

Visualization Using Zillow Economics Data - Kaggle

In keeping with my recent focus of new visualization techniques, today we're going to use the Zillow Economics Data to make a state grid map. The final result should look like this:

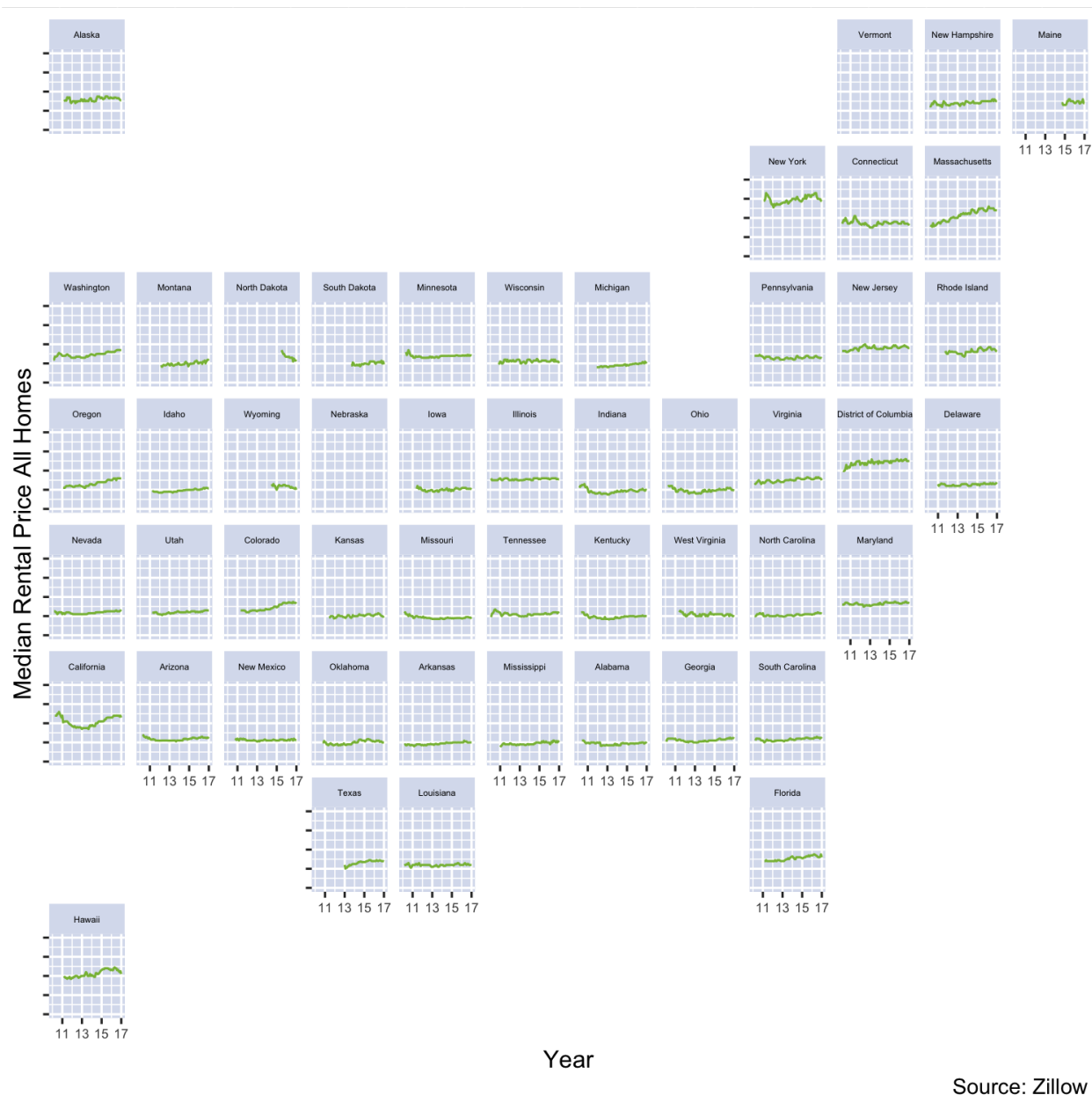


Figure 1:

Let's start as always by importing some packages we'll need along the way:

```
require( dplyr )
require( lubridate )
require( geofacet )
require( ggplot2 )
```

```
require( scales )
```

We import the Zillow Economics dataset that we'll be working with:

```
zillow.data <- read.csv( 'State_time_series.csv' , header = T , stringsAsFactors = F )
```

Let's also tip our hats to the group that donated this fine data for us to work with, by defining a custom color scheme using the Zillow color palette from the Zillow Media Room.

```
# Define specific color scheme  
#-----  
zillow.blue = rgb( 85/255, 119/255, 187/255 , alpha = 0.25 )  
zillow.green = rgb( 136/255, 187/255, 68/255 , alpha = 1 )
```

Filter & Clean The Data

For the purpose of this visualization we only want to work with a small portion of the data, so we do the following: we grab the Date, Region Name, and Median Rental Price across All Home columns. We also only want data at the individual state level, and so we eliminate rows with the 'UnitedStates' region.

```
working.data <- zillow.data[ which( zillow.data$RegionName != 'UnitedStates' )  
                             , c( 'Date'  
                                   , 'RegionName'  
                                   , 'MedianRentalPrice_AllHomes' ) ]
```

```
working.data$Date <- working.data$Date %>% as.Date
```

Next remove any missing values:

```
working.data <- na.omit( working.data )
```

This next part is a little tedious, but the state grid map package seeks to match its stored state names onto the state names of our dataset – however, in this dataset any state name with two or more words has the spaces removed. This little bit of code just cycles through and replaces each of the shortened state names in the Zillow data with its separated true name:

```
working.data$RegionName <- sapply( working.data$RegionName , FUN = function(state) {  
  if( state == 'NewYork' ) { 'New York' } else  
    if( state == 'SouthDakota' ) { 'South Dakota' } else  
      if( state == 'NorthDakota' ) { 'North Dakota' } else  
        if( state == 'NewHampshire' ) { 'New Hampshire' } else  
          if( state == 'NewJersey' ) { 'New Jersey' } else  
            if( state == 'NewMexico' ) { 'New Mexico' } else  
              if( state == 'NorthCarolina' ) { 'North Carolina' } else  
                if( state == 'RhodeIsland' ) { 'Rhode Island' } else  
                  if( state == 'SouthCarolina' ) { 'South Carolina' } else  
                    if( state == 'WestVirginia' ) { 'West Virginia' } else  
                      if( state == 'DistrictofColumbia' ) { 'District of Columbia' } else  
                        { state }  
})
```

And that's it for filtering and cleaning! On to the mapping . . .

Making The State Grid Map

The geofacet package makes this very easy, as shown below. Everything else is essentially just formatting:

```
ggplot( working.data , aes( Date , MedianRentalPrice_AllHomes ) ) +
  theme( axis.text=element_text(size = 6)
        , strip.text.x = element_text(size = 4)
        , strip.text = element_text(color = 'black' )
        , strip.background = element_rect( fill = zillow.blue )
        , panel.background = element_rect( fill = zillow.blue ,
                                           colour = zillow.blue ,
                                           size = 0.1, linetype = "solid")

  ) +
  geom_line( color = zillow.green ) +
  scale_color_discrete( guide = FALSE ) +
  facet_geo( ~ RegionName, grid = "us_state_grid2", label = "name" ) +
  scale_y_continuous(
    labels = NULL
    , limits = c(0,4000)
  ) +
  labs( caption = "Source: Zillow",
        x = "Year",
        y = "Median Rental Price All Homes" ) +
  scale_x_date(date_breaks = "2 years",
              labels=date_format("%y"),
              limits = as.Date(c('2010-01-01','2017-01-01'))
  )
```

And that's it! I like this state grid map visualization because you're allowing the user to consumer multiple dimensions simultaneously (here we are seeing time, dollar value, and location simulataneously), and yet in a format that doesn't overload their senses. Also, it's unique – never underestimate the novelty of showing the same data in a new way.

