA                     2022      52865           NA      NA
## # i 59 more variables: hhi_total_hh <dbl>, hhi_hh_w_lt_25k <dbl>,
## #   hhi_hh_w_25k_49k <dbl>, hhi_hh_w_50k_74k <dbl>, hhi_hh_w_75k_99k <dbl>,
## #   hhi_hh_w_100k_149k <dbl>, hhi_hh_w_150k_199k <dbl>,
## #   hhi_hh_w_200k_plus <dbl>, race_ethnicity_total <dbl>,
## #   race_ethnicity_white <dbl>, race_ethnicity_black <dbl>,
## #   race_ethnicity_native <dbl>, race_ethnicity_asian <dbl>,
## #   race_ethnicity_islander <dbl>, race_ethnicity_other <dbl>, ...
```

```r
colnames(or_dem)
```

```
##  [1] "name"                              "state"
##  [3] "county"                            "city"
##  [5] "most_current_pop_year"             "population"
##  [7] "pop_dens_sq_mi"                    "mhhi"
##  [9] "hhi_total_hh"                      "hhi_hh_w_lt_25k"
## [11] "hhi_hh_w_25k_49k"                  "hhi_hh_w_50k_74k"
## [13] "hhi_hh_w_75k_99k"                  "hhi_hh_w_100k_149k"
## [15] "hhi_hh_w_150k_199k"                "hhi_hh_w_200k_plus"
## [17] "race_ethnicity_total"              "race_ethnicity_white"
## [19] "race_ethnicity_black"              "race_ethnicity_native"
## [21] "race_ethnicity_asian"              "race_ethnicity_islander"
## [23] "race_ethnicity_other"              "race_ethnicity_two"
## [25] "race_ethnicity_hispanic"           "age_total"
## [27] "age_f_0_9"                         "age_f_10_19"
## [29] "age_f_20_29"                       "age_f_30_39"
## [31] "age_f_40_49"                       "age_f_50_59"
## [33] "age_f_60_69"                       "age_f_70_plus"
## [35] "age_m_0_9"                         "age_m_10_19"
## [37] "age_m_20_29"                       "age_m_30_39"
## [39] "age_m_40_49"                       "age_m_50_59"
## [41] "age_m_60_69"                       "age_m_70_plus"
## [43] "median_age"                        "edu_att_pop_25_plus"
## [45] "edu_att_no_diploma"                "edu_att_high_school"
## [47] "edu_att_some_college"              "edu_att_bachelors"
## [49] "edu_att_graduate"                  "family_hh_total"
## [51] "family_poverty_pct"                "emp_status_civ_labor_force"
## [53] "unemployment_pct"                  "housing_units"
## [55] "occupied_units"                    "owner_occupied"
## [57] "renter_occupied"                   "median_value_owner_occupied_units"
## [59] "households"                        "hh_families"
## [61] "hh_mc_families"                    "hh_mc_with_own_children_under_18"
## [63] "hh_sp_families"                    "hh_sp_with_own_children_under_18"
## [65] "hh_non_families"                   "aland_sq_m"
## [67] "geoid"
```

```r
# select only two columns, remove oregon total population, add 'county' col, convert everything to lowe
or_pop_by_county_df <- or_dem %>%
  select(name, population) %>%
  filter(name != "Oregon") %>%
  mutate(county_name = tolower(str_replace(name, " County$", ""))) %>%
  select(-name)

head(or_pop_by_county_df)
```

```
## # A tibble: 6 x 2
##   population county_name
##        <dbl> <chr>
## 1      16685 baker
## 2      95615 benton
## 3     420925 clackamas
## 4      41190 clatsop
## 5      52865 columbia
## 6      64908 coos
```

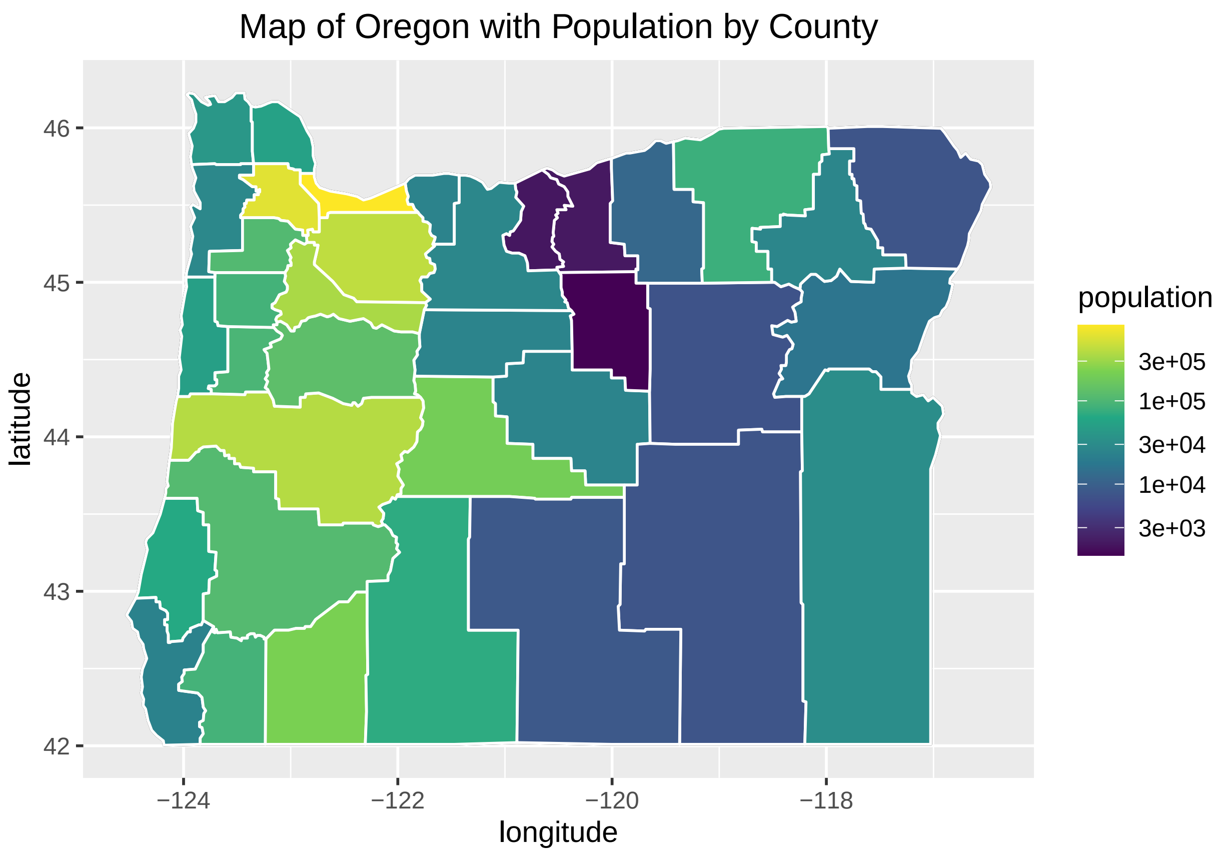
```
head(or_map_county_df)
```

```
##   longitude latitude group_id county_name
## 1 -117.2042 44.30683        1       baker
## 2 -117.4907 44.30683        1       baker
## 3 -117.4907 44.38704        1       baker
## 4 -117.5366 44.42142        1       baker
## 5 -117.5709 44.42142        1       baker
## 6 -117.5996 44.43861        1       baker
```

```
# now the county_name in or_pop_by_county_df corresponds to or_map_county_df, join them in a single df
or_map_with_population_df <- left_join(or_map_county_df, or_pop_by_county_df, by = "county_name")
head(or_map_with_population_df)
```

```
##   longitude latitude group_id county_name population
## 1 -117.2042 44.30683        1       baker      16685
## 2 -117.4907 44.30683        1       baker      16685
## 3 -117.4907 44.38704        1       baker      16685
## 4 -117.5366 44.42142        1       baker      16685
## 5 -117.5709 44.42142        1       baker      16685
## 6 -117.5996 44.43861        1       baker      16685
```

```
plot_or_w_county +
  ggtitle(glue("Map of Oregon with Population by County")) +
  geom_polygon(data = or_map_with_population_df,
  aes(x = longitude, y = latitude, fill = population, group = county_name), color = "white") +
  scale_fill_continuous(trans = "log10", type = "viridis")
```



