

Final Portfolio MT

Matthew Thompson

2025-03-03

Reflection

In this final portfolio I created 4 plots: grouped bar (horizontal), paneled line and alluvial diagram, faceted line and area plot, and a pie chart, and 1 table. I made sure to tackle major aspects such as having a basic plot, paneled plots, faceted plot, and a summary statistics table, including at least one unusual plot.

Throughout the semester besides minor feedbacks like aesthetics improvements, I noticed majority of feedbacks tend to involve categories decisions like grouping categories and which categories are unnecessary. Besides categorizing, I also need to work on better visual representations, thus showing which data, which visual to use, and which aesthetics setting to use or exclude. I think it may correspond to my tendency to try to include too much information in one place. Thus, I used this final portfolio as an opportunity to incorporate feedbacks and try to address my decision flaws with the best of my abilities.

For instance, with a pie chart and a grouped bar chart, I categorized all animals to each 4 appropriate categories without overlaps where they could belong to more than one category based on the criteria. There was a feedback that suggested to move legend labels into each slice in the pie chart rather than the list of legends. Unfortunately, I struggled on that part and could only manage to get the labels aligned partially correctly despite research I've done.

I started to understand why stacked plots can often be problematic when it comes to accurately quantifying data values, so I used grouped bar chart and less numbers of grouped data which was much more readable than stacked bar chart with over 10 categories. I decided to not include the weight class since it may be a bit overwhelming to not just read grouped bars but also to read the weight class on top of them. I also addressed too many categories and better representation of data in a faceted line and area plot and a summary statistics table for comparisons of HPI.

I also addressed the stacked style aesthetics when answering the question regarding top 5 most dangerous jobs. Instead of using a proportional stream graph, I used a line graph to better visualize the trend of fatalities over time for each job more accurately. I challenged myself to try to create an alluvial diagram I believe was a good graph to show the common causes of fatalities for each job. Unfortunately, I struggled on that part as well, especially on the texts aesthetics. It was hard to read. I considered removing the text labels and add legend labels instead. However, the main problem with alluvial diagram, I believe, is that people won't know where the flow starts at first. I decided to keep the text labels for the sake of readability and explain briefly in the captioning.

Overall, I think I've addressed too many categories, better visual representations, and aesthetics decisions in my final portfolio. According to criteria, I don't believe I had that much major flaws, thus I would grade myself an A. However, if I also take into account other factors like mastery of plots and knowledge, I'll say I deserve a pretty solid **B** to reflect my ongoing progress to mastery, good grasp of basics, and a bit of struggles despite improvements I hopefully demonstrated.

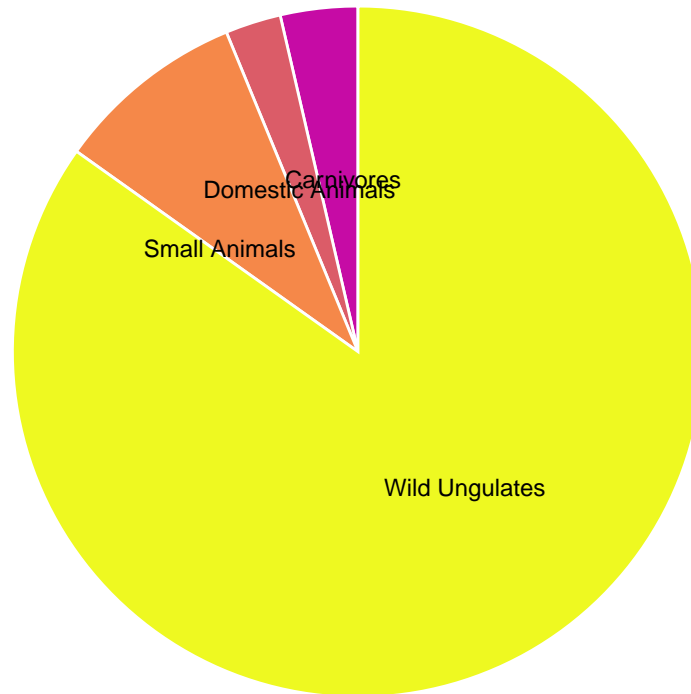


Figure 1. What types of animals do cougars kill? Animals were grouped into four major categories: Wild Ungulates, Small Animals, Domestic Animals, and Carnivores. Wild Ungulates had the most cougar kills compared to domestic animals with the least number of kills. Note: These four major categories were selected to avoid overlapping categories where animals could belong to more than one category.

