# Stocks.R

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(gganimate)
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
```

```
##
## Attaching package: 'xts'
```

```
## The following objects are masked from 'package:dplyr':
##
##     first, last
```

```
## Loading required package: TTR
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
library(gifski)
library(png)
library(forecast)
```

```
## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff
```

```
library(tibble)
library(viridis)
```

```
## Loading required package: viridisLite
```

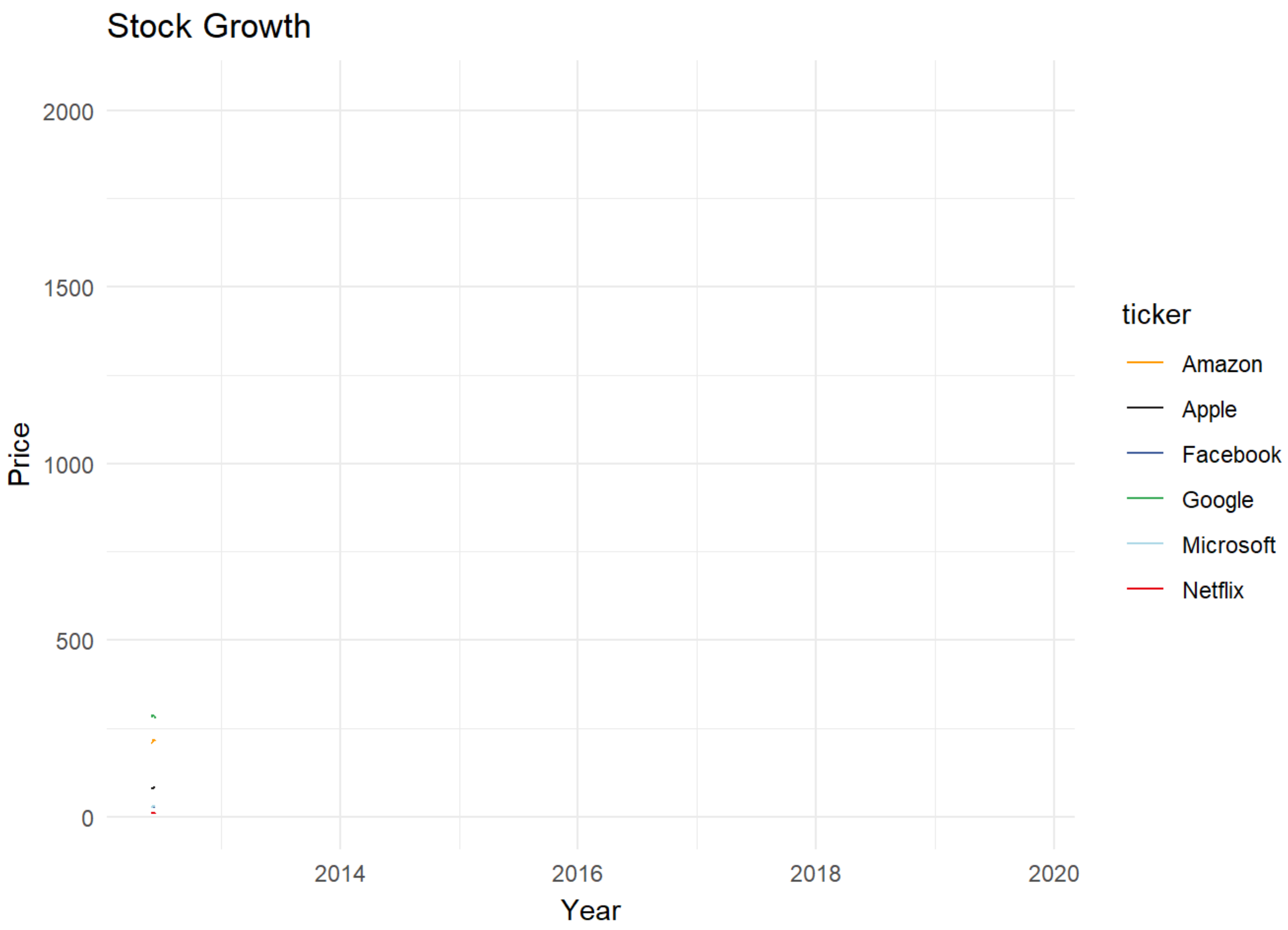
```
##
## Attaching package: 'viridis'
```

```
## The following object is masked from 'package:scales':
##
##     viridis_pal
```