```
coord_sf(datum = sf::st_crs(4326))
```

```
## <ggproto object: Class CoordSf, CoordCartesian, Coord, gg>
##     aspect: function
##     backtransform_range: function
##     clip: on
##     crs: NULL
##     datum: crs
##     default: FALSE
##     default_crs: NULL
##     determine_crs: function
##     distance: function
##     expand: TRUE
##     fixup_graticule_labels: function
##     get_default_crs: function
##     is_free: function
##     is_linear: function
##     label_axes: list
##     label_graticule:
##     labels: function
##     limits: list
##     lims_method: cross
##     modify_scales: function
##     ndiscr: 100
##     params: list
##     range: function
##     record_bbox: function
##     render_axis_h: function
##     render_axis_v: function
##     render_bg: function
##     render_fg: function
##     setup_data: function
##     setup_layout: function
##     setup_panel_guides: function
##     setup_panel_params: function
##     setup_params: function
##     train_panel_guides: function
##     transform: function
##     super:  <ggproto object: Class CoordSf, CoordCartesian, Coord, gg>
```

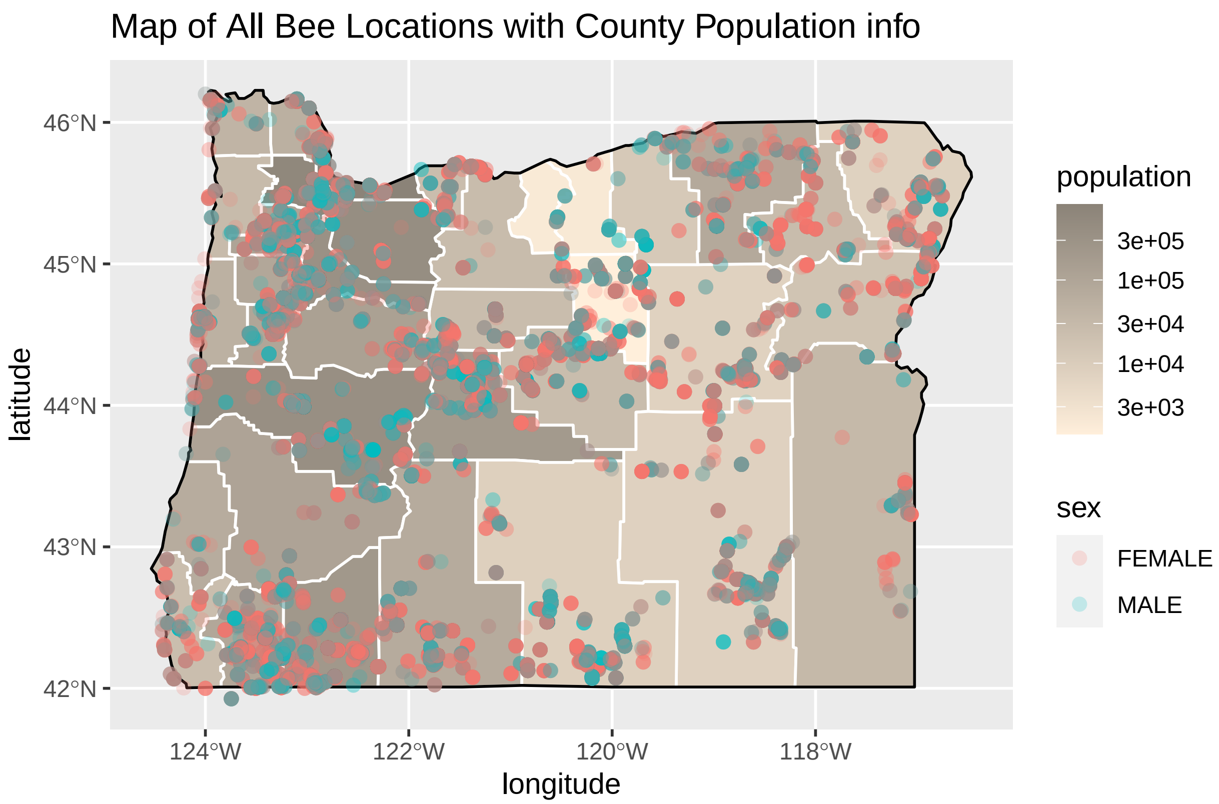
```
summary(oba_2018_2019$decimalLatitude)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   41.93   43.17   44.60   44.23   45.17   46.99      12
```

```
summary(oba_2018_2019$decimalLongitude)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
## -124.42 -123.23 -122.89 -121.97 -121.27   43.37      12
```

```
all_species_data <- oba_2018_2019 %>%
  filter((decimalLongitude < -116) & (decimalLatitude < 46.5) & (sex != "")) %>%
  select(x = decimalLongitude, y = decimalLatitude, sex)

all_species_spatial <- st_as_sf(all_species_data, coords = c("x", "y"), crs = "WGS84")

p_title <- "Map of All Bee Locations with County Population info"
p <- ggplot() +
  geom_polygon(
    data = or_map_county_df, fill = "lightgray", color = "white",
    aes(x = longitude, y = latitude, group = county_name)
  ) +
  geom_polygon(
    data = or_map_with_population_df,
    aes(x = longitude, y = latitude, fill = population, group = county_name), color = "white"
  ) +
  scale_fill_gradient(trans = "log10", low = "antiquewhite1", high = "antiquewhite4") +
  geom_polygon(
    data = or_map_base_df, fill = NA, color = "black",
    aes(x = longitude, y = latitude, group = state_name)
  ) +
  # geom_sf(data = all_species_spatial, color = "darkorange", alpha = 0.1, size = 2) +
  geom_sf(data = all_species_spatial, aes(color = sex), alpha = 0.2, size = 2) +
  # geom_segment(aes(x = -118.4, y = 42.3, xend = -117.21, yend = 42.3), linewidth = 4) +
  # annotate("text", x = -117.8, y = 42.43, label = "100 km", size = 4) +
  # theme(plot.title = element_text(hjust = 0.5)) +
  ggtitle(p_title)
p
```



```
# ggsave(glue("plots/{p_title}_by_sex.png"), dpi = 300, plot = p)
```

2. Weather and Foraging Patterns

Lane county

```
# get climate data for lane county

# Read csv files
avg_temp <- read.csv("data/lane_county_avg_temp_2018_2019.csv", skip = 3, header = 4)
avg_ppt <- read.csv("data/lane_county_precipitation_2018_2019.csv", skip = 3, header = 4)

# rename columns
colnames(avg_temp)[colnames(avg_temp) == "Date"] <- "date"
colnames(avg_temp)[colnames(avg_temp) == "Value"] <- "temp_f"
colnames(avg_ppt)[colnames(avg_ppt) == "Date"] <- "date"
colnames(avg_ppt)[colnames(avg_ppt) == "Value"] <- "ppt_inch"

# add column 'year' and 'month' from the first four string character from the 'Date' column
avg_temp$year <- substr(avg_temp$date, 1, 4)
avg_temp$month <- substr(avg_temp$date, 5, 6)
avg_temp$year <- as.numeric(avg_temp$year)
avg_temp$month <- as.numeric(avg_temp$month)
avg_temp$temp_c <- (avg_temp$temp_f - 32) * (5 / 9) # add celsius
head(avg_temp)
```

```
##     date temp_f year month   temp_c
## 1 201801   42.0 2018     1  5.555556
## 2 201802   38.2 2018     2  3.444444
## 3 201803   41.3 2018     3  5.166667
## 4 201804   46.7 2018     4  8.166667
## 5 201805   55.0 2018     5 12.777778
## 6 201806   58.4 2018     6 14.666667
```

```
avg_ppt$year <- substr(avg_ppt$date, 1, 4)
avg_ppt$month <- substr(avg_ppt$date, 5, 6)
avg_ppt$year <- as.numeric(avg_ppt$year)
avg_ppt$month <- as.numeric(avg_ppt$month)
avg_ppt$ppt_cm <- avg_ppt$ppt_inch * 2.54 # add cm ppt
head(avg_ppt)
```

```
##     date ppt_inch year month  ppt_cm
## 1 201801     8.41 2018     1 21.3614
## 2 201802     4.73 2018     2 12.0142
## 3 201803     5.84 2018     3 14.8336
## 4 201804     6.55 2018     4 16.6370
## 5 201805     0.83 2018     5  2.1082
## 6 201806     1.86 2018     6  4.7244
```

```
lane_oba_2018_2019 <- oba_2018_2019 %>%
    filter(county == "Lane")