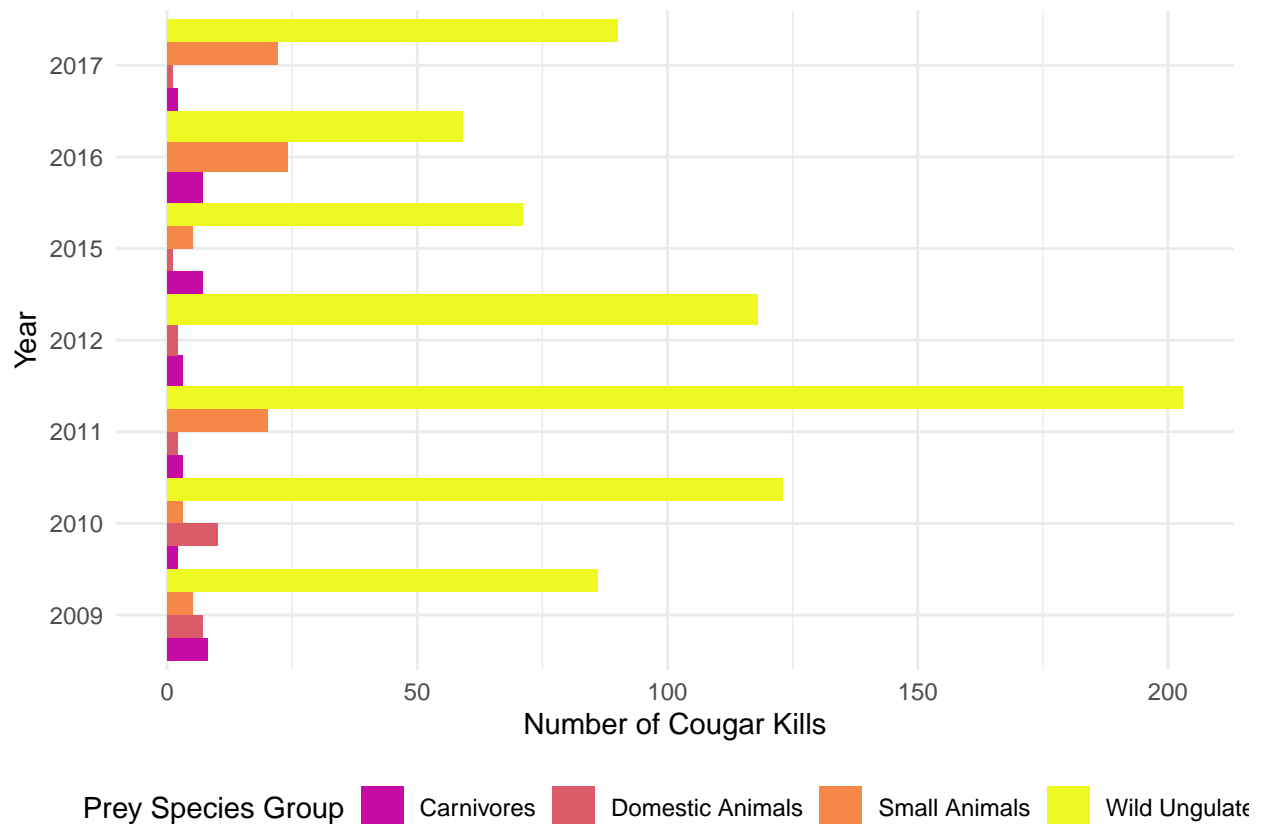


Figure 2. How are cougars kills distributed over time and how do they compare? Animals were grouped into four major categories: Wild Ungulates, Small Animals, Domestic Animals, and Carnivores and distributed across years. To better visualize the quantifying comparison between major categories, grouped bar chart was used. Wild ungulates such as mule deers, pronghorns, and other hoofed animals in the wild were the most common prey for cougars. Overall, domestic animals were the least common prey for cougars. In 2012 and 2016, no cougar kills were recorded for small animals and domestic animals, respectively. Note: there is a gap between 2012 and 2015 where no data was recorded for some reasons Source: Recolonizing carnivores: Is cougar predation behaviorally mediated by bears?