```
aapl_data <- getSymbols(Symbols = "AAPL",
                        auto.assign = FALSE,
                        src = "yahoo",
                        from = "2012-06-01",
                        to = Sys.Date())
```

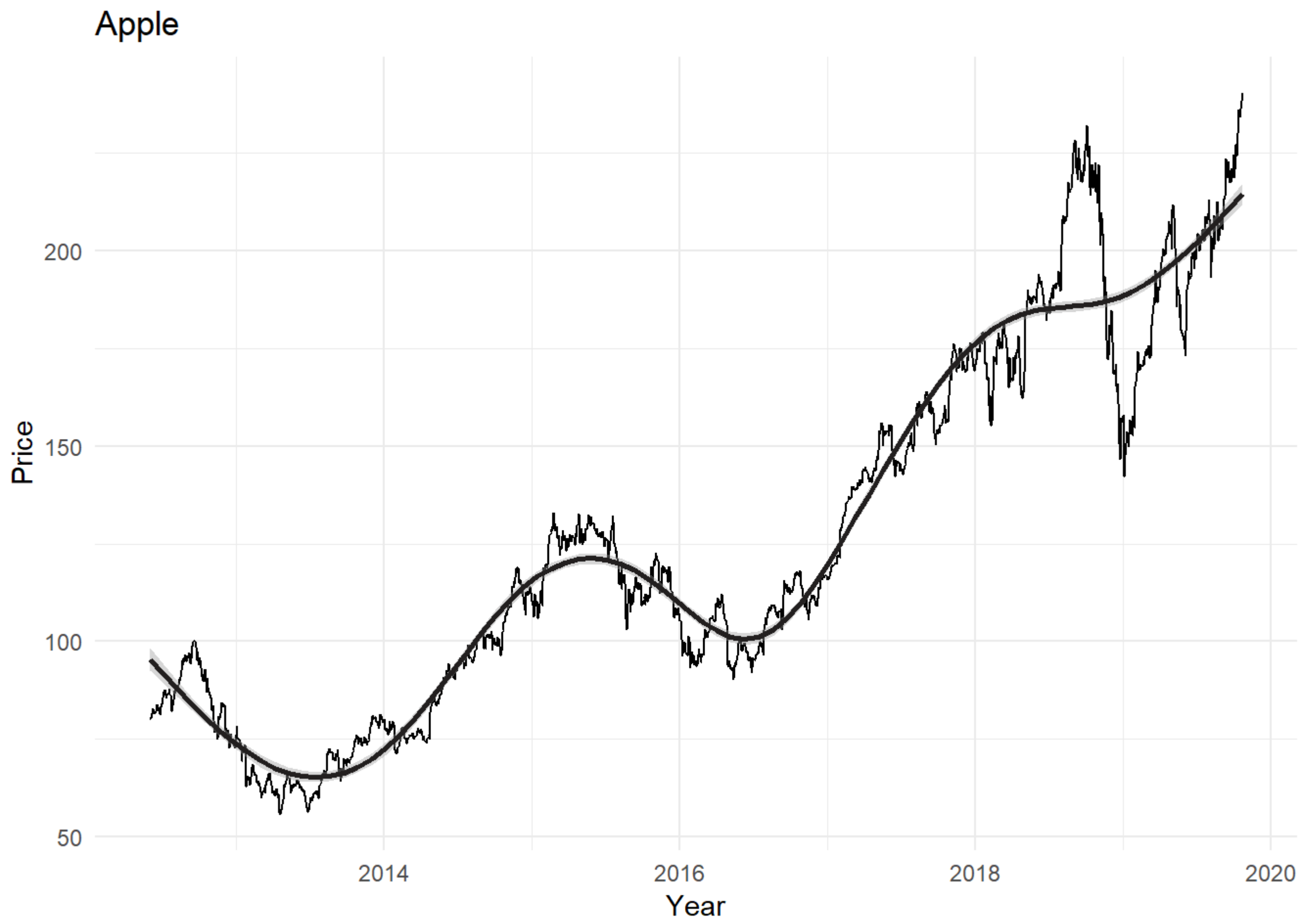
```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
amzn_data <- getSymbols(Symbols = "AMZN",
                        auto.assign = FALSE,
                        from = '2012-06-01',
                        to = Sys.Date())
googl_data <- getSymbols(Symbols = "GOOG",
                        auto.assign = FALSE,
                        from = '2012-06-01',
                        to = Sys.Date())
msft_data <- getSymbols(Symbols = "MSFT",
                        auto.assign = FALSE,
                        from = '2012-06-01',
                        to = Sys.Date())
fb_data <- getSymbols(Symbols = "FB",
                      auto.assign = FALSE,
                      from = '2012-06-01',
                      to = Sys.Date())
nflx_data <- getSymbols(Symbols = "NFLX",
                      auto.assign = FALSE,
                      from = '2012-06-01',
                      to = Sys.Date())


combined_close_tech <- data.frame(date = index(amzn_data),
                                  amzn_data,
                                  row.names = NULL) %>%
  select(date, close = AMZN.Close) %>%
  mutate(ticker = 'Amazon') %>%
  bind_rows(.,
            data.frame(date = index(googl_data),
                       googl_data,
                       row.names = NULL) %>%
              select(date, close = GOOG.Close) %>%
              mutate(ticker = 'Google')) %>%
  bind_rows(.,
            data.frame(date = index(msft_data),
                       msft_data,
                       row.names = NULL) %>%
              select(date, close = MSFT.Close) %>%
              mutate(ticker = 'Microsoft')) %>%
  bind_rows(.,
            data.frame(date = index(aapl_data),
                       aapl_data,
                       row.names = NULL) %>%
              select(date, close = AAPL.Close) %>%
              mutate(ticker = 'Apple')) %>%
  bind_rows(.,
            data.frame(date = index(nflx_data),
                       nflx_data,
                       row.names = NULL) %>%
              select(date, close = NFLX.Close) %>%
              mutate(ticker = 'Netflix')) %>%
  bind_rows(.,
            data.frame(date = index(fb_data),
                       fb_data,
                       row.names = NULL) %>%
              select(date, close = FB.Close) %>%
              mutate(ticker = 'Facebook'))


theme_bare <- theme(panel.background = element_blank(),
                    panel.border = element_blank(),
                    axis.title = element_blank(),
                    axis.ticks = element_blank(),
                    panel.grid = element_blank())


stock_colors <- c("Apple" = "#221f1f",
                  "Amazon" = "#ff9900",
                  "Facebook" = "#3b5998",
                  "Google" = "#34a853",
                  "Microsoft" = "#add8e6",
                  "Netflix" = "#e50914")