Thanks very much for checking this out – please upvote if you liked it!

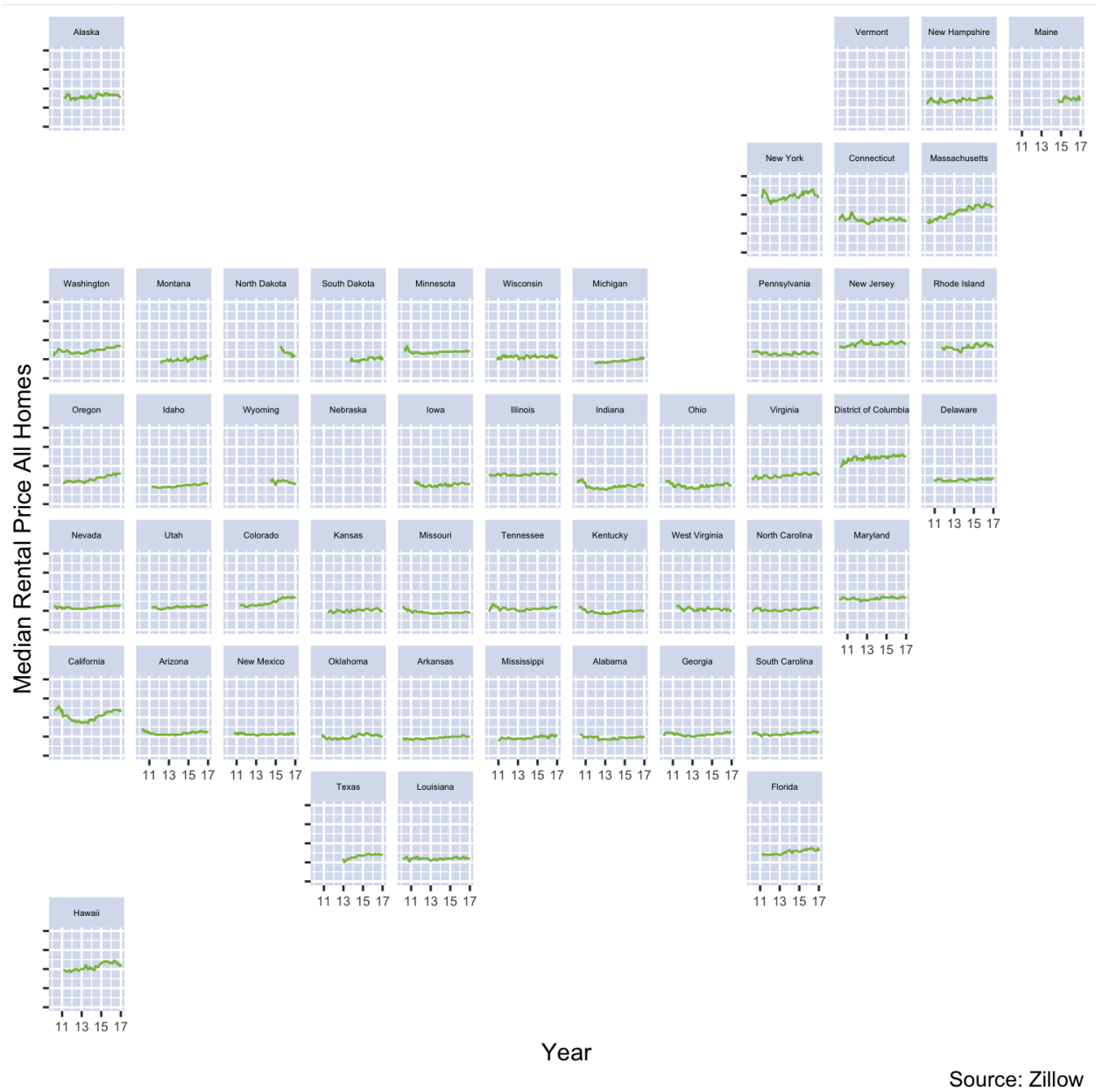


Figure 2: