

ISA 444: Business Forecasting

03: Time Series Plots

Fadel M. Megahed, PhD

Endres Associate Professor
Farmer School of Business
Miami University

 @FadelMegahed

 fmegahed

 fmegahed@miamioh.edu

 Automated Scheduler for Office Hours

Spring 2023

Quick Refresher from Last Class

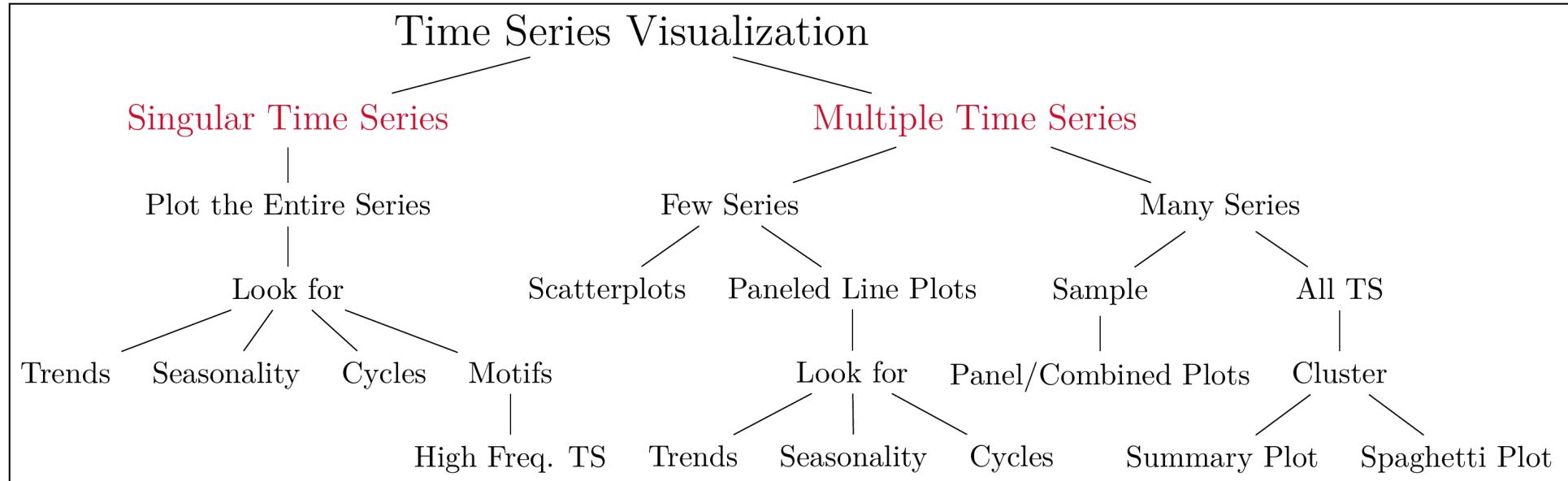
- Describe the syntax, data types, and data structures in `R`.
- Access the help for `R` functions (each help file has the following components: Description, usage, arguments, value, and examples).
- Utilize the project workflow in `R` and create `R` script.
- Access, subset, and create `ts()` objects in `R`.

Learning Objectives for Today's Class

- Examine a line chart for trends, seasonality, and cycles.
- Explain the grammar of graphics and how it can be used to create time series plots in .
- Create interactive time-series plots by using the  package .

A Taxonomy of Time Series Plots and their Interpretation

A Structured Approach for Time Series Viz

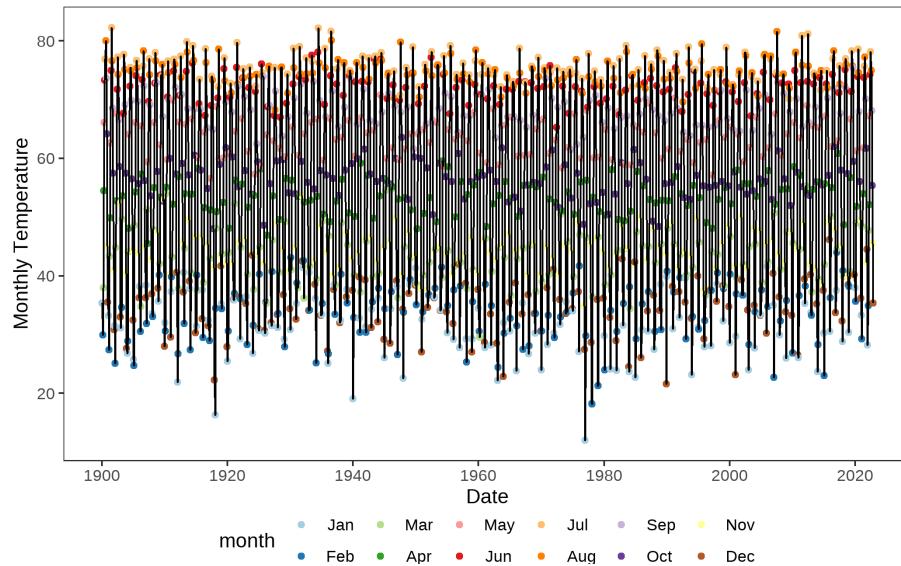


A Potential Framework for Time Series Visualization

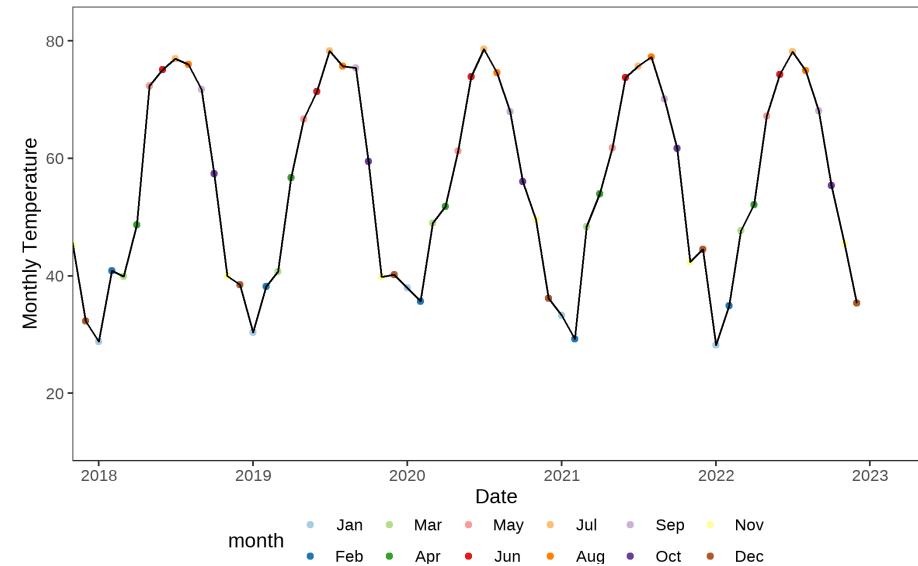
This is my best attempt to improve on the general advice provided in the previous slide. Many of the suggestions, presented in this flow chart, stem from my past and current research/consulting collaborations.

They are by no means a comprehensive list of everything that you can do.

The Line Chart



A plot of a long time-series for monthly weather in Cincinnati, with color denoting different months.



A snippet of the time-series (last 5 years) for monthly weather in Cincinnati, with color denoting different months.

The Line Chart: Practical Considerations

Things to Consider: (Insert below)

- Format your data:
- Entire time-series vs a snippet:
- On the Use of Color:
- On grouping the data:

05 : 00

On the Interpretation of Line Charts

Activity

Book Stores

GDP 1

GDP 2

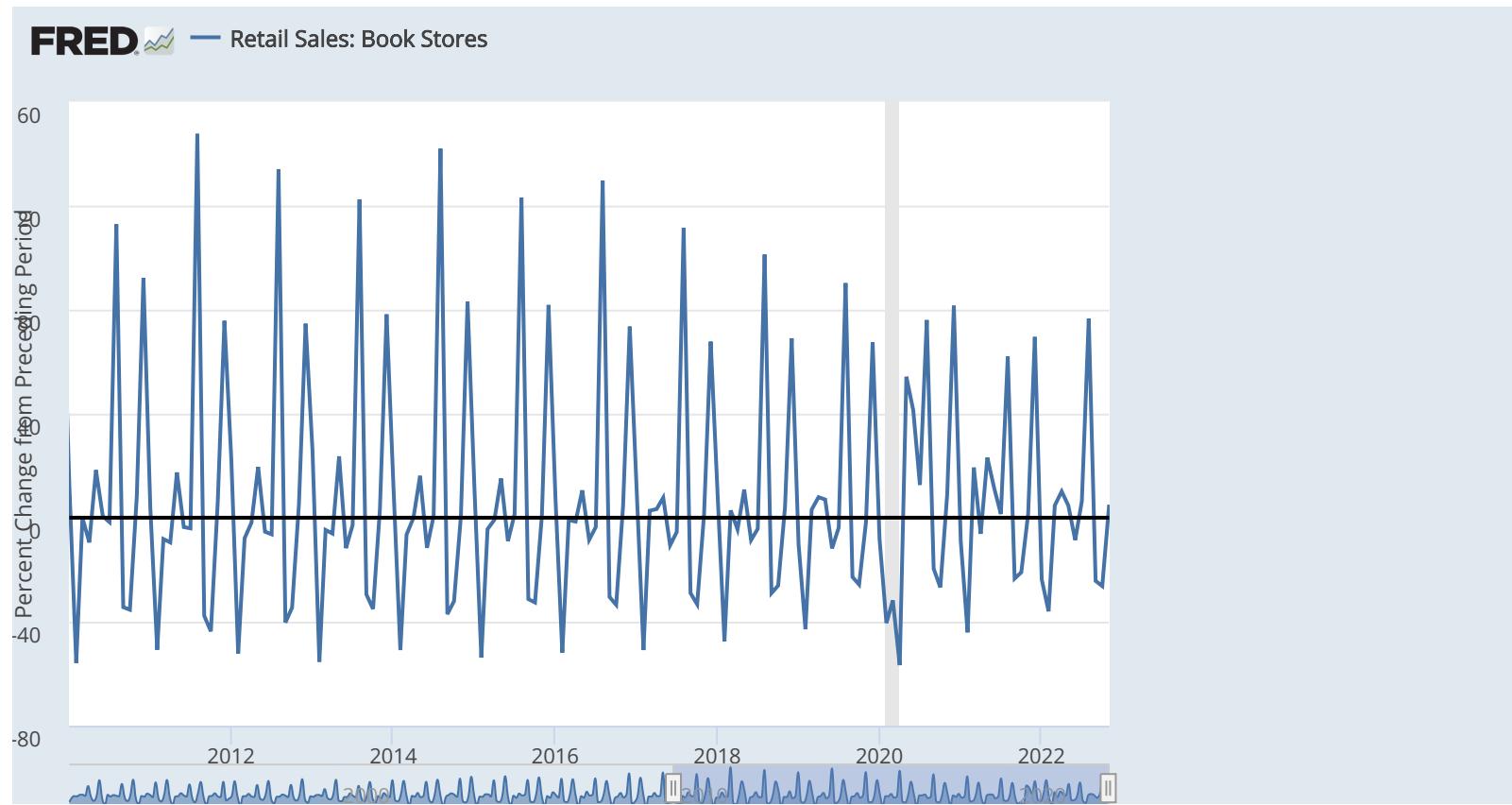
Key Points

Over the next 5 minutes, please identify what you have learned from the charts in each tab.

- Write down your answers in the last tab (it is editable).
- Discuss your answers with your neighboring classmates.
- Be prepared to share these answers with class.