# find the count of total number of 'gbifID' by year and month in lane_oba_2018_2019
net_count_per_month <- lane_oba_2018_2019 %>%
    group_by(year, month) %>%
    summarise(count = n_distinct(gbifID))
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
species_count_per_month <- lane_oba_2018_2019 %>%
    filter(species != "") %>%
    group_by(year, month) %>%
    summarise(count = n_distinct(species))
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
# get and add ppt_cm information from avg_ppt df by year and month
net_count_per_month <- net_count_per_month %>%
    left_join(avg_ppt, by = c("year", "month")) %>%
    left_join(avg_temp, by = c("year", "month"))

species_count_per_month <- species_count_per_month %>%
    left_join(avg_ppt, by = c("year", "month")) %>%
    left_join(avg_temp, by = c("year", "month"))

head(net_count_per_month)
```

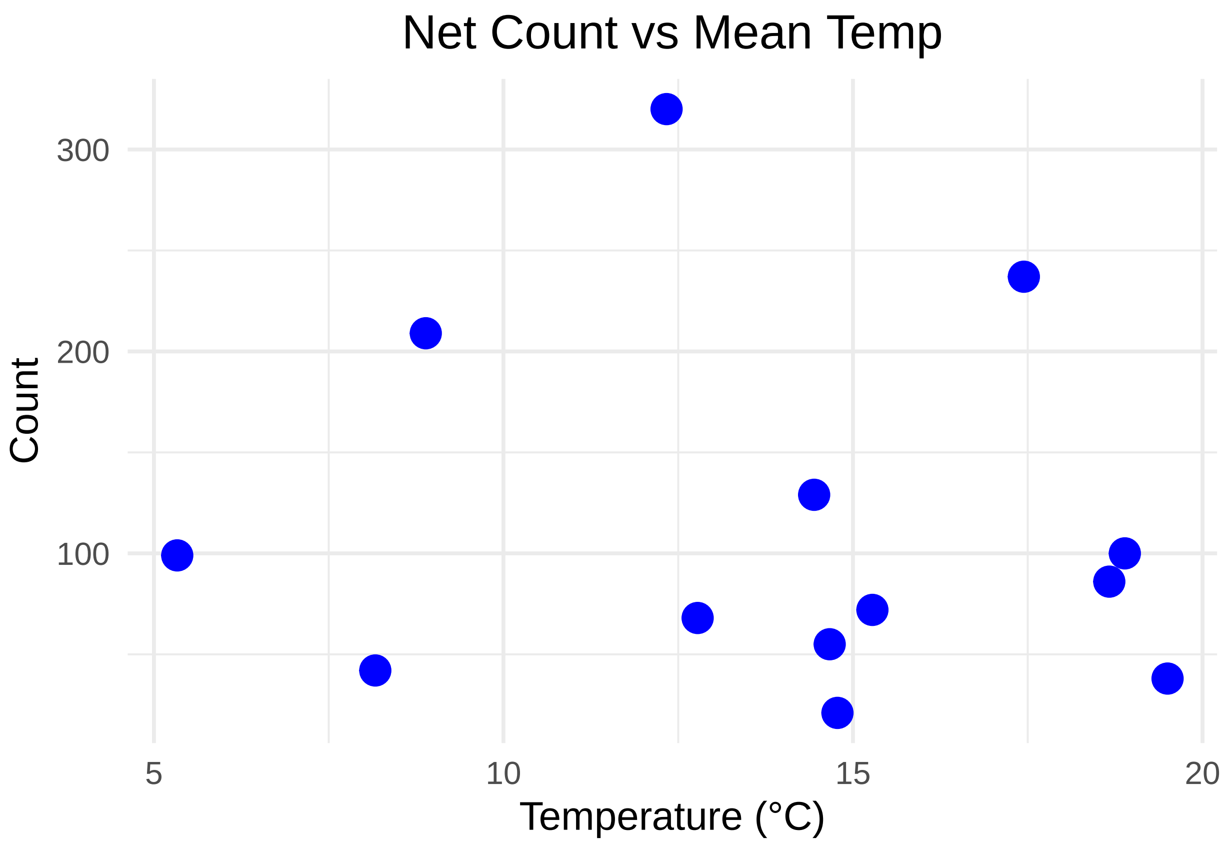
```
## # A tibble: 6 x 9
## # Groups:   year [1]
##    year month count date.x ppt_inch  ppt_cm date.y temp_f temp_c
##   <dbl> <dbl> <int>  <int>    <dbl>   <dbl>  <int>  <dbl>  <dbl>
## 1  2018     4    42 201804     6.55 16.6    201804   46.7   8.17
## 2  2018     5    68 201805     0.83  2.11   201805   55    12.8 
## 3  2018     6    55 201806     1.86  4.72   201806   58.4  14.7 
## 4  2018     7    38 201807     0.01  0.0254 201807   67.1  19.5 
## 5  2018     8    86 201808     0.13  0.330  201808   65.6  18.7 
## 6  2018     9    21 201809     0.41  1.04   201809   58.6  14.8 
```

```r
head(species_count_per_month)
```

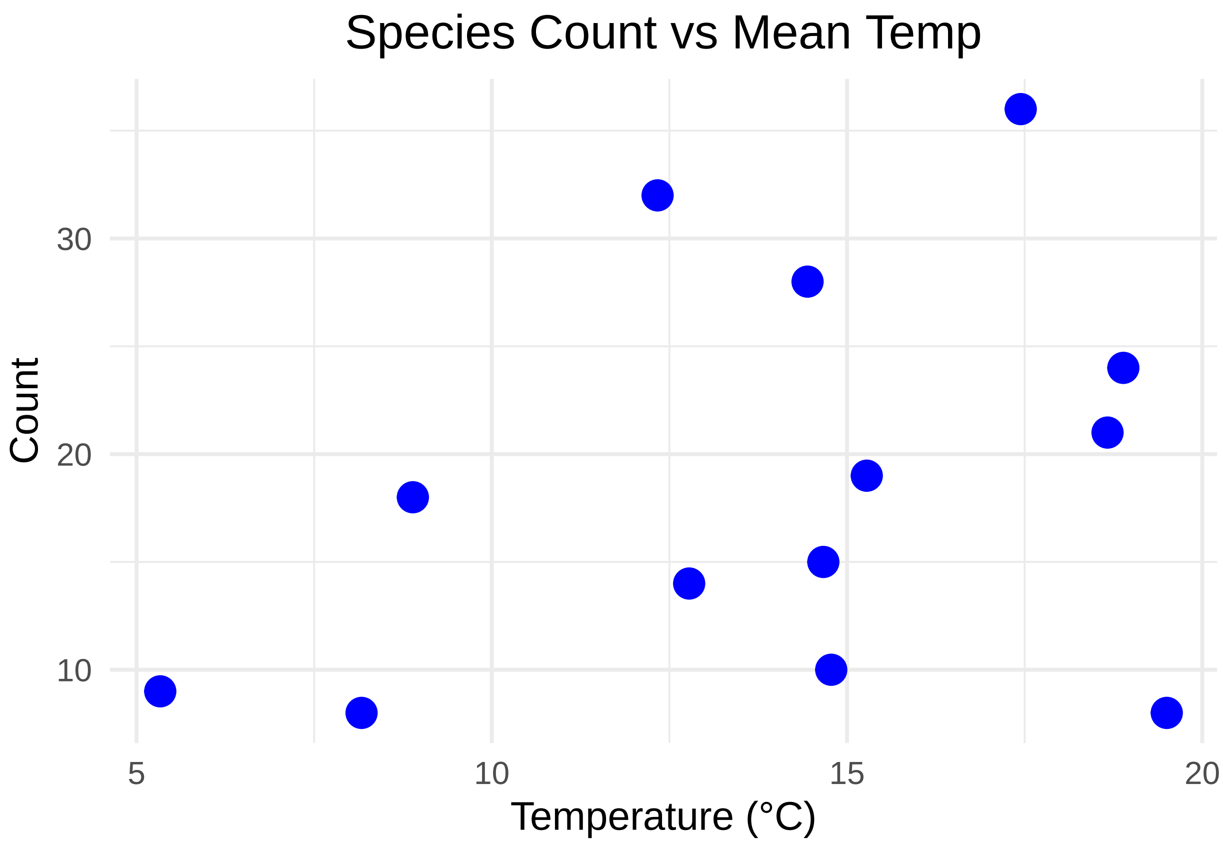
```
## # A tibble: 6 x 9
## # Groups:   year [1]
##    year month count date.x ppt_inch  ppt_cm date.y temp_f temp_c
##   <dbl> <dbl> <int>  <int>    <dbl>   <dbl>  <int>  <dbl>  <dbl>
## 1  2018     4     8 201804     6.55 16.6    201804   46.7   8.17
## 2  2018     5    14 201805     0.83  2.11   201805   55    12.8 
## 3  2018     6    15 201806     1.86  4.72   201806   58.4  14.7 
## 4  2018     7     8 201807     0.01  0.0254 201807   67.1  19.5 
## 5  2018     8    21 201808     0.13  0.330  201808   65.6  18.7 
## 6  2018     9    10 201809     0.41  1.04   201809   58.6  14.8 
```

Temperature effect

```r
# Scatter plot
ggplot(net_count_per_month) +
    geom_point(aes(x = temp_c, y = count), size = 5, color = "blue") +
    labs(x = "Temperature (°C)", y = "Count") +
    ggtitle("Net Count vs Mean Temp") +
    theme_minimal(15) +
    theme(plot.title = element_text(hjust = 0.5))
```



```r
ggplot(species_count_per_month) +
    geom_point(aes(x = temp_c, y = count), size = 5, color = "blue") +
    labs(x = "Temperature (°C)", y = "Count") +
    ggtitle("Species Count vs Mean Temp") +
    theme_minimal(15) +
    theme(plot.title = element_text(hjust = 0.5))
```



```
lin_reg <- lm(species_count_per_month$count ~ species_count_per_month$temp_c)
lin_reg_slope <- summary(lin_reg)$coefficients[2]
lin_reg_pval <- summary(lin_reg)$coefficients[7]
summary(lin_reg)
```

```
##
## Call:
## lm(formula = species_count_per_month$count ~ species_count_per_month$temp_c)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.634  -4.143  -1.032   3.030  14.851
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      8.5498     8.7324   0.979    0.349
## species_count_per_month$temp_c   0.7223     0.5999   1.204    0.254
##
## Residual standard error: 9.1 on 11 degrees of freedom
## Multiple R-squared:  0.1164, Adjusted R-squared:  0.03613
## F-statistic:  1.45 on 1 and 11 DF,  p-value: 0.2538
```

```
print(lin_reg_slope)
```

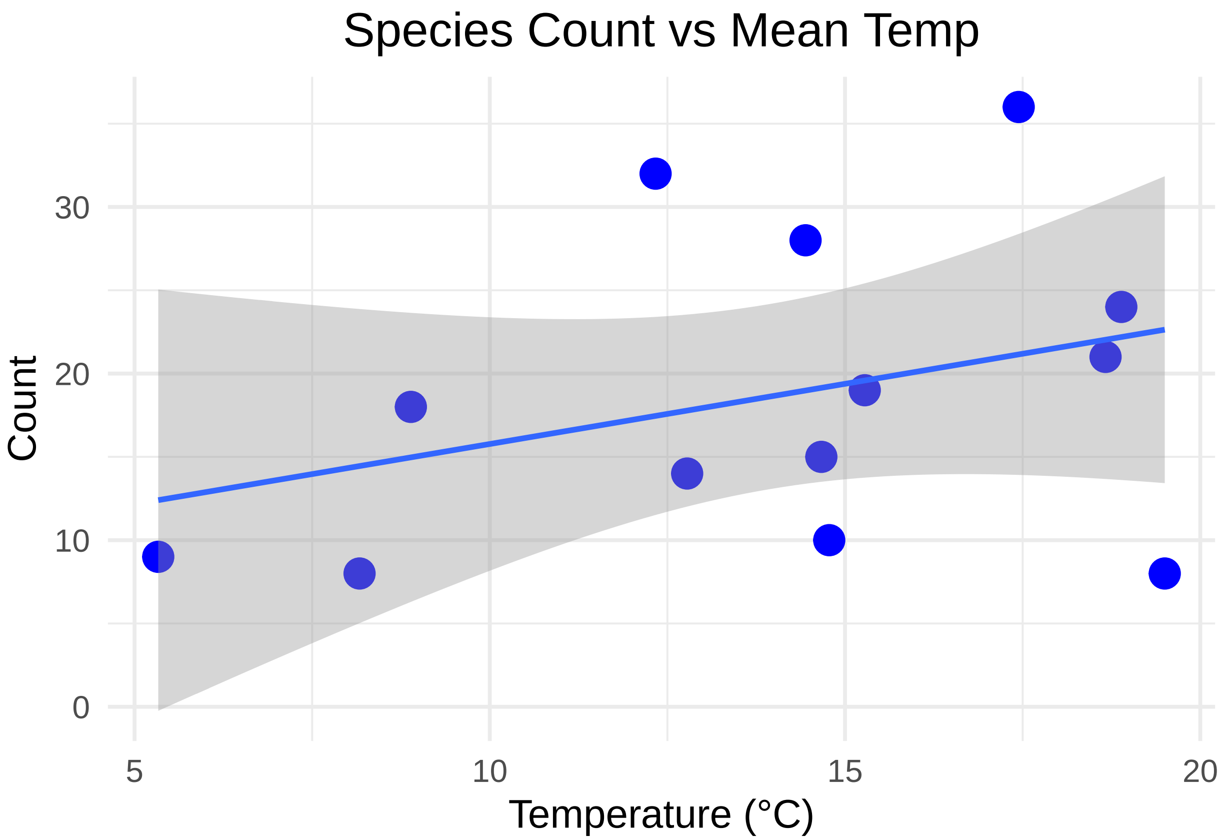
```
## [1] 0.7222794
```