ggplot(data = combined_close_tech, aes(x = date, y = close, color = ticker)) +
  geom_line() +
  scale_color_manual(values = stock_colors) +
  xlab('Year') +
  ylab('Price') +
  ggtitle("Stock Growth") +
  theme_minimal() +
  transition_reveal(date)
```



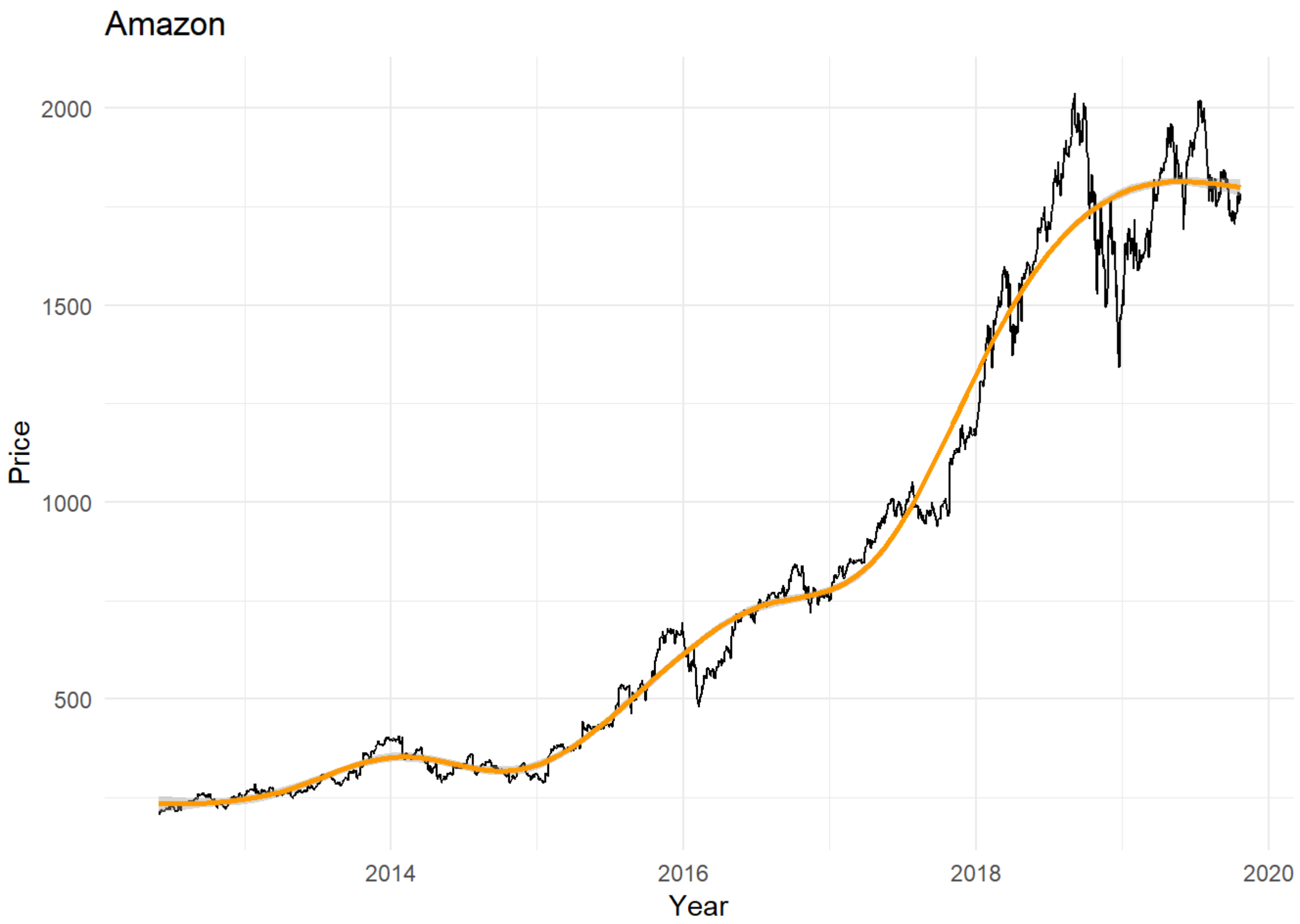
```
ggplot() +
  geom_line(data = aapl_data, aes(x = index(aapl_data), y = AAPL.Close, group = 1)) +
  stat_smooth(data = aapl_data, aes(x =index(aapl_data), y = AAPL.Close), color = "#221f1f") +
  xlab('Year') +
  ylab('Price') +
  ggtitle("Apple") +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