On the Interpretation of Line Charts

Activity	Book Stores	GDP 1	GDP 2	Key Points
----------	-------------	-------	-------	------------



05:00

On the Interpretation of Line Charts

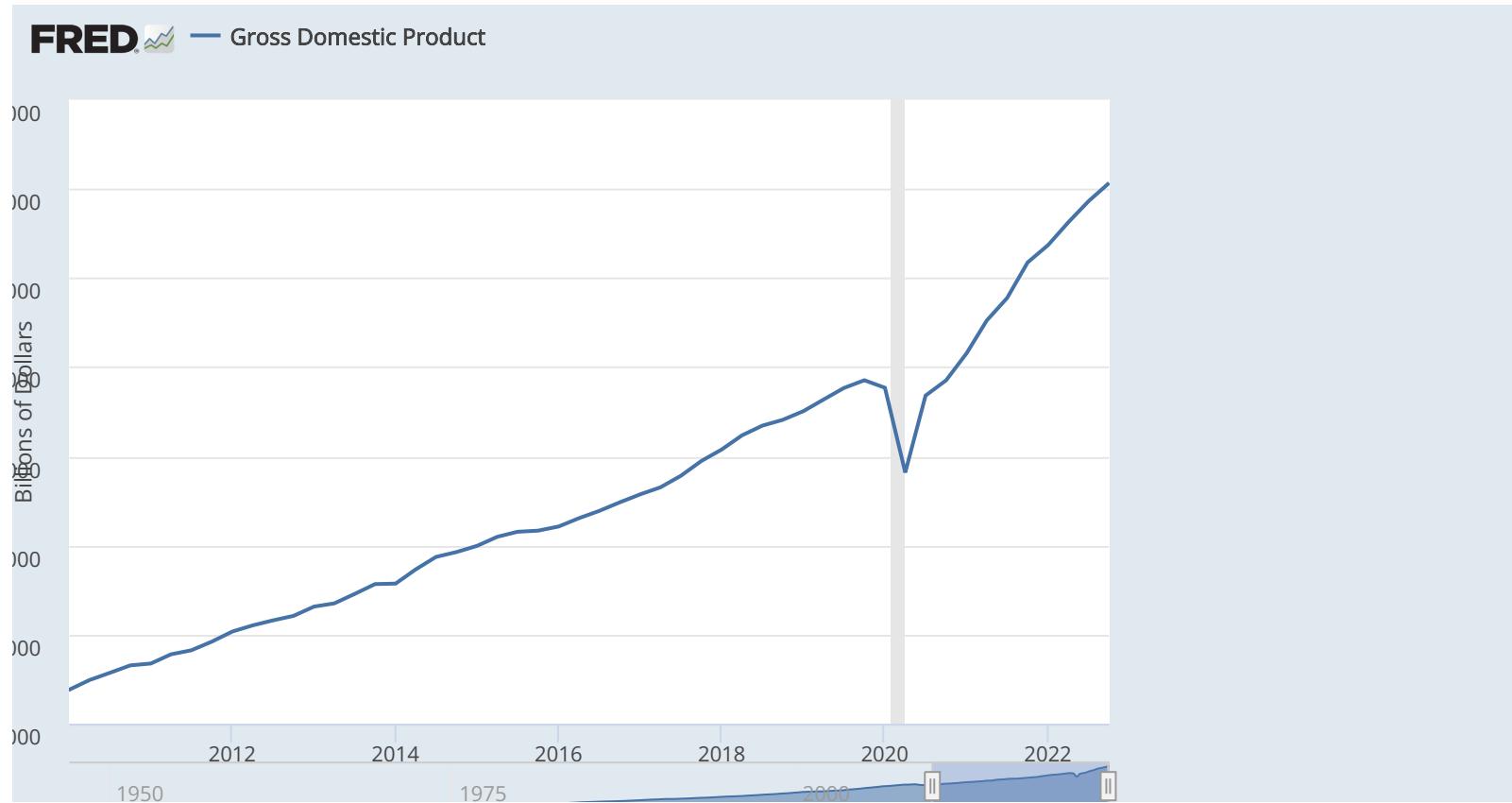
Activity

Book Stores

GDP 1

GDP 2

Key Points



05:00

On the Interpretation of Line Charts

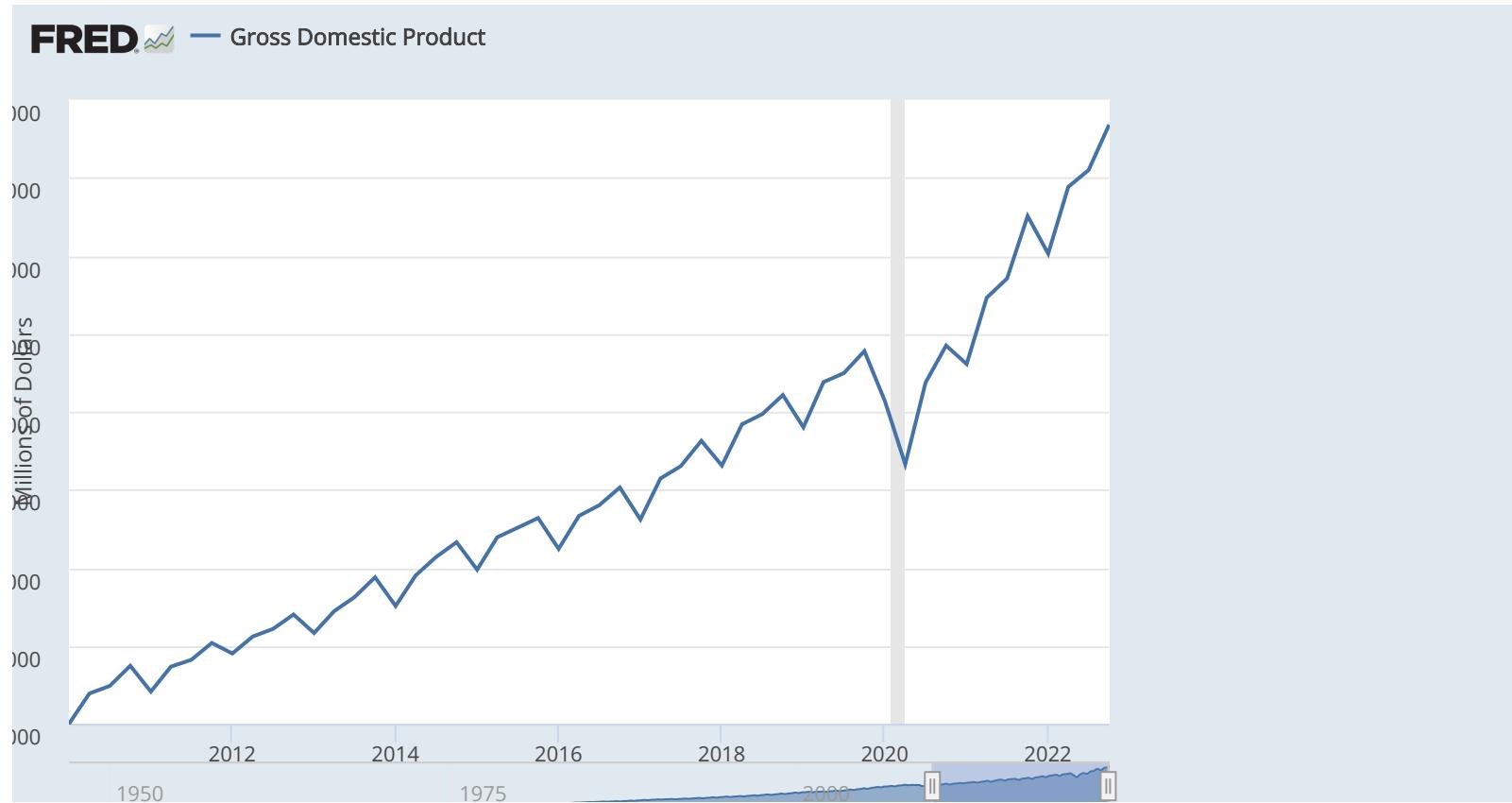
Activity

Book Stores

GDP 1

GDP 2

Key Points



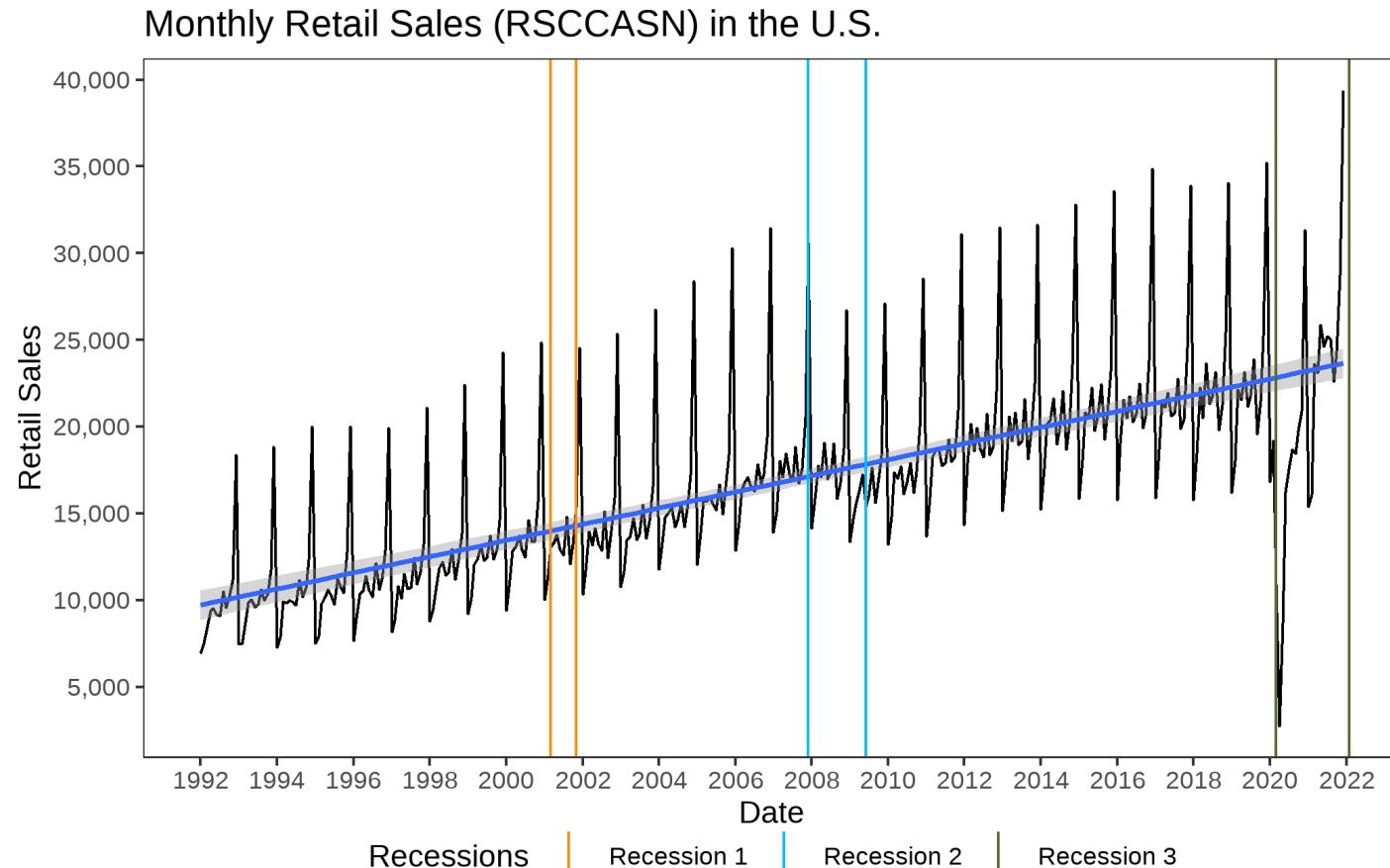
On the Interpretation of Line Charts

Activity	Book Stores	GDP 1	GDP 2	Key Points
----------	-------------	-------	-------	------------

Main Insight(s): (Insert below)

- **Book Stores:** Trend: ... | Seasonality: ... | Cycle: ...
- **GDP 1:** Trend: ... | Seasonality: ... | Cycle: ...
- **GDP 2:** Trend: ... | Seasonality: ... | Cycle: ...

Need Assistance with Trends



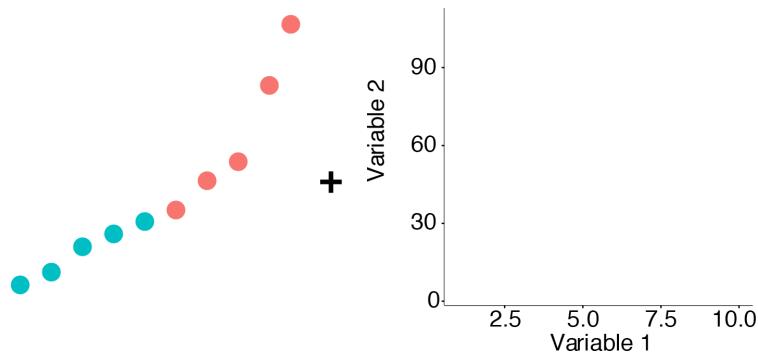
Need Assistance with Seasonality

In [Section 2.2.1 of our reference book](#), the authors presented two approaches for considering seasonality. We can replicate them easily in R. Refer to the discussion in the next section for more detail.