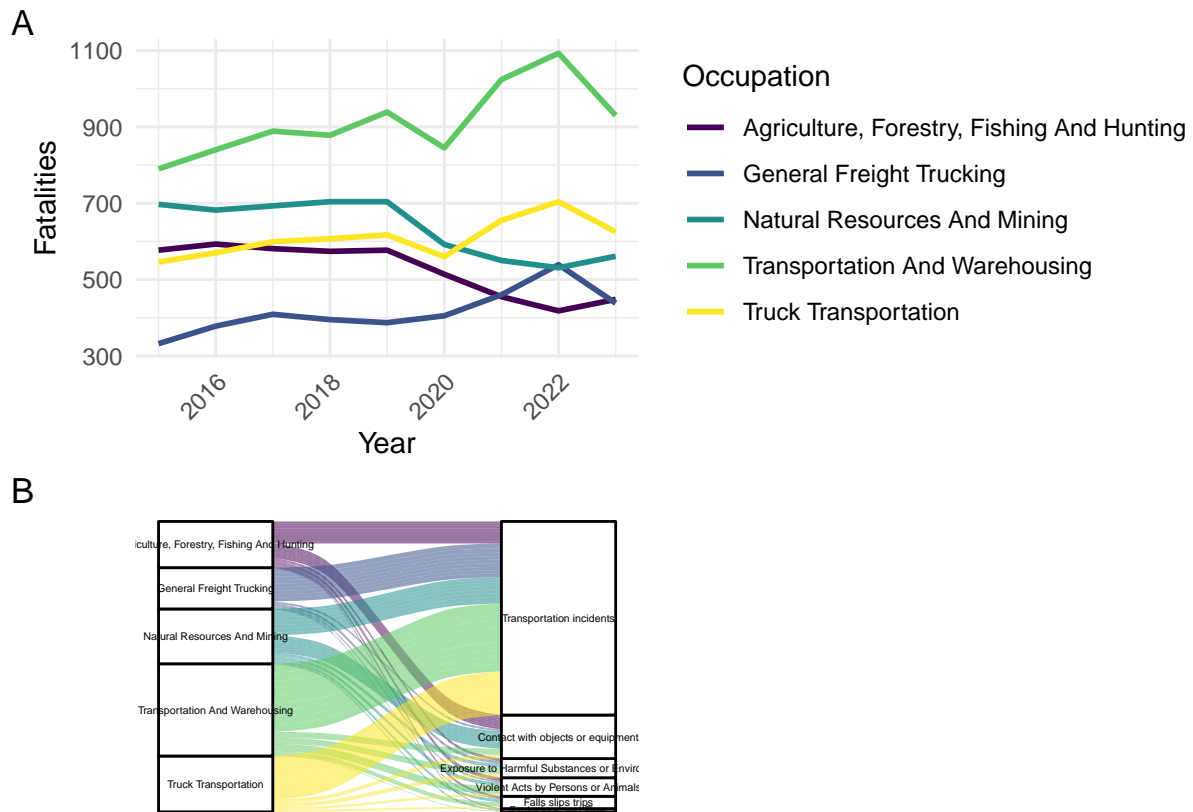


Figure 3. What are the top 5 most dangerous jobs that remain over the years from 2015 to 2023 and what are the common causes for each? The top 5 most dangerous jobs were selected throughout the years where their fatalities were all recorded from 2015 to 2023. (Panel A), a line graph was selected as the best visual representation of the total fatalities over time. The trend shows the transportation and mining as the most dangerous job, although it dropped sharply in 2023. The trend shows truck-related jobs like general freight driving, transportation & warehousing, and truck transportation generally increasing over time until they dropped in 2023. Agriculture and natural resources & mining started to drop in 2019 and then increased slightly in 2023. (Panel B) Alluvial diagram was used to visualize the causes of fatalities for each of the same 5 jobs. The causes were categorized into different categories such as transportation incidents, falls, and exposure to harmful substances. Transportation incidents were the most common cause of fatalities across all jobs compared to explosions and fires as the least common cause of fatalities.

