Housing Data Heatmap

Introduction

There are many interesting ways to look at seasonality and changes in trend over time. One often-overlooked way is to use a heatmap. For this example I'll use macroeconomic housing data, since it has both a predictable seasonal trend, as well as a lower-frequency business cycle trend.

As usual, I'll start by importing a few packages that we'll need along the way:

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

Data Manipulation

Next let's import the data we'll work with, which I've pulled from the Federal Reserve. Specifically, I'll use data for New One Family Houses Sold:

```
input_data <- read.csv( "HSN1FNSA.csv" , header = TRUE )

head( input_data )
```

```
##     DATE HSN1FNSA
## 1 1/1/63       42
## 2 2/1/63       35
## 3 3/1/63       44
## 4 4/1/63       52
## 5 5/1/63       58
## 6 6/1/63       48
```

Note that one way we could automate this would be to connect to the Federal Reserve API, which is rather easy to do and would keep us from having to manually download the CSV file from the link above.

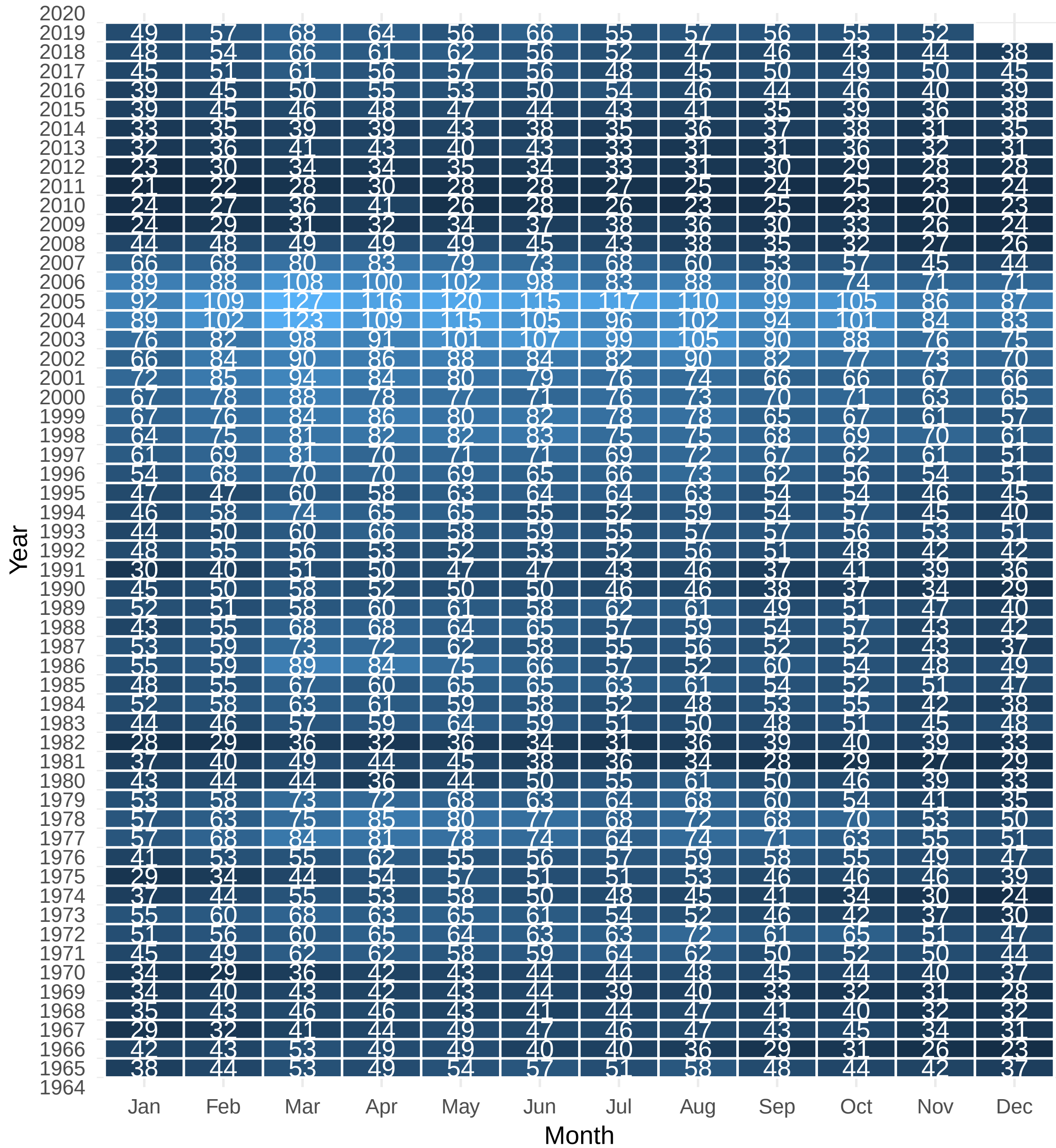
The goal of this project is primarily visualization, and so we'll do very little data manipulation here. The below lines start by converting the Excel dates into R date objects, then extract the Year and Month.

```
Manipulated_Data =
  input_data %>%
  mutate( d = format(as.Date( DATE, "%m/%d/%y"),
                     "19%y-%m-%d") ) %>%
  mutate(
    Year = year(d),
    Month = month( d , label = TRUE , abbr = TRUE ) ,
    # fix an odd feature of the Years
    Year = if_else(
      Year <= 1930 , Year +100 , Year)
  ) %>%
  setNames(c(
    'Original_Date','value','Format_Date',
```

```
    'Year','Month'
  ))
```

