Idea 2

Sonya Eason

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(patchwork)
library(lmerTest)
```

```
Loading required package: lme4
Loading required package: Matrix

Attaching package: 'Matrix'

The following objects are masked from 'package:tidyr':

    expand, pack, unpack
```

```
Warning in check_dep_version(): ABI version mismatch:
lme4 was built with Matrix ABI version 2
Current Matrix ABI version is 1
Please re-install lme4 from source or restore original 'Matrix' package
```

```
Attaching package: 'lmerTest'

The following object is masked from 'package:lme4':

    lmer

The following object is masked from 'package:stats':

    step
```

```r
library(knitr)
library(broom)
```

```r
unemployment <- read_csv("data/Unemployment.csv")
```

```
Rows: 1848 Columns: 5
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): quarter, state
dbl (3): year, month, unemployment_rate

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
price <- read_csv("data/Price and Availability Data.csv")
```

```
Rows: 1680 Columns: 18
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (3): quarter, market, internal_class
dbl (15): year, RBA, available_space, availability_proportion, internal_clas...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
occupancy <- read_csv("data/Major Market Occupancy Data-revised.csv")
```

```
Rows: 190 Columns: 6
-- Column specification --------------------------------------------------------
```

```
Delimiter: ","
chr (2): quarter, market
dbl (4): year, ending_occupancy_proportion, starting_occupancy_proportion, a...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
leases <- read_csv("data/Leases.csv")
```

```
Rows: 194685 Columns: 35
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (17): quarter, monthsigned, market, building_name, building_id, address,...
dbl (18): year, zip, leasedSF, costarID, RBA, available_space, availability_...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
income <- read_csv("data/ACSST5Y2023.S2411-Data.csv", skip = 1)
```

```
New names:
Rows: 29 Columns: 291
-- Column specification
-------------------------------------------------------- Delimiter: "," chr
(8): Geography, Geographic Area Name, Estimate!!Median earnings (dolla... dbl
(282): Estimate!!Median earnings (dollars)!!Civilian employed population... lgl
(1): ...291
i Use `spec()` to retrieve the full column specification for this data. i
Specify the column types or set `show_col_types = FALSE` to quiet this message.
* `` -> `...291`
```

```r
# Filter columns whose names contain the word "computer" (case-insensitive)
income_filtered <- income[, grep("Computer", names(income), ignore.case = TRUE)]

# View the filtered dataframe
head(income_filtered)
```

```
# A tibble: 6 x 32
  Estimate!!Median earnings (dol~1 Margin of Error!!Med~2 Estimate!!Median ear~3
                             <dbl>                  <dbl>                  <dbl>
```

```
1                            92496                   1522                  94140
2                            98669                   1287                  99819
3                           110826                   2036                 112334
4                           108910                   1420                 113034
5                           150927                   3424                 168819
6                           168523                   1448                 179740
# i abbreviated names:
#   1: `Estimate!!Median earnings (dollars)!!Civilian employed population 16 years and over
#   2: `Margin of Error!!Median earnings (dollars)!!Civilian employed population 16 years and
#   3: `Estimate!!Median earnings (dollars)!!Civilian employed population 16 years and over
# i 29 more variables:
#   `Margin of Error!!Median earnings (dollars)!!Civilian employed population 16 years and o
#   `Estimate!!Median earnings (dollars)!!Civilian employed population 16 years and over wit
```

```r
cs <- income_filtered[,1]

tech <- data.frame(location = income$`Geographic Area Name`, income = cs)

colnames(tech)[2] <- "income"
```

```r
library(dplyr)

tech$GEOID <- c(
  "04013", "06037", "06059", "06073", "06075", "06085", "08031", "11001",
  "12057", "12086", "13121", "17031", "24017", "24510", "25025", "26163",
  "34013", "36061", "37119", "37183", "42101", "47037", "48113", "48201",
  "48439", "48453", "49035", "51059", "53033"
)
```

```r
library(tigris)
```

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

```r
library(sf)
```

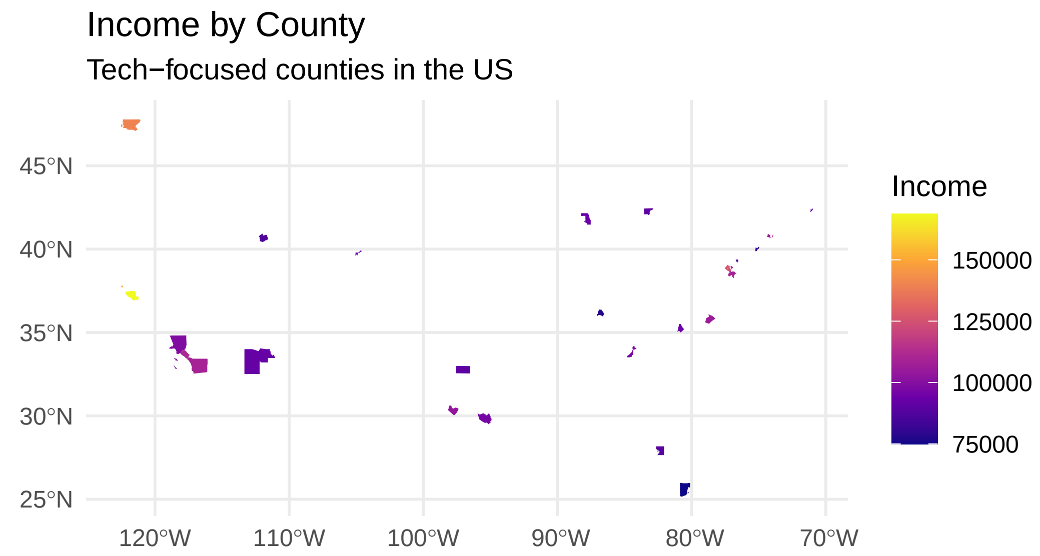
```
Linking to GEOS 3.10.2, GDAL 3.4.1, PROJ 8.2.1; sf_use_s2() is TRUE
```