```
print(lin_reg_pval)
```

```
## [1] 0.3485819
```

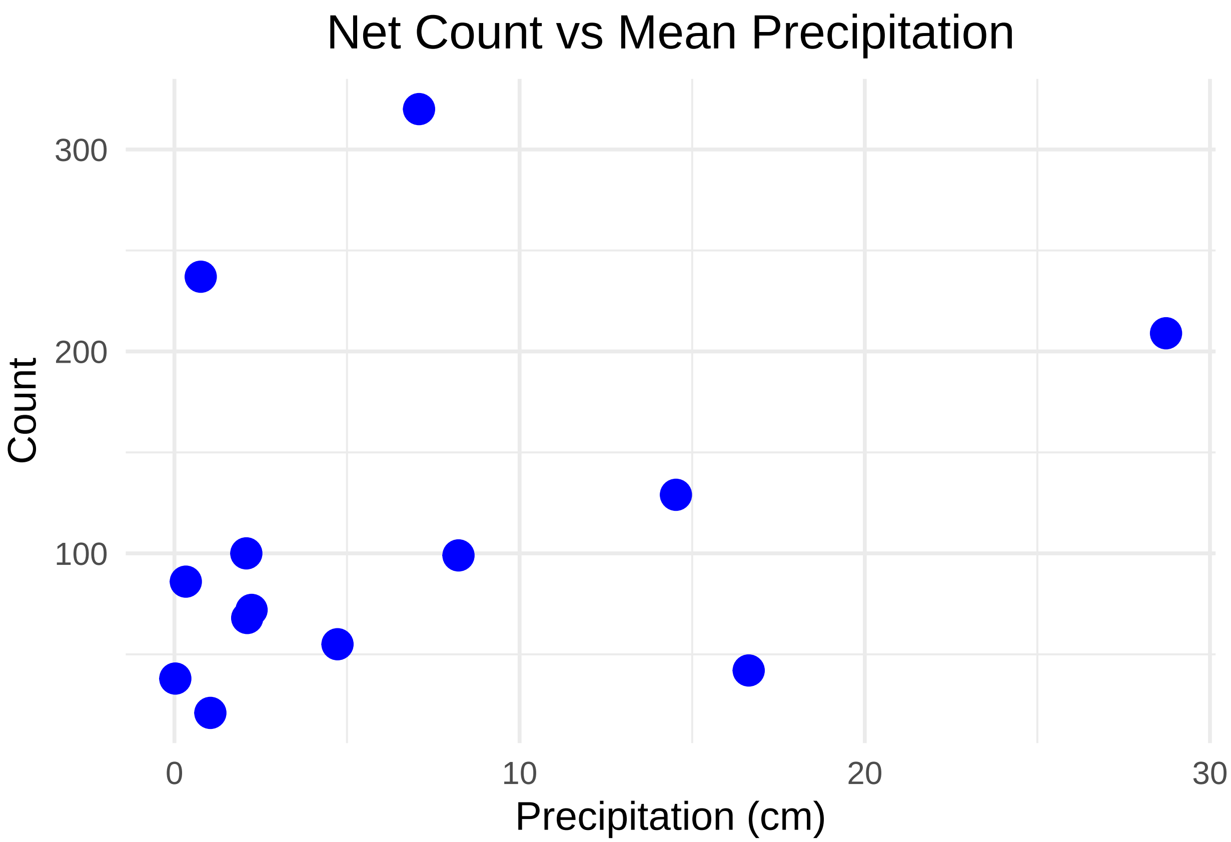
```
ggplot(species_count_per_month, aes(x = temp_c, y = count)) +
    geom_point(size = 5, color = "blue") +
    labs(x = "Temperature (°C)", y = "Count") +
    ggtitle("Species Count vs Mean Temp") +
    theme_minimal(15) +
    geom_smooth(method = "lm", se = TRUE) +
    theme(plot.title = element_text(hjust = 0.5))
```

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

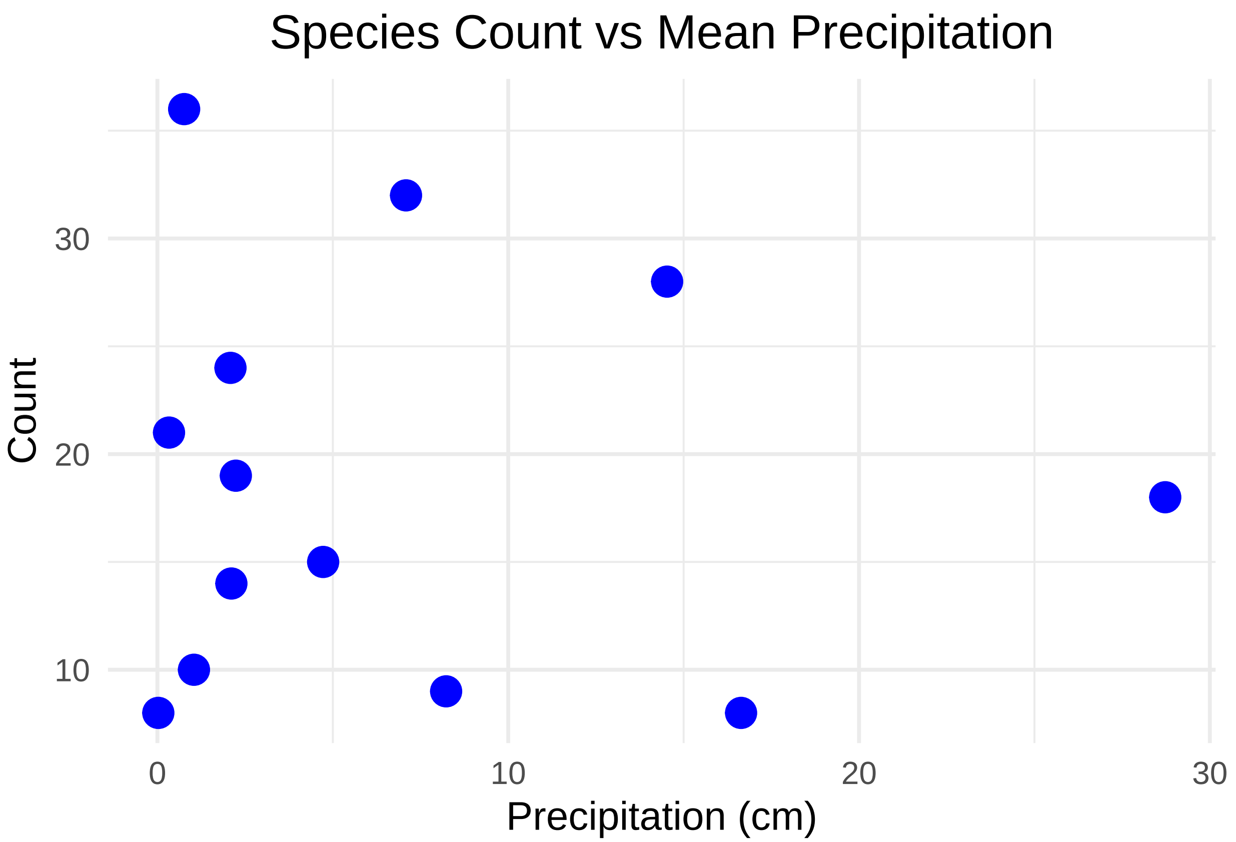


### Precipitation effect

```
# Scatter plot
ggplot(net_count_per_month) +
    geom_point(aes(x = ppt_cm, y = count), size = 5, color = "blue") +
    labs(x = "Precipitation (cm)", y = "Count") +
    ggtitle("Net Count vs Mean Precipitation") +
    theme_minimal(15)+
    theme(plot.title = element_text(hjust = 0.5))
```



```
ggplot(species_count_per_month) +
    geom_point(aes(x = ppt_cm, y = count), size = 5, color = "blue") +
    labs(x = "Precipitation (cm)", y = "Count") +
    ggtitle("Species Count vs Mean Precipitation") +
    theme_minimal(15) +
    theme(plot.title = element_text(hjust = 0.5))
```



```
lin_reg <- lm(net_count_per_month$count ~ species_count_per_month$ppt_cm)
lin_reg_slope <- summary(lin_reg)$coefficients[2]
lin_reg_pval <- summary(lin_reg)$coefficients[7]
summary(lin_reg)
```

```
##
## Call:
## lm(formula = net_count_per_month$count ~ species_count_per_month$ppt_cm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -101.189  -52.249  -18.824    0.721  205.624
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      92.995     32.139   2.893   0.0146 *
## species_count_per_month$ppt_cm    3.017      3.023   0.998   0.3398
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 88.99 on 11 degrees of freedom
## Multiple R-squared:  0.08301,    Adjusted R-squared:  -0.0003575
## F-statistic: 0.9957 on 1 and 11 DF,  p-value: 0.3398
```

```
print(lin_reg_slope)
```

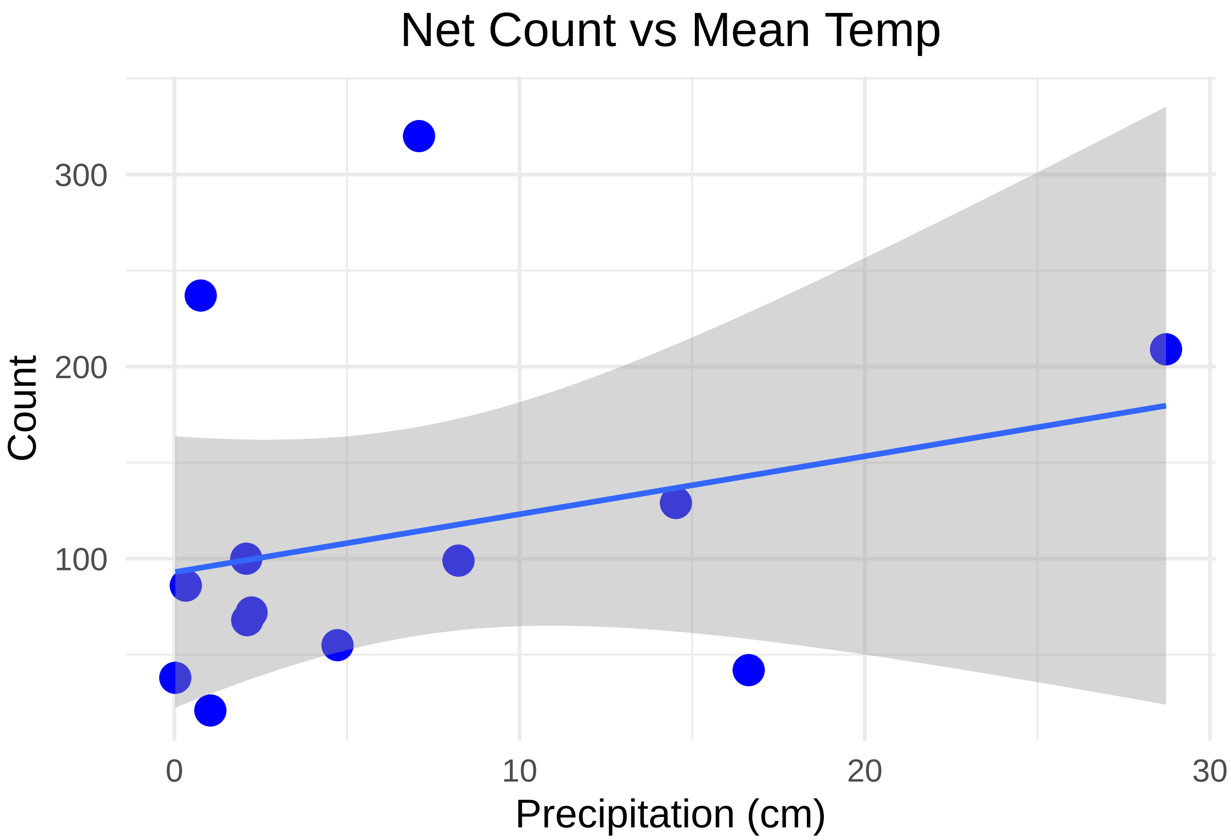
```
## [1] 3.01698
```