The Grammar of Graphics and the ggplot2 package

A Visual Introduction to Graph Layers

Aesthetics



Layer 1

Axis

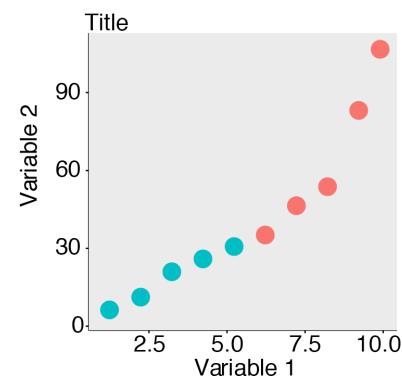
Theme

Output



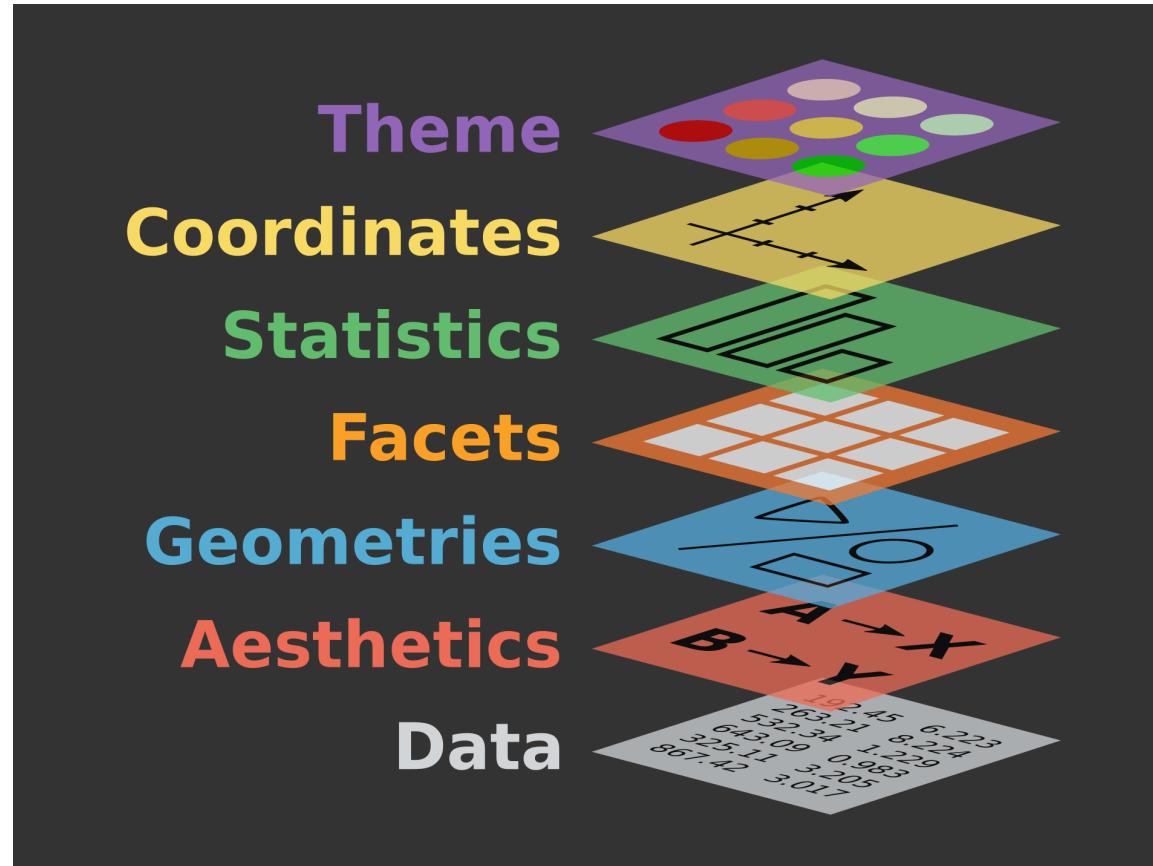
Layer 2

Layer 3



Schematic of some distinct layers in the Grammar of Graphics.

The Grammar of Graphics



An overview of the layers introduced in the Grammar of Graphics.

Grammar of Graphics Layers: Data

- Data needs to be in a **tidy** format (see next two slides).
- The `dplyr` and `tidyr`  can help with **tidying** your data.

“**TIDY DATA** is a standard way of mapping the meaning of a dataset to its structure.”

-HADLEY WICKHAM

In tidy data:

- each variable forms a column
- each observation forms a row
- each cell is a single measurement

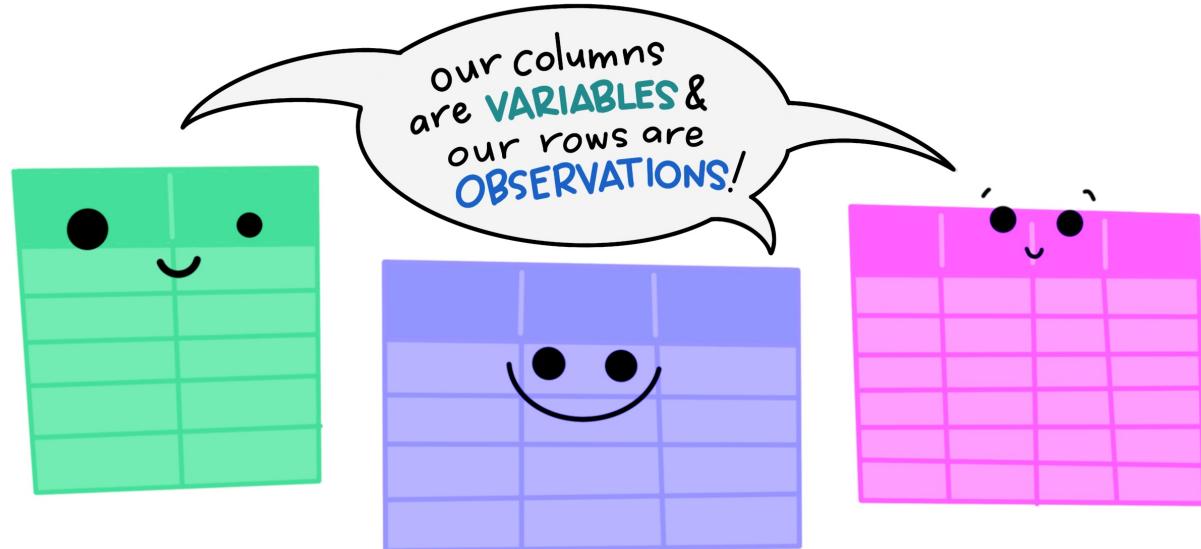
each column a variable

each row an observation

id	name	color
1	floof	gray
2	max	black
3	cat	orange
4	donut	gray
5	merlin	black
6	panda	calico

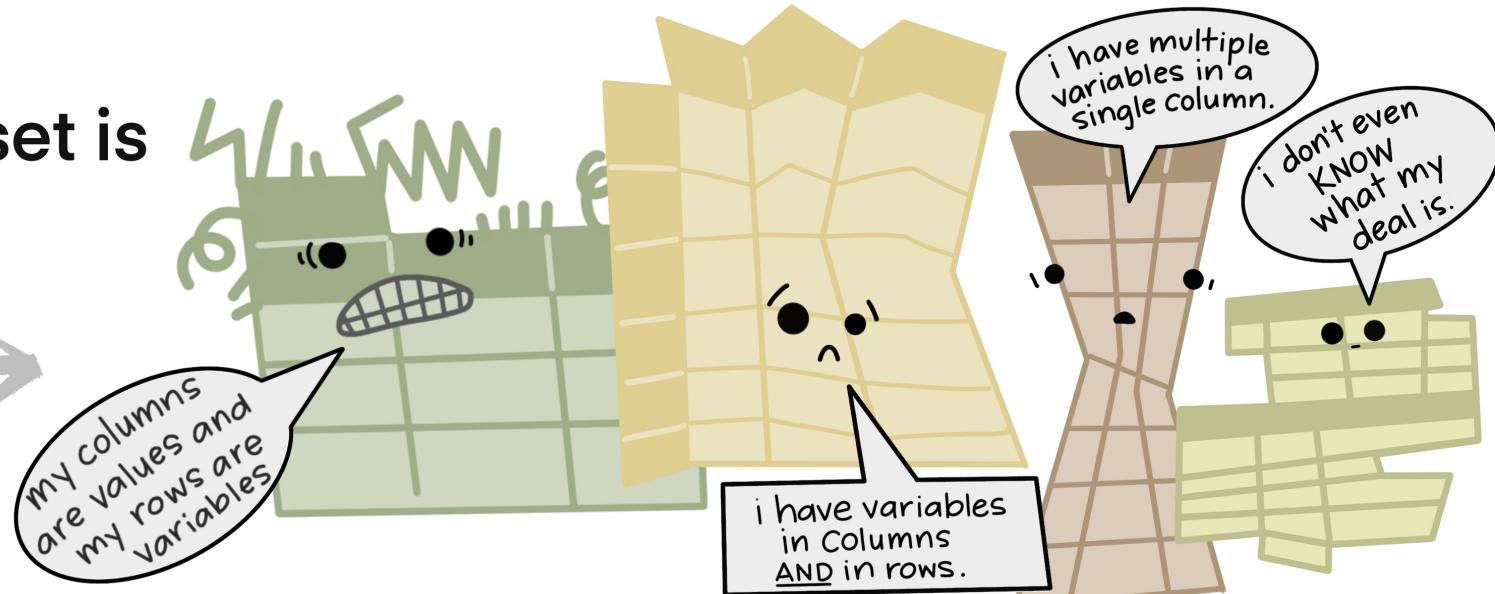
Wickham, H. (2014). Tidy Data. Journal of Statistical Software 59 (10). DOI: 10.18637/jss.v059.i10

The standard structure of
tidy data means that
“tidy datasets are all alike...”



“...but every messy dataset is
messy in its own way.”

-HADLEY WICKHAM



Grammar of Graphics Layers: Aesthetics

Aesthetics (`ggplot2::aes()`) are used to make data visible. For example:

- `x`, `y`: variable to be plotted along the x and y axes.
- `color`: color of geoms (i.e., points, lines, etc) according to the data.
- `fill`: the inside color of the geom (useful for bar charts).
- `group`: what group a geom belongs to (useful in multiple ts).
- `shape`: the shape of the plotted point (circle, triangle, filled circle, etc).
- `linetype`: the type of line used (solid, dashed, etc).
- `size`: size scaling for an extra dimension.
- `alpha`: the transparency of the geom