Let's see what the basic version of this plot would give us:

```
ggplot(Manipulated_Data, aes(x=Month,
                             y=Year,
                             fill=value,
                             label=value  )) +
  geom_tile( color = 'white', size = 0.5 ) +
  scale_y_continuous( limits = c(1964,2020),
                      expand=c(0,0),
                      breaks = 1964:2020) +
  geom_text( color = 'white' ) +
  theme_minimal() +
  theme( legend.position = 'topright',
         legend.direction = 'horizontal',
         plot.caption = element_text(hjust=0),
         panel.grid.major.y = element_blank(),
         legend.key.width = unit(1.5,"cm")
  )
```



Sure, this is useful, and we can see some variation here, but it's not the best visual representation of the data. So we'll try to make this more impactful by doing nothing more than changing the color pallette.

To do this we'll need to define some custom functions:

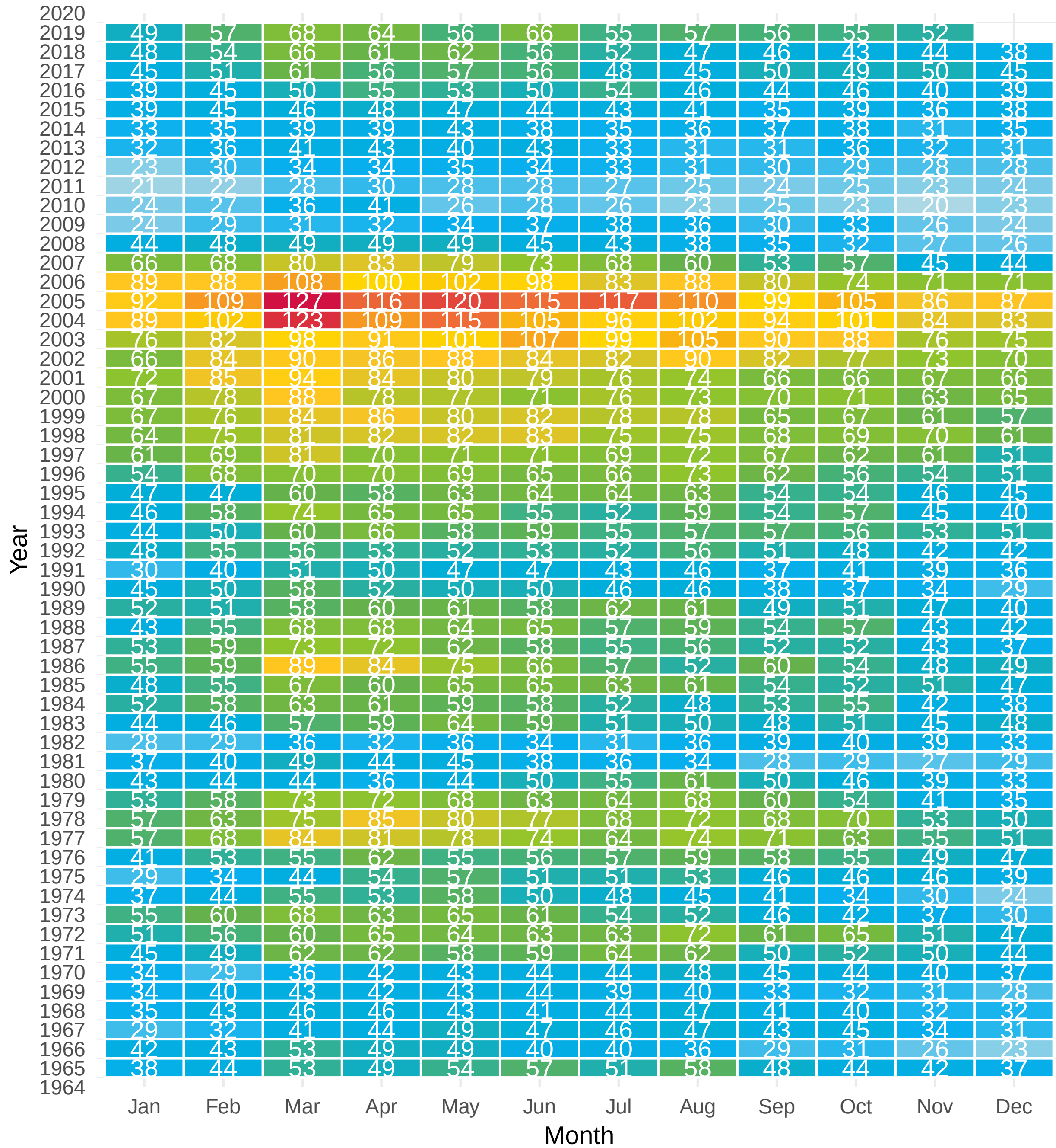
```
# this function is simply a list of the colors we would like to use
# and has an argument that would could use to make it reversible if we wanted
MY_PAL <- function( reverse = FALSE) {
  pal <- c(
    "lightblue2" = rgb(173,216,230, maxColorValue = 256),
    "lightblue"  =  rgb(9, 177,240, maxColorValue = 256),
    'blue'       = "#00aedb",
```

```
    "green"      = rgb(103,180,75, maxColorValue = 256),
    "green2"      = rgb(147,198,44, maxColorValue = 256),
    'yellow'     = "#ffc425",
    'gold'       = "#FFD700",
    'orange'     = "#f37735",
    'red'        = "#d11141"      )
  if (reverse) pal <- rev(pal)
  colorRampPalette(pal)
}

SCALE_FILL_MYCOL <- function(
  reverse = FALSE,
  ...
) {
  pal <- MY_PAL(reverse = reverse)
  scale_fill_gradientn(colours = pal(256), ...)
}
```