```r
print(lin_reg_pval)
```

```
## [1] 0.01461421
```

```r
ggplot(net_count_per_month, aes(x = ppt_cm, y = count)) +
    geom_point(size = 5, color = "blue") +
    labs(x = "Precipitation (cm)", y = "Count") +
    ggtitle("Net Count vs Mean Temp") +
    theme_minimal(15) +
    geom_smooth(method = "lm", se = TRUE) +
    theme(plot.title = element_text(hjust = 0.5))
```

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



## Oregon

```r
# get climate data for Oregon
# Read csv files
avg_temp <- read.csv("data/oregon_avg_temp_2018_2019.csv", skip = 3, header = 4)
avg_ppt <- read.csv("data/oregon_precipitation_2018_2019.csv", skip = 3, header = 4)

# rename columns
colnames(avg_temp)[colnames(avg_temp) == "Date"] <- "date"
colnames(avg_temp)[colnames(avg_temp) == "Value"] <- "temp_f"
colnames(avg_ppt)[colnames(avg_ppt) == "Date"] <- "date"
colnames(avg_ppt)[colnames(avg_ppt) == "Value"] <- "ppt_inch"

# add column 'year' and 'month' from the first four string character from the 'Date' column
```

```r
avg_temp$year <- substr(avg_temp$date, 1, 4)
avg_temp$month <- substr(avg_temp$date, 5, 6)
avg_temp$year <- as.numeric(avg_temp$year)
avg_temp$month <- as.numeric(avg_temp$month)
avg_temp$temp_c <- (avg_temp$temp_f - 32) * (5 / 9) # add celsius
head(avg_temp)
```

```
##     date temp_f year month     temp_c
## 1 201801   37.5 2018     1  3.0555556
## 2 201802   33.2 2018     2  0.6666667
## 3 201803   38.4 2018     3  3.5555556
## 4 201804   45.0 2018     4  7.2222222
## 5 201805   55.4 2018     5 13.0000000
## 6 201806   58.5 2018     6 14.7222222
```

```r
avg_ppt$year <- substr(avg_ppt$date, 1, 4)
avg_ppt$month <- substr(avg_ppt$date, 5, 6)
avg_ppt$year <- as.numeric(avg_ppt$year)
avg_ppt$month <- as.numeric(avg_ppt$month)
avg_ppt$ppt_cm <- avg_ppt$ppt_inch * 2.54 # add cm ppt
head(avg_ppt)
```

```
##     date ppt_inch year month  ppt_cm
## 1 201801     4.14 2018     1 10.5156
## 2 201802     2.29 2018     2  5.8166
## 3 201803     3.42 2018     3  8.6868
## 4 201804     3.26 2018     4  8.2804
## 5 201805     1.27 2018     5  3.2258
## 6 201806     1.13 2018     6  2.8702
```

```r
# find the count of total number of 'gbifID' by year and month in oba_2018_2019
net_count_per_month <- oba_2018_2019 %>%
    group_by(year, month) %>%
    summarise(count = n_distinct(gbifID))
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```r
species_count_per_month <- oba_2018_2019 %>%
    filter(species != "") %>%
    group_by(year, month) %>%
    summarise(count = n_distinct(species))
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```r
# get and add ppt_cm information from avg_ppt df by year and month
net_count_per_month <- net_count_per_month %>%
    left_join(avg_ppt, by = c("year", "month")) %>%
    left_join(avg_temp, by = c("year", "month"))

species_count_per_month <- species_count_per_month %>%
    left_join(avg_ppt, by = c("year", "month")) %>%
    left_join(avg_temp, by = c("year", "month"))

head(net_count_per_month)
```

```
## # A tibble: 6 x 9
## # Groups:   year [1]
##    year month count date.x ppt_inch ppt_cm date.y temp_f temp_c
##   <dbl> <dbl> <int>  <int>    <dbl>  <dbl>  <int>  <dbl>  <dbl>
## 1  2017     3     1     NA       NA     NA     NA     NA     NA
## 2  2017     5    31     NA       NA     NA     NA     NA     NA
## 3  2017     6    14     NA       NA     NA     NA     NA     NA
## 4  2017     7    70     NA       NA     NA     NA     NA     NA
## 5  2017     8    70     NA       NA     NA     NA     NA     NA
## 6  2017     9    43     NA       NA     NA     NA     NA     NA
```

```
head(species_count_per_month)
```

```
## # A tibble: 6 x 9
## # Groups:   year [1]
##    year month count date.x ppt_inch ppt_cm date.y temp_f temp_c
##   <dbl> <dbl> <int>  <int>    <dbl>  <dbl>  <int>  <dbl>  <dbl>
## 1  2017     5     4     NA       NA     NA     NA     NA     NA
## 2  2017     6     7     NA       NA     NA     NA     NA     NA
## 3  2017     7    15     NA       NA     NA     NA     NA     NA
## 4  2017     8    20     NA       NA     NA     NA     NA     NA
## 5  2017     9    12     NA       NA     NA     NA     NA     NA
## 6  2017    10     2     NA       NA     NA     NA     NA     NA
```

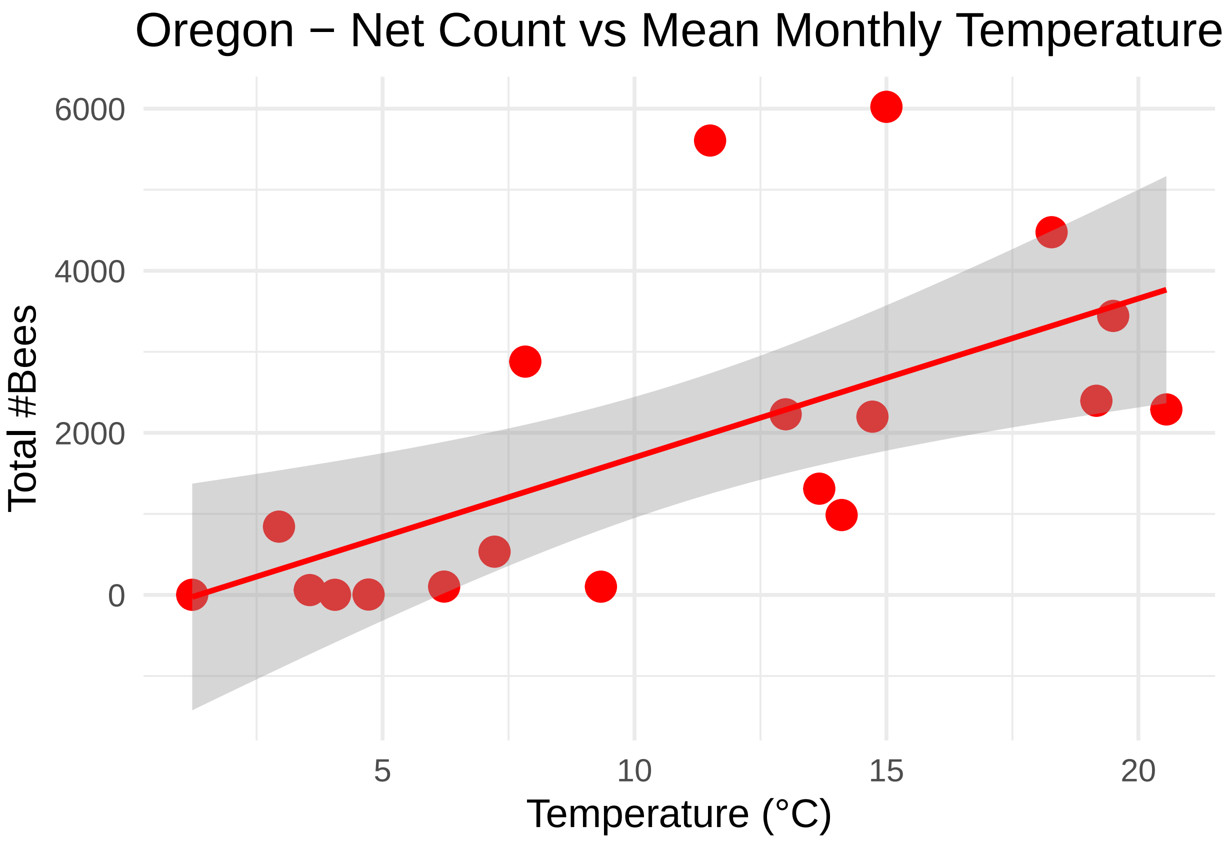
Temperature Effect

