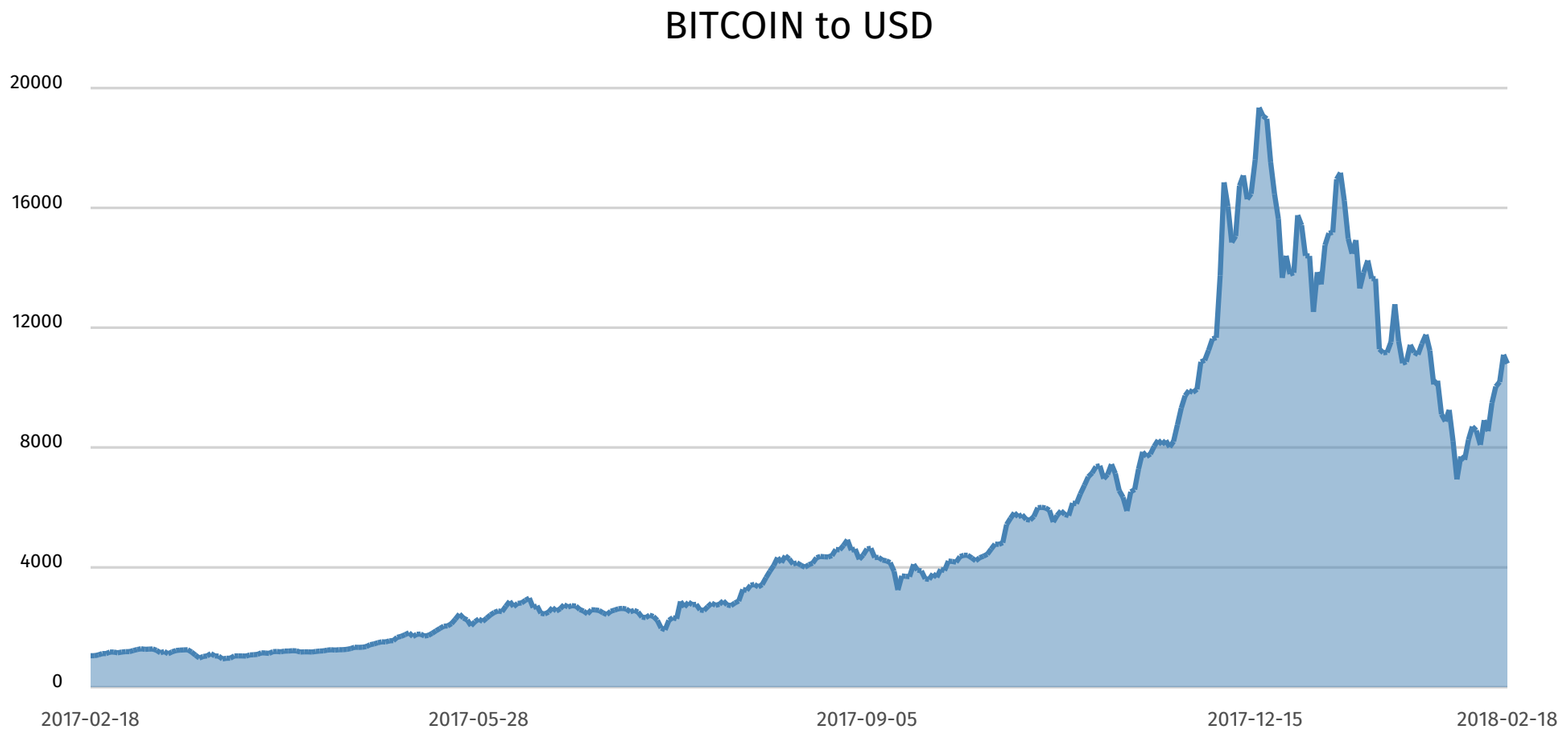
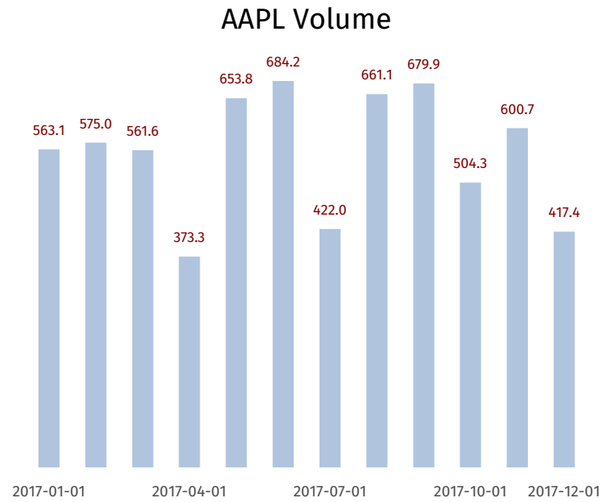
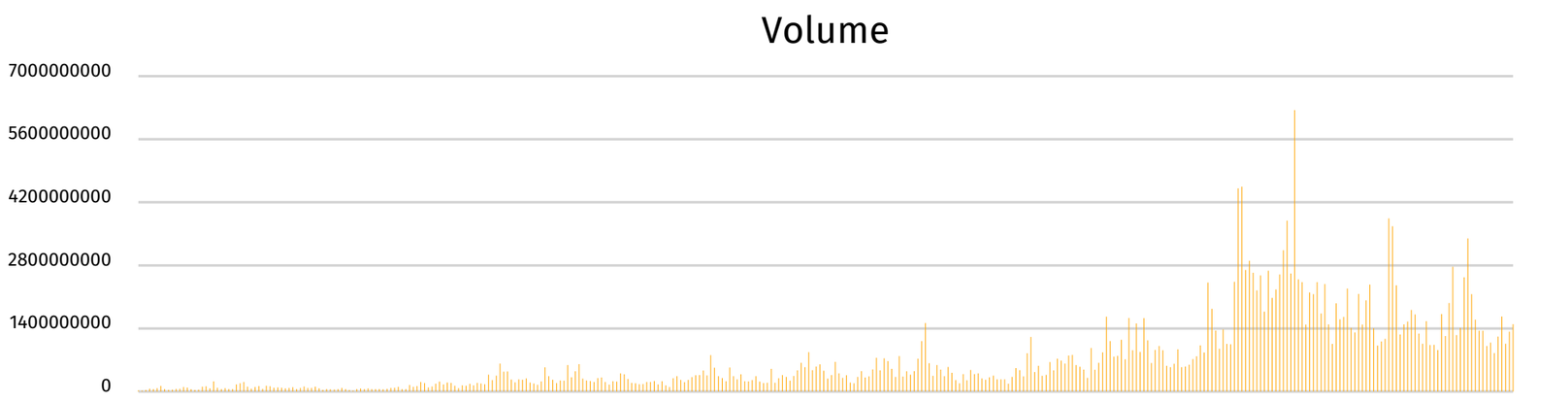
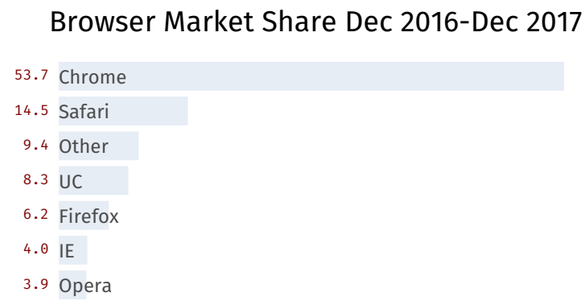
# dchart