```
library(dplyr)
library(ggplot2)
library(stringr)
```

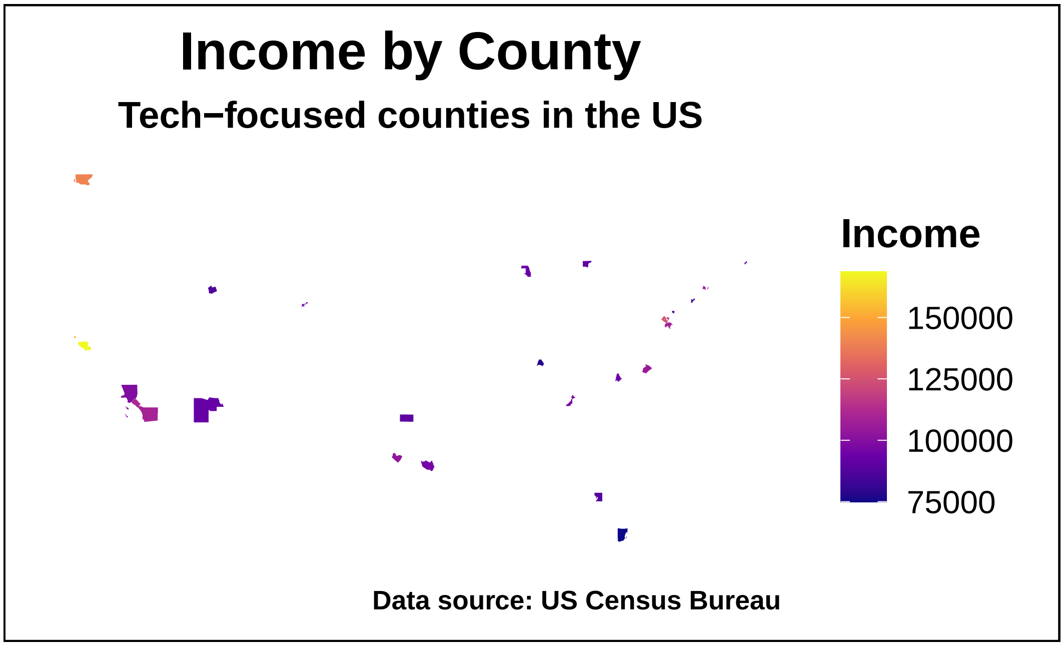
```
library(sf)


shapefile_path <- "data/cb_2022_us_county_5m.shp"
counties_sf <- st_read(shapefile_path)
```

```
Reading layer `cb_2022_us_county_5m' from data source
  `/home/guest/GLMs/Datafest/data/cb_2022_us_county_5m.shp' using driver `ESRI Shapefile'
Simple feature collection with 3235 features and 12 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -179.1473 ymin: -14.55255 xmax: 179.7785 ymax: 71.35256
Geodetic CRS:  NAD83
```

```
# Join your data with the shapefile
map_df <- counties_sf %>%
  left_join(tech, by = "GEOID") %>%
  filter(!is.na(income)) # filter only counties in your data

# Plot
ggplot(map_df) +
  geom_sf(aes(fill = income), color = NA) +
  scale_fill_viridis_c(option = "plasma", name = "Income", na.value = "grey90") +
  theme_minimal() +
  labs(title = "Income by County", subtitle = "Tech-focused counties in the US")
```



```
# Join your data with the shapefile
map_df <- counties_sf %>%
  left_join(tech, by = "GEOID") %>%
  filter(!is.na(income))  # filter only counties in your data

# Plot with bolded black titles and labels
ggplot(map_df) +
  geom_sf(aes(fill = income), color = NA) +
  scale_fill_viridis_c(option = "plasma", name = "Income", na.value = "grey90") +
  theme_minimal(base_size = 15) +  # Set base size for consistency
  labs(
    title = "Income by County",
    subtitle = "Tech-focused counties in the US",
    caption = "Data source: US Census Bureau"
  ) +
  theme(
    plot.title = element_text(face = "bold", size = 20, color = "black", hjust = 0.5),
    plot.subtitle = element_text(face = "bold", size = 14, color = "black", hjust = 0.5),
    plot.caption = element_text(face = "bold", size = 10, color = "black", hjust = 1),
    axis.text = element_blank(),  # Removing axis text
    axis.title = element_blank(),  # Removing axis titles
    legend.title = element_text(face = "bold", color = "black"),
    legend.text = element_text(color = "black"),
    panel.grid = element_blank(),  # Remove gridlines for a cleaner look
    plot.margin = margin(10, 10, 10, 10),  # Add some space around the plot
    plot.background = element_rect(fill = "white")  # Clean white background
  )
```

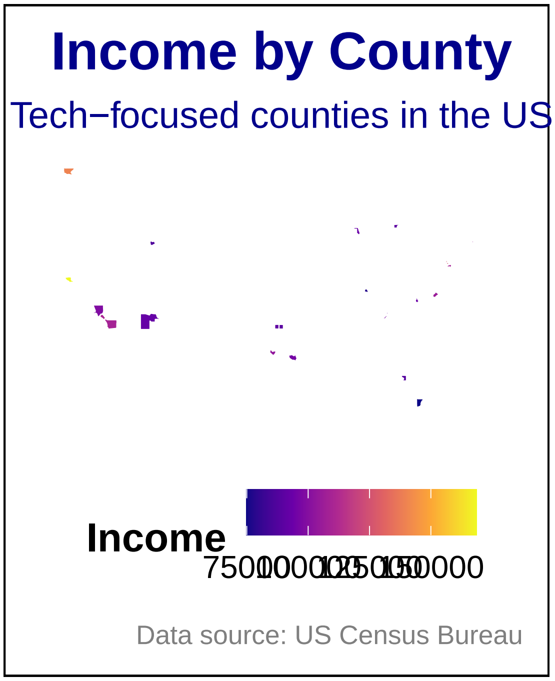


Income by County

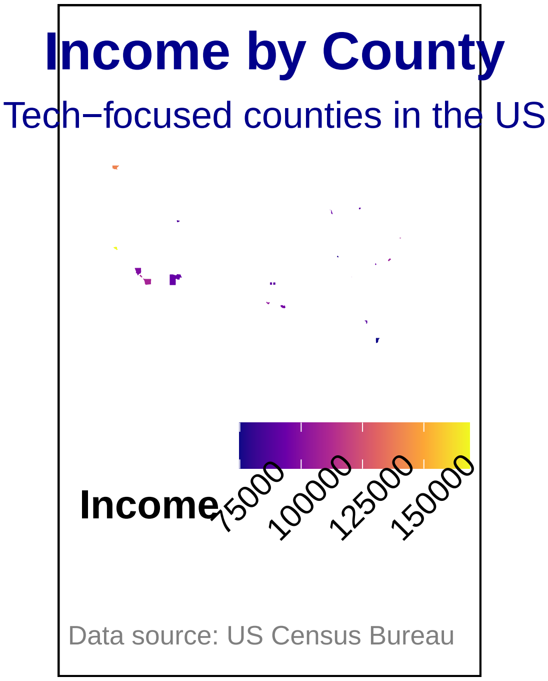
Tech−focused counties in the US

Data source: US Census Bureau