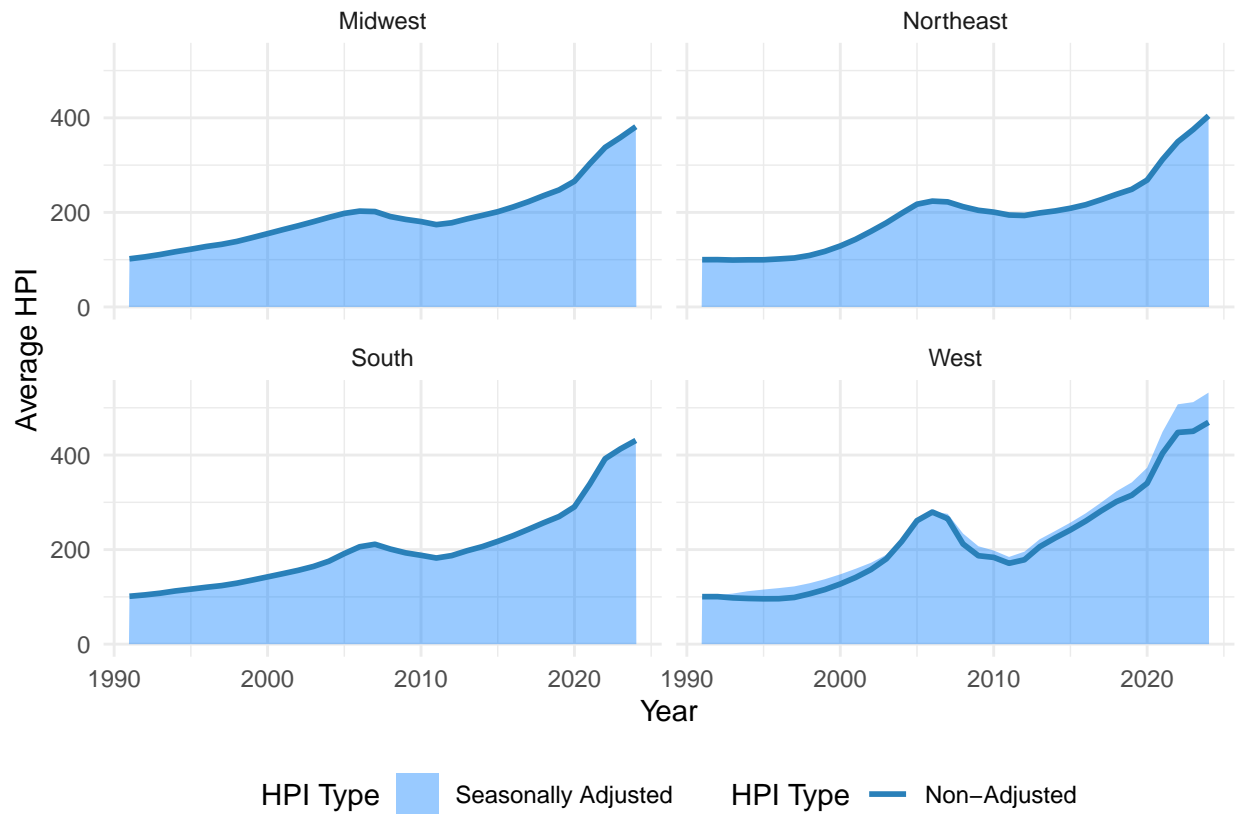


Figure 4. How do seasonally-adjusted and non-adjusted housing price index (HPI) compare across four major regions in the US over time? HPI was faceted across four major regions in the United States: Northeast, Midwest, South, and West. The HPI was further divided into seasonally adjusted and non-adjusted data. The seasonally adjusted data was represented by filled areas, while the non-adjusted data was represented by lines. Midwest, Northwest, and the South seem to have a pretty similar trend in HPI while West is a bit more prevalent. For instance, seasonally adjusted and non-adjusted HPI are about the same across three regions except for the West where the seasonally adjusted HPI is sometimes higher than the non-adjusted HPI at some occasions. However, the overall trend is the same for all regions where HPI increased until past 2005 when HPI fell. Then, it started rising again after 2010. Note: USA HPI was excluded due to the redundancy since it is basically an overall HPI of the entire US.

Time	HPI	
	Min HPI	Max HPI
Years		
1991-1995	95	138
1996-2000	95	173
2001-2005	128	278
2006-2010	158	294
2011-2015	153	283
2016-2020	182	441
2021-2023*	255	596

Table 1. How does HPI compare over the years in terms of maximum and minimum? HPI remain untouched, although years were grouped each 5 years and rounded up as whole numbers. Minimum and maximum HPI were selected across every 5 years and then showcased in the table. HPI continued to increase until within year of 2011-2015, it decreased slightly. Then, it started increasing again in the next interval (2016-2023). Overall trend show the increasing HPI ever since then except for 2011-2015 interval.

Note: Asterick(*) by 2021-2023 indicates an incomplete interval. 2024 have incomplete data where it lacks full 12 months, thus it was excluded from the table.

