exploration

2024-11-20

Bees and wildfires

This project aims to answer the question "how does wildfire severity affect bees?" We start by loading in the fire rasters and making sure the scales are consistent

Load Libraries

Load rasts into variables

```
# Load 2020 fires
beachie_2020 = rast("data/2020/beachie creek willamette 2020/BeachieCreek_SBS_final.tif")
freemont_2020 = rast("data/2020/freemont winema 2020/CRANE_sbs.tif")
holiday_2020 = rast("data/2020/holiday farm willamette 2020/HolidayFarm_SBS_final.tif")
hood_2020 = rast("data/2020/mt hood 2020/Riverside_SBS_final.tif")
umpqua_2020 = rast("data/2020/umpqua 2020/ArchieCreek_SBS_final.tif")

ben_young_2020 = rast("data/BEN_YOUNG_sbs/BEN_YOUNG_sbs.tif")
brattain_2020 = rast("data/BRATTAIN_sbs/BRATTAIN_sbs.tif")
lionshead_2020 = rast("data/lionshead_sbs/Lionshead_SBS_final.tif")
thielsen_2020 = rast("data/thielsen_sbs/Thielsen_SBS_final.tif")
two_four_2020 = rast("data/TWO_FOUR_TWO_sbs/TWO_FOUR_TWO_sbs.tif")
whiteriver_2020 = rast("data/whiteriver_sbs/WhiteRiver_SBS_final.tif")
```

Clean our raster data to have the same scale and extents

```
# Adjust the scale of all fire rasters to be 1-4
adjust_scale <- function(raster) {
    raster[raster > 4] <- NA
    return(raster)
}

# Align raster extents to a reference raster
align_extent <- function(raster, reference_raster) {
  return(extend(raster, ext(reference_raster)))
}

# Apply the function and reassign
beachie_2020 <- adjust_scale(beachie_2020)
freemont_2020 <- adjust_scale(freemont_2020)
holiday_2020 <- adjust_scale(holiday_2020)
hood_2020 <- adjust_scale(hood_2020)
umpqua_2020 <- adjust_scale(umpqua_2020)
ben_young_2020 <- adjust_scale(ben_young_2020)
```

```r
brattain_2020 <- adjust_scale(brattain_2020)
lionshead_2020 <- adjust_scale(lionshead_2020)
thielsen_2020 <- adjust_scale(thielsen_2020)
two_four_2020 <- adjust_scale(two_four_2020)
whiteriver_2020 <- adjust_scale(whiteriver_2020)
```

Continue cleaning by reprojecting and converting the rasters to continuous dataframes

```r
# Reproject raster data function
reproject_raster <- function(raster, reference_raster) {
  return(project(raster, crs(reference_raster), res = res(reference_raster)))
}

# Convert raster to dataframe function
raster_to_dataframe <- function(raster) {
  return(as.data.frame(raster, xy = TRUE))
}

# Assign nuneric values function for categorical variables for consistency
transform_soil_burn <- function(df) {
  # Check if 'SoilBurnSe' exists in the dataframe
  if ("SoilBurnSe" %in% colnames(df)) {
    df$SoilBurnSe <- as.character(df$SoilBurnSe)
    df$SoilBurnSe[df$SoilBurnSe == "High"] <- "4"
    df$SoilBurnSe[df$SoilBurnSe == "Moderate"] <- "3"
    df$SoilBurnSe[df$SoilBurnSe == "Low"] <- "2"
    df$SoilBurnSe[df$SoilBurnSe == "Unburned"] <- "1"
    df$SoilBurnSe <- as.factor(df$SoilBurnSe)
  } else{}
  return(df)
}

# Load and transform to a shapefile function
load_and_transform_shapefile <- function(shapefile_path, reference_raster) {
  shapefile <- st_read(shapefile_path)
  return(st_transform(shapefile, crs(reference_raster)))
}

# Function to process a list of rasters
process_rasters <- function(raster, reference_raster) {
  # Reproject the raster
  raster <- reproject_raster(raster, reference_raster)

  # Convert raster to data frame
  raster_df <- raster_to_dataframe(raster)

  return(raster_df)
}
```

Apply cleaning functions created

```
# Apply the function for each fire
beachie_2020_df <- process_rasters(beachie_2020, beachie_2020)
freemont_2020_df <- process_rasters(freemont_2020, beachie_2020)
holiday_2020_df <- process_rasters(holiday_2020, beachie_2020)
hood_2020_df <- process_rasters(hood_2020, beachie_2020)
umpqua_2020_df <- process_rasters(umpqua_2020, beachie_2020)

ben_young_2020_df <- process_rasters(ben_young_2020, beachie_2020)
brattain_2020_df <- process_rasters(brattain_2020, beachie_2020)
lionshead_2020_df <- process_rasters(lionshead_2020, beachie_2020)
thielsen_2020_df <- process_rasters(thielsen_2020, beachie_2020)
two_four_2020_df <- process_rasters(two_four_2020, beachie_2020)
whiteriver_2020_df <- process_rasters(whiteriver_2020, beachie_2020)

# Convert the columns to numeric (continuous) if they aren't already
beachie_2020_df$Layer_1 <- as.numeric(beachie_2020_df$Layer_1)
freemont_2020_df$Layer_1 <- as.numeric(freemont_2020_df$Layer_1)
holiday_2020_df$Layer_1 <- as.numeric(holiday_2020_df$Layer_1)
hood_2020_df$Layer_1 <- as.numeric(hood_2020_df$Layer_1)
umpqua_2020_df$Layer_1 <- as.numeric(umpqua_2020_df$Layer_1)

ben_young_2020_df$Layer_1 <- as.numeric(ben_young_2020_df$Layer_1)
brattain_2020_df$Layer_1 <- as.numeric(brattain_2020_df$Layer_1)
lionshead_2020_df$Layer_1 <- as.numeric(lionshead_2020_df$Layer_1)
thielsen_2020_df$Layer_1 <- as.numeric(thielsen_2020_df$Layer_1)
two_four_2020_df$Layer_1 <- as.numeric(two_four_2020_df$Layer_1)
whiteriver_2020_df$Layer_1 <- as.numeric(whiteriver_2020_df$Layer_1)
```

Load ecoregions dataset

```
oregon_ecoregions <- st_read("data/OR-ecoregions/Ecoregions_OregonConservationStrategy.shp")
```

```
## Reading layer `Ecoregions_OregonConservationStrategy' from data source
##   `/Users/madikloberdanz/Documents/bee_repo/data/OR-ecoregions/Ecoregions_OregonConservationStrategy
##   using driver `ESRI Shapefile'
## Simple feature collection with 9 features and 6 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 183871.7 ymin: 88600.88 xmax: 2345213 ymax: 1675043
## Projected CRS: NAD83 / Oregon GIC Lambert (ft)
```

```
oregon_ecoregions <- st_transform(oregon_ecoregions, crs(beachie_2020))
```