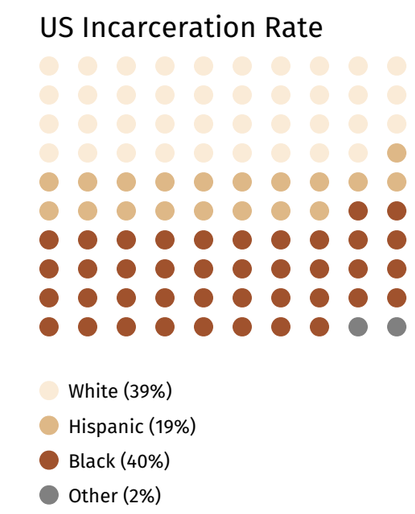
## charts for deck



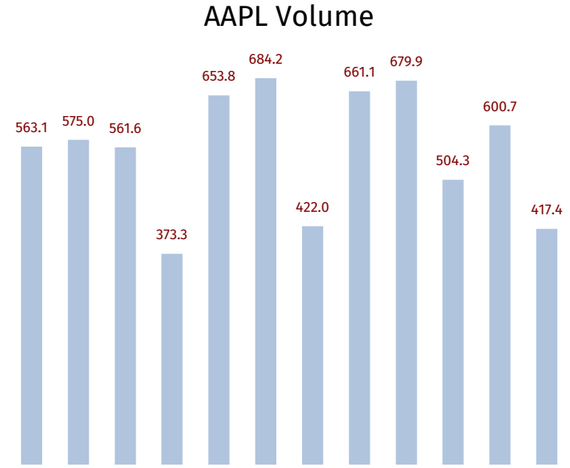



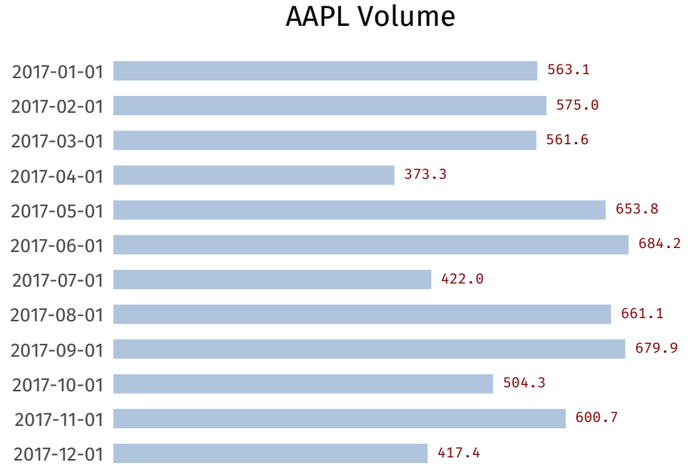



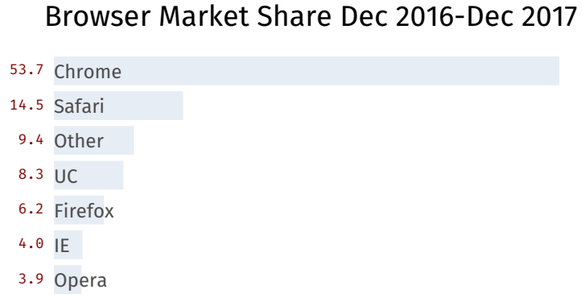

Anthony Starks



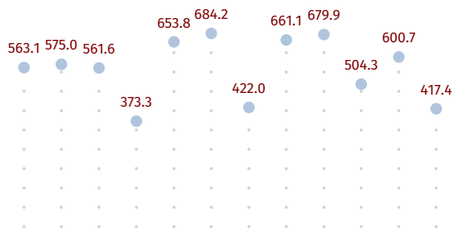

bar



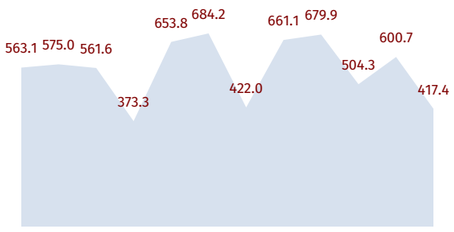

hbar



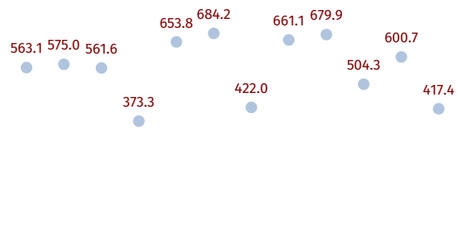

wbar



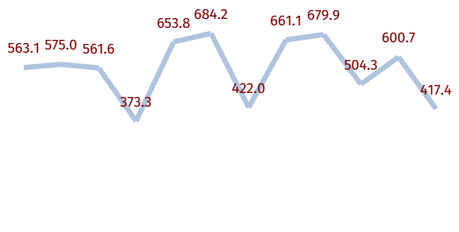

dot



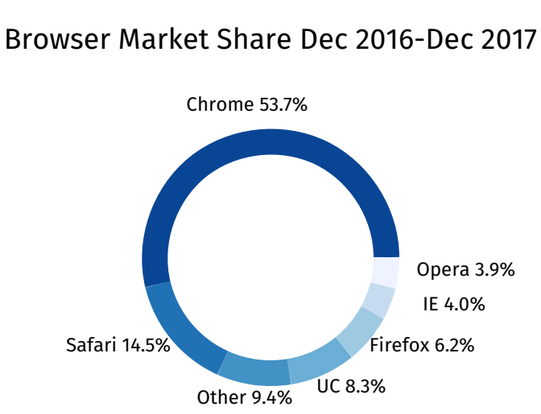

vol



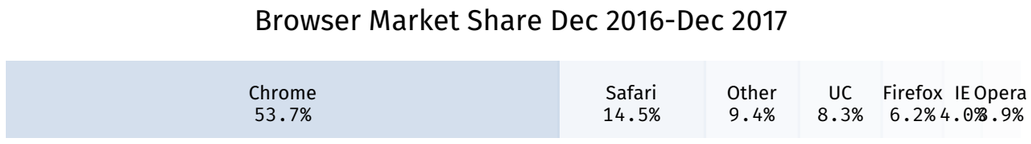

scatter



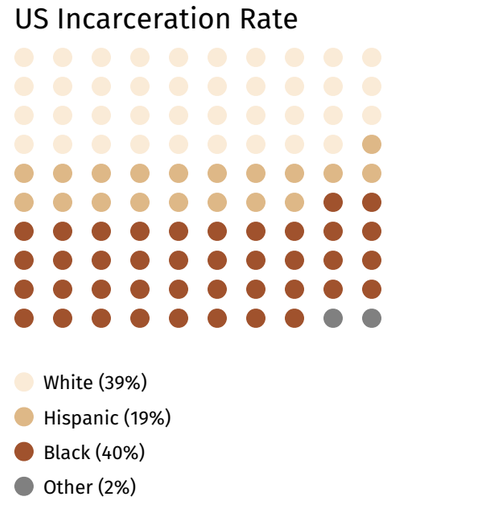

line




donut




pmap




pgrid

# Chart Types

| | |
|---|---|
| `-bar` | bar chart (default true) |
| `-wbar` | word bar chart (default false) |
| `-hbar` | horizontal bar chart (default false) |
| `-scatter` | scatter chart (default false) |
| `-dot` | dot plot (default false) |
| `-line` | line chart (default false) |
| `-vol` | volume plot (default false) |
| `-pgrid` | proportional grid (default false) |
| `-pmap` | proportional map (default false) |
| `-donut` | donut chart (default false) |