R Entire Script

```
library(ggplot2)
library(dplyr)
library(tidyverse)
library(lubridate)
library(reshape2)
library(readxl)
library(viridis)
library(ggstream)
library(ggalluvial)
library(gt)

#####
# Basic Plots of Cougars - Pie and Histogram
#####

# load the data
cougars <- read_xlsx("Cougars Killsites.xlsx")

# data cleaning & transformations
cougars_NA <- cougars %>%
  filter(is.na(Year)) %>%
  summarize(n = n())

cougars_NA_2 <- cougars %>%
  filter(is.na(PreySpeciesSingle)) %>%
  summarize(n = n())
#no NA values generally, but I need to fix the labeling prey species first

#Fixing preyspeciessingle names
cougar_species <- cougars %>%
  mutate(
    PreySpeciesSingle = str_trim(str_to_title(PreySpeciesSingle)),
    # trimmed spaces and converted first letter of each word to be capitalized
    PreyWeightClass = str_trim(str_to_title(PreyWeightClass))
    # standardized weight class
  ) %>%
  mutate(
    PreySpeciesSingle = case_when(
      PreySpeciesSingle == "Feral Horsse" ~ "Feral Horse",
      # Correct spelling
      TRUE ~ PreySpeciesSingle # Keep other values unchanged
    )
  )

cougar_species_check <- cougar_species %>% #checking species names
  group_by(PreySpeciesSingle) %>%
  unique() %>%
  summarize(n = n())

#group preyspecies into major groups
cougarspecies_grouped <- cougar_species %>%
  mutate(
```

```

PreySpeciesGroup = case_when(
  PreySpeciesSingle %in% c("Mule Deer", "Bighorn Sheep", "Feral Horse",
    "Pronghorn", "Scavenged - Feral Horse",
    "Scavenged - Mule Deer", "Unknown Yearling Ungulate",
    "Unknown Yoy Ungulate") ~ "Wild Ungulates",
  PreySpeciesSingle %in% c("Bird", "Small Mammal", "Porcupine",
    "Bird", "Beaver", "Badger") ~ "Small Animals",
  PreySpeciesSingle %in% c("Domestic Dog", "Domestic Sheep",
    "Scavenged - Domestic Cow",
    "Domestic Cattle") ~ "Domestic Animals",
  PreySpeciesSingle %in% c("Black Bear", "Coyote", "Red Fox",
    "Bobcat") ~ "Carnivores")
)

# Summarize the data by PreySpeciesGroup for pie chart position
cougarspecies_grouped_check <- cougarspecies_grouped %>%
  group_by(PreySpeciesGroup) %>%
  summarise(n = n()) %>% #count kills
  mutate(csum = cumsum(n), # Cumulative sum for stacking
    pos = csum - n / 4) # Position for text labels (center of each slice)
# Source: https://r-charts.com/part-whole/pie-chart-labels-outside-ggplot2/

# created pie chart of prey species groups
ggplot(cougarspecies_grouped_check, aes(x = "", y = n, fill = PreySpeciesGroup)) +
  geom_bar(stat = "identity", width = 1, color = "white") +
  coord_polar("y", start = 0) +
  geom_text(aes(label = PreySpeciesGroup, y = pos),
    size = 3,
    position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c(
    "#C60BA5",
    "#DB5C68",
    "#F58849",
    "#EEF921"
  )) +
  theme_void() + # Remove gridlines and background
  theme(legend.position = "none")

#Used ChatGPT and a few sources online to try to code legend labels inside a pie chart

#####
# Create a histogram of the number of cougar kills by year
# grouped by species group and weight class
# is stacked and horizontal
#####

cougar_year_check <- cougarspecies_grouped %>%
  group_by(Year, PreySpeciesGroup) %>%
  summarise(Count = n())

ggplot(cougarspecies_grouped, aes(x = factor(Year), fill = PreySpeciesGroup)) +
  geom_bar(position = "dodge", stat = "count", width=1) +
  labs(x = "Year", y = "Number of Cougar Kills", fill = "Prey Species Group") +

```



```

scale_fill_manual(values = c(
  "Carnivores" = "#C60BA5", # Magenta
  "Domestic Animals" = "#DB5C68", # Muted Pink
  "Small Animals" = "#F58849", # Coral Red
  "Wild Ungulates" = "#EEF921" # Bright Yellow
)) +
theme_minimal() +
theme(legend.position = "bottom") +
coord_flip()

#####
#Unusual Plot: panels
#####

#importing and cleaning
dangerous_job <- read_csv('Dangerous Jobs.csv')

#removed dupes for all columns
removed_dupes <- dangerous_job %>%
  group_by_all() %>%
  distinct()

removed_dupes0.1 <- dangerous_job %>%
  group_by(MajorGroup, MinorGroup,
    BroadOccupation, DetailedOccupation,
    Cause) %>%
  #avoids distinct() effect on year, NAICS, and Fatalities
  distinct(Fatalities, .keep_all = TRUE)
#removes duplicates that were set apart by slightly diff NAIC

#renamed object
data_prep <- removed_dupes0.1

#replaces NA to unknown for other major groups
#if detailedoccupaiton has a value and fatalities to 0
replaced_NA <- data_prep %>%
  mutate(NAICS = as.character(NAICS)) %>%
  mutate(NAICS = ifelse(is.na(NAICS), "UNKNOWN", NAICS),
    MajorGroup = ifelse(is.na(MajorGroup), "UNKNOWN", MajorGroup),
    MinorGroup = ifelse(is.na(MinorGroup), "UNKNOWN", MinorGroup),
    BroadOccupation = ifelse(is.na(BroadOccupation), "UNKNOWN", BroadOccupation),
    Fatalities = ifelse(is.na(Fatalities), 0, Fatalities))

# Filter and summarize fatalities per DetailedOccupation per Year
grouped_data <- replaced_NA %>%
  filter(Cause == "Total.Fatalities" & DetailedOccupation != "Total") %>%
  dplyr::select(Year, DetailedOccupation, Fatalities) %>%
  group_by(DetailedOccupation, Year) %>%
  summarize(Fatalities = sum(Fatalities, na.rm = TRUE)) %>%
  ungroup()

# Now, filtered out only occupations that have fatalities
# in every year (2015-2023)