```
ggplot(map_df) +
  geom_sf(aes(fill = income), color = "white", size = 0.1) +  # Adding borders with white col
  scale_fill_viridis_c(option = "plasma", name = "Income", na.value = "grey90") +  # Better
  labs(
    title = "Income by County",
    subtitle = "Tech-focused counties in the US",
    caption = "Data source: US Census Bureau"
  ) +
  theme_minimal(base_size = 15) +  # Bigger font size for better readability
  theme(
    plot.title = element_text(face = "bold", size = 20, hjust = 0.5, color = "darkblue"),
    plot.subtitle = element_text(size = 14, hjust = 0.5, color = "darkblue"),
    plot.caption = element_text(size = 10, hjust = 1, color = "gray50"),
    axis.text = element_blank(),
    axis.title = element_blank(),
    legend.position = "bottom",  # Move the legend to the bottom
    legend.title = element_text(face = "bold"),
    legend.text = element_text(size = 12),
    panel.grid = element_blank(),  # Remove gridlines for a cleaner look
    plot.margin = margin(10, 10, 10, 10)  # Add some space around the plot
  ) +
  # Add a border around the plot
  coord_sf() +
  theme(plot.background = element_rect(fill = "white"))
```

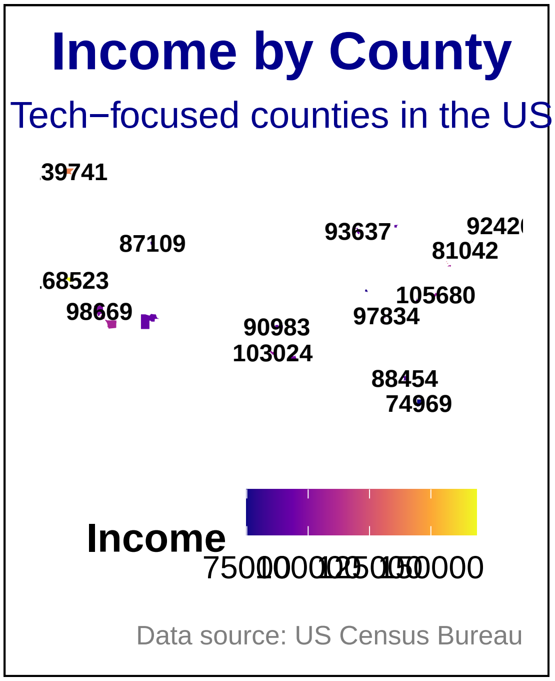

Income by County
Tech−focused counties in the US
Income
75000100000125000150000
Data source: US Census Bureau

```
ggplot(map_df) +
  geom_sf(aes(fill = income), color = "white", size = 0.1) +  # Adding borders with white col
  scale_fill_viridis_c(option = "plasma", name = "Income", na.value = "grey90") +  # Better
  labs(
    title = "Income by County",
    subtitle = "Tech-focused counties in the US",
    caption = "Data source: US Census Bureau"
  ) +
  theme_minimal(base_size = 15) +  # Bigger font size for better readability
  theme(
    plot.title = element_text(face = "bold", size = 20, hjust = 0.5, color = "darkblue"),
    plot.subtitle = element_text(size = 14, hjust = 0.5, color = "darkblue"),
    plot.caption = element_text(size = 10, hjust = 1, color = "gray50"),
    axis.text = element_blank(),
    axis.title = element_blank(),
    legend.position = "bottom",  # Move the legend to the bottom
    legend.title = element_text(face = "bold"),
    legend.text = element_text(size = 12, angle = 45),  # Tilt legend labels by 45 degrees
    panel.grid = element_blank(),  # Remove gridlines for a cleaner look
    plot.margin = margin(10, 10, 10, 10)  # Add some space around the plot
  ) +
  # Add a border around the plot
  coord_sf() +
  theme(plot.background = element_rect(fill = "white"))
```


Income by County
Tech−focused counties in the US
Income
75000
100000
125000
150000
Data source: US Census Bureau

```
ggplot(map_df) +
  geom_sf(aes(fill = income), color = "white", size = 0.1) +  # Adding borders with white col
  scale_fill_viridis_c(option = "plasma", name = "Income", na.value = "grey90",
                       limits = c(min(map_df$income, na.rm = TRUE), max(map_df$income, na.rm
                       oob = scales::squish) +  # Adjust color limits and squish outliers
  labs(
    title = "Income by County",
    subtitle = "Tech-focused counties in the US",
    caption = "Data source: US Census Bureau"
  ) +
  theme_minimal(base_size = 15) +  # Bigger font size for better readability
  theme(
    plot.title = element_text(face = "bold", size = 20, hjust = 0.5, color = "darkblue"),
    plot.subtitle = element_text(size = 14, hjust = 0.5, color = "darkblue"),
    plot.caption = element_text(size = 10, hjust = 1, color = "gray50"),
    axis.text = element_blank(),
    axis.title = element_blank(),
    legend.position = "bottom",  # Move the legend to the bottom
    legend.title = element_text(face = "bold"),
    legend.text = element_text(size = 12),
    panel.grid = element_blank(),  # Remove gridlines for a cleaner look
    plot.margin = margin(10, 10, 10, 10),  # Add some space around the plot
    plot.background = element_rect(fill = "white")  # Clean white background
  ) +
  # Optionally add county labels to show income (you can remove or adjust this)
  geom_sf_text(aes(label = round(income, 0)), size = 3, color = "black",
               fontface = "bold", check_overlap = TRUE) +
  coord_sf()
```

```
Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
give correct results for longitude/latitude data
```



```r
biz_income_filtered <- income[, grep("Business", names(income), ignore.case = TRUE)]