08:00

Identify the Aesthetics Used in the Charts

Activity

Line Chart 1

Line Chart 2

Seasonal Chart 1

Seasonal Chart 2

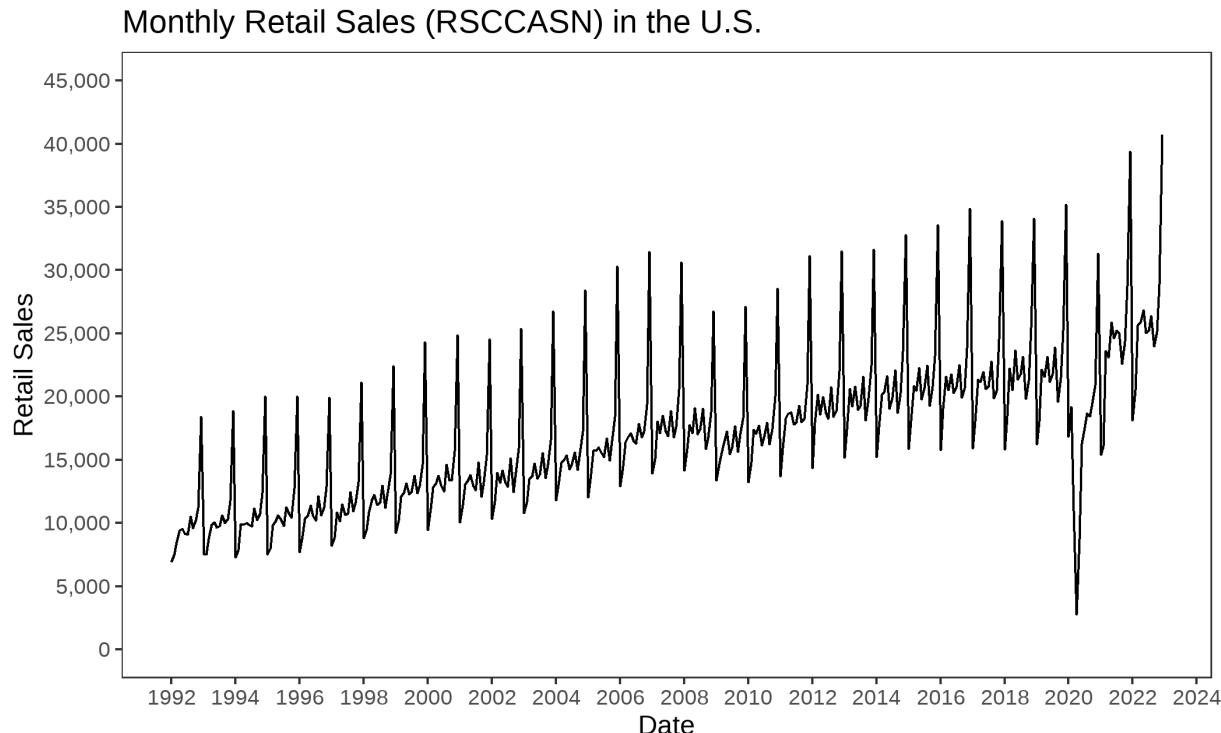
Over the next 8 minutes, please identify the aesthetics used in each chart.

- Write down your answers in the right-side of each tab (it is editable).
- You can discuss your answers with your neighboring classmates.
- Be prepared to share these answers with class.

08:00

Identify the Aesthetics Used in the Charts

Activity	Line Chart 1	Line Chart 2	Seasonal Chart 1	Seasonal Chart 2
----------	--------------	--------------	------------------	------------------



Main Aesthetics: (Insert below)

- **x**: and its class is:
- **y**: and its class is:
- **group**:
- **color**:

08:00

Identify the Aesthetics Used in the Charts

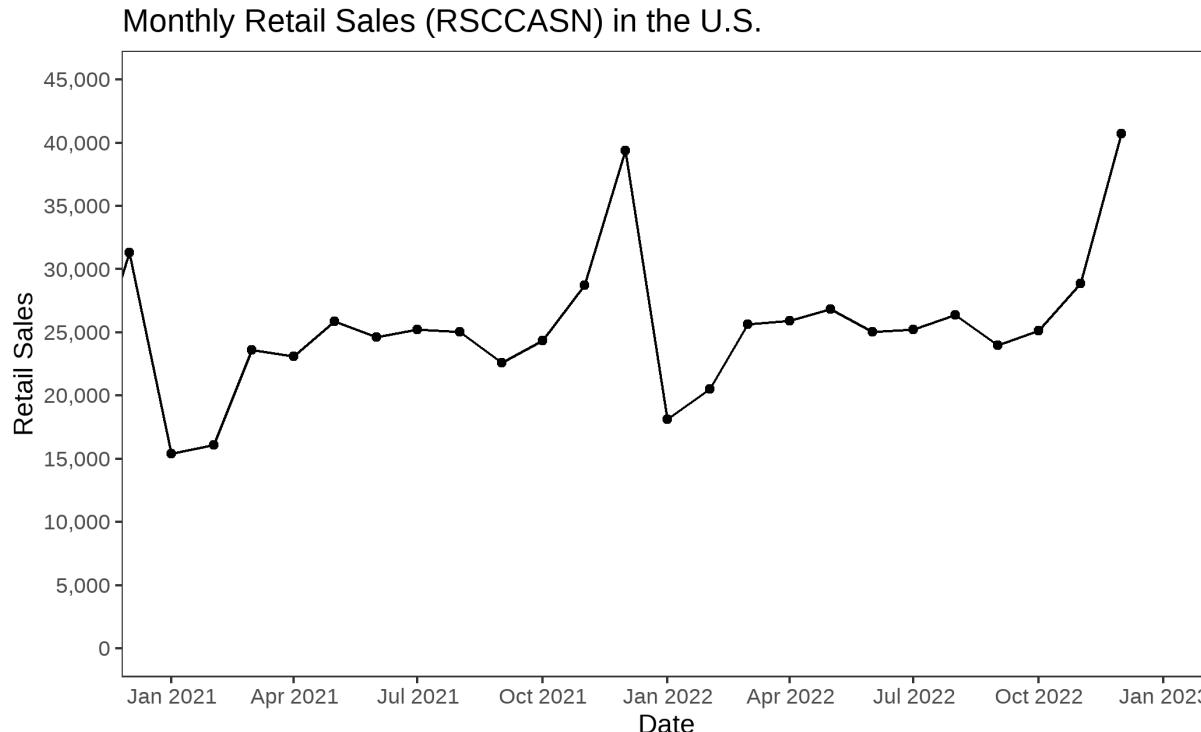
Activity

Line Chart 1

Line Chart 2

Seasonal Chart 1

Seasonal Chart 2



Main Aesthetics: (Insert below)

- **x**: and its class is:
- **y**: and its class is:
- **group**:
- **color**:

08:00

Identify the Aesthetics Used in the Charts

Activity

Line Chart 1

Line Chart 2

Seasonal Chart 1

Seasonal Chart 2



Main Aesthetics: (Insert below)

- **x**: and its class is:
- **y**: and its class is:
- **group**:
- **color**:

Identify the Aesthetics Used in the Charts

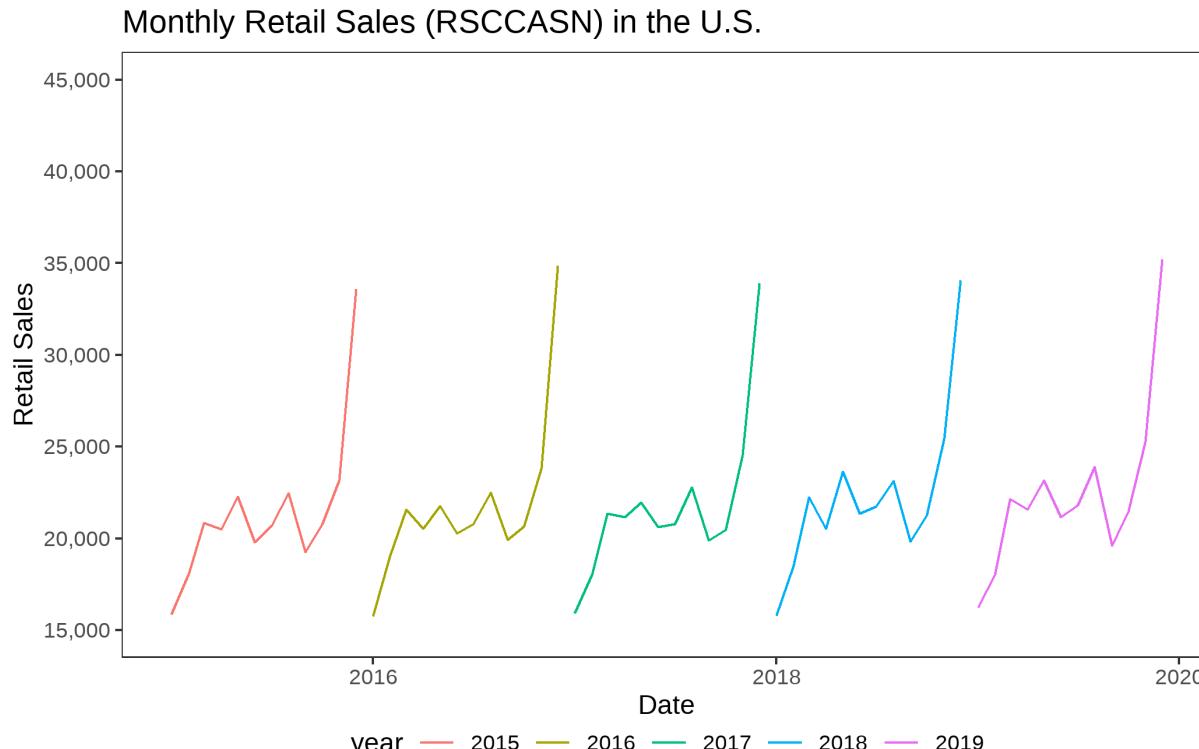
Activity

Line Chart 1

Line Chart 2

Seasonal Chart 1

Seasonal Chart 2



Main Aesthetics: (Insert below)

- **x**: and its class is:
- **y**: and its class is:
- **group**:
- **color**:

Grammar of Graphics Layers: Aesthetics

- Assigned **globally** to the entire plot via `ggplot2::ggplot(ggplot2::aes())`, or to **specific geoms** (e.g., `ggplot2::geom_point(ggplot2::aes())`).



The color is passed globally through `ggplot(aes(x=date, y=price, color=year))`.



The color is passed as an argument within the layer as `geom_point(aes(color = year))`.

Grammar of Graphics Layers: Individual Geoms

Geometric objects i.e., geoms help determine the type of plot. In this class, we will typically use one or more of the following *geoms*:

- `ggplot2::geom_point()`: scatterplot or points in a line graph.
- `ggplot2::geom_line()`: lines connecting points by increasing value of x.
- `ggplot2::geom_smooth()`: to fit a function line (e.g., linear regression line) based on data.

Grammar of Graphics Layers: Facets

We can use `ggplot2::facet_wrap()` to create small multiples based on a single variable. Arguments for `ggplot2::facet_wrap()` include:

- `facets` which takes the variable of interest in quotes (i.e., `facets = 'symbol'`);
- `nrow` and/or `ncol` which take numeric inputs for the number of rows and columns; and
- `scales`, where we typically use `free_y` to denote that different `ylim` for each panel.

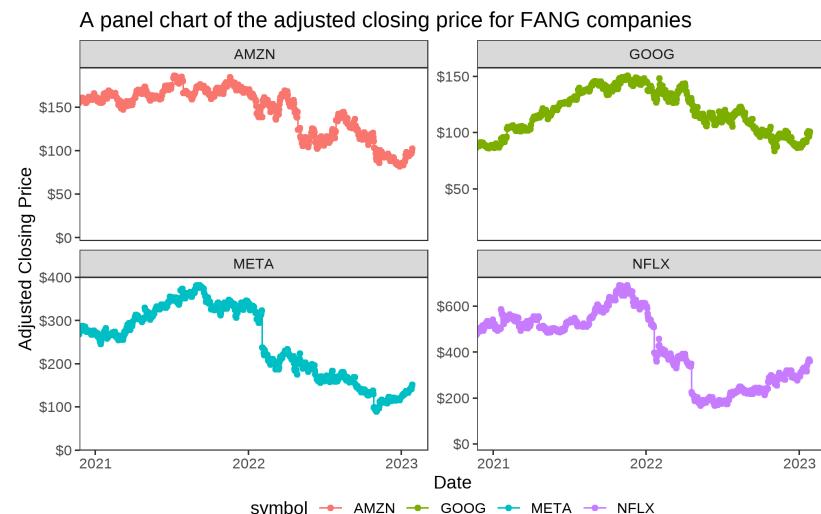
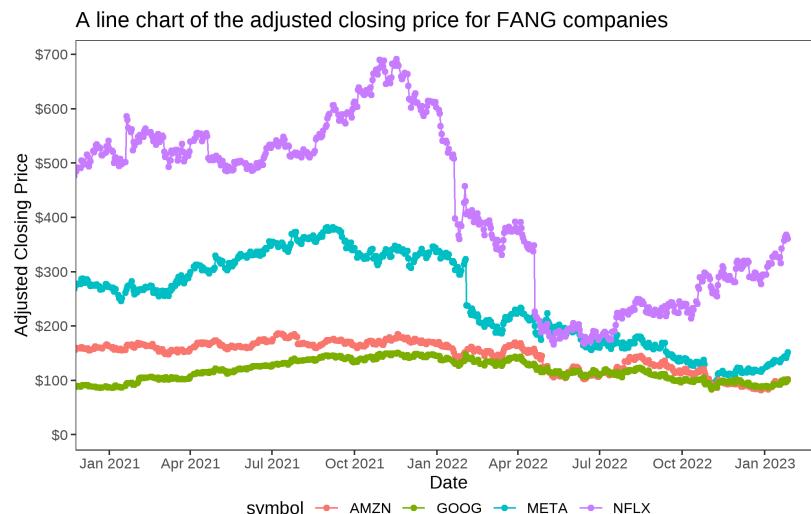
```
fang_df = tidyquant::tq_get(  
  x = c('META', 'AMZN', 'NFLX', 'GOOG'),  
  from = '2010-01-01', to = Sys.Date()  
)  
  
colnames(fang_df)
```

```
## [1] "symbol"    "date"      "open"       "high"       "low"        "close"      "volume"  
## [8] "adjusted"
```

Grammar of Graphics Layers: Facets

We can use `ggplot2::facet_wrap()` to create small multiples based on a single variable. Arguments for `ggplot2::facet_wrap()` include:

- `facets` which takes the variable of interest in quotes (i.e., `facets = 'symbol'`);
- `nrow` and/or `ncol` which take numeric inputs for the number of rows and columns; and
- `scales`, where we typically use `free_y` to denote that different `ylim` for each panel.