```
# Scatter plot
lin_reg <- lm(net_count_per_month$count ~ net_count_per_month$temp_c)
# lin_reg_slope <- summary(lin_reg)$coefficients[2]
# lin_reg_pval <- summary(lin_reg)$coefficients[7]
summary(lin_reg)
```

```
##
## Call:
## lm(formula = net_count_per_month$count ~ net_count_per_month$temp_c)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1516.5  -975.3  -422.3   278.2  3616.4
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -264.18     724.20  -0.365  0.71977
## net_count_per_month$temp_c   196.07      58.28   3.364  0.00368 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1528 on 17 degrees of freedom
##   (8 observations deleted due to missingness)
## Multiple R-squared:  0.3997, Adjusted R-squared:  0.3644
## F-statistic: 11.32 on 1 and 17 DF,  p-value: 0.003681
```

```
p_title = "Oregon - Net Count vs Mean Monthly Temperature"
p <- ggplot(net_count_per_month, aes(x = temp_c, y = count)) +
    geom_point(size = 5, color = "#ff0000") +
    labs(x = "Temperature (°C)", y = "Total #Bees") +
    ggtitle(p_title) +
    theme_minimal(15) +
    geom_smooth(method = "lm", se = TRUE, color="#ff0000") +
    theme(plot.title = element_text(hjust = 0.5))
p
```

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

## Warning: Removed 8 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 8 rows containing missing values (`geom_point()`).
```



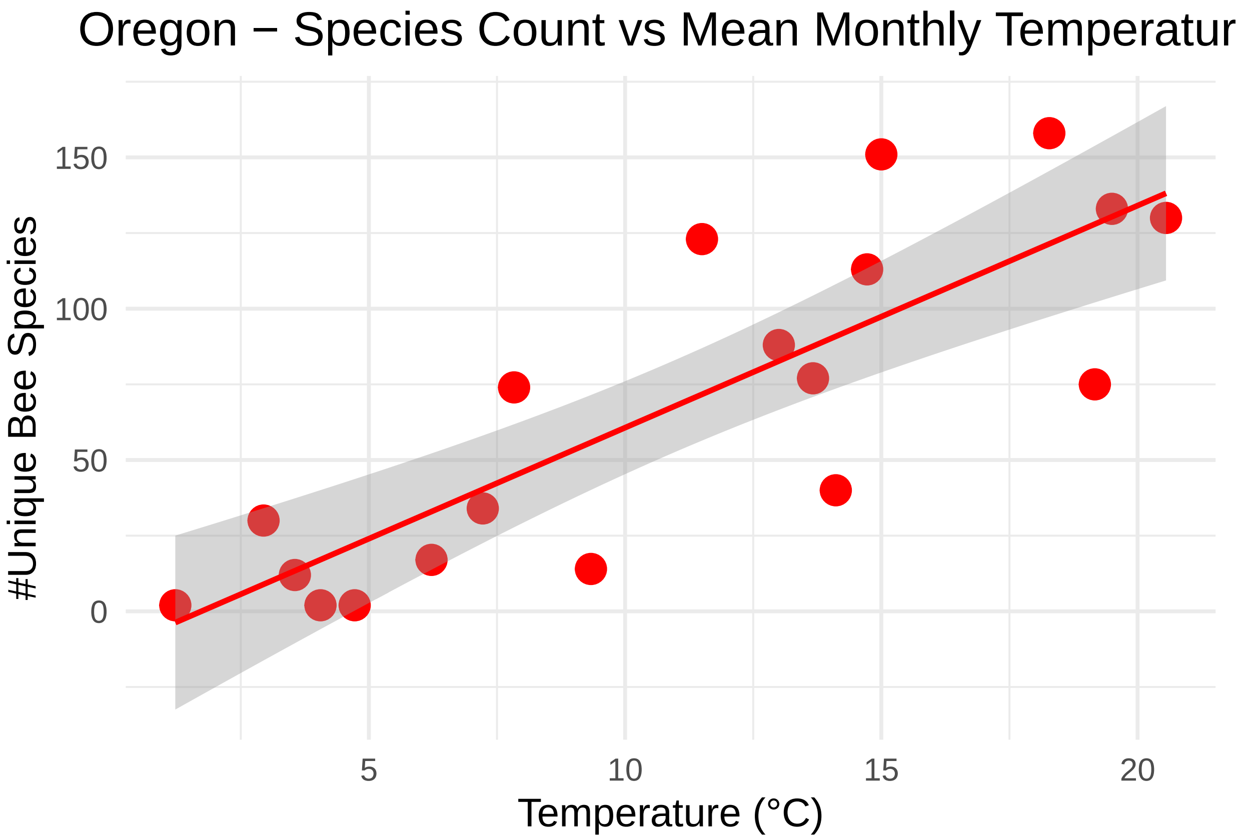
```
# ggsave(glue("plots/{p_title}.png"), dpi = 300, plot = p)


lin_reg <- lm(species_count_per_month$count ~ species_count_per_month$temp_c)
summary(lin_reg)
```

```
##
## Call:
## lm(formula = species_count_per_month$count ~ species_count_per_month$temp_c)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -52.952 -15.521  -1.403  19.370  53.621
##
```

```
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -12.687     14.883  -0.852    0.406
## species_count_per_month$temp_c    7.338      1.198   6.126 1.12e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.4 on 17 degrees of freedom
##   (7 observations deleted due to missingness)
## Multiple R-squared:  0.6883, Adjusted R-squared:  0.6699
## F-statistic: 37.53 on 1 and 17 DF,  p-value: 1.119e-05
```

```
p_title <- "Oregon - Species Count vs Mean Monthly Temperature"
p <- ggplot(species_count_per_month, aes(x = temp_c, y = count)) +
    geom_point(size = 5, color = "#ff0000") +
    labs(x = "Temperature (°C)", y = "#Unique Bee Species") +
    ggtitle(p_title) +
    theme_minimal(15) +
    geom_smooth(method = "lm", se = TRUE, color = "#ff0000") +
    theme(plot.title = element_text(hjust = 0.5))
p
```

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

## Warning: Removed 7 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 7 rows containing missing values (`geom_point()`).
```



```
# ggsave(glue("plots/{p_title}.png"), dpi = 300, plot = p)
```

Precipitation effect

```
lin_reg <- lm(net_count_per_month$count ~ net_count_per_month$ppt_cm)
# lin_reg_slope <- summary(lin_reg)$coefficients[2]
# lin_reg_pval <- summary(lin_reg)$coefficients[7]
summary(lin_reg)
```

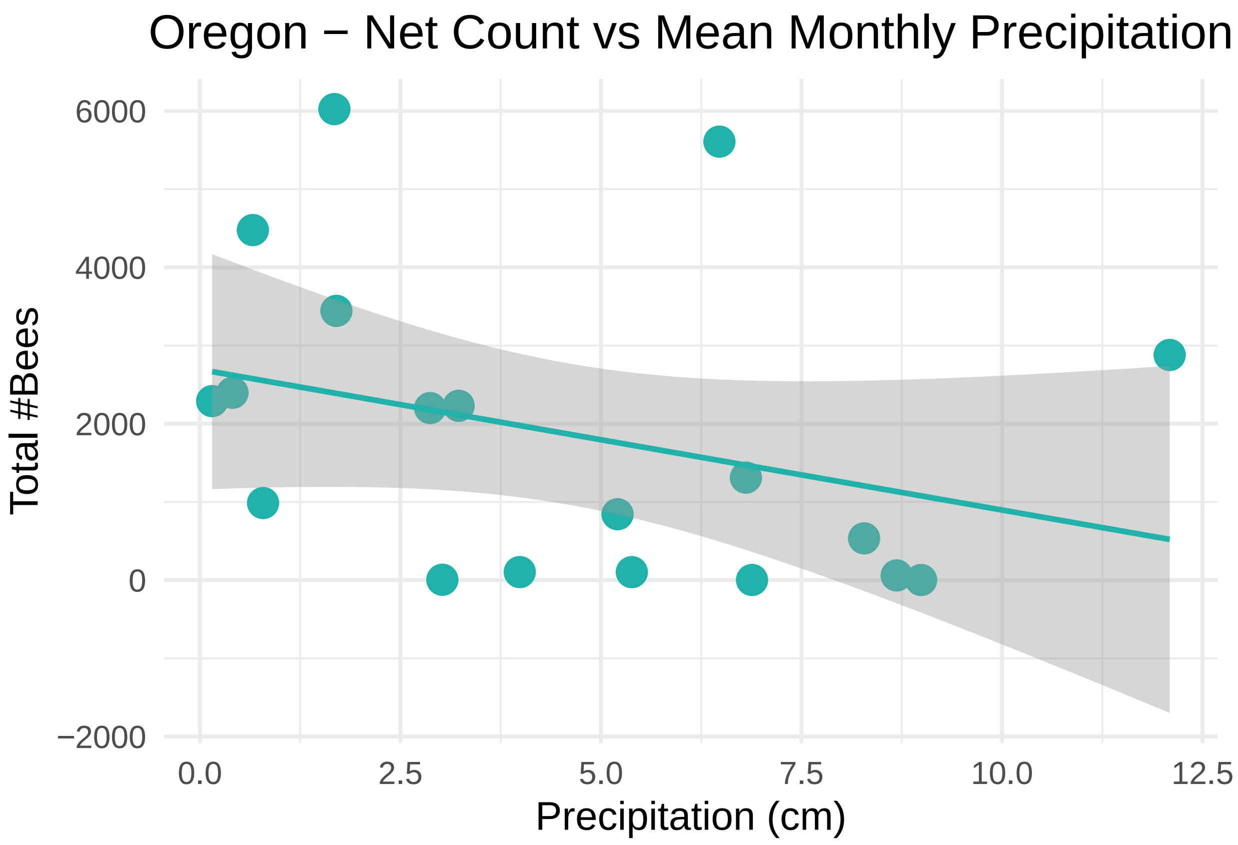
```
##
## Call:
## lm(formula = net_count_per_month$count ~ net_count_per_month$ppt_cm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2145.4 -1265.1  -377.3   585.1  4077.4
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   2693.7      727.9   3.700  0.00178 **
## net_count_per_month$ppt_cm    -179.7      128.1  -1.403  0.17865
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1867 on 17 degrees of freedom
##   (8 observations deleted due to missingness)
## Multiple R-squared:  0.1038, Adjusted R-squared:  0.05104
## F-statistic: 1.968 on 1 and 17 DF,  p-value: 0.1786
```

