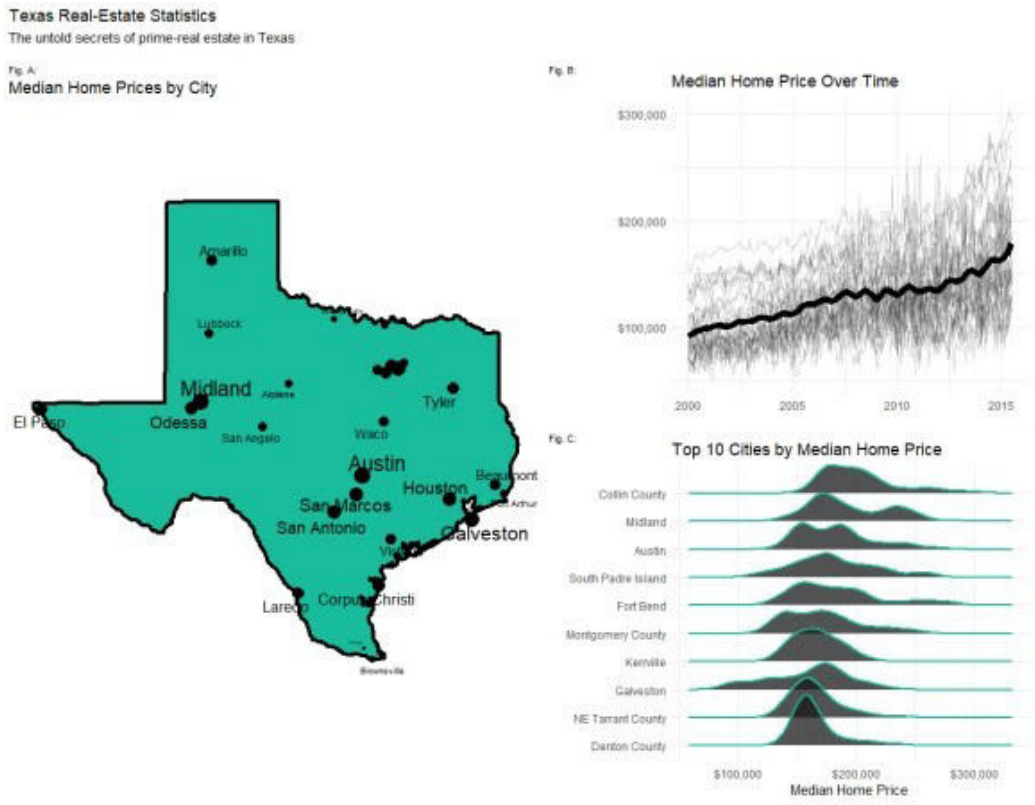
patchwork
Combine multiple ggplots into one.



A Texas Real Estate Storyboard

What we're making today combines 3 plots into 1 illustrative storyboard.

1. Map Plot
2. Time Series Plot
3. Ridgeline Plot

The **ggplot2 package** is great for single plots, but what about creating a storyboard for illustrating ideas, making persuasive arguments, and storytelling?

Now you can **make publication-ready storyboards using patchwork**, the composer of ggplot2. Patchwork makes it ridiculously simple to combine separate ggplots into the same graphic, arranging plots into a grid and adding figures, labels, and annotations.

The `patchwork` package aims to make it easy to combine ggplots by:

- Using a ggplot2 syntax for using a grammar of plot-layout operations.
- Extends the amazing ggplot2 package.