```

```

required_years <- 2015:2023
valid_occupations <- grouped_data %>%
  group_by(DetailedOccupation) %>%
  filter(all(required_years %in% Year)) %>%
  ungroup()

#rearranged into top 5 with total kills
top_5 <- valid_occupations %>%
  group_by(DetailedOccupation) %>%
  summarize(Total_Fatalities = sum(Fatalities)) %>%
  top_n(5, Total_Fatalities) %>%
  ungroup()

#top 5 occupations:
#agriculture, truck transportation, natural resources and mining,
# general freight trucking, and transportation and warehousing

#selects only these detailed occupations
top_5_occupations <- valid_occupations %>%
  filter(DetailedOccupation %in% c("Agriculture, Forestry, Fishing And Hunting",
    "Truck Transportation",
    "Natural Resources And Mining",
    "General Freight Trucking",
    "Transportation And Warehousing"))

# shows causes for each of these occupation and exclude total.fatalities in Cause
more_cause <- replaced_NA %>%
  filter(DetailedOccupation %in% c("Agriculture, Forestry, Fishing And Hunting",
    "Truck Transportation",
    "Natural Resources And Mining",
    "General Freight Trucking",
    "Transportation And Warehousing") & Cause != "Total.Fatalities")

# create a proportional steam graph
# plot_1 <- ggplot(top_5_occupations, aes(x = Year, y = Fatalities,
#   group = DetailedOccupation,
#   fill = DetailedOccupation)) +
#   geom_stream(type = "proportional", stat = "identity", alpha = 1) +
#   theme_minimal() +
#   labs(x = "Year",
#     y = "Fatalities",
#     fill = "Occupation") +
#   scale_fill_viridis(discrete = TRUE) +
#   theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
#   theme(legend.position = 'right',
#     legend.direction = "vertical",
#     legend.box = "horizontal") +
#   guides(fill = guide_legend(ncol = 1))
# # Arranged legend into one column

#using line plot instead
plot_1_alternative <- ggplot(top_5_occupations, aes(x = Year, y = Fatalities,
  group = DetailedOccupation,

```

```

        color = DetailedOccupation)) +
geom_line(size = 1) +
theme_minimal() +
labs(x = "Year",
     y = "Fatalities",
     color = "Occupation") +
scale_color_viridis(discrete = TRUE) +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
theme(legend.position = 'right',
      legend.direction = "vertical",
      legend.box = "horizontal") +
guides(color = guide_legend(ncol = 1))

#alluvial diagram
# sorted data so that largest causes are at the top
sorted_data <- more_cause %>%
  group_by(Cause) %>%
  summarise(Total_Fatalities = sum(Fatalities)) %>%
  arrange(desc(Total_Fatalities)) # Sort largest to smallest

# converted Cause to a factor with custom ordering
more_cause$Cause <- factor(more_cause$Cause, levels = sorted_data$Cause)

plot_2 <- ggplot(more_cause, aes(axis1 = DetailedOccupation, axis2 = Cause, y = Fatalities)) +
  geom_alluvium(aes(fill = DetailedOccupation, alpha = 0.9)) +
  geom_stratum() +
  geom_text(stat = "stratum", aes(label = after_stat(stratum)), size = 1.25) +
  scale_fill_viridis(discrete = TRUE, guide = guide_legend(title = NULL)) +
  theme_void() +
  theme(
    axis.text = element_blank(),
    axis.ticks = element_blank(),
    panel.grid = element_blank(),
    legend.position = "none"
  )

library(patchwork)
combined_plot <- (plot_1 / plot_2) +
  #annotate the plots
  plot_annotation(
    tag_levels = 'A')

# Display the combined plot
print(combined_plot)

#####
# Housing Price Faceted
#####

housing_price <- read_csv("Housing Price Index.xlsx - Data.csv")

#adding date columns
housing_price2 <- housing_price %>%