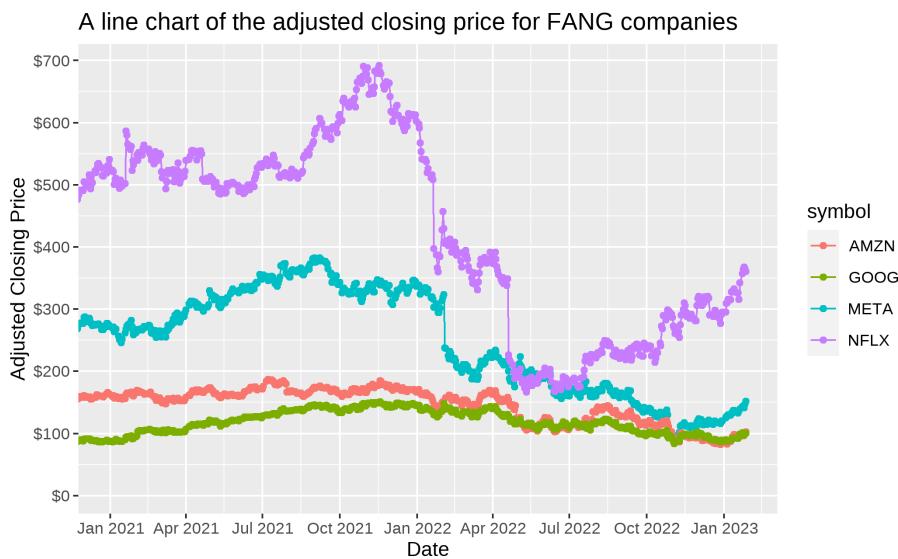
Grammar of Graphics Layers: Coordinates

In class, we will use the following two functions to create **snapshots** of the data:

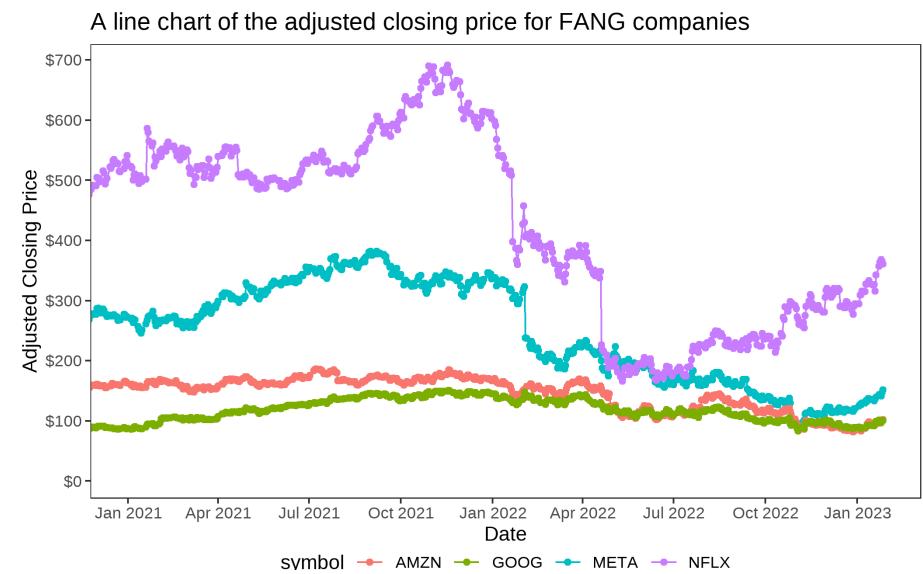
- `ggplot2::coord_cartesian()` to set limits. We will specifically use its `xlim` argument to create a snapshot when the x axis contains a continuous variable (e.g., year). See [ggplot2 documentation](#) for more detail.
- `tidyquant::coord_x_date()` to set limits. We will specifically use its `xlim` argument to create a snapshot when the x axis contains a date variable (with a `class` of date). See [tidyquant documentation](#) for more detail.

Grammar of Graphics: Themes

Themes control the overall visual defaults. There are some themes built within the `ggplot2` 📁 (see the [complete themes guide](#)). For additional themes, please feel free to play with the `ggthemes` 📁.



Default ggplot2 theme



A modified theme (no gridlines, no gray background and caption is place below).

Putting it all together

A Singular TS: AAPL's Adj. Close

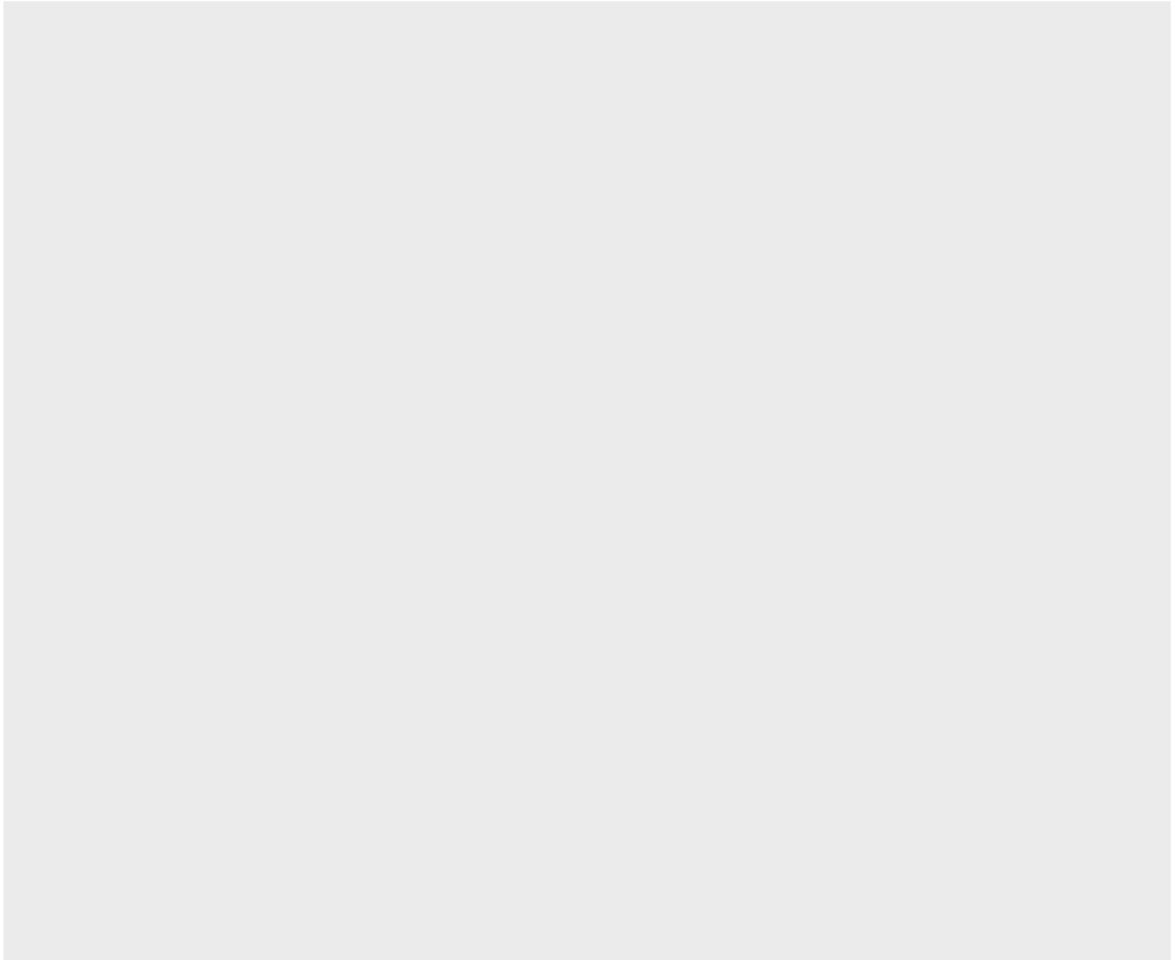
```
if(require(tidyquant)==F) install.packages("tidyquant") # install if needed
if(require(tidyverse)==F) install.packages('tidyverse') # install if needed

aapl = # get AAPL stock data from 1st trading day after Jan 1, 2020 to now
  tidyquant::tq_get(x = 'AAPL', from = '2020-01-01', to = Sys.Date() ) |>
    # select (i.e., keep) only the variables below
    dplyr::select( c(date, symbol, adjusted) ) |>
    # create the following variables: year and month
    dplyr::mutate(
      # date has to be of class Date if not use lubridate::ymd (mdy, dmy, etc)
      # to convert the string variable to date
      year = lubridate::year(date),
      month = lubridate::month(date, label = T)
    )
tail(aapl, n = 1) # print the last obs to see what we have
```

```
## # A tibble: 1 × 5
##   date     symbol adjusted  year month
##   <date>    <chr>     <dbl> <dbl> <ord>
## 1 2023-01-27 AAPL       146.  2023 Jan
```

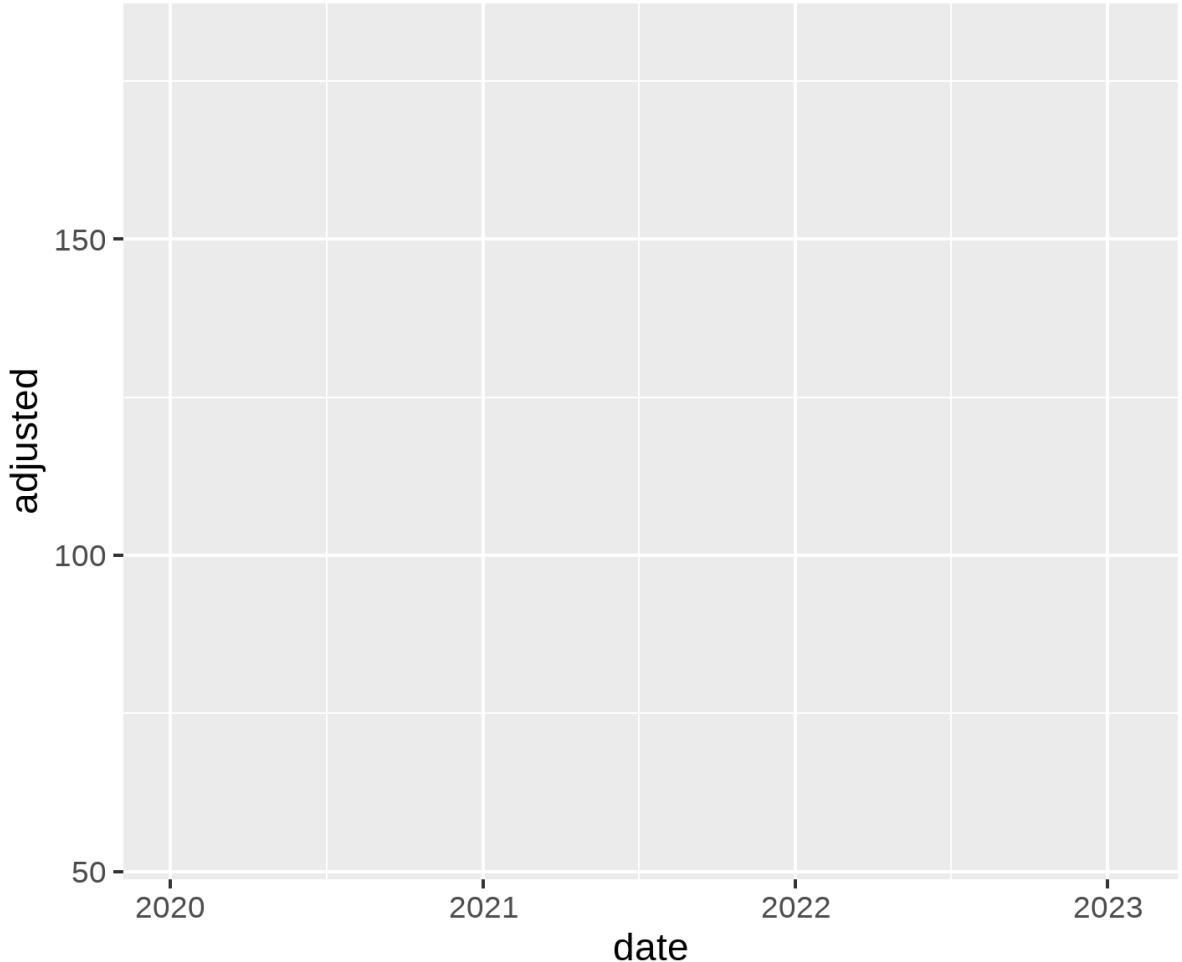
A Singular TS: The GG Layers

```
# layers are + in ggplot2  
ggplot2::ggplot(aapl)
```