How patchwork works

Here's the simplified patchwork syntax. That's all we need to create a two-column layout with the right-column containing two rows is in the simple syntax. Let's break it down.

```
100
101 # 3.0 Patchwork ----
102
103 gg_tx_map + (gg_tx_timeseries / gg_tx_ridge)
104
```

Patchwork has a very simple syntax where we can create layouts super easily. Here's the general syntax that combines:

1. Two-Column Layout using the **Plus Sign `+`**
2. **Parenthesis `()`** to create a subplot group.
3. Two-Row Layout using the **Division Sign `\`**

This is the basic layout for the Texas Real-Estate Storyboard shown in the chart above. 📊☝️

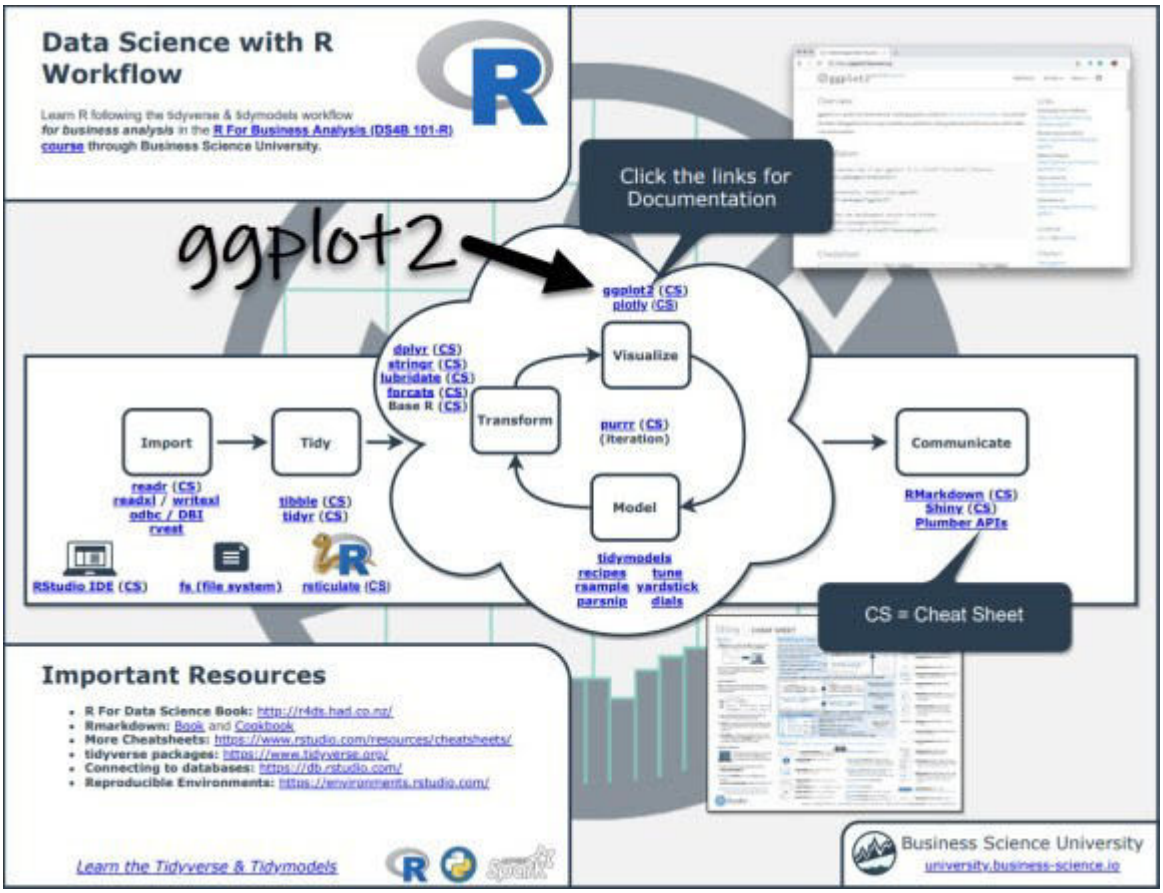
The most important component to patchwork is that **you need to be good at developing `ggplot2` plots** if you want to make your storyboard look great. For that, I'll defer to our Ultimate R Cheat Sheet to help you get up to speed.