# Chart Elements

| | |
|---|---|
| `-grid` | show gridlines on the y axis (default false) |
| `-val` | show values (default true) |
| `-valpos` | value position (t=top, b=bottom, m=middle) (default "t") |
| `-yaxis` | show a y axis (default true) |
| `-yrange` | specify the y axis labels (min,max,step) |
| `-fulldeck` | generate full deck markup (default true) |
| `-title` | show the title (default true) |
| `-chartitle` | specify the title (overiding title in the data) |
| `-xlabel` | x axis label interval (default 1, 0 to supress all labels) |
| `-xlast` | show the last x label |
| `-hline` | horizontal line at value with label |

# Position and Scaling

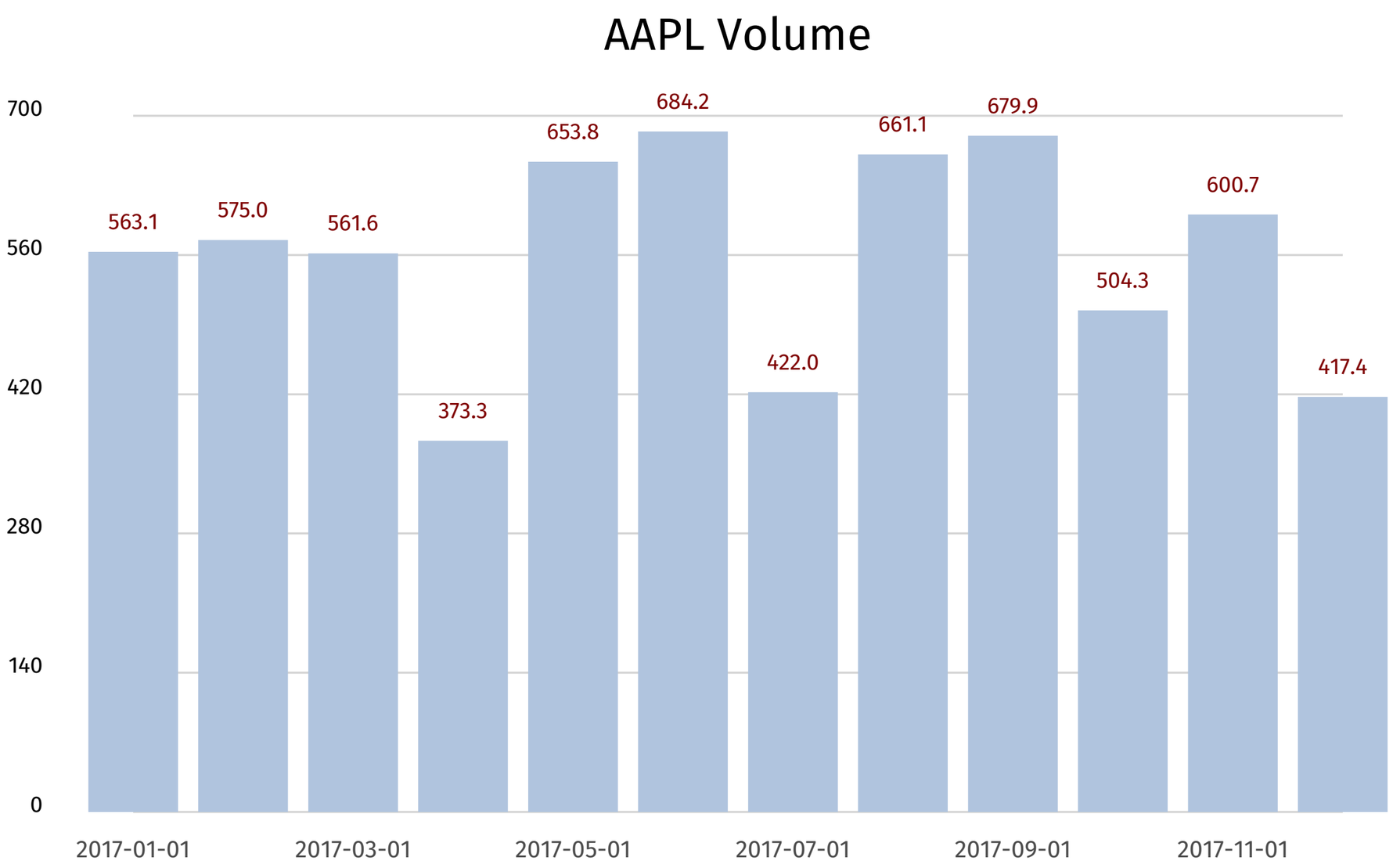
| | |
|---|---|
| `-top` | top of the plot (default 80) |
| `-bottom` | bottom of the plot (default 30) |
| `-left` | left margin (default 20) |
| `-right` | right margin (default 80) |
| `-min` | set the minimum value |
| `-max` | set the maximum value |
| `-dmin` | data minimum (default false, min=0) |