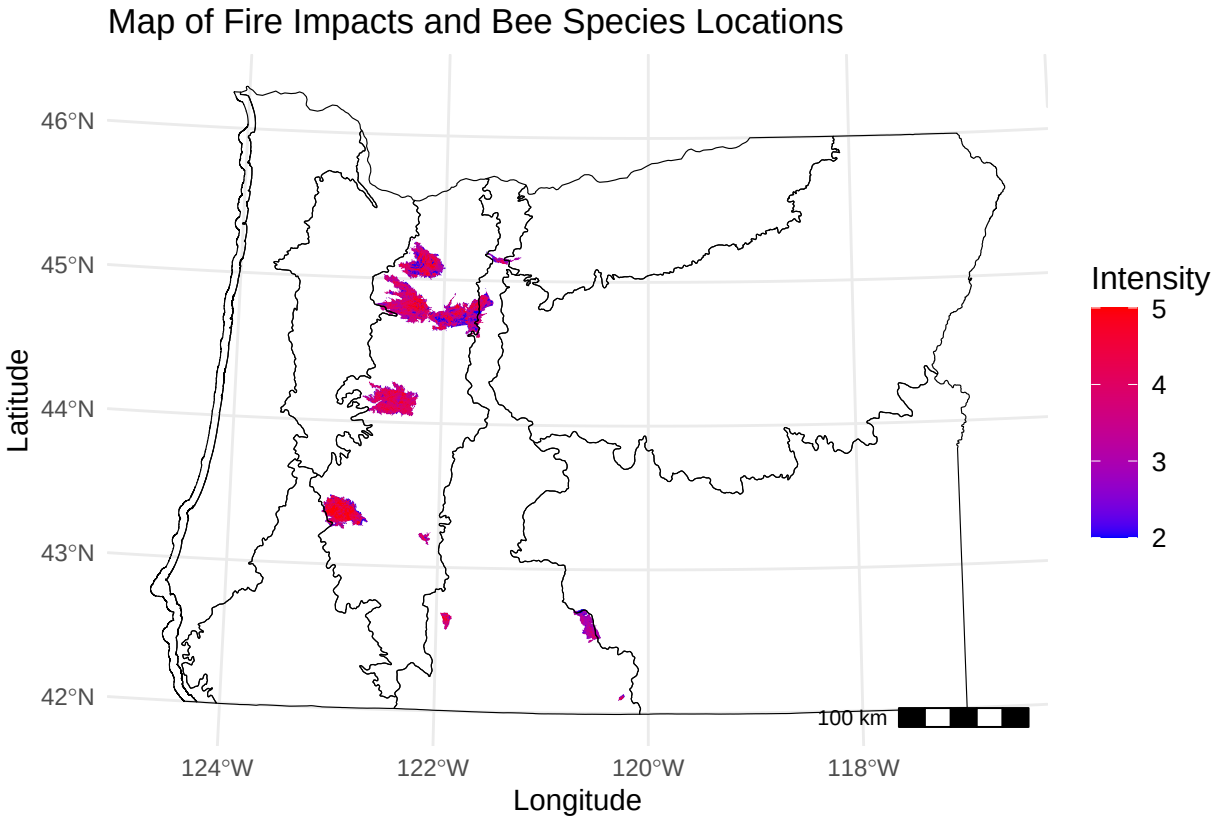
Initial visualization of fire rasts and ecoregions datasets

Plot all of the 2020 fires on the oregon map to see the severities and general regions.

```
ggplot() +
  geom_raster(data = beachie_2020_df, aes(x = x, y = y, fill = Layer_1)) +
  geom_raster(data = freemont_2020_df, aes(x = x, y = y, fill = Layer_1)) +
  geom_raster(data = holiday_2020_df, aes(x = x, y = y, fill = Layer_1)) +
```

```
  geom_raster(data = hood_2020_df, aes(x = x, y = y, fill = Layer_1)) +
  geom_raster(data = umpqua_2020_df, aes(x = x, y = y, fill = Layer_1)) +
  geom_raster(data = ben_young_2020_df, aes(x = x, y = y, fill = Layer_1)) +
  geom_raster(data = brattain_2020_df, aes(x = x, y = y, fill = Layer_1)) +
  geom_raster(data = lionshead_2020_df, aes(x = x, y = y, fill = Layer_1)) +
  geom_raster(data = thielsen_2020_df, aes(x = x, y = y, fill = Layer_1)) +
  geom_raster(data = two_four_2020_df, aes(x = x, y = y, fill = Layer_1)) +
  geom_raster(data = whiteriver_2020_df, aes(x = x, y = y, fill = Layer_1)) +

  # Add the Oregon ecoregions as a boundary layer
  geom_sf(data = oregon_ecoregions, fill = NA, color = "black") +

  # Customize colors
  scale_fill_gradient(low = "blue", high = "red", name = "Intensity") +

  # Add labels and scale
  labs(
    title = "Map of Fire Impacts and Bee Species Locations",
    x = "Longitude",
    y = "Latitude"
  ) +

  # Add annotation for scale
  annotation_scale(location = "br", width_hint = 0.2) +
  theme_minimal() +
  theme(legend.title = element_text(size = 12), legend.text = element_text(size = 10))
```

```
## Warning: Raster pixels are placed at uneven horizontal intervals and will be shifted
## i Consider using `geom_tile()` instead.
## Raster pixels are placed at uneven horizontal intervals and will be shifted
## i Consider using `geom_tile()` instead.
## Raster pixels are placed at uneven horizontal intervals and will be shifted
## i Consider using `geom_tile()` instead.
```