# View the filtered dataframe
head(biz_income_filtered)
```

```
# A tibble: 6 x 128
  Estimate!!Median earnings (dol~1 Margin of Error!!Med~2 Estimate!!Median ear~3
                             <dbl>                  <dbl>                  <dbl>
1                            72567                    581                  81942
2                            79997                    504                  85983
3                            91593                    756                  99499
4                            86450                    909                  91054
5                           125099                   2933                 138637
6                           136030                   1830                 139866
# i abbreviated names:
#   1: `Estimate!!Median earnings (dollars)!!Civilian employed population 16 years and over w
#   2: `Margin of Error!!Median earnings (dollars)!!Civilian employed population 16 years and
#   3: `Estimate!!Median earnings (dollars)!!Civilian employed population 16 years and over w
# i 125 more variables:
#   `Margin of Error!!Median earnings (dollars)!!Civilian employed population 16 years and ov
#   `Estimate!!Median earnings (dollars)!!Civilian employed population 16 years and over with
```

```r
biz <- income_filtered[,1]
business <- data.frame(location = income$`Geographic Area Name`, income = biz)

colnames(business)[2] <- "income"
```

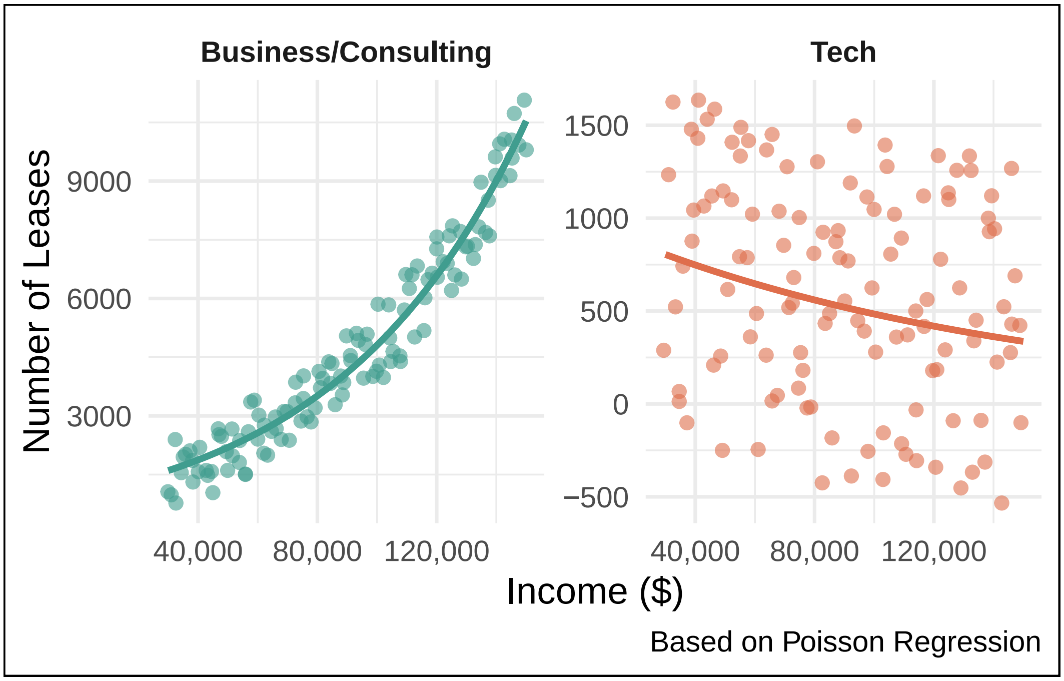
```r
library(dplyr)

# Mapping of counties to cities
county_to_city <- c(
  "Maricopa County, Arizona" = "Phoenix",
  "Los Angeles County, California" = "Los Angeles",
  "Orange County, California" = "Orange County",
  "San Diego County, California" = "San Diego",
  "San Francisco County, California" = "San Francisco",
  "Santa Clara County, California" = "South Bay/San Jose",
  "Denver County, Colorado" = "Denver",
  "District of Columbia, District of Columbia" = "Washington D.C.",
  "Hillsborough County, Florida" = "Tampa",
  "Miami-Dade County, Florida" = "South Florida",
  "Fulton County, Georgia" = "Atlanta",
  "Cook County, Illinois" = "Chicago",
  "Charles County, Maryland" = "Southern Maryland",
  "Baltimore city, Maryland" = "Baltimore",
  "Suffolk County, Massachusetts" = "Boston",
  "Wayne County, Michigan" = "Detroit",
  "Essex County, New Jersey" = "Northern New Jersey",
  "New York County, New York" = "Manhattan",
  "Mecklenburg County, North Carolina" = "Charlotte",
  "Wake County, North Carolina" = "Raleigh/Durham",
  "Philadelphia County, Pennsylvania" = "Philadelphia",
  "Davidson County, Tennessee" = "Nashville",
  "Dallas County, Texas" = "Dallas/Ft Worth",
  "Harris County, Texas" = "Houston",
  "Tarrant County, Texas" = "Dallas/Ft Worth",
  "Travis County, Texas" = "Austin",
  "Salt Lake County, Utah" = "Salt Lake City",
  "Fairfax County, Virginia" = "Northern Virginia",
  "King County, Washington" = "Seattle"
)

# Add a new column to tech dataframe with the matched city based on county
business <- business %>%
  mutate(city = county_to_city[business$location])
```

```
leases |>
  filter(internal_industry%in%c("Technology, Advertising, Media, and Information", "Business
  group_by(internal_industry, market) |>
  count()
```