```

```

mutate(Date = as.POSIXct(Date, format = "%m/%d/%Y")) %>%
  #had to adjust to match the format
mutate(Year = year(Date)) %>%
mutate(Month = month(Date)) %>%
mutate(Day = day(Date))

#preparing for faceted plot (line + area)
mean_acrossdivisons <- housing_price2 %>%
  group_by(Year) %>%
  summarize(across(matches("(SA)|(NSA)"), mean, na.rm = TRUE))

#combined all divisions mean into each 4 major regions
#for SA and NSA before I can plot
major_regionsmean <- mean_acrossdivisons %>%
  mutate(
    Northeast_SA = rowMeans(dplyr::select(., matches("New\\.England\\(SA\\)|Middle\\.Atlantic\\(SA\\)")),
    Midwest_SA = rowMeans(dplyr::select(., matches("East\\.North\\.Central\\(SA\\)|West\\.North\\.Centr\\(SA\\)")),
    South_SA = rowMeans(dplyr::select(., matches("South\\.Atlantic\\(SA\\)|East\\.South\\.Central\\(SA\\)")),
    West_SA = rowMeans(dplyr::select(., matches("Mountain\\(SA\\)|Pacific\\(SA\\)")), na.rm = TRUE),

    Northeast_NSA = rowMeans(dplyr::select(., matches("New\\.England\\(NSA\\)|Middle\\.Atlantic\\(NSA\\)")),
    Midwest_NSA = rowMeans(dplyr::select(., matches("East\\.North\\.Central\\(NSA\\)|West\\.North\\.Centr\\(NSA\\)")),
    South_NSA = rowMeans(dplyr::select(., matches("South\\.Atlantic\\(NSA\\)|East\\.South\\.Central\\(NSA\\)")),
    West_NSA = rowMeans(dplyr::select(., matches("Mountain\\(NSA\\)|Pacific\\(NSA\\)")), na.rm = TRUE),
  )

#selected only necessary columns before plotting
final_housing_transformations <- major_regionsmean %>%
  dplyr::select(Year, West_NSA, Midwest_NSA, South_NSA, Northeast_NSA,
               West_SA, Midwest_SA, South_SA, Northeast_SA)

plot_data <- major_regionsmean %>%
  pivot_longer(cols = c(Northeast_SA, Midwest_SA, South_SA, West_SA,
                       Northeast_NSA, Midwest_NSA, South_NSA, West_NSA),
               names_to = "Type",
               values_to = "HPI") %>%
#reformatted values as column "HPI" and type for each region
  mutate(
    Region = case_when(
      grepl("Northeast", Type) ~ "Northeast",
      grepl("Midwest", Type) ~ "Midwest",
      grepl("South", Type) ~ "South",
      grepl("West", Type) ~ "West"
    ),
    Adjusted = ifelse(grepl("_SA", Type), "Seasonally Adjusted", "Non-Adjusted")
  ) %>%
  dplyr::select(Year, Type, Region, HPI, Adjusted)

#plotting the faceted line + area
ggplot() +
#for each major region
  #plots the seasonally adjusted data only with the area filled
  geom_area(data = filter(plot_data, Adjusted == "Seasonally Adjusted"),

```

```

    aes(x = Year, y = HPI, fill = Adjusted, group = Type),
    alpha = 0.4) +
#plots the non-seasonally adjusted data only in lines
geom_line(data = filter(plot_data, Adjusted == "Non-Adjusted"),
    aes(x = Year, y = HPI, color = Adjusted, group = Type),
    size = 1) +

#sets the color for fill area and line
    scale_color_manual(values = c(
        "Non-Adjusted" = "#2980b9"           # line color for NSA
    )) +
    scale_fill_manual(values = c(
        "Seasonally Adjusted" = "#007bff"    #fill color for SA
    )) +
#facet across regions
facet_wrap(~Region, scales = "fixed") +
labs(
    x = "Year",
    y = "Average HPI",
    fill = "HPI Type",
    color = "HPI Type"
) +
theme_minimal() +
theme(legend.position = "bottom")

#####
# Housing Price Summary Table
#####

housing_price <- read_csv("Housing Price Index.xlsx - Data.csv")

#adding date columns
housing_price2 <- housing_price %>%
mutate(Date = as.POSIXct(Date, format = "%m/%d/%Y")) %>% #had to adjust to match the format
mutate(Year = year(Date)) %>%
mutate(Month = month(Date)) %>%
mutate(Day = day(Date))

#Now I need to add month names and then merge
mon <- c("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December")
monthnames <- data.frame(Month = 1:12, longnames = mon)

monthnames1 <- monthnames %>%
mutate(longnames = factor(longnames, levels = mon))
#ensures correct alphabetical order in case instead of
#default alphabetical order

months_housingprices <- merge(housing_price2, monthnames, by = "Month", all = TRUE)
#merges months into the original data

#Renaming longnames to Month
month_renamed <- months_housingprices %>%
    rename("Months" = "longnames")