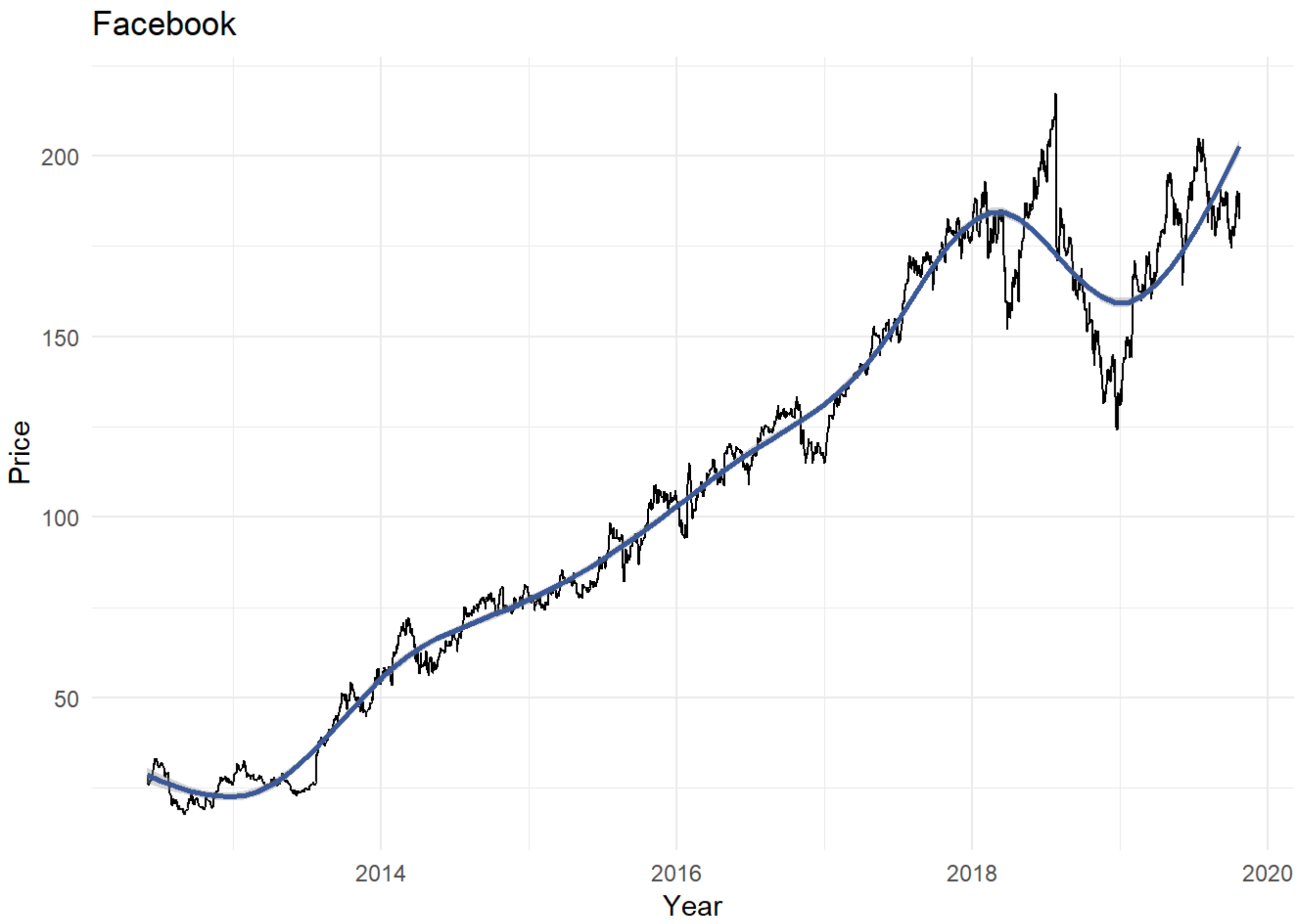
```
ggplot() +
  geom_line(data = amzn_data, aes(x = index(amzn_data), y = AMZN.Close, group = 1)) +
  stat_smooth(data = amzn_data, aes(x =index(amzn_data), y = AMZN.Close), color = "#ff9900") +
  xlab('Year') +
  ylab('Price') +
  ggtitle("Amazon") +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



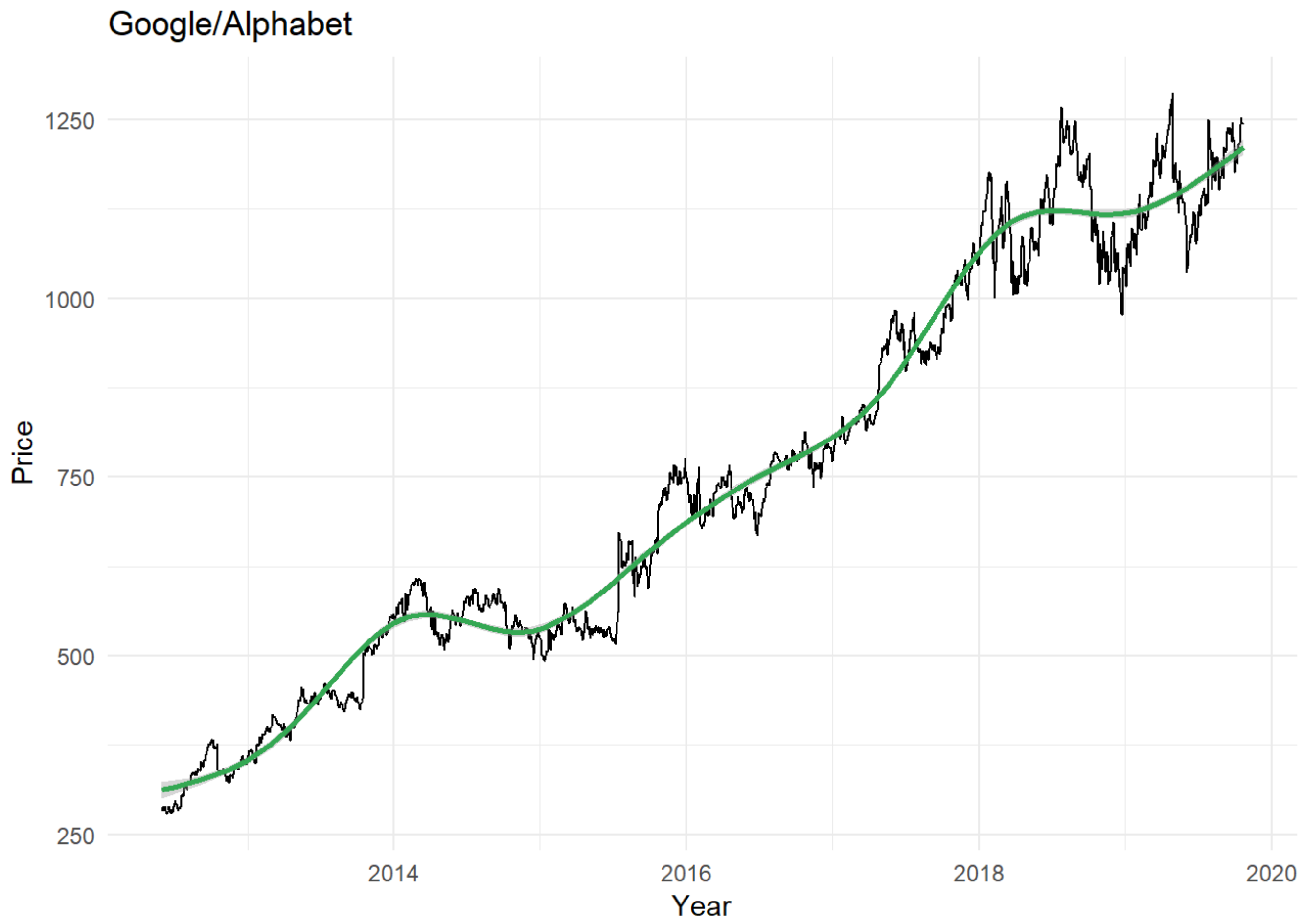
```
ggplot() +
  geom_line(data = fb_data, aes(x = index(fb_data), y = FB.Close, group = 1)) +
  stat_smooth(data = fb_data, aes(x =index(fb_data), y = FB.Close), color = "#3b5998") +
  xlab('Year') +
  ylab('Price') +
  ggtitle("Facebook") +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