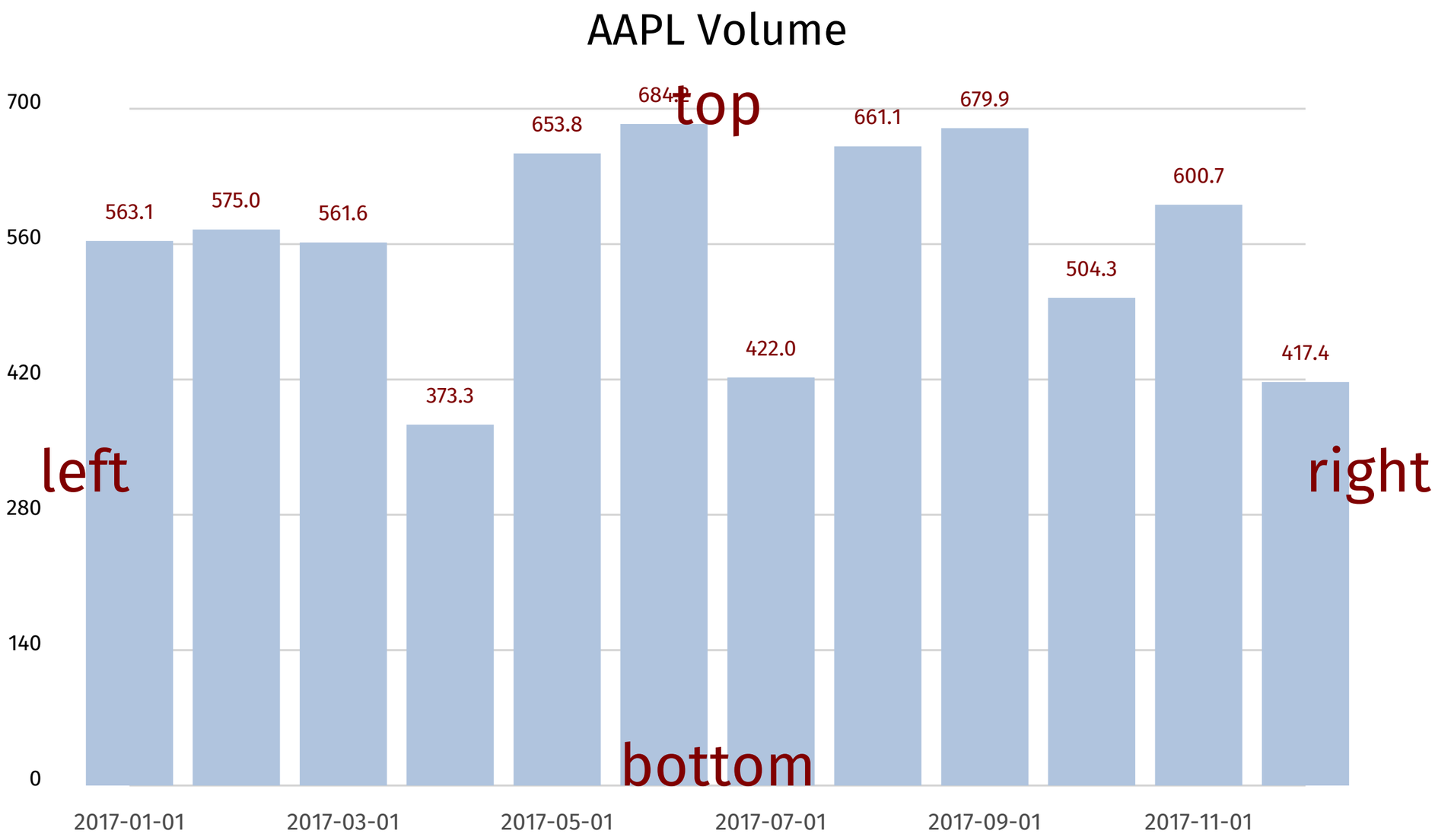
# Measures and Attributes

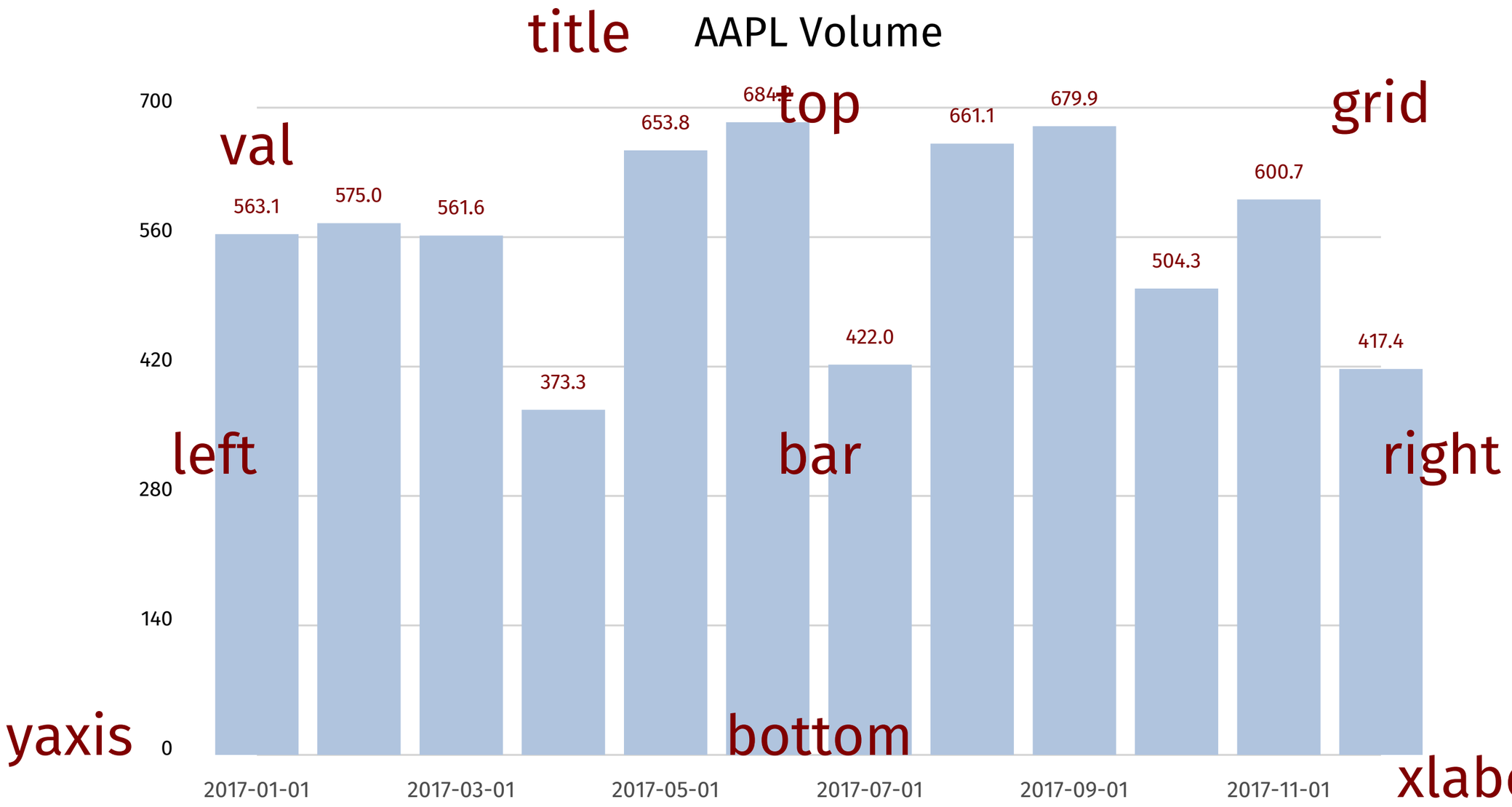
| | |
|---|---|
| `-barwidth` | barwidth (default computed from data size) |
| `-ls` | linespacing (default 2.4) |
| `-textsize` | text size (default 1.5) |
| `-color` | data color (default "lightsteelblue") |
| `-vcolor` | value color (default "rgb(127,0,0)") |
| `-datafmt` | data format for values (default "%.1f") |
| `-psize` | diameter of the donut (default 30) |
| `-pwidth` | width of the donut or proportional map (default 3) |

# CSV

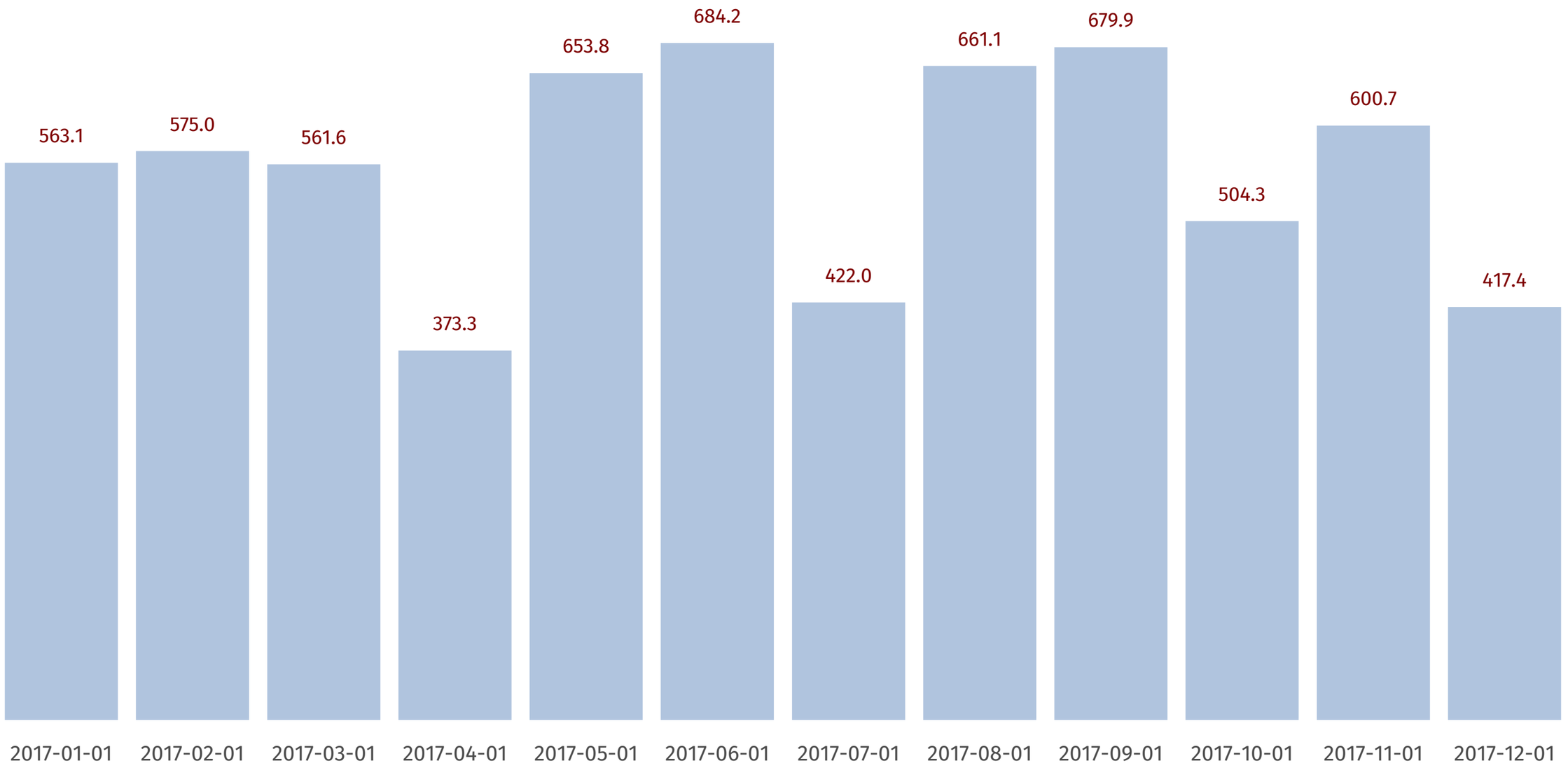
| | |
|---|---|
| `-csv` | read CSV files (default false) |
| `-csvcol` | specify the columns to use for label,value |