```
# A tibble: 58 x 3
# Groups:   internal_industry, market [58]
   internal_industry                                                market     n
   <chr>                                                            <chr>  <int>
 1 Business, Professional, and Consulting Services (except Financi~ Atlan~   127
 2 Business, Professional, and Consulting Services (except Financi~ Austin    40
 3 Business, Professional, and Consulting Services (except Financi~ Balti~    59
 4 Business, Professional, and Consulting Services (except Financi~ Boston    68
 5 Business, Professional, and Consulting Services (except Financi~ Charl~    45
 6 Business, Professional, and Consulting Services (except Financi~ Chica~   124
 7 Business, Professional, and Consulting Services (except Financi~ Chica~    83
 8 Business, Professional, and Consulting Services (except Financi~ Dalla~    93
 9 Business, Professional, and Consulting Services (except Financi~ Denver    68
10 Business, Professional, and Consulting Services (except Financi~ Detro~    41
# i 48 more rows
```

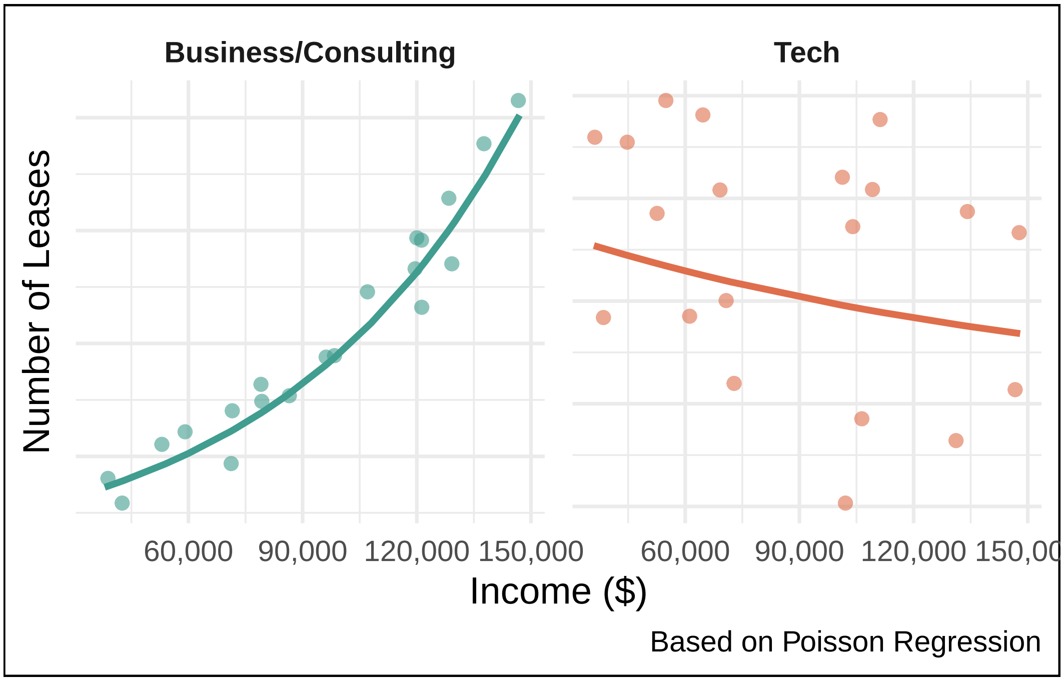
```r
# Load necessary libraries
library(ggplot2)
library(dplyr)

# Simulate the data for the two types of companies (business/consulting and tech)
set.seed(123)

# Income range (for simplicity, I'm assuming a range from 30k to 150k)
income_range <- seq(30000, 150000, by = 1000)

# Business/Managerial/Consulting companies: 17% increase in leases per 10k increase in income
business_lease <- 1000 * (1.17)^(income_range / 10000)

# Tech companies: 7% decrease in leases per 10k increase in income
tech_lease <- 1000 * (1 - 0.07)^(income_range / 10000)

# Create a data frame
data <- data.frame(
  income = rep(income_range, 2),
  leases = c(business_lease, tech_lease),
  company_type = rep(c("Business/Consulting", "Tech"), each = length(income_range))
)

# Define custom colors using the provided hex codes
custom_colors <- c("Business/Consulting" = "#409D8F", "Tech" = "#DF6E4C")

# Plot
ggplot(data, aes(x = income, y = leases, color = company_type)) +
  geom_line(size = 1.2) +  # Line plot for leases
  geom_jitter(width = 1000, height = 1000, alpha = 0.6, size = 2) +  # Add jittered points wi
  facet_wrap(~ company_type, scales = "free_y") +  # Facet by company type
  scale_x_continuous(labels = scales::comma) +
  scale_color_manual(values = custom_colors) +
  labs(
    x = "Income ($)",
    y = "Number of Leases",
    caption = "Based on Poisson Regression"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    strip.text = element_text(face = "bold"),
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(size = 12),
    legend.position = "none",  # Remove legend since we have facet labels
    plot.background = element_rect(fill = "white")  # Clean background
  )
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```