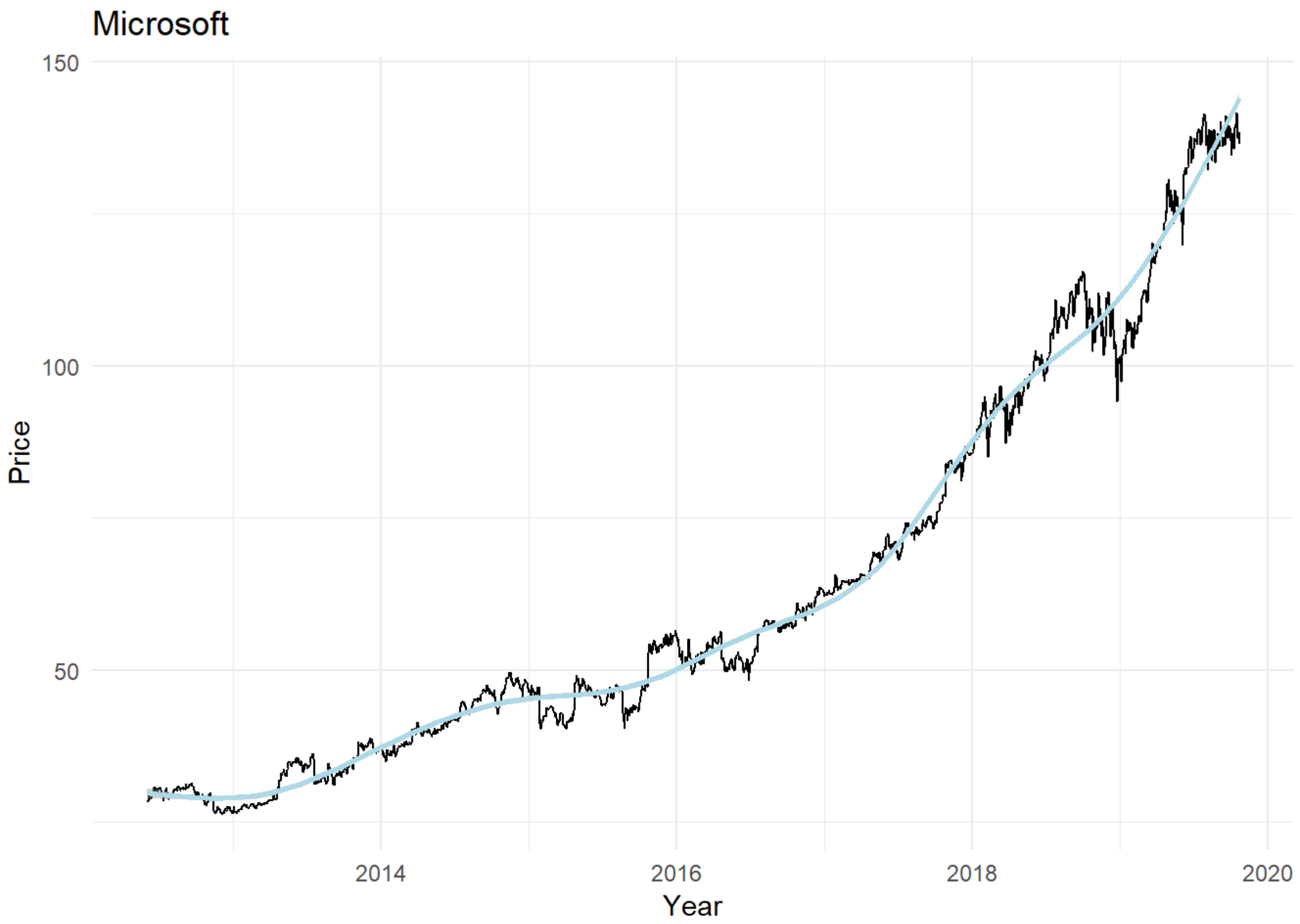
```
ggplot() +
  geom_line(data =googl_data, aes(x = index(googl_data), y = GOOG.Close, group = 1)) +
  stat_smooth(data = googl_data, aes(x =index(googl_data), y = GOOG.Close), color = "#34a853") +
  xlab('Year') +
  ylab('Price') +
  ggtitle("Google/Alphabet") +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



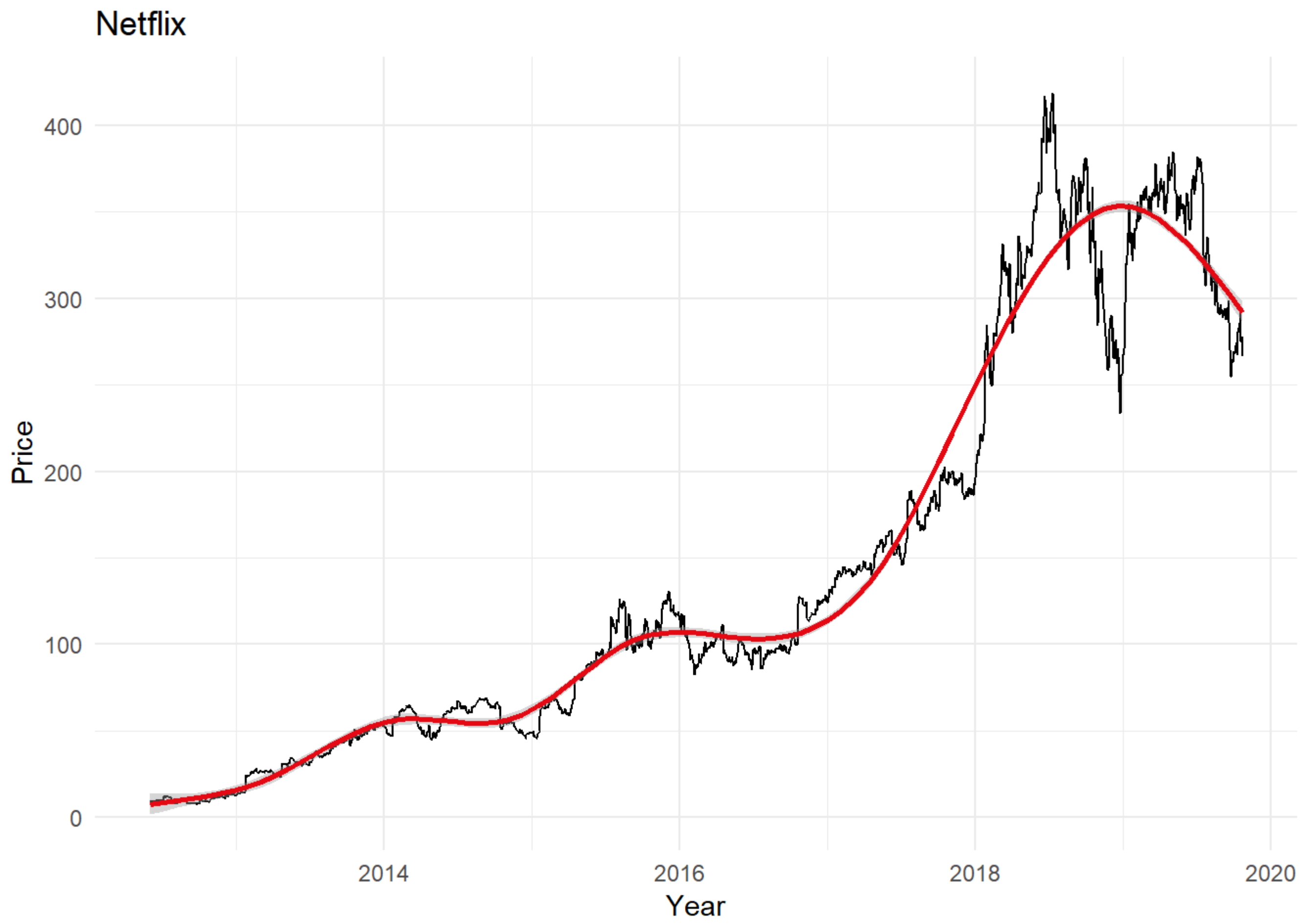
```
ggplot() +
  geom_line(data =msft_data, aes(x = index(msft_data), y = MSFT.Close, group = 1)) +
  stat_smooth(data = msft_data, aes(x =index(msft_data), y = MSFT.Close), color = "#add8e6") +
  xlab('Year') +
  ylab('Price') +
  ggtitle("Microsoft") +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