AAPL Volume
700
560
420
280
140
0
563.1
575.0
561.6
373.3
653.8
684.2
422.0
661.1
679.9
504.3
600.7
417.4
2017-01-01
2017-03-01
2017-05-01
2017-07-01
2017-09-01
2017-11-01

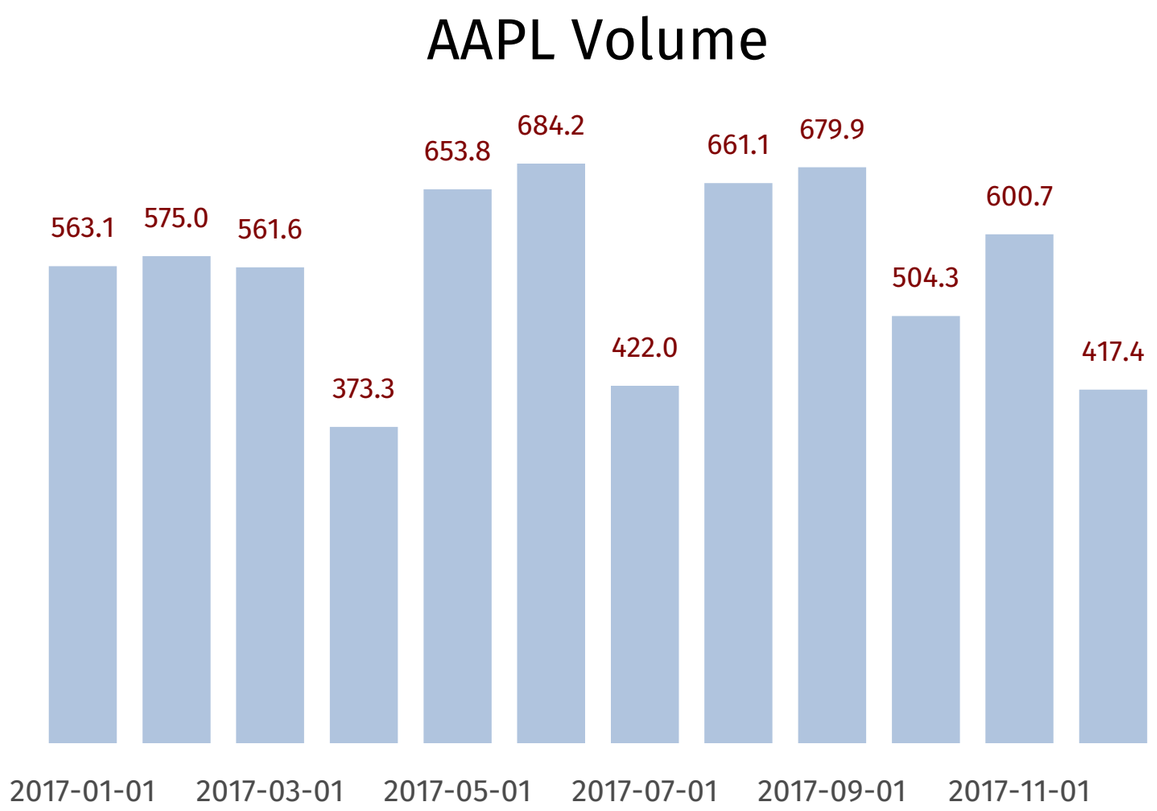

AAPL Volume
700
560
420
280
140
0
563.1
575.0
561.6
373.3
653.8
684.3
422.0
661.1
679.9
504.3
600.7
417.4
2017-01-01
2017-03-01
2017-05-01
2017-07-01
2017-09-01
2017-11-01
top
left
right
bottom



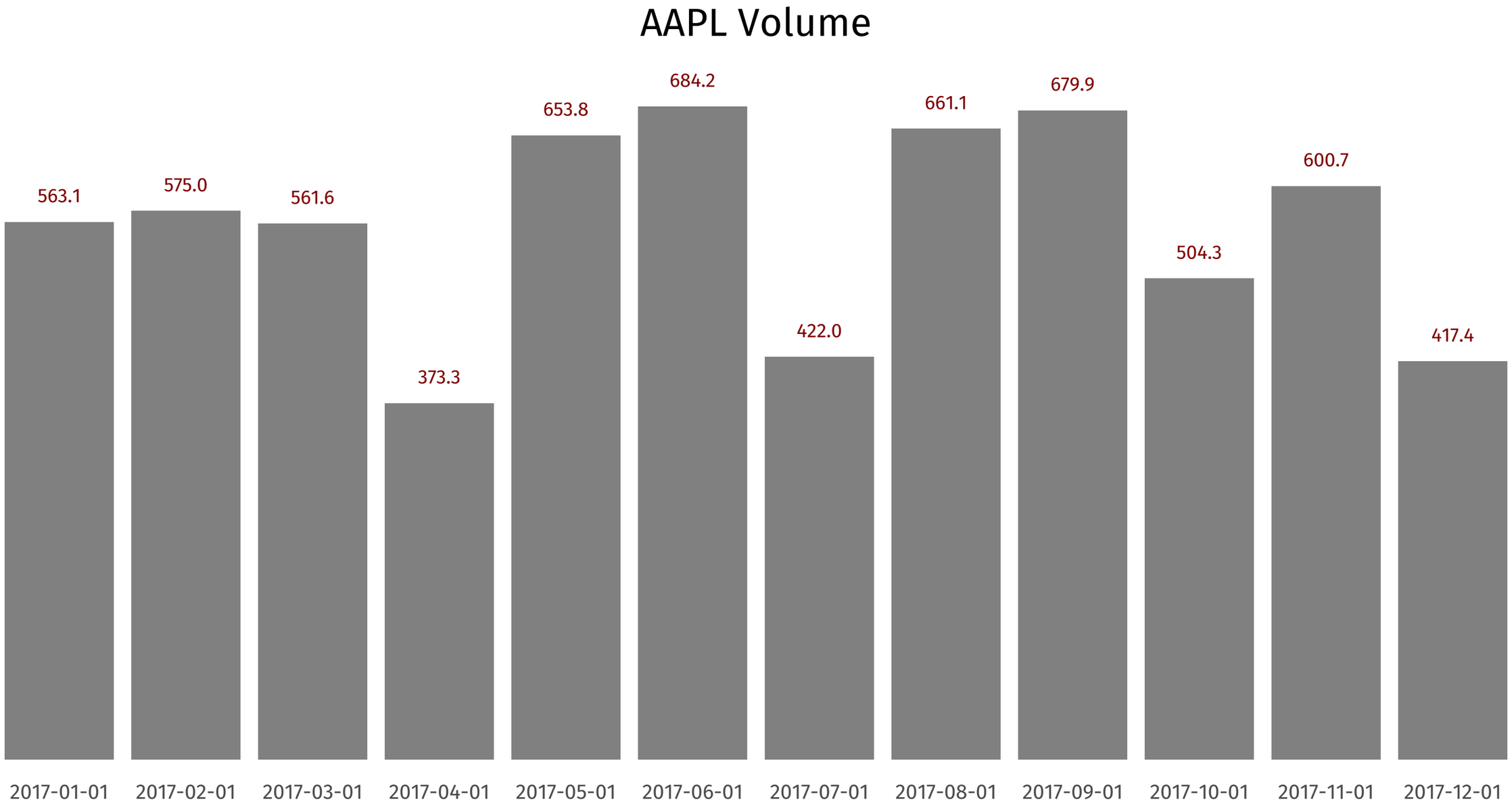

```
dchart -left=20 -right=80 -top=75 -yaxis -xlabel=2 -val -grid AAPL.d
```