```
p_title <- "Oregon - Net Count vs Mean Monthly Precipitation"
p <- ggplot(net_count_per_month, aes(x = ppt_cm, y = count)) +
    geom_point(size = 5, color = "lightseagreen") +
    labs(x = "Precipitation (cm)", y = "Total #Bees") +
    ggtitle(p_title) +
    theme_minimal(15) +
    geom_smooth(method = "lm", se = TRUE, color = "lightseagreen") +
    theme(plot.title = element_text(hjust = 0.5))
p
```

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

```
## Warning: Removed 8 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 8 rows containing missing values (`geom_point()`).
```

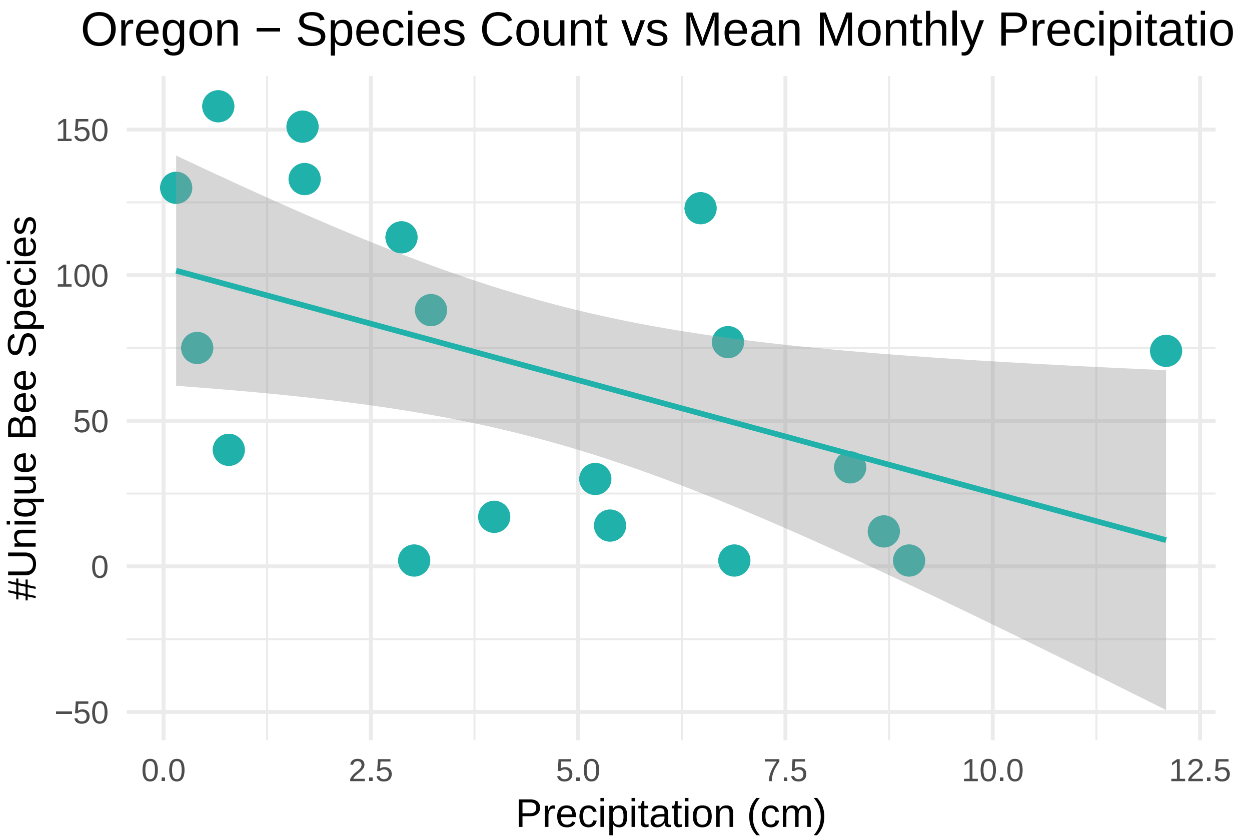


```r
# ggsave(glue("plots/{p_title}.png"), dpi = 300, plot = p)

lin_reg <- lm(species_count_per_month$count ~ species_count_per_month$ppt_cm)
summary(lin_reg)
```

```
##
## Call:
## lm(formula = species_count_per_month$count ~ species_count_per_month$ppt_cm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -77.292 -39.670  -4.536  37.999  70.485
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     102.721     19.151   5.364 5.16e-05 ***
## species_count_per_month$ppt_cm   -7.751      3.370  -2.300   0.0344 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 49.11 on 17 degrees of freedom
##   (7 observations deleted due to missingness)
## Multiple R-squared:  0.2373, Adjusted R-squared:  0.1924
## F-statistic:  5.29 on 1 and 17 DF,  p-value: 0.03439
```

```r
p_title <- "Oregon - Species Count vs Mean Monthly Precipitation"
p <- ggplot(species_count_per_month, aes(x = ppt_cm, y = count)) +
    geom_point(size = 5, color = "lightseagreen") +
    labs(x = "Precipitation (cm)", y = "#Unique Bee Species") +
    ggtitle(p_title) +
    theme_minimal(15) +
    geom_smooth(method = "lm", se = TRUE, color = "lightseagreen") +
    theme(plot.title = element_text(hjust = 0.5))
p
```

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

## Warning: Removed 7 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 7 rows containing missing values (`geom_point()`).
```



```
# ggsave(glue("plots/{p_title}.png"), dpi = 300, plot = p)
```

Results

1. A. Effect of Sampling Protocols: Are different sampling protocols more likely to capture bees of one sex than another?

- Yes! Trap nests capture a higher percentage of male bees. For example, in a colony, the male-to-female ratio is typically 1:100, while for solitary bees, it's closer to 1:1 (Willmer, 2011).

1. B. Effect of Human Population (Marker for Urbanization): How does the human population level affect bee distribution and sex ratio?

- There is a higher occurrence of bees near population centers. However, this observation comes with a caveat, as it can be influenced by sampling bias. My studies have not shown a clear difference in the sex ratio of bees in urbanized areas compared to other habitats.

2. Effect of Local Weather on Bee Populations: How does the number of foraging bees change with temperature and precipitation?

- In my study, I found that the number of unique bee species significantly increases with increasing average monthly temperature. Conversely, the number of foraging bees tends to significantly decrease with increasing average monthly precipitation. These findings highlight the intricate relationship between local weather patterns and bee populations, emphasizing the importance of considering climatic factors in bee conservation efforts.