Initial look at the Oregon Bee Atlas data

First, let's read the OBA data and see what columns we have

```
oba_df <- read.csv("data/OBA_2018-2023.csv")
head(oba_df)
```

```
##                   Observation.No. Voucher.No. user_id     user_login
## 1 Andony_Melathopoulos:18.001.001              429964 amelathopoulos
## 2 Andony_Melathopoulos:18.002.001              429964 amelathopoulos
## 3 Andony_Melathopoulos:18.002.002              429964 amelathopoulos
## 4 Andony_Melathopoulos:18.002.003              429964 amelathopoulos
## 5 Andony_Melathopoulos:18.002.004              429964 amelathopoulos
## 6 Andony_Melathopoulos:18.002.005              429964 amelathopoulos
##   Collector...First.Name Collector...First.Initial Collector...Last.Name
## 1                 Andony                         A.          Melathopoulos
## 2                 Andony                         A.          Melathopoulos
## 3                 Andony                         A.          Melathopoulos
## 4                 Andony                         A.          Melathopoulos
## 5                 Andony                         A.          Melathopoulos
## 6                 Andony                         A.          Melathopoulos
##        Collectors taxon_kingdom_name Associated.plant...genus..species url
## 1 A.Melathopoulos
## 2 A.Melathopoulos
## 3 A.Melathopoulos
## 4 A.Melathopoulos
## 5 A.Melathopoulos
## 6 A.Melathopoulos
```

```
##   Sample.ID Specimen.ID Collection.Day.1 Month.1  MonthJul MonthAb Year.1
## 1                   NA              18     iii     March       3   2018
## 2                   NA              20     iii     March       3   2018
## 3                   NA              20     iii     March       3   2018
## 4                   NA              20     iii     March       3   2018
## 5                   NA               2      ix September       9   2018
## 6                   NA               2      ix September       9   2018
##   Collection.Date Time.1 Collection.Day.2 Month.2 Year.2 Collection.Day.2.Merge
## 1       3/18/2018
## 2       3/20/2018
## 3       3/20/2018
## 4       3/20/2018
## 5        9/2/2018
## 6        9/2/2018
##   Time.2    Collection.ID Position.of.1st.digit Collection.No. Sample.No.
## 1        A Melathopoulos                                   1          1
## 2         A Melathopoulos                                  2          1
## 3         A Melathopoulos                                  2          2
## 4         A Melathopoulos                                  2          3
## 5         A Melathopoulos                                  2          4
## 6         A Melathopoulos                                  2          5
##   Country  State  County                                    Location
## 1     USA Oregon  Benton                      Corvallis, NW Orchard Ave
## 2     USA Oregon  Benton                      Corvallis, NW Orchard Ave
## 3     USA Oregon  Benton                      Corvallis, NW Orchard Ave
## 4     USA Oregon  Benton                      Corvallis, NW Orchard Ave
## 5     USA Oregon Clatsop Clatskanie, Big Creek Mainline, Knob Point Road
## 6     USA Oregon Clatsop Clatskanie, Big Creek Mainline, Knob Point Road
##           Abbreviated.Location Collection.Site.Description          Team
## 1    Astoria Maggie Johnson Rd                              Melathopoulos
## 2 Big Crk. Mainline Knob Pt Rd                              Melathopoulos
## 3 Big Crk. Mainline Knob Pt Rd                              Melathopoulos
## 4 Big Crk. Mainline Knob Pt Rd                              Melathopoulos
## 5 Big Crk. Mainline Knob Pt Rd                              Melathopoulos
## 6 Big Crk. Mainline Knob Pt Rd                              Melathopoulos
##   Habitat Elevation..m. Dec..Lat. Dec..Long.  X Collectionmethod
## 1                        44.556   -123.285 NA              Net
## 2                        44.567   -123.283 NA              Net
## 3                        44.567   -123.283 NA              Net
## 4                        44.567   -123.283 NA              Net
## 5                        46.102   -123.506 NA              Net
## 6                        46.102   -123.506 NA              Net
##   Collection.method.merge.field Associated.plant...family
## 1
## 2
## 3
## 4
## 5
## 6
##   Associated.plant...genus..species.1 Associated.plant...Inaturalist.URL
## 1
## 2
## 3
## 4
```

```
## 5
## 6
##   Associated.plant Assoc.plant.merge.field          Collectors.1
## 1                                           Andony Melathopoulos
## 2                                           Andony Melathopoulos
## 3                                           Andony Melathopoulos
## 4                                           Andony Melathopoulos
## 5                                          Andony Melathopoulos
## 6                                          Andony Melathopoulos
##   Collector.1.abreviation Collector.2 Collector.3 Genus Species sex caste
## 1         A Melathopoulos           NA          NA
## 2          A Melathopoulos          NA          NA
## 3          A Melathopoulos          NA          NA
## 4          A Melathopoulos          NA          NA
## 5          A Melathopoulos          NA          NA
## 6          A Melathopoulos          NA          NA
##   vol.det.Genus vol.det.Species vol.det.sex.caste Determined.By Date.Determined
## 1                                                                           NA
## 2                                                                           NA
## 3                                                                           NA
## 4                                                                           NA
## 5                                                                           NA
## 6                                                                           NA
##   Verified.By Other.Determiner.s. Other.Dets.Sci..Name.s. Other.Dets..Date.s.
## 1          NA                                         NA                  NA
## 2          NA                                         NA                  NA
## 3          NA                                         NA                  NA
## 4          NA                                         NA                  NA
## 5          NA                                         NA                  NA
## 6          NA                                         NA                  NA
##   Additional.Notes X.1
## 1                   NA
## 2                   NA
## 3                   NA
## 4                   NA
## 5                   NA
## 6                   NA
```

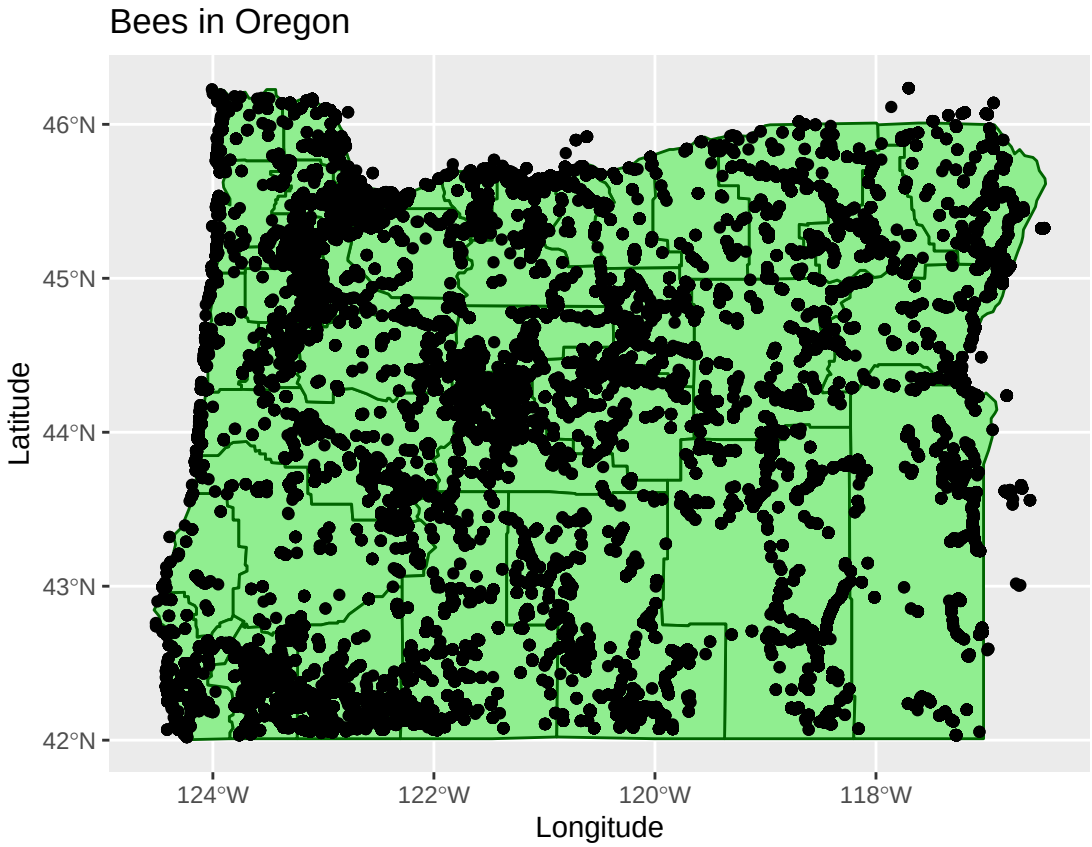
Now, let's plot all of the bees that are within Oregon on top of a map to see potential areas of interest. We initially discovered that not all of the OBA data is within the oregon bounds.

```
# Step 1: Retrieve Oregon county polygon data
or <- map_data("county", "oregon") %>%
  select(lon = long, lat, group, id = subregion)

# Convert the Oregon polygon to an sf object (simple feature)
or_sf <- st_as_sf(or, coords = c("lon", "lat"), crs = 4326)

# Step 2: Convert your bee data (`oba_df`) to an sf object
## Also going to normalize Genus and Species values
bees_sf <- oba_df %>%
  select(Dec..Lat., Dec..Long., Genus, Species) %>%
  rename(Latitude = Dec..Lat., Longitude = Dec..Long.) %>%
  mutate(
    Genus = tolower(trimws(Genus)), Species = tolower(trimws(Species))
```

```
  ) %>%
  filter(!is.na(Latitude) & !is.na(Longitude)) %>%
  st_as_sf(coords = c("Longitude", "Latitude"), crs = 4326)


# Step 3: Crop the bee data to the Oregon bounds
bees_in_oregon <- st_crop(bees_sf, or_sf)
```

```
## Warning: attribute variables are assumed to be spatially constant throughout
## all geometries
```

```
# Step 4: Plot the polygon (oregon), then plot the spatial fetaures (bee locations) on top
ggplot() +
  # Plot Oregon polygons
  geom_polygon(data = or, aes(x = lon, y = lat, group = group),
               fill = "lightgreen", color = "darkgreen") +
  # Overlay bee locations with coloring by genus
  geom_sf(data = bees_in_oregon) +
  labs(title = "Bees in Oregon", x = "Longitude", y = "Latitude") +
  theme(
    legend.key.size = unit(0.4, "cm"), # Reduce key size
    legend.text = element_text(size = 4), # Reduce text size
    legend.title = element_text(size = 8) # Reduce title size
  ) +
  guides(color = guide_legend(ncol = 3))
```



Plot the fires seperately

We want to: - Plot the bees in the general region the year before the fire - Then plot the bees in the general region of the fire after the fire year - Going to look at fires from just 2020

MonthAb contains the month as a number. Year.1 contains the year as a YYYY number.

```
# Look at the columns of the bee stuff first
oba_df <- oba_df %>%
  rename(
    year = Year.1,
    month = MonthAb,
    full_date = Collection.Date,
    latitude = Dec..Lat.,
    longitude = Dec..Long.,
    plant = Associated.plant
  )

oba_selection <- oba_df %>%
  select(year, month, Genus, Species, plant, latitude, longitude) %>%
  filter(!is.na(Genus), Genus != "")

head(oba_selection)
```

```
##   year month    Genus     Species              plant latitude longitude
## 1 2018     4   Bombus  flavifrons      Ceanothus sp.   44.568  -123.283
## 2 2018     4   Bombus rufocinctus      Ceanothus sp.   44.568  -123.283
## 3 2018     4   Bombus      mixtus      Ceanothus sp.   44.568  -123.283
## 4 2018     4   Bombus      mixtus      Ceanothus sp.   44.568  -123.283
## 5 2018     5 Andrena              Ceanothus cuneatus   44.560  -123.289
## 6 2018     5 Andrena              Ceanothus cuneatus   44.560  -123.289
```

Split OBA data into two sets (24 months before and after the fire)

To be as precise as possible and take full advantage of the data, we created a function to grab the data in 2018, 2019, and 2020 that is 24 months before a given fire in 2020. We did the opposite for the 24 months after a given fire in 2020. Note that this only works for 2020 functions, but could be easily adjusted for any year.

```
# Function to get bee data 24 months before the fire for a fire in 2020 that occured in fire_month
bees_before_fire <- function(fire_month) {
  oba_selection %>%
    filter(
      # Ex: fire in Sept 2020 would want to get data from Sept 2018 up to Sept 2020
      (year == 2020 & month < fire_month) |  # Bees in Jan - Oct 2020
      (year == 2019) |                       # Bees in all of 2019
      (year == 2018 & month >= fire_month)   # Bees in Sept - Dec 2018
    )
}

# Function to get bee data 24 months after the fire for a fire in 2020 that occured in fire_month
bees_after_fire <- function(fire_month) {
  oba_selection %>%
    filter(
      # Ex: fire in Sept 2020 would want to get data from Sept 2020 up to Sept 2022
```

```
      (year == 2020 & month > fire_month) |   # Bees in Oct - Dec 2020
      (year == 2021) |                        # Bees in all of 2021
      (year == 2022 & month <= fire_month)    # Bees in Jan - Sept 2022
    )
}
before_sept_20 <- bees_before_fire(9)
head(before_sept_20)
```

```
##   year month        Genus      Species        plant latitude longitude
## 1 2019     6 Lasioglossum   ruidosense Sidalcea sp.   45.250  -123.111
## 2 2020     7     Halictus               Sidalcea sp.   45.250  -123.111
## 3 2018     9       Bombus vosnesenskii                45.258  -123.074
## 4 2018     9       Bombus vosnesenskii                45.258  -123.074
## 5 2018     9     Ceratina      acantha                45.258  -123.074
## 6 2018     9     Ceratina      acantha                45.258  -123.074
```

Main function for plotting fire buffers and bees inside

We decided to look at the bees within 25 km of the fire region to get a better sample size and to account for the fact that the fire could affect the ecosystem within a certain radius.

```
plot_bees_in_fire_region <- function(bee_data, fire_sp_obj, fire_name) {
  ### This function will take in bee data (as a dataframe) and a fire raster
  ### and plot the bees found within a 25 km radius of the provided fire
  ### it will also provide a count of the bees within the radius
  ### The goal is to be able to compare the bees in the radius before and after the fire


  ## Deal with the base map of Oregon
  or <- map_data("county", "oregon") %>%
    select(lon = long, lat = lat, group = group, id = subregion)

  or_sf <- st_as_sf(or, coords = c("lon", "lat"), crs = 4326, remove = FALSE)

  ## Make sure fire spatial obj matches Oregon map's CRS
  fire_sp_obj <- project(fire_sp_obj, crs(or_sf))

  ## Convert bee data to an sf object & make sure CRS matches
  bees_sf <- bee_data %>%
    mutate(
      Genus = tolower(trimws(Genus)),
      Species = tolower(trimws(Species))
    ) %>%
    filter(!is.na(latitude) & !is.na(longitude)) %>%
    st_as_sf(coords = c("longitude", "latitude"), crs = 4326)


  bees_sf <- st_transform(bees_sf, crs = st_crs(or_sf))

  ## Use the fire's bounds to create a buffer (25 km) & capture all the bees within it by cropping
  fire_bbox <- st_as_sfc(st_bbox(fire_sp_obj))
  fire_buffer <- st_buffer(fire_bbox, dist = 25000)

  bees_in_fire_region <- st_intersection(bees_sf, fire_buffer)
```

```
  ## Total number of bees in the buffer for comparison (before / after fire)
  total_bees <- nrow(bees_in_fire_region)

  ## Plot the Oregon polygon and the bee locations in the fire region
  ggplot() +
    geom_polygon(data = or, aes(x = lon, y = lat, group = group),
                 fill = "lightgreen", color = "darkgreen") +

    geom_sf(data = fire_buffer, fill = NA, color = "red", linetype = "dashed") +

    geom_sf(data = bees_in_fire_region, size = 2, alpha = 0.7) +

    labs(
      title = paste("Bees in Fire Region:", fire_name),
      subtitle = paste("Total bees found in the buffer region:", total_bees),
      x = "Longitude",
      y = "Latitude"
    )
}
```