AAPL Volume
563.1
575.0
561.6
373.3
653.8
684.2
422.0
661.1
679.9
504.3
600.7
417.4
2017-01-01
2017-02-01
2017-03-01
2017-04-01
2017-05-01
2017-06-01
2017-07-01
2017-08-01
2017-09-01
2017-10-01
2017-11-01
2017-12-01

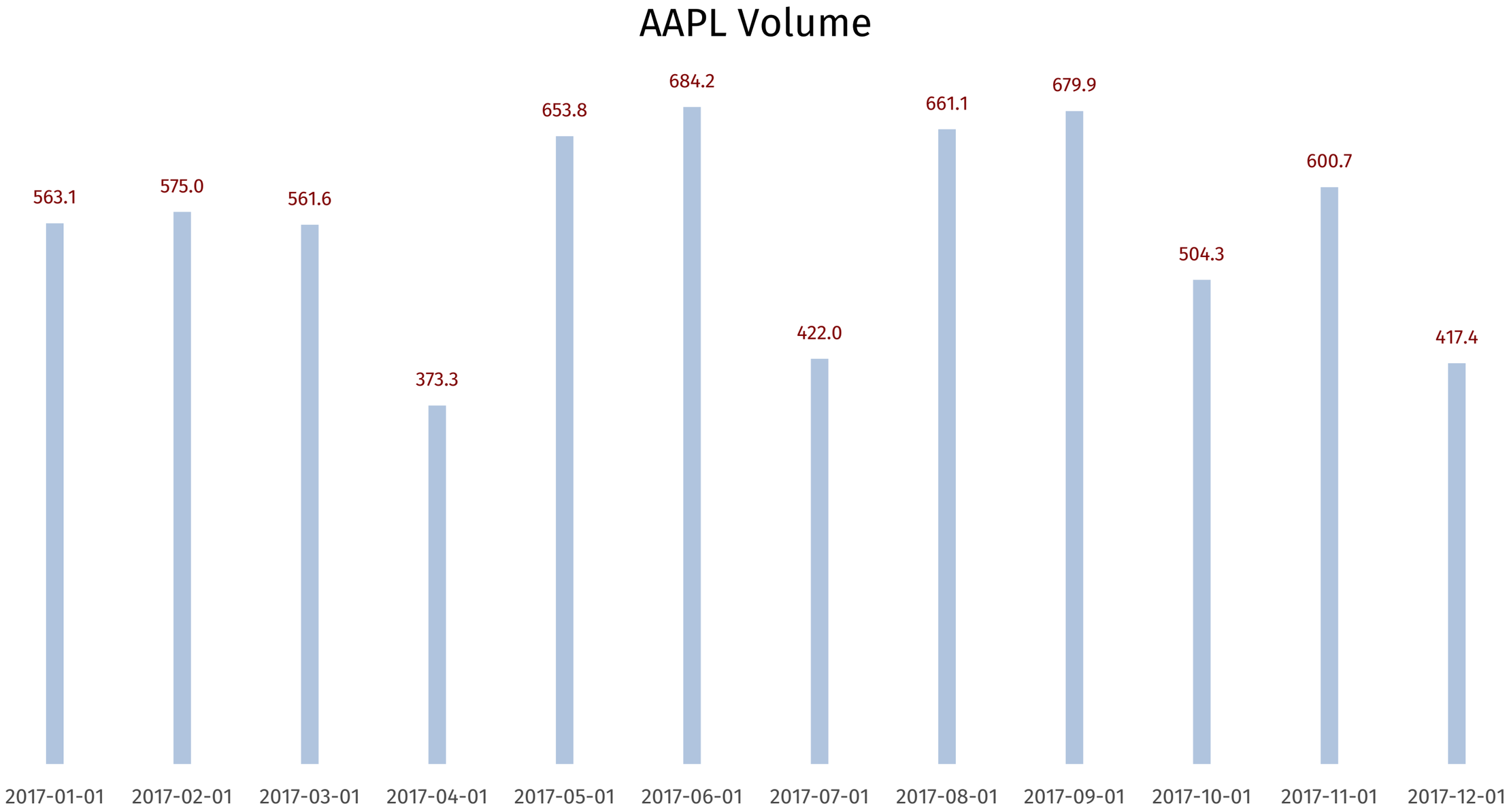
# bar


AAPL Volume
563.1
575.0
561.6
373.3
653.8
684.2
422.0
661.1
679.9
504.3
600.7
417.4
2017-01-01
2017-03-01
2017-05-01
2017-07-01
2017-09-01
2017-11-01