Before we get started, get the Cheat Sheet

`patchwork` is great for combining plots. But, you'll still need to learn how to wrangle data with dplyr and visualize data with ggplot2. For those topics, I'll use the Ultimate R Cheat Sheet to refer to `dplyr` and `ggplot2` code in my workflow.

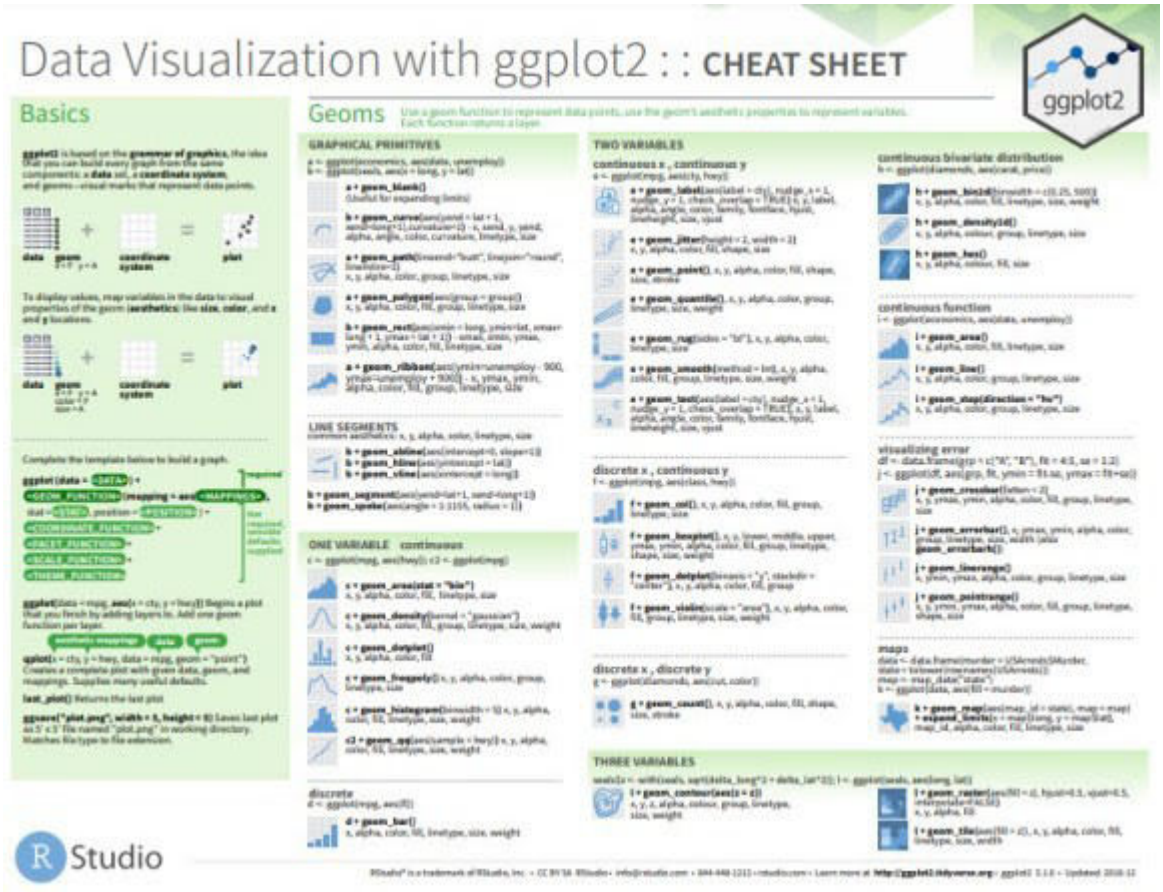
Quick Example:

Download the Ultimate R Cheat Sheet. **Then Click the "CS" next to "ggplot2"** opens the Data Visualization with Dplyr Cheat Sheet.



Now you're ready to quickly reference `ggplot2` functions.

Onto the tutorial.