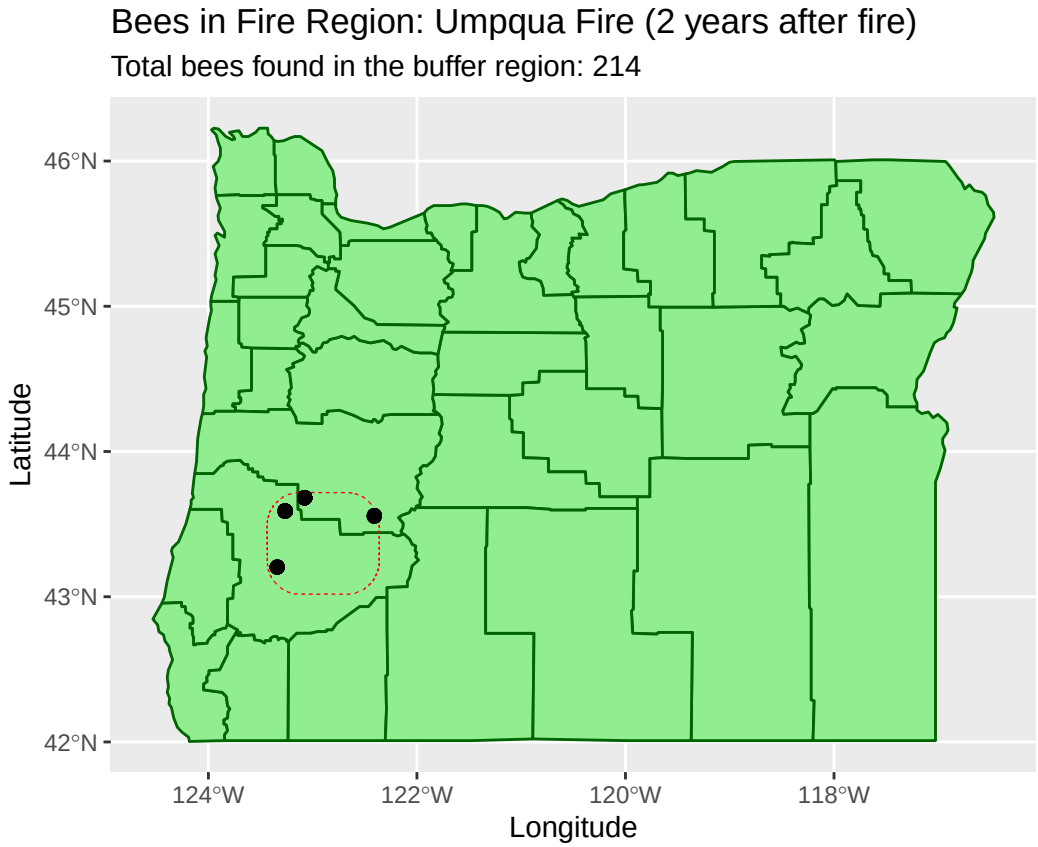
This cell allows us to visually compare the change in bees before and after the fire.

```
umpqua_fire_month =
# Test the function with different fire spatial objects
plot_bees_in_fire_region(bees_before_fire(9), umpqua_2020, "Umpqua Fire (2 years before fire)")
```

```
## Warning: attribute variables are assumed to be spatially constant throughout
## all geometries
```

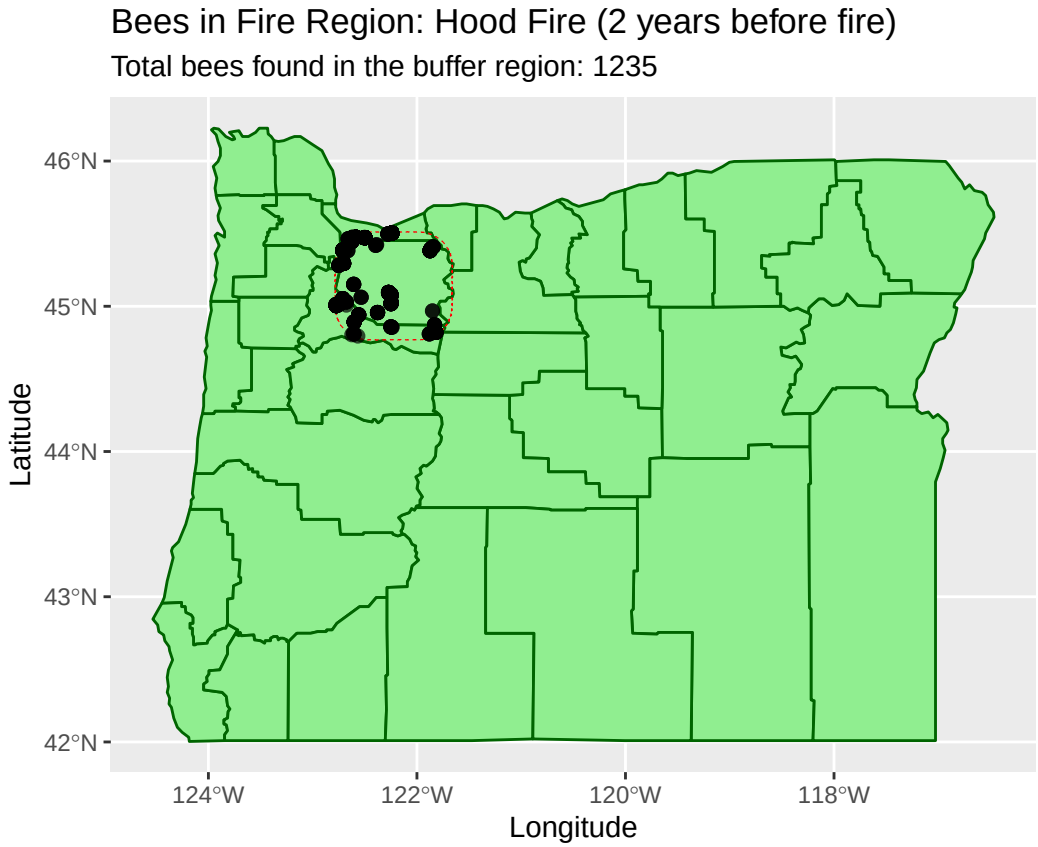
```
plot_bees_in_fire_region(bees_after_fire(9), umpqua_2020, "Umpqua Fire (2 years after fire)")
```

```
## Warning: attribute variables are assumed to be spatially constant throughout
## all geometries
```