```

```

HPI_clean <- month_renamed %>%
  select(-c(`USA(NSA)`, `USA(SA)`))
#removed USA(NSA) and USA(SA) since they are not regions
# and also to avoid max and min selecting values
# in these columns

Grouped_Years <- HPI_clean %>%
  mutate(Years = case_when(
    Year %in% c("1991", "1992", "1993", "1994", "1995") ~ "1991-1995",
    Year %in% c("1996", "1997", "1998", "1999", "2000") ~ "1996-2000",
    Year %in% c("2001", "2002", "2003", "2004", "2005") ~ "2001-2005",
    Year %in% c("2006", "2007", "2008", "2009", "2010") ~ "2006-2010",
    Year %in% c("2011", "2012", "2013", "2014", "2015") ~ "2011-2015",
    Year %in% c("2016", "2017", "2018", "2019", "2020") ~ "2016-2020",
    Year %in% c("2021", "2022", "2023") ~ "2021-2023*"
  ))
#excluded 2024 since it is incomplete at the moment

#find min and max across rows
Min_MaxHPI <- Grouped_Years %>%
  mutate(
    pmin = do.call(pmin, c(select(., matches("SA|NSA")), list(na.rm = TRUE))),
    pmax = do.call(pmax, c(select(., matches("SA|NSA")), list(na.rm = TRUE)))
  )

# Summarize the data by Years in min and max across HPIs
Min_MaxHPI2 <- Min_MaxHPI %>%
  group_by(Years) %>%
  #round to whole number with min and max
  summarize(Min_HPI = round(min(pmin, na.rm = TRUE), 0),
    Max_HPI = round(max(pmax, na.rm = TRUE), 0))

Min_MaxHPI3 <- Min_MaxHPI2 %>%
  #removes NA values in Years
  filter(!is.na(Years))

#renamed Min_HPI to Min HPI and Max_HPI to Max HPI
Min_MaxHPI4 <- Min_MaxHPI3 %>%
  rename("Min HPI" = "Min_HPI",
    "Max HPI" = "Max_HPI")

#####
#Making Table

library(gt)

#set as data frame to avoid potential issues with grouping variables and gt()
Min_MaxHPI4 <- as.data.frame(Min_MaxHPI4)

#making a table
Final_MinMaxHPI_Output <- Min_MaxHPI4 %>%
  gt() %>%
  tab_spanner(label = "Time", columns = "Years") %>%

```

```

#bolds Time and HPI
tab_style(style = cell_text(weight = "bold"),
          locations = cells_column_spanners(spanners = "Time")) %>%
tab_style(style = cell_text(align = "center"),
          locations = cells_column_labels(columns = "Years")) %>%
#centers years interval (1991-2023) to the center
tab_style(style = cell_text(align = "center"),
          locations = cells_body(columns = "Years")) %>%
tab_spanner(label = "HPI", columns = c("Min HPI", "Max HPI")) %>%
tab_style(style = cell_text(weight = "bold"),
          locations = cells_column_spanners(spanners = "HPI")) %>%
tab_footnote(
  "How does HPI compare over the years in terms of maximum and minimum? HPI remain untouched, although
tab_source_note(
  "Note: Asterick(*) by 2021-2023 indicates an incomplete interval. 2024 have incomplete data where it .
) %>%
#stylize footnote and source_note
tab_style(style = cell_text(size = 'small', color = 'black'),
          locations = list(
cells_footnotes(),
cells_source_notes ()
)) %>%
#colors the footnote and source_note bottom of cell to gray
tab_style(
  style = cell_fill(color = "gray"),
  locations = list(
    cells_footnotes(),
    cells_source_notes()
  )
)

gtsave(Final_MinMaxHPI_Output, "Final_MinMaxHPI_Output.pdf")
knitr::include_graphics("Final_MinMaxHPI_Output.pdf") #to pull it back in

```