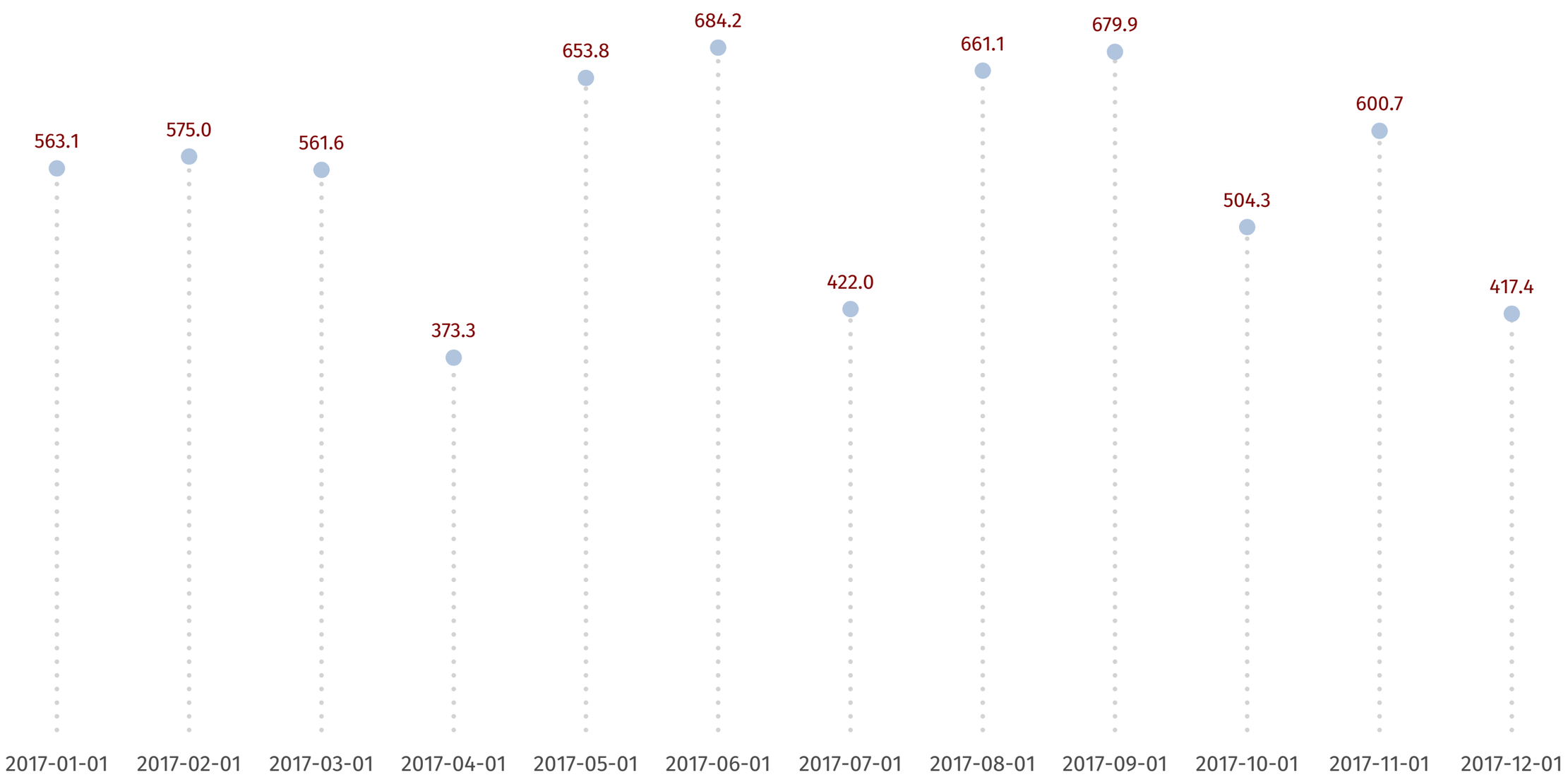
# scaled bar



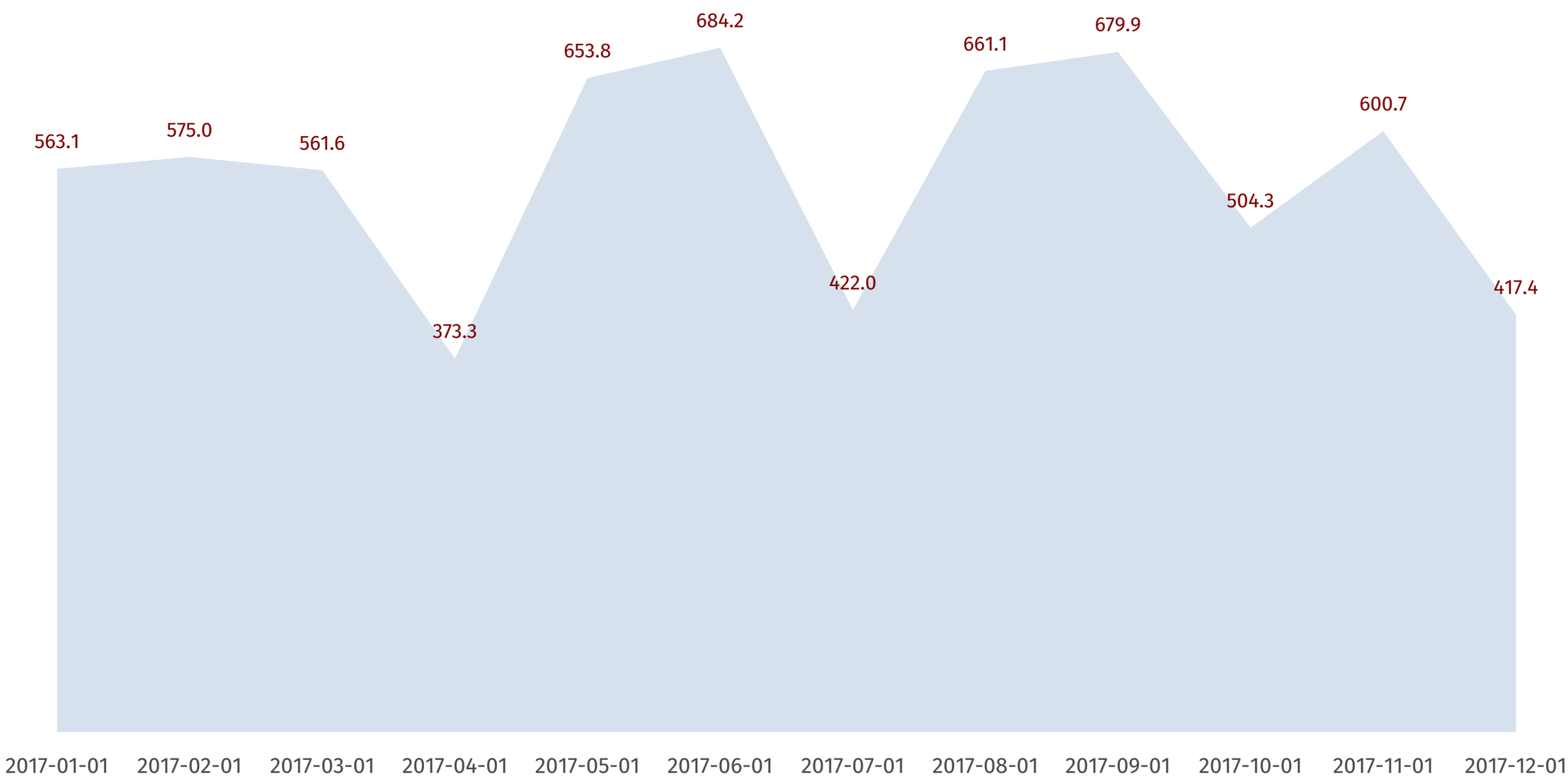

color change



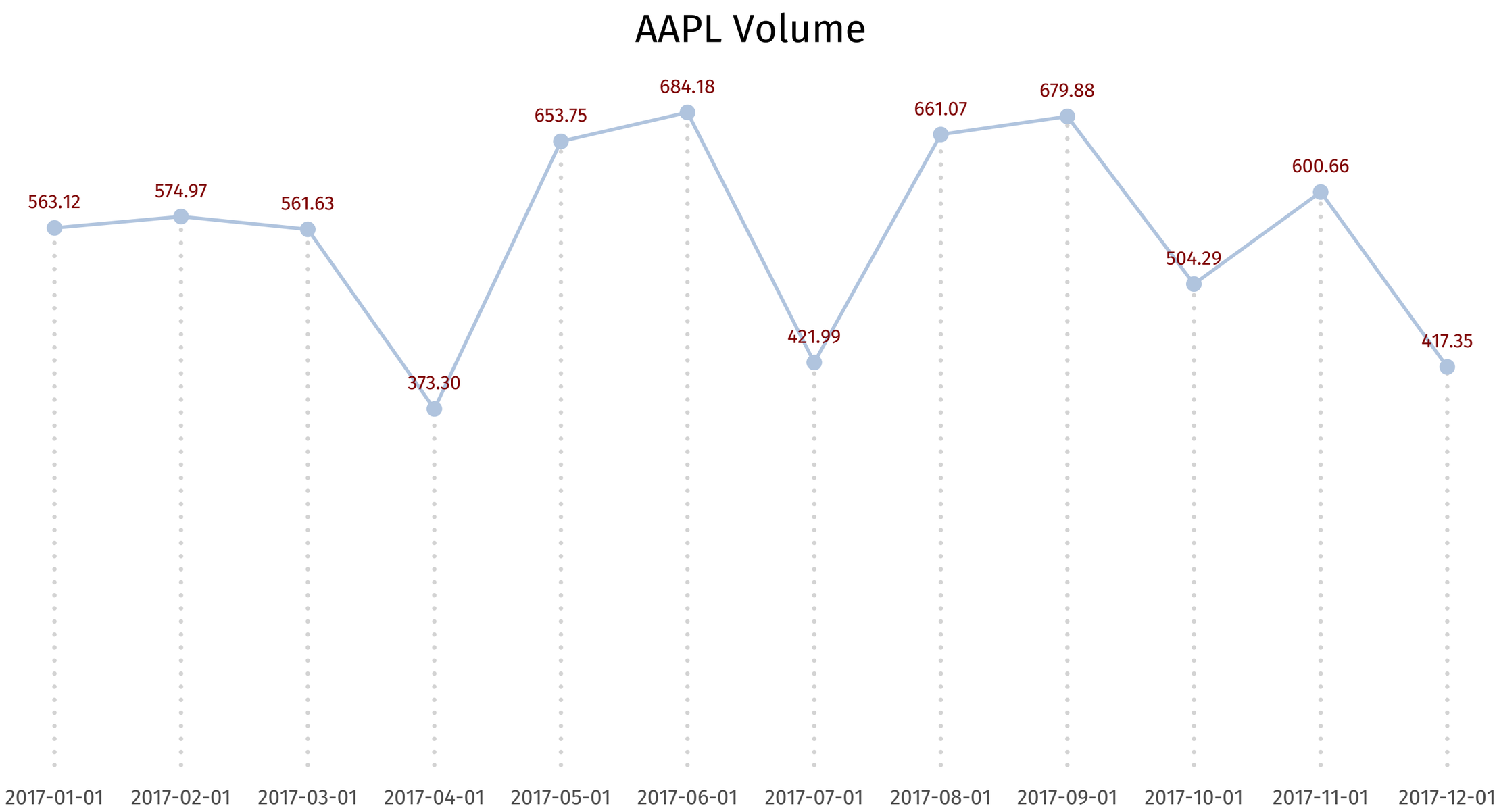

# barwith change


AAPL Volume
563.1
575.0
561.6
373.3
653.8
684.2
422.0
661.1
679.9
504.3
600.7
417.4
2017-01-01
2017-02-01
2017-03-01
2017-04-01
2017-05-01
2017-06-01