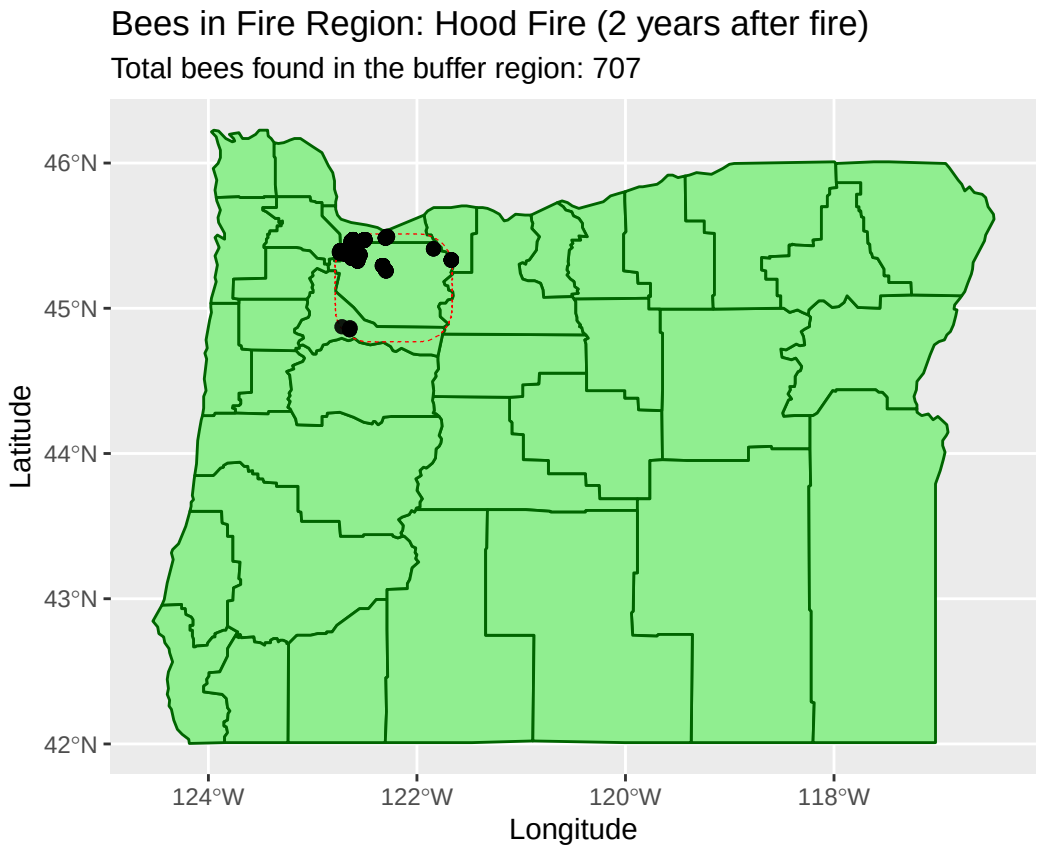

```
plot_bees_in_fire_region(bees_before_fire(9), hood_2020, "Hood Fire (2 years before fire)")
```

```
## Warning: attribute variables are assumed to be spatially constant throughout
## all geometries
```




```
plot_bees_in_fire_region(bees_after_fire(9), hood_2020, "Hood Fire (2 years after fire)")
```

```
## Warning: attribute variables are assumed to be spatially constant throughout
## all geometries
```




```
# plot_bees_in_fire_region(bees_before_fire(9), holiday_2020, "Holiday Fire (2 years before fire)")
# plot_bees_in_fire_region(bees_after_fire(9), holiday_2020, "Holiday Fire (2 years after fire))")
#
# plot_bees_in_fire_region(bees_before_fire(8), beachie_2020, "Beachie Fire (2 years before fire)")
# plot_bees_in_fire_region(bees_after_fire(8), beachie_2020, "Beachie Fire (2 years after fire)")
#
# plot_bees_in_fire_region(bees_before_fire(8), freemont_2020, "Freemont Fire (2 years before fire)")
# plot_bees_in_fire_region(bees_after_fire(8), freemont_2020, "Freemont Fire (2 years after fire)")
#
# plot_bees_in_fire_region(bees_before_fire(7), ben_young_2020, "Ben Young Fire (2 years before fire)")
# plot_bees_in_fire_region(bees_after_fire(7), ben_young_2020, "Ben Young Fire (2 years after fire)")
#
# plot_bees_in_fire_region(bees_before_fire(8), lionshead_2020, "Lionshead Fire (2 years before fire)")
# plot_bees_in_fire_region(bees_after_fire(8), lionshead_2020, "Lionshead Fire (2 years after fire)")
#
# plot_bees_in_fire_region(bees_before_fire(9), brattain_2020, "Brattain Fire (2 years before fire)")
# plot_bees_in_fire_region(bees_after_fire(9), brattain_2020, "Brattain Fire (2 years after fire)")
#
# plot_bees_in_fire_region(bees_before_fire(9), thielsen_2020, "Thielsen Fire (2 years before fire)")
# plot_bees_in_fire_region(bees_after_fire(9), thielsen_2020, "Thielsen Fire (2 years after fire)")
#
# plot_bees_in_fire_region(bees_before_fire(9), two_four_2020, "Two Four Fire (2 years before fire)")
# plot_bees_in_fire_region(bees_after_fire(9), two_four_2020, "Two Four Fire (2 years after fire)")
#
# plot_bees_in_fire_region(bees_before_fire(8), whiteriver_2020, "Whiteriver Fire (2 years before fire)
# plot_bees_in_fire_region(bees_after_fire(8), whiteriver_2020, "Whiteriver Fire (2 years after fire)")
```

Define the "average severity" for a given fire

Since we are looking to see if average severity is related to change in bee count, we want to determine all 11 fire's average severities.

```
define_avg_severity <- function(fire_df) {
  ### This function takes all the severity data ("Layer_1" in the raster)
  ### returns the average severity for the fire
  prop_severe <- sum(fire_df$Layer_1 == 4 | fire_df$Layer_1 == 5) / nrow(fire_df)
  avg_severity <- sum(fire_df$Layer_1) / nrow(fire_df)
  median_severity <- median(fire_df$Layer_1)
  mode_severity <- as.numeric(names(sort(table(fire_df$Layer_1), decreasing = TRUE)[1]))
  return(avg_severity)
}

define_avg_severity(umpqua_2020_df)
```

```
## [1] 4.009
```

```
define_avg_severity(hood_2020_df)
```

```
## [1] 3.512376
```

```
define_avg_severity(holiday_2020_df)
```

```
## [1] 3.762565
```

```
define_avg_severity(beachie_2020_df)
```

```
## [1] 3.687295
```

```
define_avg_severity(freemont_2020_df)
```

```
## [1] 3.362752
```

```
define_avg_severity(ben_young_2020_df)
```

```
## [1] 3.325592
```

```
define_avg_severity(brattain_2020_df)
```

```
## [1] 3.059134
```

```
define_avg_severity(lionshead_2020_df)
```

```
## [1] 3.413909
```

```
define_avg_severity(thielsen_2020_df)
```

```
## [1] 3.449503
```

```
define_avg_severity(two_four_2020_df)
```

```
## [1] 3.781873
```

```
define_avg_severity(whiteriver_2020_df)
```

```
## [1] 3.086412
```

Statistical test to evaluate significance in change

Steps here: - Count the bees in the 25km buffer - Calculate the percent change, look at dataframe of severity and percent change in bees - Run a correlation analysis - Evaluate statistical significance

```
bees_in_buffer_zone <- function(bee_data, fire_sp_obj) {

  ## Deal with the base map of Oregon
  or <- map_data("county", "oregon") %>%
    select(lon = long, lat = lat, group = group, id = subregion)

  or_sf <- st_as_sf(or, coords = c("lon", "lat"), crs = 4326, remove = FALSE)

  ## Make sure fire spatial obj matches Oregon map's CRS
  fire_sp_obj <- project(fire_sp_obj, crs(or_sf))

  ## Convert bee data to an sf object & make sure CRS matches
  bees_sf <- bee_data %>%
    filter(!is.na(latitude) & !is.na(longitude)) %>%
    st_as_sf(coords = c("longitude", "latitude"), crs = 4326)

  bees_sf <- st_transform(bees_sf, crs = st_crs(or_sf))

  ## Use the fire's bounds to create a buffer (50 km) & capture all the bees within it by cropping
  fire_bbox <- st_as_sfc(st_bbox(fire_sp_obj))
  fire_buffer <- st_buffer(fire_bbox, dist = 25000)

  bees_in_fire_region <- st_intersection(bees_sf, fire_buffer)

  ## Total number of bees in the buffer for comparison
  total_bees <- nrow(bees_in_fire_region)
  return(total_bees)
}
```

Feature engineering calculation of percent change

```
# Calculate bees in buffer zone before and after the fire for Umpqua fire
percent_change <- function(fire_rast, bees_before, bees_after) {
  before_fire <- bees_in_buffer_zone(bees_before, fire_rast)
  after_fire <- bees_in_buffer_zone(bees_after, fire_rast)
  percent_change_value <- ((after_fire - before_fire) / before_fire) * 100
  return(percent_change_value)}

umpqua_perc_change <- suppressWarnings(percent_change(umpqua_2020, bees_before_fire(9), bees_after_fire
hood_perc_change <- suppressWarnings(percent_change(hood_2020, bees_before_fire(9), bees_after_fire(9))
holiday_perc_change <- suppressWarnings(percent_change(holiday_2020, bees_before_fire(9), bees_after_fi
beachie_perc_change <- suppressWarnings(percent_change(beachie_2020, bees_before_fire(8), bees_after_fi
freemont_perc_change <- suppressWarnings(percent_change(freemont_2020, bees_before_fire(8), bees_after_

ben_young_perc_change <- suppressWarnings(percent_change(ben_young_2020, bees_before_fire(7), bees_afte
brattain_perc_change <- suppressWarnings(percent_change(brattain_2020, bees_before_fire(9), bees_after_
lionshead_perc_change <- suppressWarnings(percent_change(lionshead_2020, bees_before_fire(8), bees_afte
thielsen_perc_change <- suppressWarnings(percent_change(thielsen_2020, bees_before_fire(9), bees_after_
```

```r
two_four_perc_change <- suppressWarnings(percent_change(two_four_2020, bees_before_fire(9), bees_after_
whiteriver_perc_change <- suppressWarnings(percent_change(whiteriver_2020, bees_before_fire(8), bees_af
```