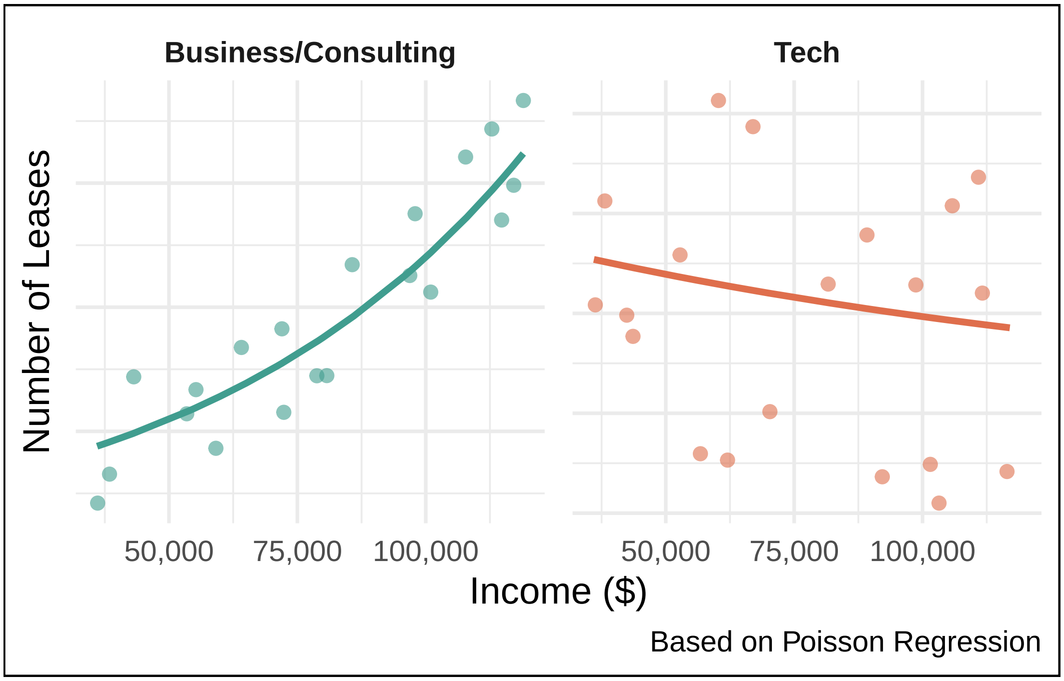
```
# Load necessary libraries
library(ggplot2)
library(dplyr)

# Simulate the data for the two types of companies (business/consulting and tech)
set.seed(123)

# Income range (for simplicity, I'm assuming a range from 30k to 150k)
income_range <- seq(30000, 150000, by = 1000)
```

```r
# Business/Managerial/Consulting companies: 17% increase in leases per 10k increase in income
business_lease <- 1000 * (1.17)^(income_range / 10000)

# Tech companies: 7% decrease in leases per 10k increase in income
tech_lease <- 1000 * (1 - 0.07)^(income_range / 10000)

# Create a data frame
data <- data.frame(
  income = rep(income_range, 2),
  leases = c(business_lease, tech_lease),
  company_type = rep(c("Business/Consulting", "Tech"), each = length(income_range))
)

# Subset the data to select only 20 points from each company type
data_subset <- data %>%
  group_by(company_type) %>%
  sample_n(20)  # Take a random sample of 20 points for each company type

# Define custom colors using the provided hex codes
custom_colors <- c("Business/Consulting" = "#409D8F", "Tech" = "#DF6E4C")

# Plot
ggplot(data_subset, aes(x = income, y = leases, color = company_type)) +
  geom_line(size = 1.2) +  # Line plot for leases
  geom_jitter(width = 1000, height = 1000, alpha = 0.6, size = 2) +  # Add jittered points wi
  facet_wrap(~ company_type, scales = "free_y") +  # Facet by company type
  scale_x_continuous(labels = scales::comma) +
  scale_color_manual(values = custom_colors) +
  labs(
    x = "Income ($)",
    y = "Number of Leases",
    caption = "Based on Poisson Regression"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    strip.text = element_text(face = "bold"),
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(size = 12),
    legend.position = "none",  # Remove legend since we have facet labels
    axis.text.y = element_blank(),  # Remove y-axis labels
    plot.background = element_rect(fill = "white")  # Clean background
  )
```



```
# Load necessary libraries
library(ggplot2)
library(dplyr)

# Simulate the data for the two types of companies (business/consulting and tech)
set.seed(123)

# Income range (for simplicity, I'm assuming a range from 30k to 150k)
income_range <- seq(30000, 120000, by = 1000)

# Business/Managerial/Consulting companies: 17% increase in leases per 10k increase in income
business_lease <- 1000 * (1.17)^(income_range / 10000)

# Tech companies: 7% decrease in leases per 10k increase in income
tech_lease <- 1000 * (1 - 0.07)^(income_range / 10000)

# Create a data frame
data <- data.frame(
  income = rep(income_range, 2),
  leases = c(business_lease, tech_lease),
  company_type = rep(c("Business/Consulting", "Tech"), each = length(income_range))
)
```

```r
# Subset the data to select only 20 points from each company type
data_subset <- data %>%
  group_by(company_type) %>%
  sample_n(20)  # Take a random sample of 20 points for each company type

# Define custom colors using the provided hex codes
custom_colors <- c("Business/Consulting" = "#409D8F", "Tech" = "#DF6E4C")

# Plot
ggplot(data_subset, aes(x = income, y = leases, color = company_type)) +
  geom_line(size = 1.2) +  # Line plot for leases
  geom_jitter(width = 1000, height = 1000, alpha = 0.6, size = 2) +  # Add jittered points wi
  facet_wrap(~ company_type, scales = "free_y") +  # Facet by company type
  scale_x_continuous(labels = scales::comma) +
  scale_color_manual(values = custom_colors) +
  labs(
    x = "Income ($)",
    y = "Number of Leases",
    caption = "Based on Poisson Regression"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    strip.text = element_text(face = "bold"),
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(size = 12),
    legend.position = "none",  # Remove legend since we have facet labels
    axis.text.y = element_blank(),  # Remove y-axis labels
    plot.background = element_rect(fill = "white"),  # Clean background
    panel.border = element_blank()  # Remove the border around the plot
  )
```

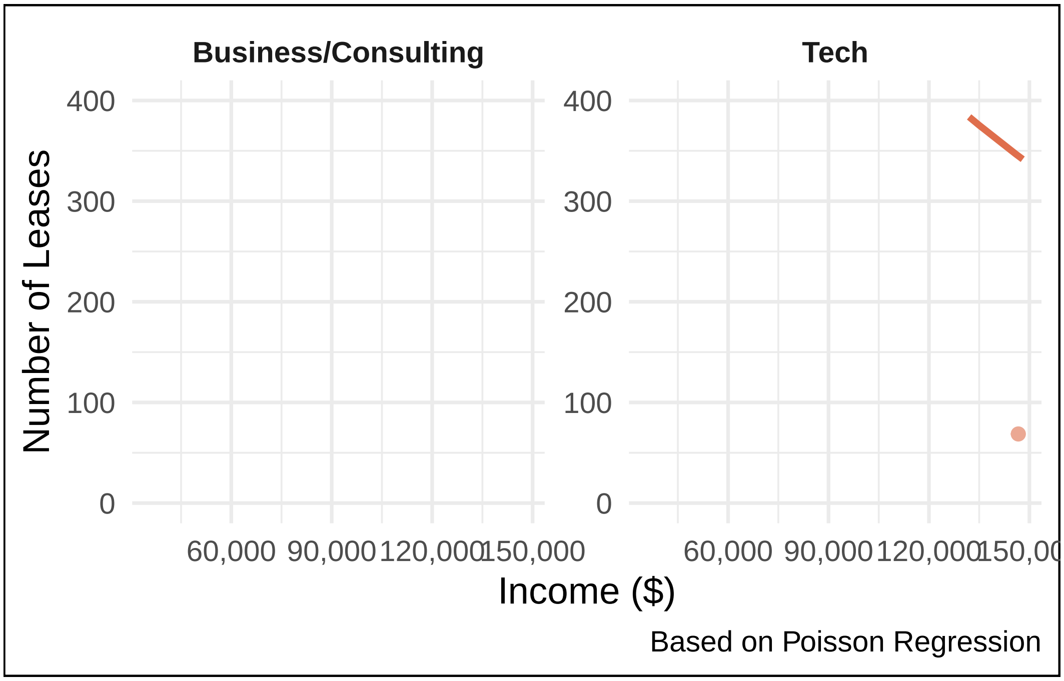


```
# Load necessary libraries
library(ggplot2)
library(dplyr)

# Simulate the data for the two types of companies (business/consulting and tech)
set.seed(123)

# Income range (for simplicity, I'm assuming a range from 30k to 150k)
income_range <- seq(30000, 150000, by = 1000)

# Business/Managerial/Consulting companies: 17% increase in leases per 10k increase in income
business_lease <- 1000 * (1.17)^(income_range / 10000)

# Tech companies: 7% decrease in leases per 10k increase in income
tech_lease <- 1000 * (1 - 0.07)^(income_range / 10000)

# Create a data frame
data <- data.frame(
  income = rep(income_range, 2),
  leases = c(business_lease, tech_lease),
  company_type = rep(c("Business/Consulting", "Tech"), each = length(income_range))
)
```

```r
# Subset the data to select only 20 points from each company type
data_subset <- data %>%
  group_by(company_type) %>%
  sample_n(20)  # Take a random sample of 20 points for each company type

# Define custom colors using the provided hex codes
custom_colors <- c("Business/Consulting" = "#409D8F", "Tech" = "#DF6E4C")

# Plot
ggplot(data_subset, aes(x = income, y = leases, color = company_type)) +
  geom_line(size = 1.2) +  # Line plot for leases
  geom_jitter(width = 1000, height = 1000, alpha = 0.6, size = 2) +  # Add jittered points wi
  facet_wrap(~ company_type, scales = "free_y") +  # Facet by company type
  scale_x_continuous(labels = scales::comma) +
  scale_y_continuous(limits = c(0, 400)) +  # Limit y-axis to 400
  scale_color_manual(values = custom_colors) +
  labs(
    x = "Income ($)",
    y = "Number of Leases",
    caption = "Based on Poisson Regression"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    strip.text = element_text(face = "bold"),
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(size = 12),
    legend.position = "none",  # Remove legend since we have facet labels
    plot.background = element_rect(fill = "white")  # Clean background
  )
```