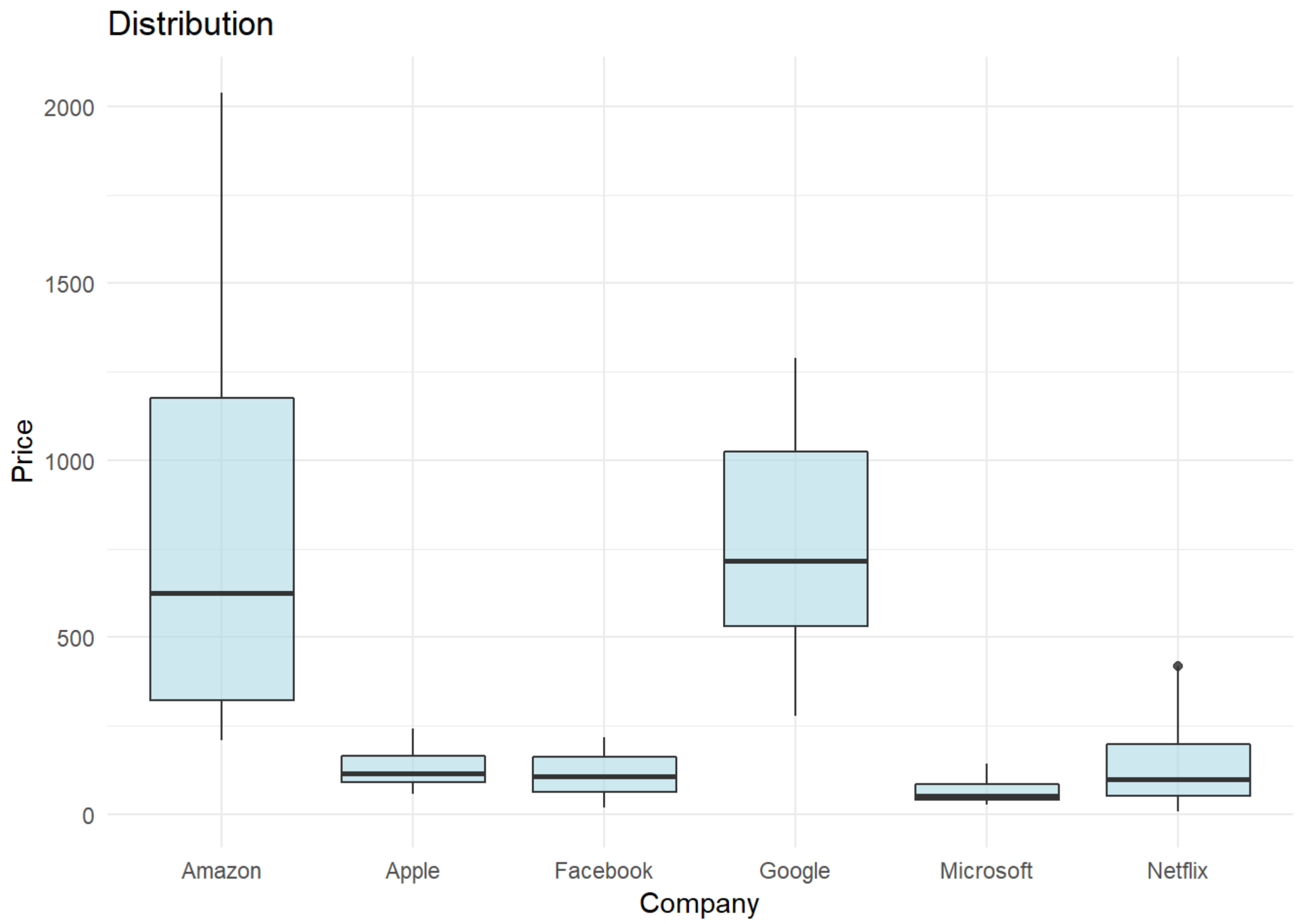


```
ggplot() +
  geom_line(data =nflx_data, aes(x = index(nflx_data), y = NFLX.Close, group = 1)) +
  stat_smooth(data = nflx_data, aes(x =index(nflx_data), y = NFLX.Close), color = "#e50914") +
  xlab('Year') +
  ylab('Price') +
  ggtitle("Netflix") +
  theme_minimal()
```