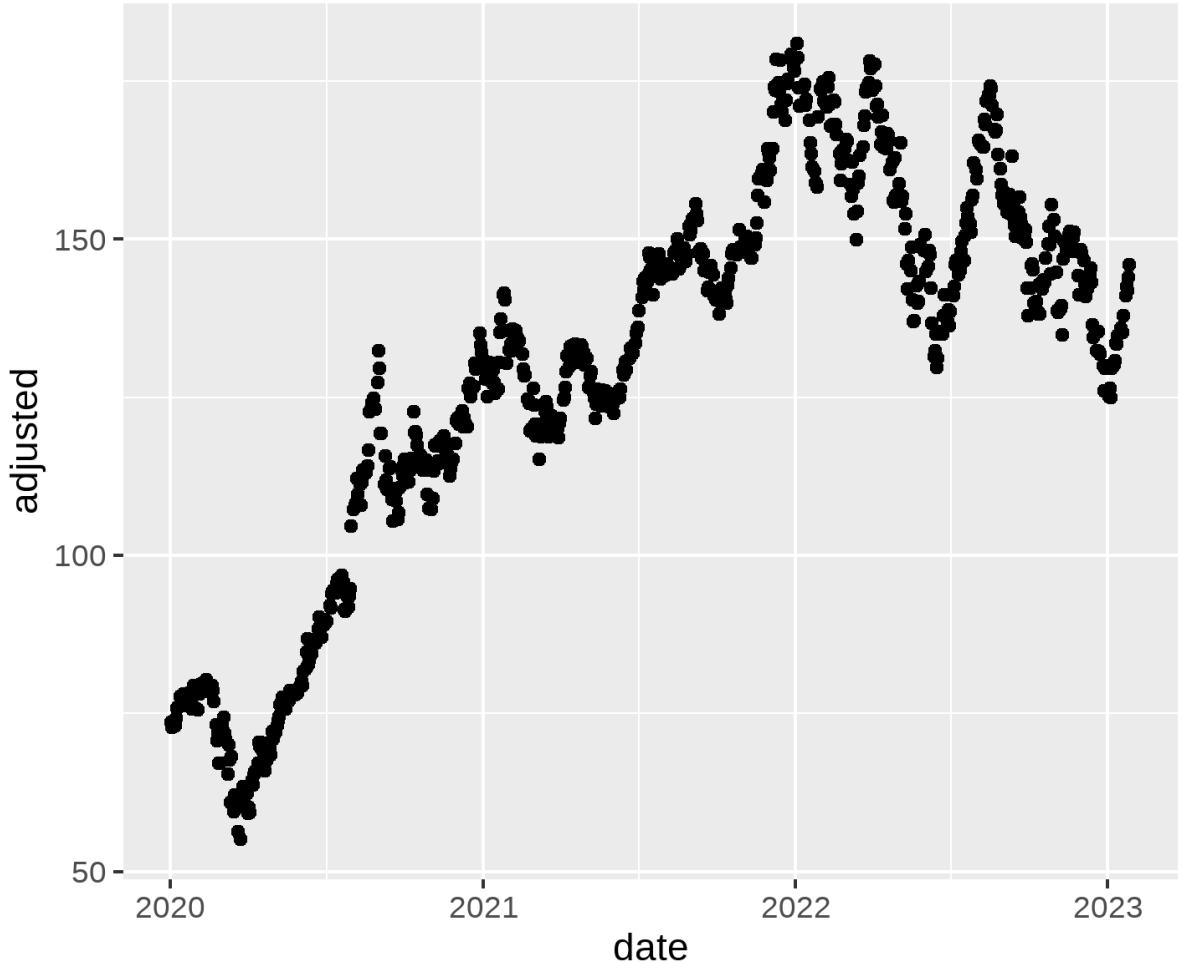
A Singular TS: The GG Layers

```
# layers are + in ggplot2
ggplot2::ggplot(
  aapl,
  ggplot2::aes(x = date, y = adjusted)
)
```



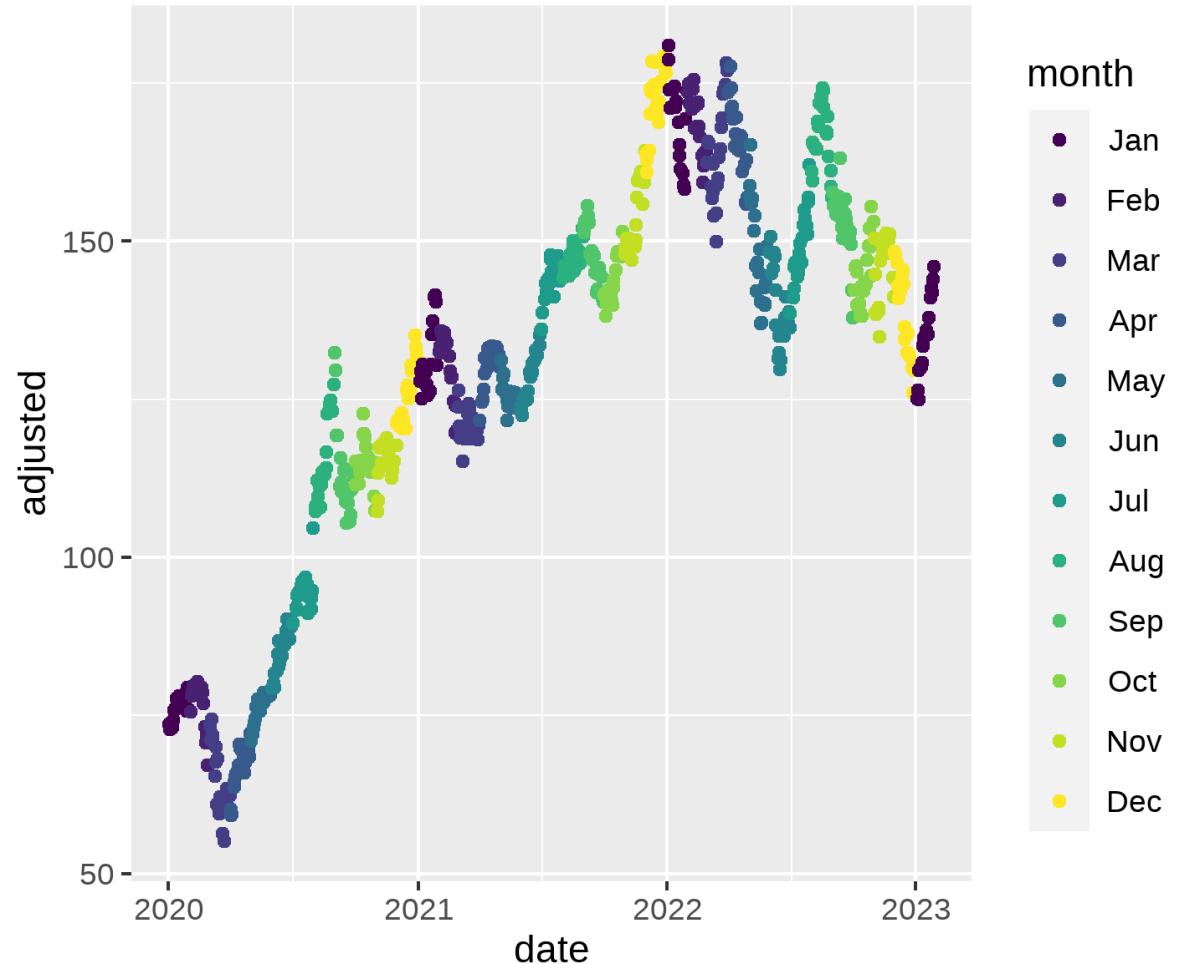
A Singular TS: The GG Layers

```
# layers are + in ggplot2
ggplot2::ggplot(
  aapl,
  ggplot2::aes(x = date, y = adjusted)
) +
  ggplot2::geom_point()
```



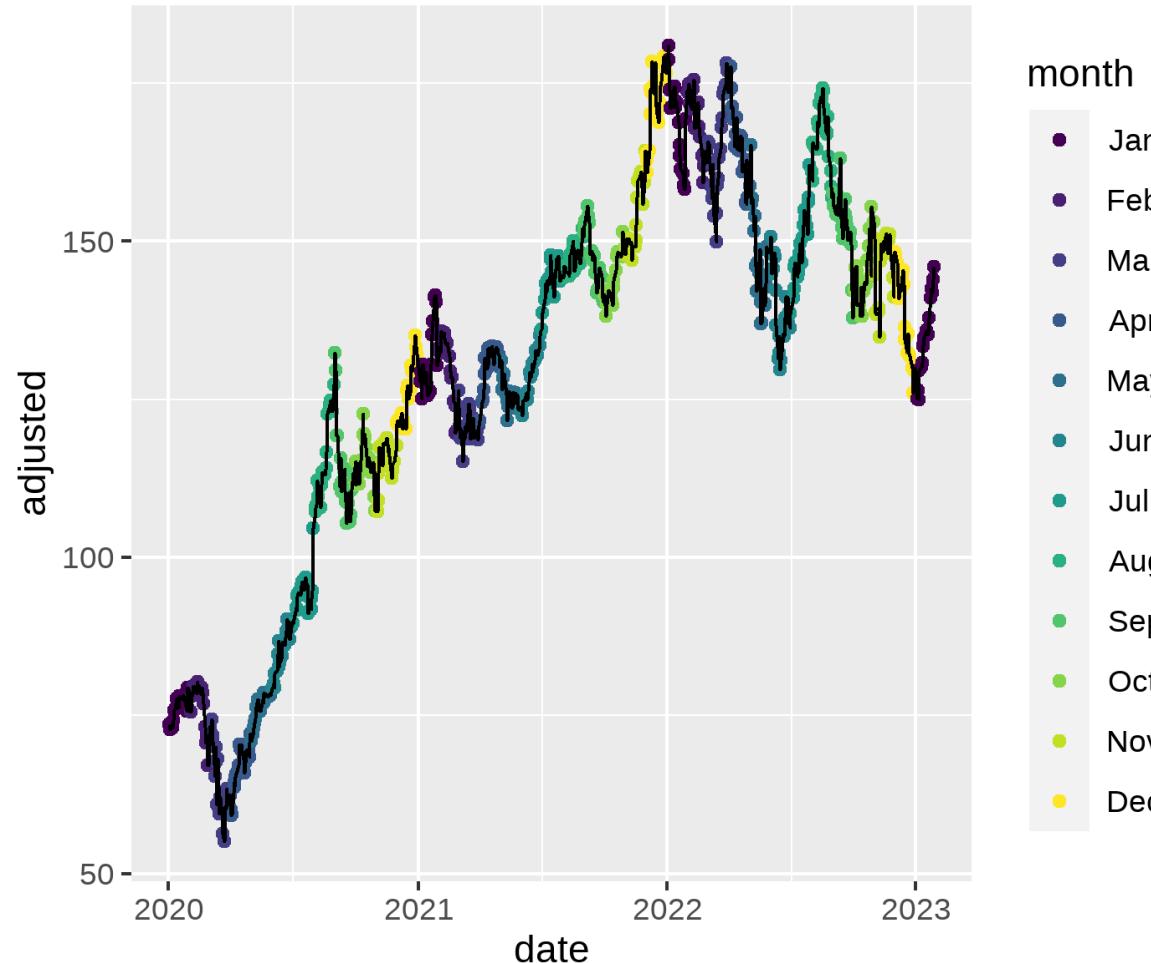
A Singular TS: The GG Layers

```
# layers are + in ggplot2
ggplot2::ggplot(
  aapl,
  ggplot2::aes(x = date, y = adjusted)
) +
  ggplot2::geom_point(
    ggplot2::aes(color = month)
)
```