Step 1: Load Libraries & Data

The libraries we'll need today are patchwork, ggridges, ggrepel, maps, tidyverse, and lubridate. All packages are available on CRAN and can be installed with `install.packages()`.

```
6  # LIBRARIES ----
7
8  library(patchwork)
9  library(ggridges)
10 library(ggrepel)
11 library(maps)
12 library(tidyverse)
13 library(lubridate)
14
```

The dataset is the txhousing data that comes with ggplot2.

```
> txhousing
# A tibble: 8,602 x 9
   city     year month sales   volume median listings inventory  date
   <chr>   <int> <int> <dbl>    <dbl>  <dbl>    <dbl>     <dbl> <dbl>
 1 Abilene  2000     1    72  5380000  71400      701       6.3 2000
 2 Abilene  2000     2    98  6505000  58700      746       6.6 2000.
 3 Abilene  2000     3   130  9285000  58100      784       6.8 2000.
 4 Abilene  2000     4    98  9730000  68600      785       6.9 2000.
 5 Abilene  2000     5   141 10590000  67300      794       6.8 2000.
 6 Abilene  2000     6   156 13910000  66900      780       6.6 2000.
 7 Abilene  2000     7   152 12635000  73500      742       6.2 2000.
 8 Abilene  2000     8   131 10710000  75000      765       6.4 2001.
 9 Abilene  2000     9   104  7615000  64500      771       6.5 2001.
10 Abilene  2000    10   101  7040000  59300      764       6.6 2001.
# ... with 8,592 more rows
```

Step 2: Make our Sub-Plots

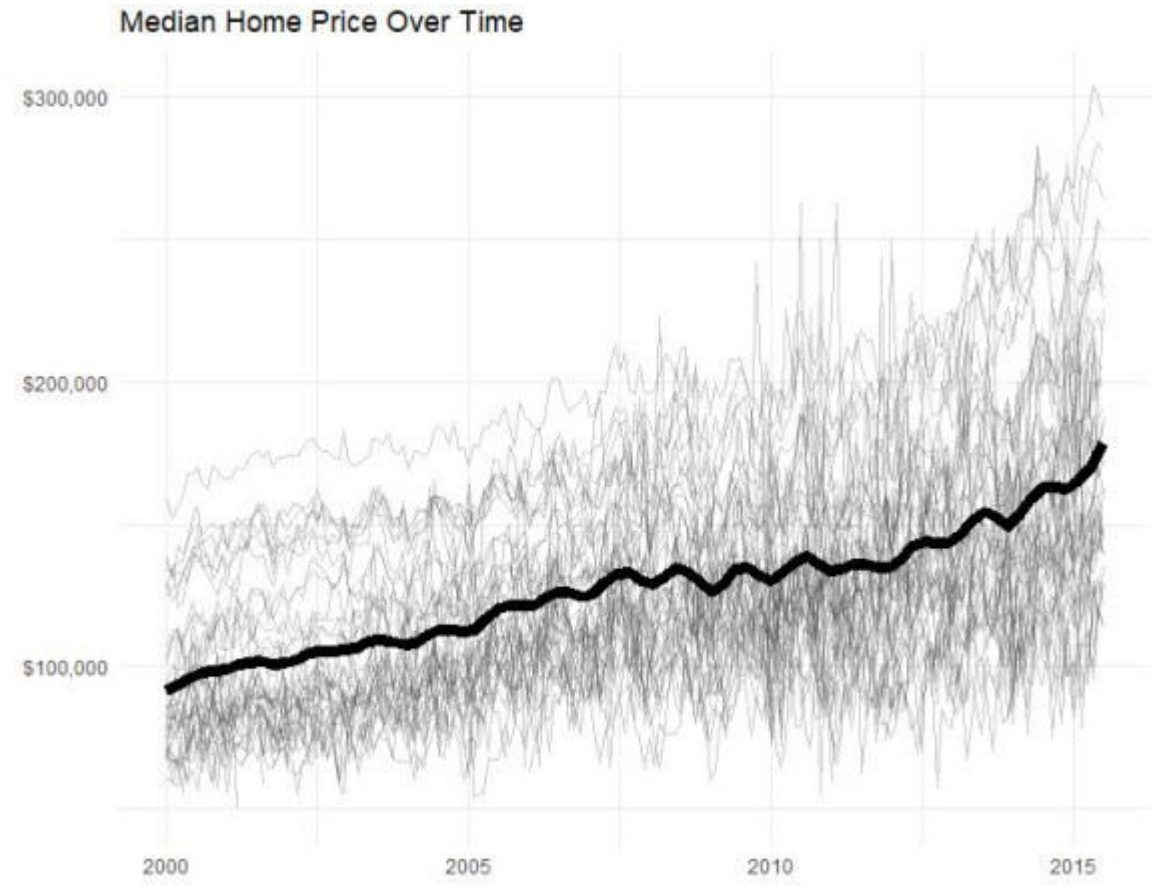
We'll start by making individual plots that are components of our final patchwork storyboard.