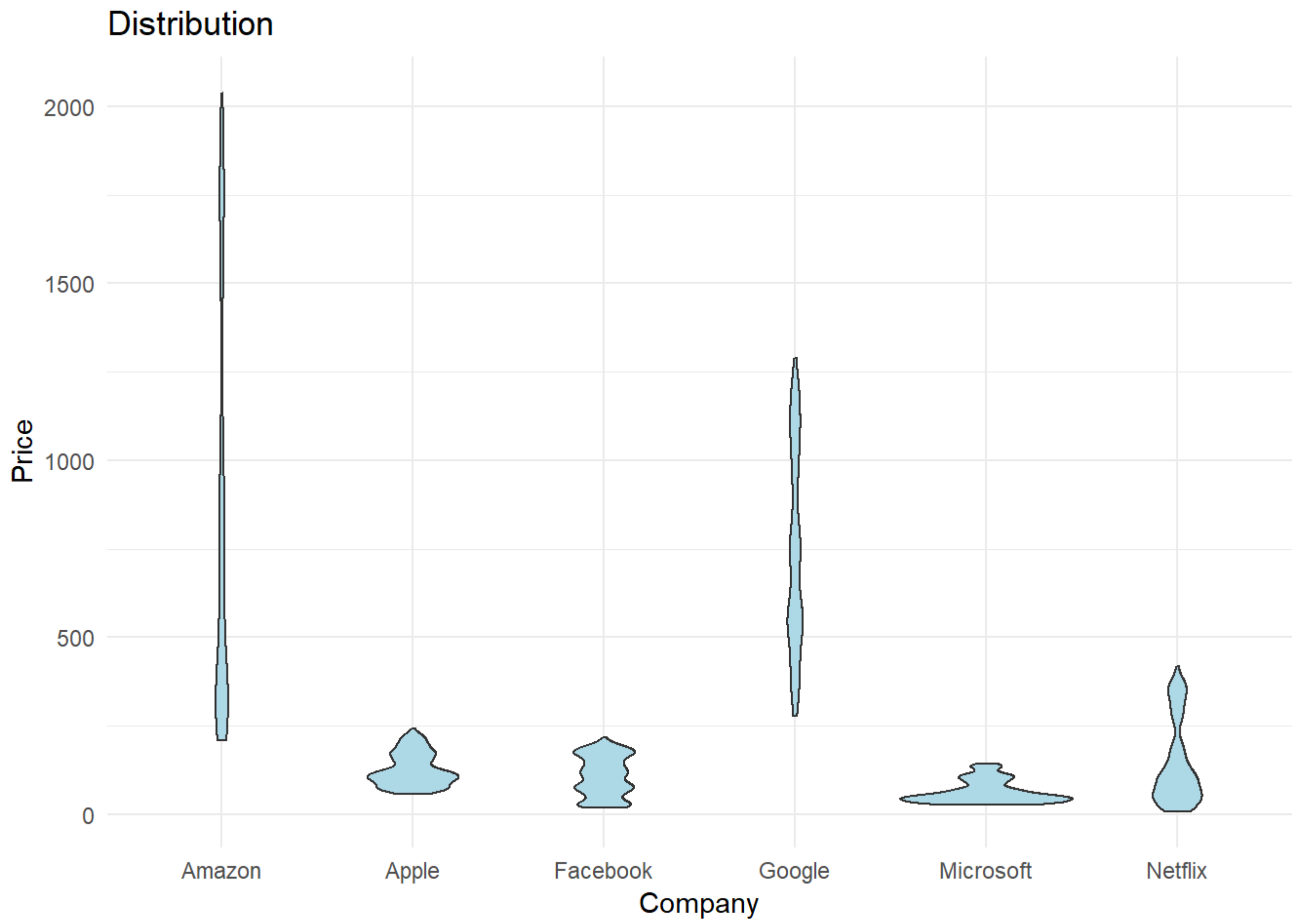
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