Print Results

```r
# Create a data frame with average severities and percent changes for all fires
fires_data <- data.frame(
  fire = c("Umpqua", "Hood", "Holiday", "Beachie", "Lionshead", "Thielsen", "Freemont", "Ben Young", "Br
  avg_severity = c(define_avg_severity(umpqua_2020_df),
                   define_avg_severity(hood_2020_df),
                   define_avg_severity(holiday_2020_df),
                   define_avg_severity(beachie_2020_df),
                   define_avg_severity(lionshead_2020_df),
                   define_avg_severity(thielsen_2020_df),
                   define_avg_severity(freemont_2020_df),
                   define_avg_severity(ben_young_2020_df),
                   define_avg_severity(brattain_2020_df),
                   define_avg_severity(two_four_2020_df),
                   define_avg_severity(whiteriver_2020_df)),
  percent_change = c(umpqua_perc_change,
                     hood_perc_change,
                     holiday_perc_change,
                     beachie_perc_change,
                     lionshead_perc_change,
                     thielsen_perc_change,
                     freemont_perc_change,
                     ben_young_perc_change,
                     brattain_perc_change,
                     two_four_perc_change,
                     whiteriver_perc_change))
head(fires_data)
```

```
##        fire avg_severity percent_change
## 1    Umpqua     4.009000      -80.18519
## 2      Hood     3.512376      -42.75304
## 3   Holiday     3.762565      -56.84790
## 4   Beachie     3.687295      -75.21531
## 5 Lionshead     3.413909      -46.02235
## 6  Thielsen     3.449503      -89.63731
```

Apply Pearson's Correlation Analysis

```r
# Correlation Analysis
correlation <- cor(fires_data$avg_severity, fires_data$percent_change, method = "pearson")
cat("Pearson correlation coefficient:", correlation, "\n")
```

```
## Pearson correlation coefficient: -0.4804525
```

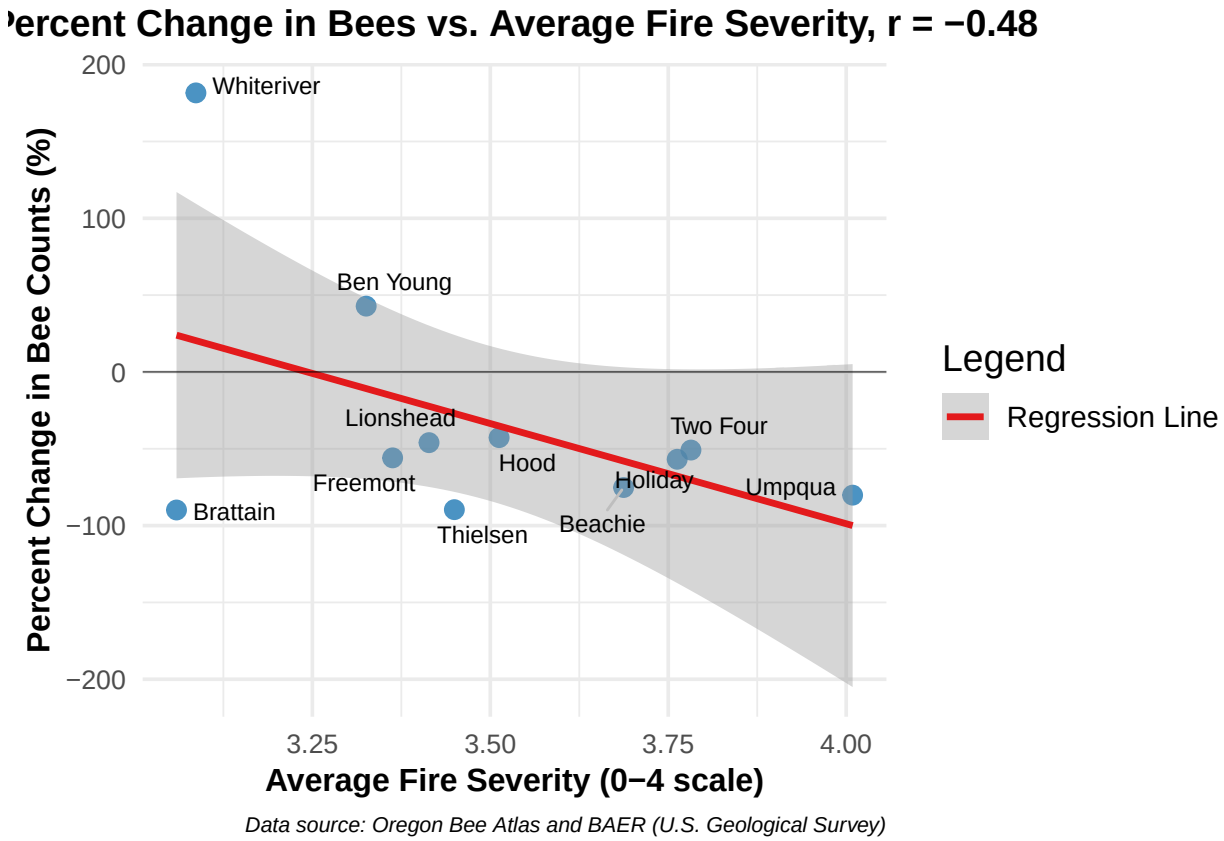
```r
# Regression Plot with Legend and Units
ggplot(fires_data, aes(x = avg_severity, y = percent_change, label = fire)) +
  geom_point(size = 3, color = "#1f78b4", alpha = 0.8) +
  geom_smooth(method = "lm", aes(color = "Regression Line"), se = TRUE, size = 1.2) +
```

```
  geom_text_repel(
    size = 3, color = "black", box.padding = 0.35, max.overlaps = 10,
    segment.size = 0.5, segment.color = "gray"
  ) +
  geom_hline(yintercept = 0, size = 0.1) +
  labs(
    title = paste("Percent Change in Bees vs. Average Fire Severity, r =", round(correlation, 2)),
    x = "Average Fire Severity (0-4 scale)",
    y = "Percent Change in Bee Counts (%)",
    caption = "Data source: Oregon Bee Atlas and BAER (U.S. Geological Survey)"
  ) +
  scale_color_manual(
    name = "Legend",
    values = c("Regression Line" = "#e31a1c")
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
    plot.subtitle = element_text(hjust = 0.5, size = 12, face = "italic"),
    axis.title = element_text(face = "bold", size = 12),
    axis.text = element_text(size = 10),
    plot.caption = element_text(size = 8, hjust = 1, face = "italic")
  )
```

```
## 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.
```