Plot 1. Time Series Plot

First, we make a time series plot that shows the smoothed trend in median home prices of all the Texas cities along with their individual trends in median price.

```r
# 2.1 Time Series ----

gg_tx_timeseries <- txhousing_tbl %>%
    ggplot(aes(date, median, group = city)) +
    geom_line(color = "gray20", alpha = 0.25) +
    geom_smooth(
        aes(group = NULL),
        method = "loess",
        span = 0.1,
        se = FALSE,
        size = 2.5,
        color = "black"
    ) +
    theme_minimal() +
    scale_y_continuous(labels = scales::dollar_format()) +
    labs(y = "", x = "", title = "Median Home Price Over Time")
gg_tx_timeseries
```

Get the Code

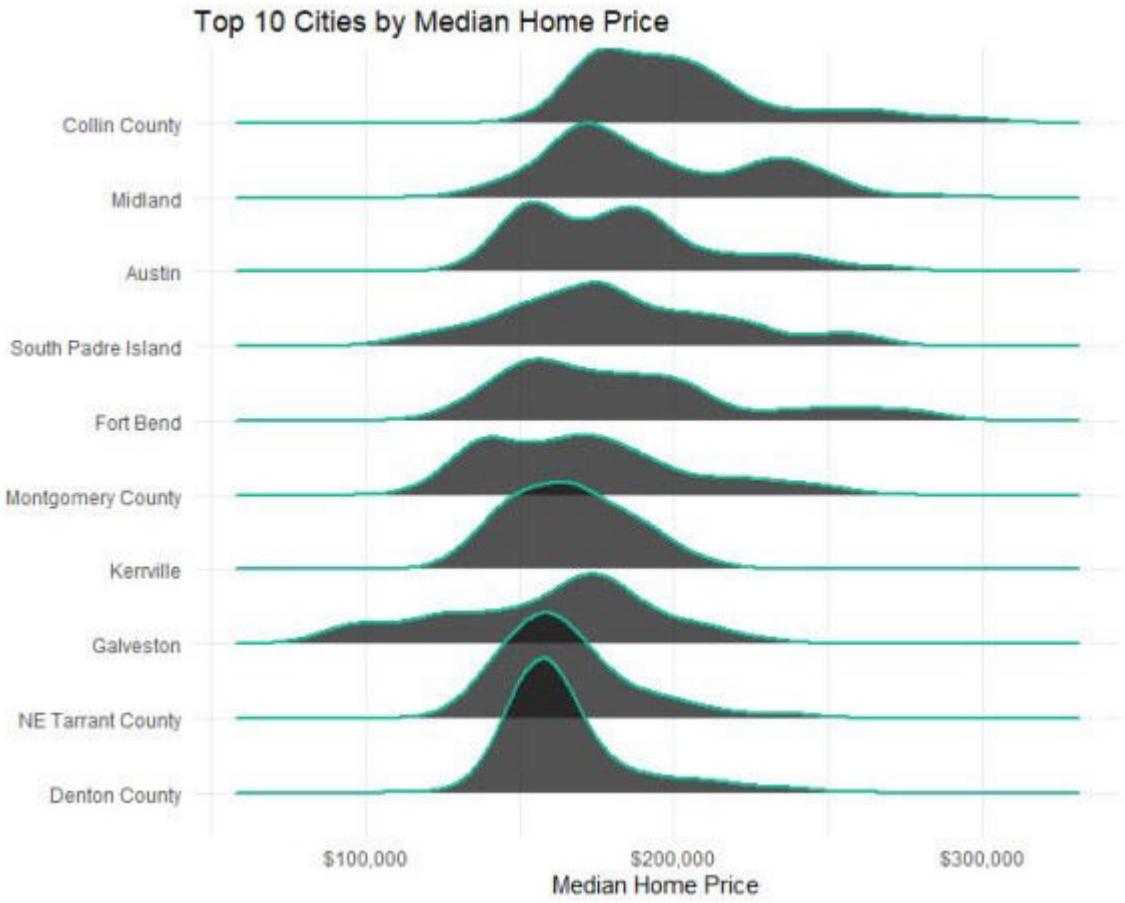


Plot 2. Ridgeline Plot

Next is creating a ridgeline plot, a special plot that shows distribution of median home prices by top-10 cities in an aesthetically pleasing visual. PS. I teach data visualization with ggplot2 in-depth in the R for Business Analysis course.

```
gg_tx_ridge <- txhousing_tbl %>%
    drop_na() %>%
    mutate(city = factor(city) %>% fct_reorder(median) %>% fct_rev()) %>%
    filter(as.numeric(city) %in% (1:10)) %>%

    ggplot(aes(x = median, y = fct_rev(city))) +
    geom_density_ridges(
        color = "#18BC9C",
        fill  = "gray10",
        alpha = 0.75,
        size  = 1
    ) +
    scale_x_continuous(labels = scales::dollar_format()) +
    theme_minimal() +
    labs(x = "Median Home Price", y = "", title = "Top 10 Cities by Median Home Price")

gg_tx_ridge
```

Get the Code

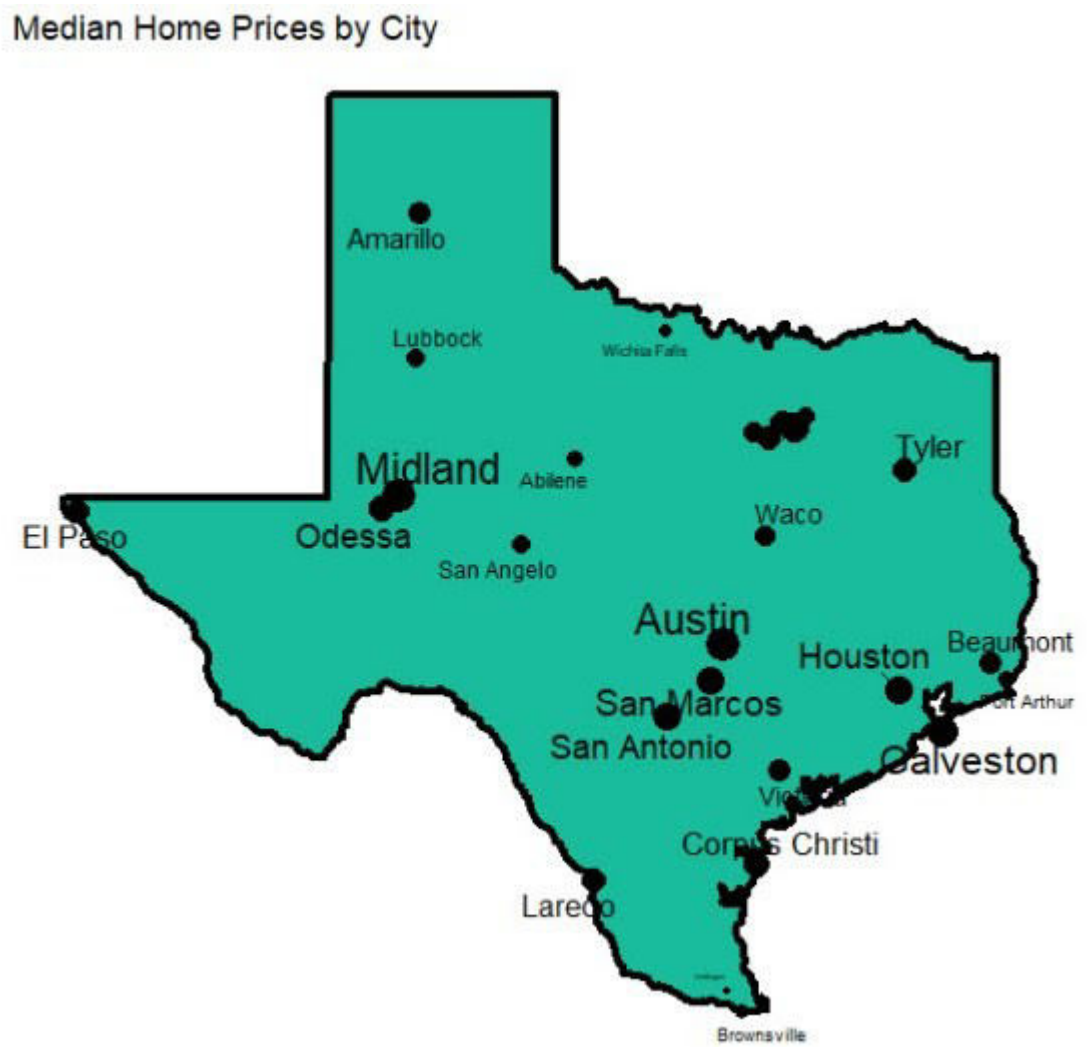


Plot 3. The Map Plot

The final plot is a geographic map that showcases the summary of which cities have the highest median home prices in Texas.

```r
texas_housing_tbl <- txhousing_tbl %>%
    group_by(city) %>%
    summarise(median = median(median, na.rm = T)) %>%
    ungroup() %>%
    mutate(city = str_to_lower(city))

texas_cities_tbl <- us.cities %>%
    filter(country.etc == "TX") %>%
    mutate(name = name %>%
                str_sub(end = nchar(name) - 3) %>%
                str_to_lower() %>%
                str_trim()
    ) %>%
    left_join(texas_housing_tbl, by = c("name" = "city"))

texas_outline_tbl <- map_data("state", region = "texas") %>% as_tibble()

gg_tx_map <- texas_cities_tbl %>%
    drop_na() %>%
    ggplot(aes(x = long, y = lat, size = median)) +
    geom_polygon(
        data  = texas_outline_tbl,
        aes(x = long, y = lat, group = group, size = NULL),
        color = "black",
        size = 1.5,
        fill = "#18bc9c"
    ) +
    geom_point() +
    geom_text_repel(aes(label = str_to_title(name)), max.overlaps = 5) +
    # scale_color_viridis_c() +
    coord_map() +
    theme_void() +
    labs(title = "Median Home Prices by City", x = "", y = "") +
    theme(legend.position = "none")

gg_tx_map
```

Get the Code

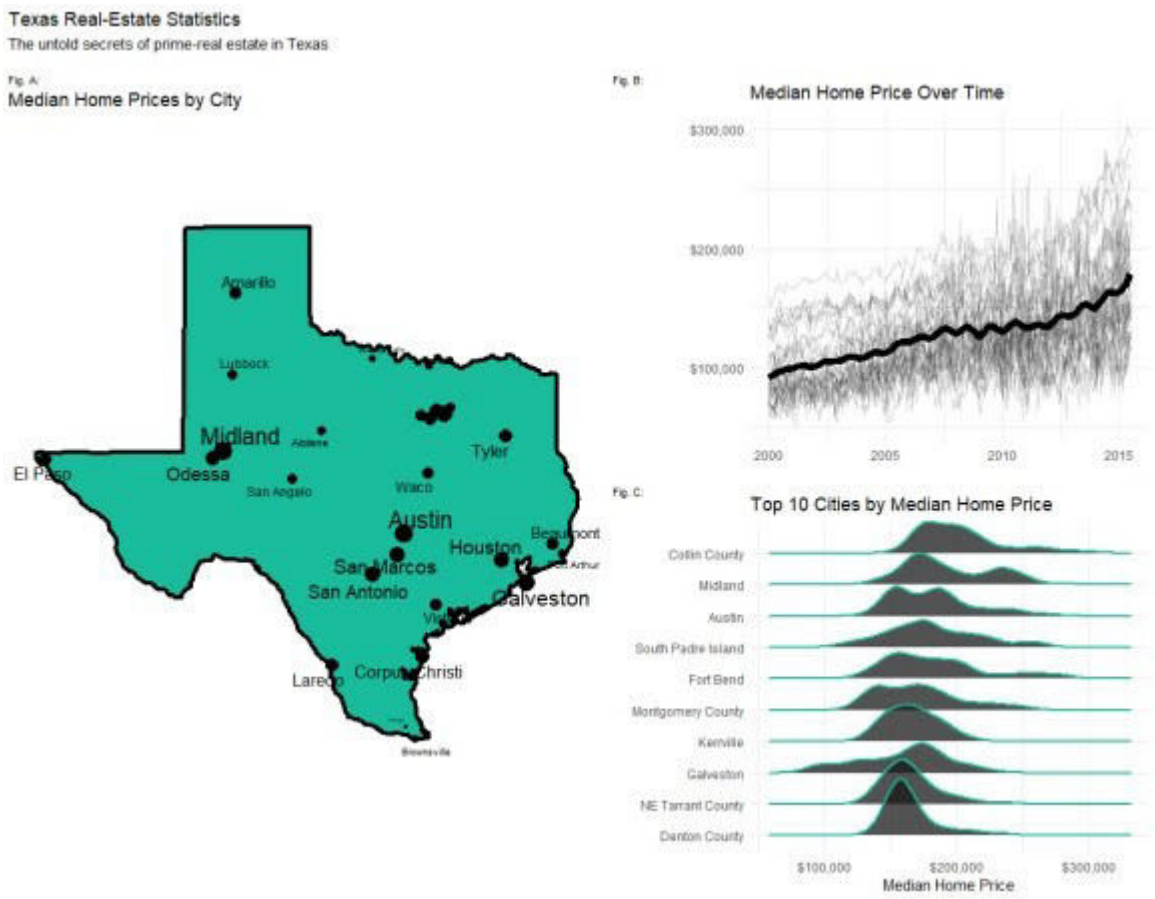


Step 3: Compose with patchwork

The YouTube tutorial does this code justice. Check it out.

```
gg_tx_map + (gg_tx_timeseries / gg_tx_ridge) +
    plot_layout(widths = c(3,2), tag_level = "new") +
    plot_annotation(
        title      = "Texas Real-Estate Statistics",
        subtitle   = "The untold secrets of prime-real estate in th
        tag_levels = "A",
        tag_prefix = "Fig. ",
        tag_suffix = ":"
    ) &
    theme(plot.tag.position = c(0, 1),
          plot.tag = element_text(size = 8, hjust = 0, vjust = 0))

```

Get the Code



In Summary

You just quickly made a professional storyboard using the ggplot2 and patchwork. Fantastic!

You should be proud.