Now let's try that same plot again but using our custom functions:

```
ggplot(Manipulated_Data, aes(x=Month,
                             y=Year,
                             fill=value,
                             label=value  )) +
  SCALE_FILL_MYCOL(reverse=FALSE,
                   name="New Home Sales (1000s,not seasonally adjusted)")+
  geom_tile( color = 'white', size = 0.5 ) +
  scale_y_continuous( limits = c(1964,2020),
                      expand=c(0,0),
                      breaks = 1964:2020) +
  geom_text( color = 'white' ) +
  theme_minimal() +
  theme( legend.position = 'topright',
         legend.direction = 'horizontal',
         plot.caption = element_text(hjust=0),
         panel.grid.major.y = element_blank(),
         legend.key.width = unit(1.5,"cm")
  )
```



Now this figure is *much* more useful! We can see the "hot" periods in the housing market (reds and oranges) as well as the "cold" periods (blues and greens).

This also allows us to return to our original motivation of seasonality vs. business cycles.

If we look vertically - that is, across years - we can also visualize the ups and downs of the macroeconomic business cycle. As we go up the chart we can see, for example, the housing boom in red in 2005, followed by the drop into blue during the 2008 Financial Crisis, followed by a return to greens in recent years during the recovery.

Similarly, if we look horizontally - across months - we can see a repeating pattern, regardless of year, that housing sales tend to peak in the Springs and Summers, but then slows down in the Falls and Winters.

Conclusion

The key takeaway here is that simply by choosing the right visualization technique, we were able to identify key patterns in this dataset, without fitting any models or really doing any math at all.