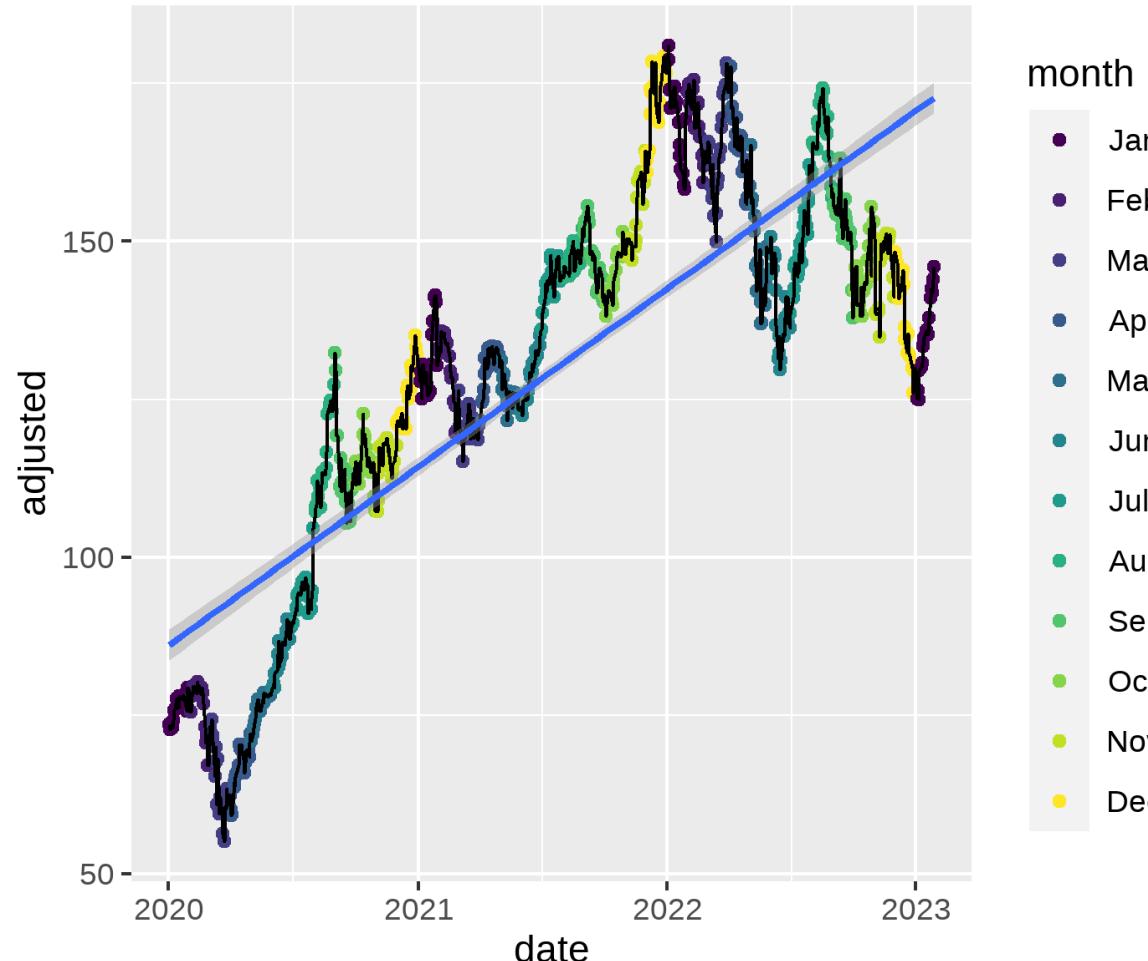
A Singular TS: The GG Layers

```
# layers are + in ggplot2
ggplot2::ggplot(
  aapl,
  ggplot2::aes(x = date, y = adjusted)
) +
  ggplot2::geom_point(
    ggplot2::aes(color = month)
) +
  ggplot2::geom_line()
```



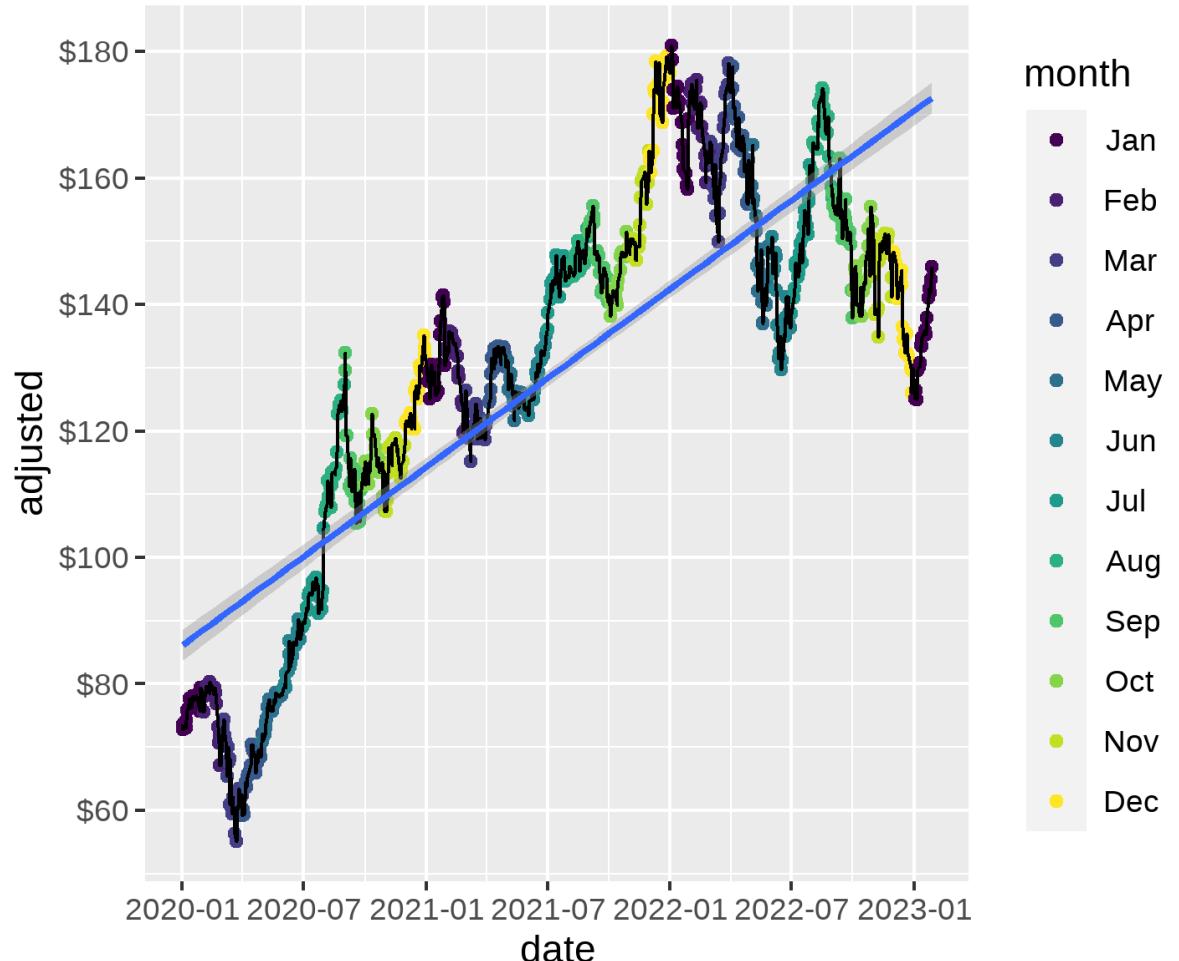
A Singular TS: The GG Layers

```
# layers are + in ggplot2
ggplot2::ggplot(
  aapl,
  ggplot2::aes(x = date, y = adjusted)
) +
  ggplot2::geom_point(
    ggplot2::aes(color = month)
) +
  ggplot2::geom_line() +
  ggplot2::geom_smooth(
    method = lm, formula = 'y ~ x'
)
```



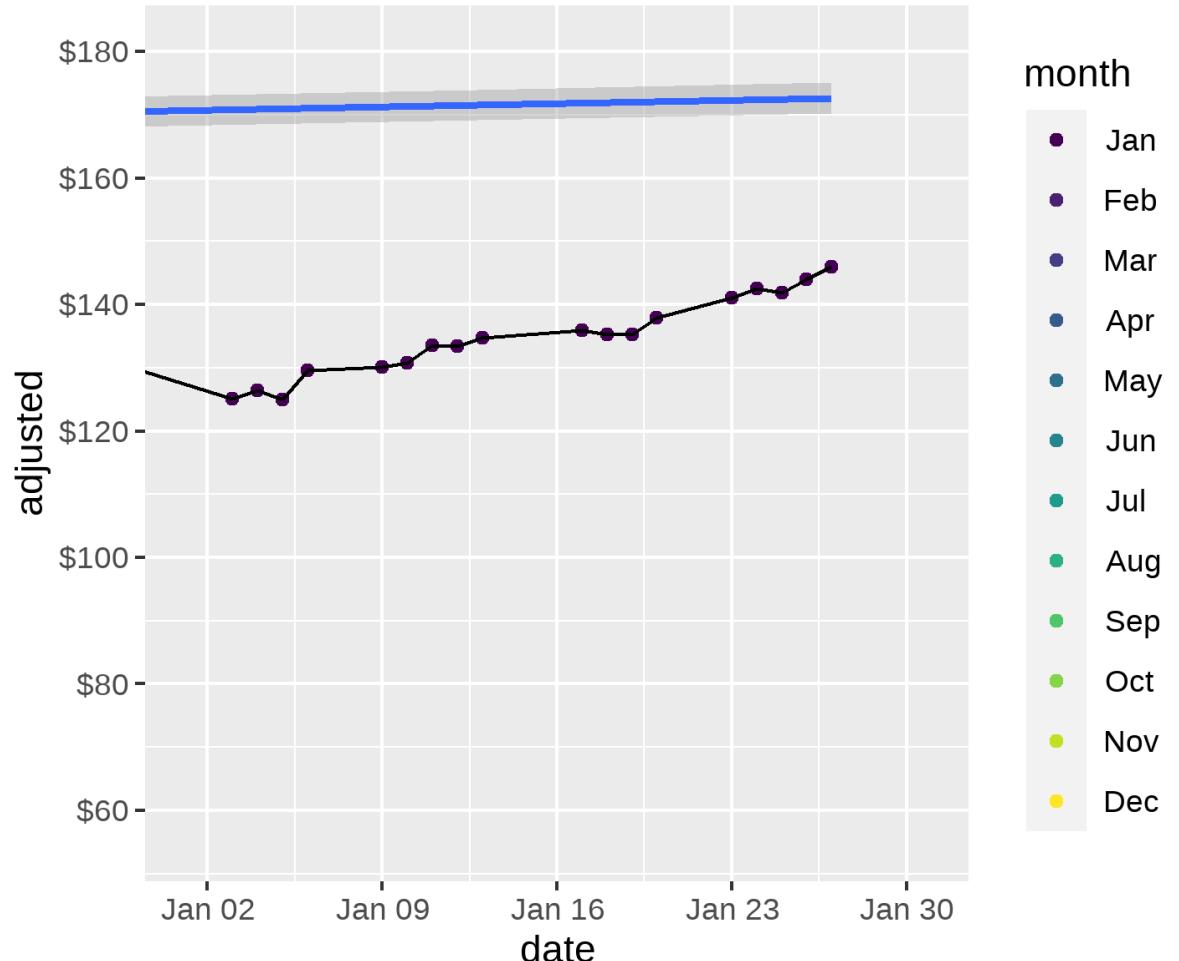
A Singular TS: The GG Layers

```
# layers are + in ggplot2
ggplot2::ggplot(
  aapl,
  ggplot2::aes(x = date, y = adjusted)
) +
  ggplot2::geom_point(
    ggplot2::aes(color = month)
) +
  ggplot2::geom_line() +
  ggplot2::geom_smooth(
    method = lm, formula = 'y ~ x'
) +
  ggplot2::scale_x_date(
    breaks = scales::pretty_breaks(n=8)
) +
  ggplot2::scale_y_continuous(
    breaks = scales::pretty_breaks(n=6),
    labels = scales::dollar)
```