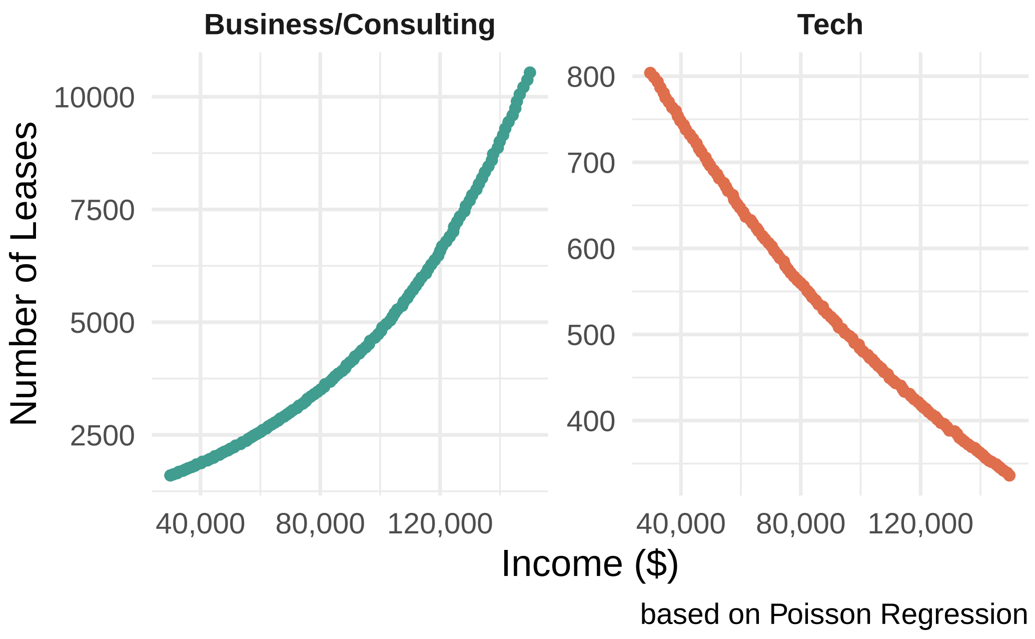
```
Warning: Removed 36 rows containing missing values or values outside the scale range
(`geom_line()`).
```

```
Warning: Removed 39 rows containing missing values or values outside the scale range
(`geom_point()`).
```

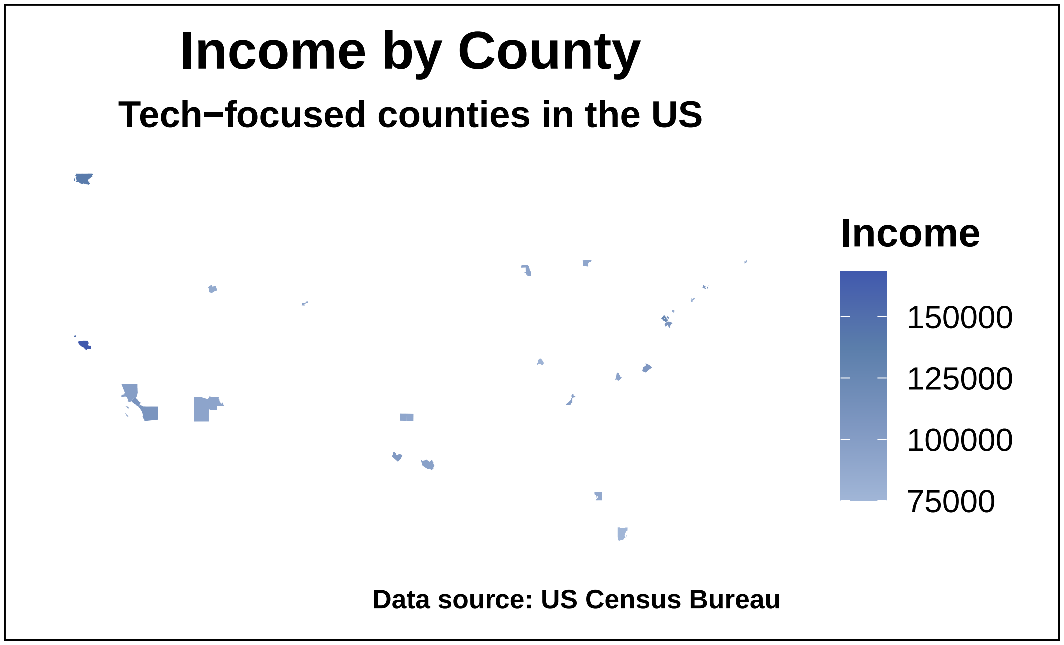


```r
# Load necessary libraries
library(ggplot2)
library(dplyr)

# Simulate the data for the two types of companies (business/consulting and tech)
set.seed(123)

# Income range (for simplicity, I'm assuming a range from 30k to 150k)
income_range <- seq(30000, 150000, by = 1000)

# Business/Managerial/Consulting companies: 17% increase in leases per 10k increase in income
business_lease <- 1000 * (1.17)^(income_range / 10000)

# Tech companies: 7% decrease in leases per 10k increase in income
tech_lease <- 1000 * (1 - 0.07)^(income_range / 10000)

# Create a data frame
data <- data.frame(
  income = rep(income_range, 2),
  leases = c(business_lease, tech_lease),
  company_type = rep(c("Business/Consulting", "Tech"), each = length(income_range))
)
```