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

```
## Warning: The following aesthetics were dropped during statistical transformation: label.
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



Test our fire severity Null Hypothesis

Null Hypothesis: There is no linear relationship between fire soil burn severity and % change in bee count. r = 0 Alternative Hypothesis: There is a linear relationship between fire soil burn severity and % change in bee count. r != 0

```
# Perform the Pearson correlation test
cor_test <- cor.test(fires_data$avg_severity, fires_data$percent_change, method = "pearson")

# Print the results
cat("Pearson correlation coefficient:", cor_test$estimate, "\n")
```

```
## Pearson correlation coefficient: -0.4804525
```

```
cat("P-value:", cor_test$p.value, "\n")
```

```
## P-value: 0.134699
```

```
# Check significance level
if (cor_test$p.value < 0.05) {
  cat("Result: Reject the null hypothesis. There is evidence of a linear relationship.\n")
} else {
  cat("Result: Fail to reject the null hypothesis. No evidence of a linear relationship.\n")
}
```

```
## Result: Fail to reject the null hypothesis. No evidence of a linear relationship.
```

Test our percent change Null Hypothesis

Another hypotheis to look at, since we found no significant linear relationship between, we can also look at if a fire, regardless of severity, does affect bee counts.

Null Hypothesis: There is no relationship between fires and bee count. Alternative Hypothesis: There is a relationship between fires and bee count.

```
# Extract number of rows for before fire data
umpqua_before_fire <- suppressWarnings(bees_in_buffer_zone(bees_before_fire(9), umpqua_2020))
hood_before_fire <- suppressWarnings(bees_in_buffer_zone(bees_before_fire(9), hood_2020))
holiday_before_fire <- suppressWarnings(bees_in_buffer_zone(bees_before_fire(9), holiday_2020))
beachie_before_fire <- suppressWarnings(bees_in_buffer_zone(bees_before_fire(8), beachie_2020))
freemont_before_fire <- suppressWarnings(bees_in_buffer_zone(bees_before_fire(8), freemont_2020))
ben_young_before_fire <- suppressWarnings(bees_in_buffer_zone(bees_before_fire(7), ben_young_2020))
brattain_before_fire <- suppressWarnings(bees_in_buffer_zone(bees_before_fire(9), brattain_2020))
lionshead_before_fire <- suppressWarnings(bees_in_buffer_zone(bees_before_fire(8), lionshead_2020))
thielsen_before_fire <- suppressWarnings(bees_in_buffer_zone(bees_before_fire(9), thielsen_2020))
two_four_before_fire <- suppressWarnings(bees_in_buffer_zone(bees_before_fire(9), two_four_2020))
whiteriver_before_fire <- suppressWarnings(bees_in_buffer_zone(bees_before_fire(8), whiteriver_2020))

# Extract number of rows for after fire data
umpqua_after_fire <- suppressWarnings(bees_in_buffer_zone(bees_after_fire(9), umpqua_2020))
hood_after_fire <- suppressWarnings(bees_in_buffer_zone(bees_after_fire(9), hood_2020))
holiday_after_fire <- suppressWarnings(bees_in_buffer_zone(bees_after_fire(9), holiday_2020))
beachie_after_fire <- suppressWarnings(bees_in_buffer_zone(bees_after_fire(8), beachie_2020))
freemont_after_fire <- suppressWarnings(bees_in_buffer_zone(bees_after_fire(8), freemont_2020))
ben_young_after_fire <- suppressWarnings(bees_in_buffer_zone(bees_after_fire(7), ben_young_2020))
brattain_after_fire <- suppressWarnings(bees_in_buffer_zone(bees_after_fire(9), brattain_2020))
lionshead_after_fire <- suppressWarnings(bees_in_buffer_zone(bees_after_fire(8), lionshead_2020))
thielsen_after_fire <- suppressWarnings(bees_in_buffer_zone(bees_after_fire(9), thielsen_2020))
two_four_after_fire <- suppressWarnings(bees_in_buffer_zone(bees_after_fire(9), two_four_2020))
whiteriver_after_fire <- suppressWarnings(bees_in_buffer_zone(bees_after_fire(8), whiteriver_2020))
```

```
# Combine before and after data into vectors
before_fire <- c(
  umpqua_before_fire, hood_before_fire, holiday_before_fire,
  beachie_before_fire, freemont_before_fire, ben_young_before_fire,
  brattain_before_fire, lionshead_before_fire, thielsen_before_fire,
  two_four_before_fire, whiteriver_before_fire
)

after_fire <- c(
  umpqua_after_fire, hood_after_fire, holiday_after_fire,
  beachie_after_fire, freemont_after_fire, ben_young_after_fire,
  brattain_after_fire, lionshead_after_fire, thielsen_after_fire,
  two_four_after_fire, whiteriver_after_fire
)

before_fire
```

```
##  [1] 1080 1235 1453 2090  919  453 1440 1521  193  708  250
```

```
after_fire
```

```
##  [1] 214 707 627 518 405 647 146 838  20 348 704
```

Calculate T Test to identify p value

```
# Perform a paired t-test
t_test <- t.test(before_fire, after_fire, paired = TRUE)
p_value <- t_test$p.value

# Print the results
cat("Paired T-Test Results:\n")
```

```
## Paired T-Test Results:
```

```
cat("T-statistic:", t_test$statistic, "\n")
```

```
## T-statistic: 3.129215
```

```
cat("P-value:", t_test$p.value, "\n")
```

```
## P-value: 0.01070477
```

```
cat("Confidence Interval:", t_test$conf.int, "\n")
```

```
## Confidence Interval: 161.4648 959.9897
```

```
cat("Mean difference:", t_test$estimate, "\n")
```

```
## Mean difference: 560.7273
```

```
# Interpretation
if (t_test$p.value < 0.05) {
  cat("Result: Reject the null hypothesis. Bee counts differ significantly before and after fires.\n")
} else {
  cat("Result: Fail to reject the null hypothesis. No significant difference in bee counts before and a
}
```

```
## Result: Reject the null hypothesis. Bee counts differ significantly before and after fires.
```

Final visualization

Show before and after fire counts in a bar chart.

```
# Create a data frame for visualization
visual_data <- data.frame(
  Location = rep(
    c("Umpqua", "Hood", "Holiday", "Beachie", "Freemont",
      "Ben Young", "Brattain", "Lionshead", "Thielsen",
      "Two Four", "White River"),
    times = 2
  ),
  Count = c(before_fire, after_fire),
  Time = rep(c("Before Fire", "After Fire"), each = length(before_fire))
)

# Factor the Time variable to ensure "Before Fire" comes first
visual_data$Time <- factor(visual_data$Time, levels = c("Before Fire", "After Fire"))

# Create the grouped bar plot
ggplot(visual_data, aes(x = Location, y = Count, fill = Time)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.7), color = "black") +  # Slight gap
```

```
  scale_fill_manual(values = c("Before Fire" = "#1f78b4", "After Fire" = "#e31a1c")) +
  labs(
    title = "Bee Counts Before and After Fires",
    x = "Fire Location",
    y = "Bee Count (number of bees)",
    fill = "Time",
    caption = "Data source: Oregon Bee Atlas"
  ) +
  annotate(
    "text",
    x = 9, y = max(c(before_fire, after_fire)) * 1.05,
    label = paste("P-value:", format(p_value, digits = 2)),
    size = 5, color = "black", fontface = "italic"
  ) +  # Add p-value annotation in the upper-right corner
  theme_minimal(base_size = 14) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
    axis.title = element_text(face = "bold", size = 12),
    plot.caption = element_text(size = 10, hjust = 1, face = "italic")
  )
```