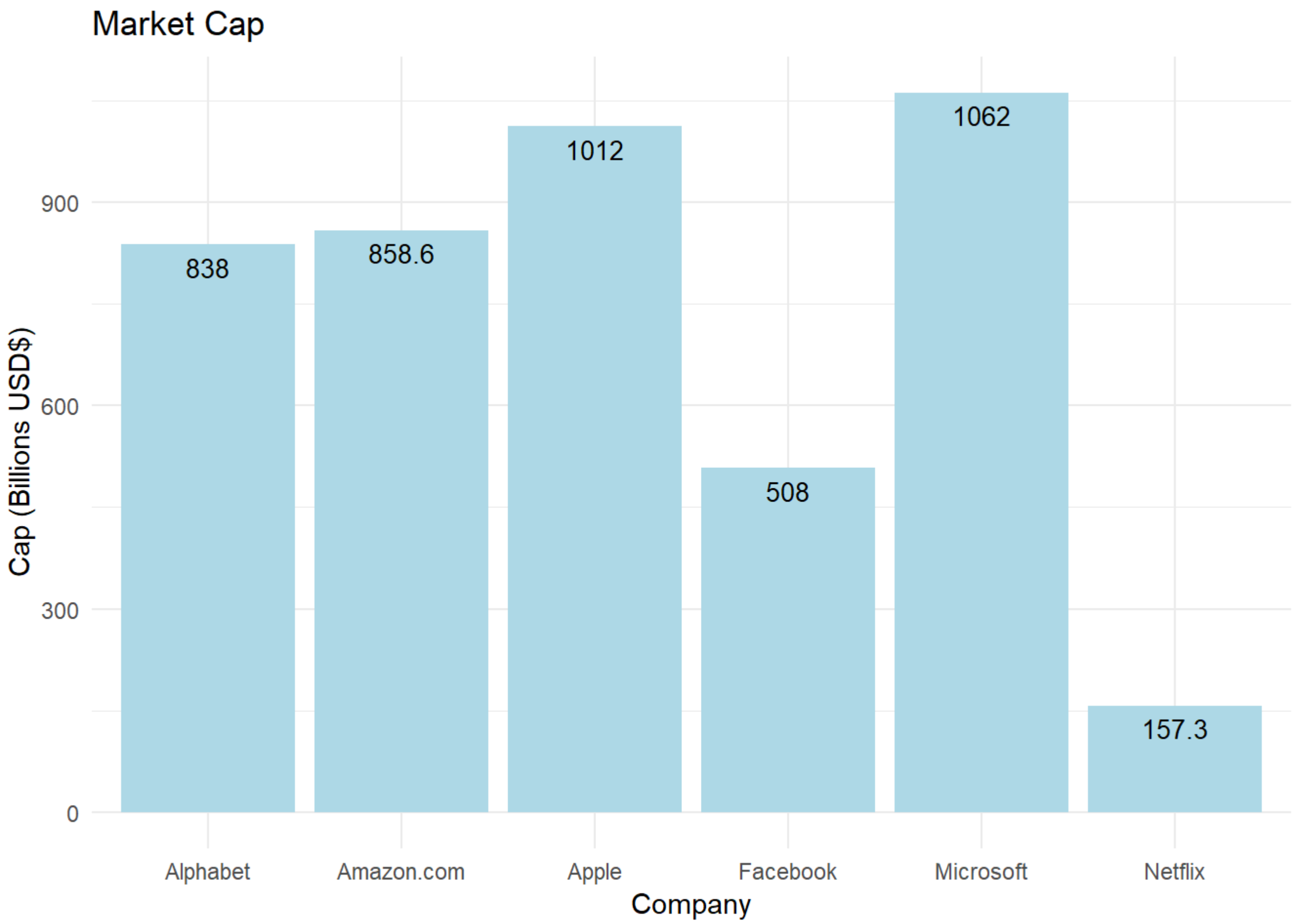
```
ggplot(combined_close_tech, aes(x=as.factor(ticker), y=close)) +
  geom_boxplot(alpha=0.6, fill="lightblue") +
  theme_minimal() +
  xlab('Company') +
  ylab('Price') +
  ggtitle("Distribution")
```



```
ggplot(combined_close_tech, aes(x=as.factor(ticker), y=close )) +
  geom_violin(fill="lightblue") +
  scale_fill_viridis(discrete = TRUE, alpha=0.6) +
  theme(
    plot.title = element_text(size=11)
  ) +
  theme_minimal() +
  xlab('Company') +
  ylab('Price') +
  ggtitle("Distribution")
```



```
marketvalue <- read.csv("marketvalue.csv")
View(marketvalue)


ggplot(data=marketvalue, aes(x=company, y=market_value)) +
  geom_bar(stat="identity", fill="lightblue" ) +
  theme_minimal() +
  xlab('Company') +
  ylab('Cap (Billions USD$)') +
  ggtitle("Market Cap") +
  geom_text(aes(label=market_value), vjust=1.6,
            color="black", size=3.5)
```


Market Cap
Cap (Billions USD$)
900
600
300
0
838
858.6
1012
508
1062
157.3
Alphabet
Amazon.com
Apple
Facebook
Microsoft
Netflix
Company