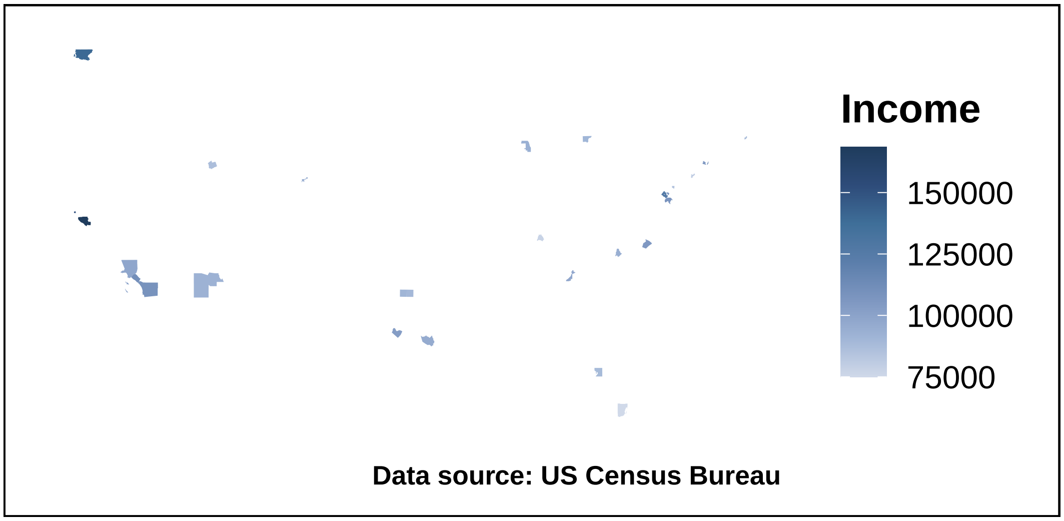
```
# Define custom colors using the provided hex codes
custom_colors <- c("Business/Consulting" = "#409D8F", "Tech" = "#DF6E4C")

# Plot
ggplot(data, aes(x = income, y = leases, color = company_type)) +
  geom_line(size = 1.2) +
  geom_jitter() +# Line plot for leases
  facet_wrap(~ company_type, scales = "free_y") +  # Facet by company type
  scale_x_continuous(labels = scales::comma) +
  scale_color_manual(values = custom_colors) +
  labs(
    x = "Income ($)",
    y = "Number of Leases",
    caption = "based on Poisson Regression"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    strip.text = element_text(face = "bold"),
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(size = 12),
    legend.position = "none"  # Remove legend since we have facet labels
  )
```



```r
# Join your data with the shapefile
map_df <- counties_sf %>%
  left_join(tech, by = "GEOID") %>%
  filter(!is.na(income))  # filter only counties in your data

# Custom blue shades for the gradient
blue_shades <- c("#A1B6D7", "#7E97C1", "#5B7EAB", "#4059AD")  # Lighter to darker blue

# Plot with bolded black titles and labels
ggplot(map_df) +
  geom_sf(aes(fill = income), color = NA) +
  scale_fill_gradientn(colors = blue_shades, name = "Income", na.value = "grey90") +  # Using
  theme_minimal(base_size = 15) +  # Set base size for consistency
  labs(
    title = "Income by County",
    subtitle = "Tech-focused counties in the US",
    caption = "Data source: US Census Bureau"
  ) +
  theme(
    plot.title = element_text(face = "bold", size = 20, color = "black", hjust = 0.5),
    plot.subtitle = element_text(face = "bold", size = 14, color = "black", hjust = 0.5),
    plot.caption = element_text(face = "bold", size = 10, color = "black", hjust = 1),
    axis.text = element_blank(),  # Removing axis text
    axis.title = element_blank(),  # Removing axis titles
    legend.title = element_text(face = "bold", color = "black"),
    legend.text = element_text(color = "black"),
    panel.grid = element_blank(),  # Remove gridlines for a cleaner look
    plot.margin = margin(10, 10, 10, 10),  # Add some space around the plot
    plot.background = element_rect(fill = "white")  # Clean white background
  )
```


Income by County
Tech−focused counties in the US
Income
150000
125000
100000
75000
Data source: US Census Bureau

```
# Join your data with the shapefile
map_df <- counties_sf %>%
  left_join(tech, by = "GEOID") %>%
  filter(!is.na(income))  # filter only counties in your data

# Dramatic blue gradient shades
blue_shades <- c("#D0D9E9", "#A1B6D7", "#7E97C1", "#5B7EAB", "#3F6F99", "#2E4C7A", "#1F3C5D"

# Plot with bolded black titles and labels
ggplot(map_df) +
  geom_sf(aes(fill = income), color = NA) +
  scale_fill_gradientn(colors = blue_shades, name = "Income", na.value = "grey90") +  # Usin
  theme_minimal(base_size = 15) +  # Set base size for consistency
  labs(
    caption = "Data source: US Census Bureau"
  ) +
  theme(
    plot.title = element_text(face = "bold", size = 20, color = "black", hjust = 0.5),
    plot.subtitle = element_text(face = "bold", size = 14, color = "black", hjust = 0.5),
    plot.caption = element_text(face = "bold", size = 10, color = "black", hjust = 1),
    axis.text = element_blank(),  # Removing axis text
    axis.title = element_blank(),  # Removing axis titles
    legend.title = element_text(face = "bold", color = "black"),
    legend.text = element_text(color = "black"),
    panel.grid = element_blank(),  # Remove gridlines for a cleaner look
    plot.margin = margin(10, 10, 10, 10),  # Add some space around the plot
    plot.background = element_rect(fill = "white")  # Clean white background
  )
```



Data source: US Census Bureau