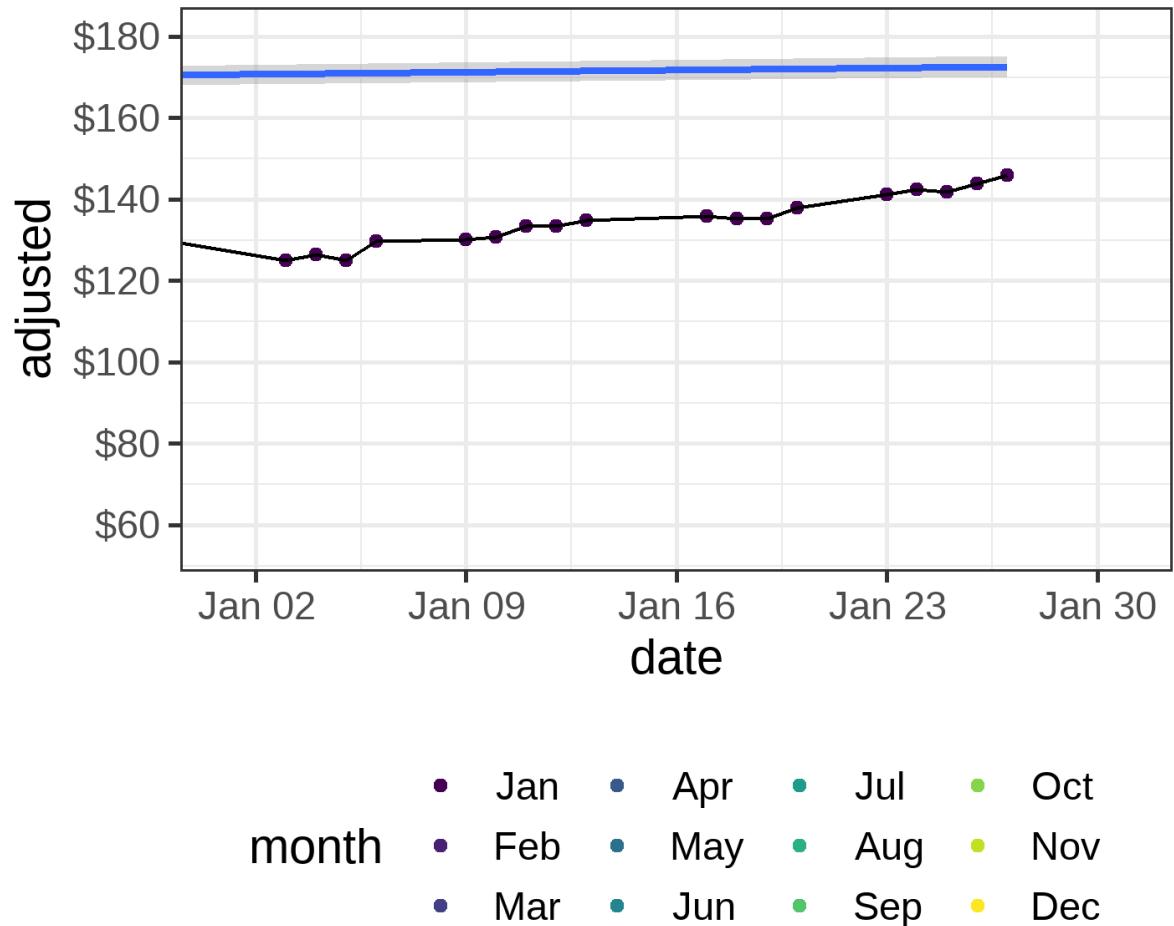
A Singular TS: The GG Layers

```
# layers are + in ggplot2
ggplot2::ggplot(
  aapl,
  ggplot2::aes(x = date, y = adjusted)
) +
  ggplot2::geom_point(
    ggplot2::aes(color = month)
  ) +
  ggplot2::geom_line() +
  ggplot2::geom_smooth(
    method = lm, formula = 'y ~ x'
  ) +
  ggplot2::scale_x_date(
    breaks = scales::pretty_breaks(n=8)
  ) +
  ggplot2::scale_y_continuous(
    breaks = scales::pretty_breaks(n=6),
    labels = scales::dollar) +
  tidyquant::coord_x_date(
    xlim = c('2023-01-01', '2023-01-31')
)
```



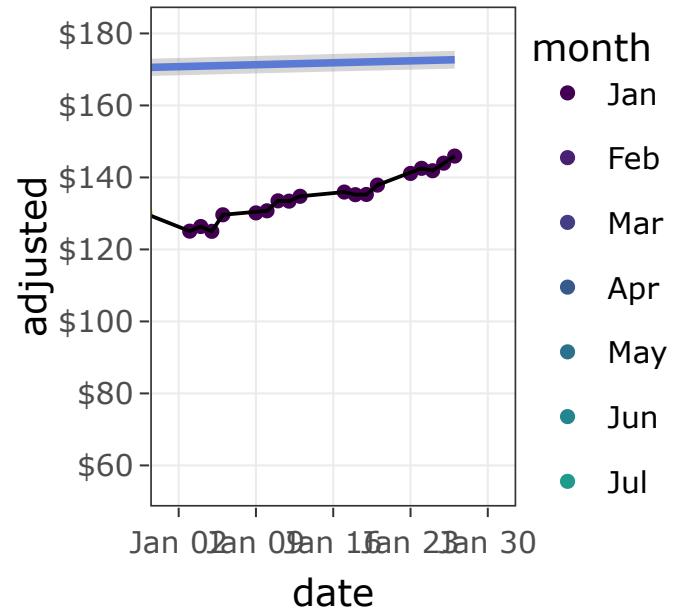
A Singular TS: The GG Layers

```
# layers are + in ggplot2
ggplot2::ggplot(
  aapl,
  ggplot2::aes(x = date, y = adjusted)
) +
  ggplot2::geom_point(
    ggplot2::aes(color = month)
  ) +
  ggplot2::geom_line() +
  ggplot2::geom_smooth(
    method = lm, formula = 'y ~ x'
  ) +
  ggplot2::scale_x_date(
    breaks = scales::pretty_breaks(n=8)
  ) +
  ggplot2::scale_y_continuous(
    breaks = scales::pretty_breaks(n=6),
    labels = scales::dollar) +
  tidyquant::coord_x_date(
    xlim = c('2023-01-01', '2023-01-31')
  ) +
  ggplot2::theme_bw(base_size = 14) +
  ggplot2::theme(
    legend.position = 'bottom'
  )
```



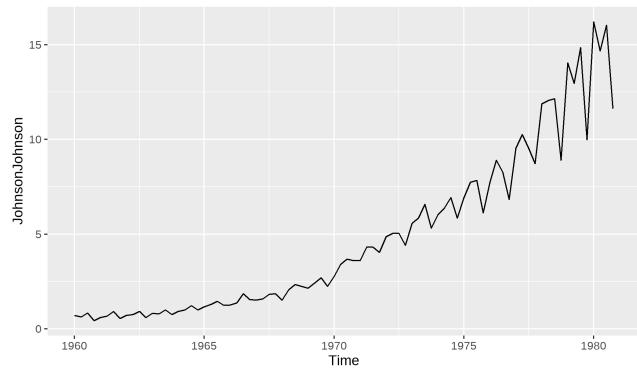
A Singular TS: The GG Layers

```
# layers are + in ggplot2
ggplot2::ggplot(
  aapl,
  ggplot2::aes(x = date, y = adjusted)
) +
  ggplot2::geom_point(
    ggplot2::aes(color = month)
  ) +
  ggplot2::geom_line() +
  ggplot2::geom_smooth(
    method = lm, formula = 'y ~ x'
  ) +
  ggplot2::scale_x_date(
    breaks = scales::pretty_breaks(n=8)
  ) +
  ggplot2::scale_y_continuous(
    breaks = scales::pretty_breaks(n=6),
    labels = scales::dollar) +
  tidyquant::coord_x_date(
    xlim = c('2023-01-01', '2023-01-31')
  ) +
  ggplot2::theme_bw(base_size = 14) +
  ggplot2::theme(
    legend.position = 'bottom'
  ) -> aapl_plot
plotly::ggplotly(p = aapl_plot)
```

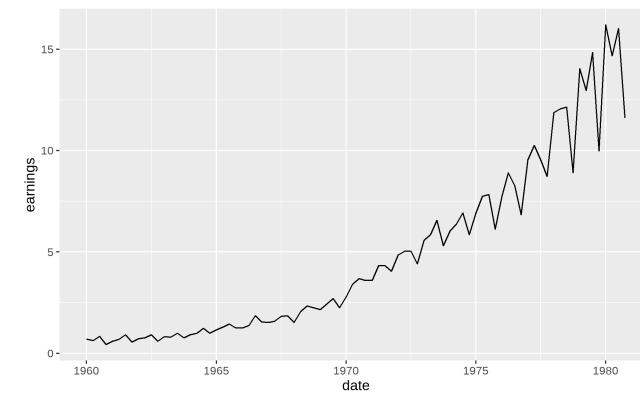


So How to Handle `ts()` Objects?

```
forecast::autoplot(JohnsonJohnson)
```



```
jj_df = data.frame(  
  date = zoo::as.Date(x = time(JohnsonJohnson)) ,  
  earnings = JohnsonJohnson  
)  
  
ggplot2::ggplot(  
  jj_df, ggplot2::aes(x = date, y = earnings) )  
  ggplot2::geom_line()
```



Multiple TS: A Demo

Let us build a panel seasonal plot of your choice with FRED data.

Recap

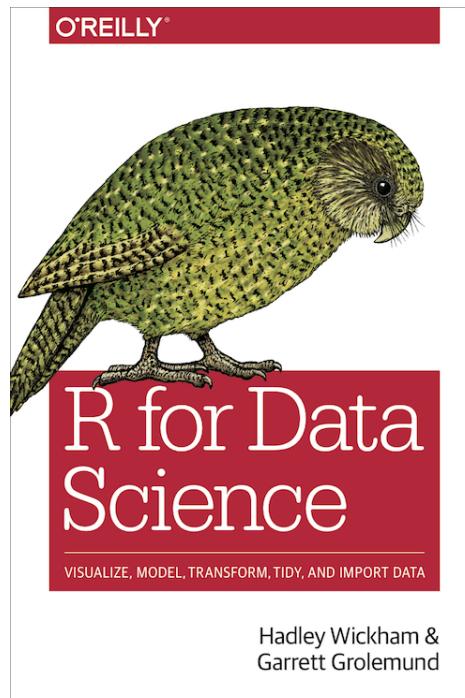
Summary of Main Points

By now, you should be able to do the following:

- Examine a line chart for trends, seasonality, and cycles.
- Explain the grammar of graphics and how it can be used to create time series plots in .
- Create interactive time-series plots by using the `plotly` package .

Things to Do to Prepare for Our Next Class

- Go over your notes, read the **references below**, and **complete** the self-paced R tutorial.
- Complete [Assignment 02](#) and [Assignment 03](#) on Canvas.



- Data Visualization
- Graphics for Communication
- Dates and Times