2017-07-01
2017-08-01
2017-09-01
2017-10-01
2017-11-01
2017-12-01

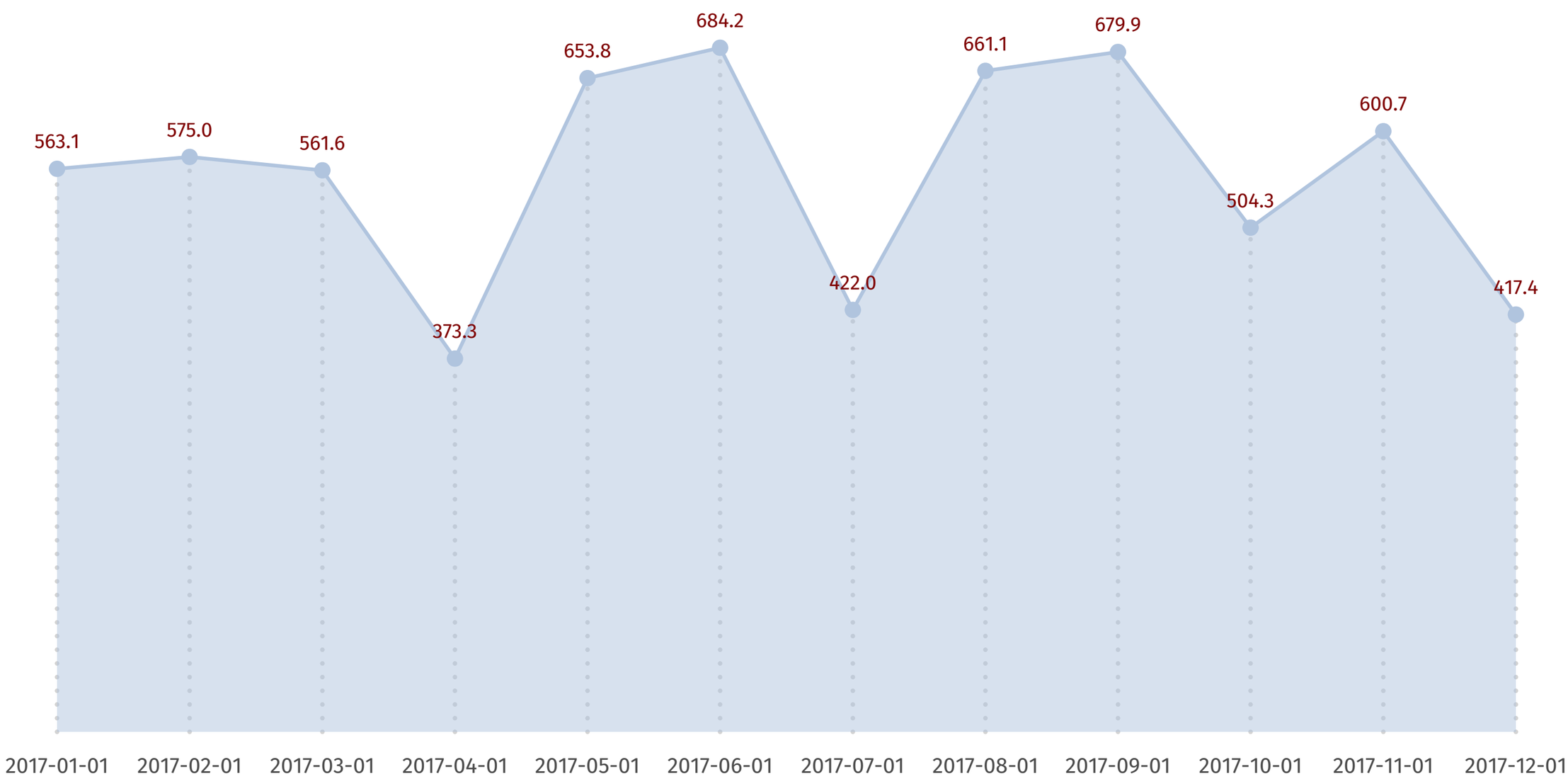
# dot


AAPL Volume
563.1
575.0
561.6
373.3
653.8
684.2
422.0
661.1
679.9
504.3
600.7
417.4
2017-01-01
2017-02-01
2017-03-01
2017-04-01
2017-05-01
2017-06-01
2017-07-01
2017-08-01
2017-09-01
2017-10-01
2017-11-01
2017-12-01

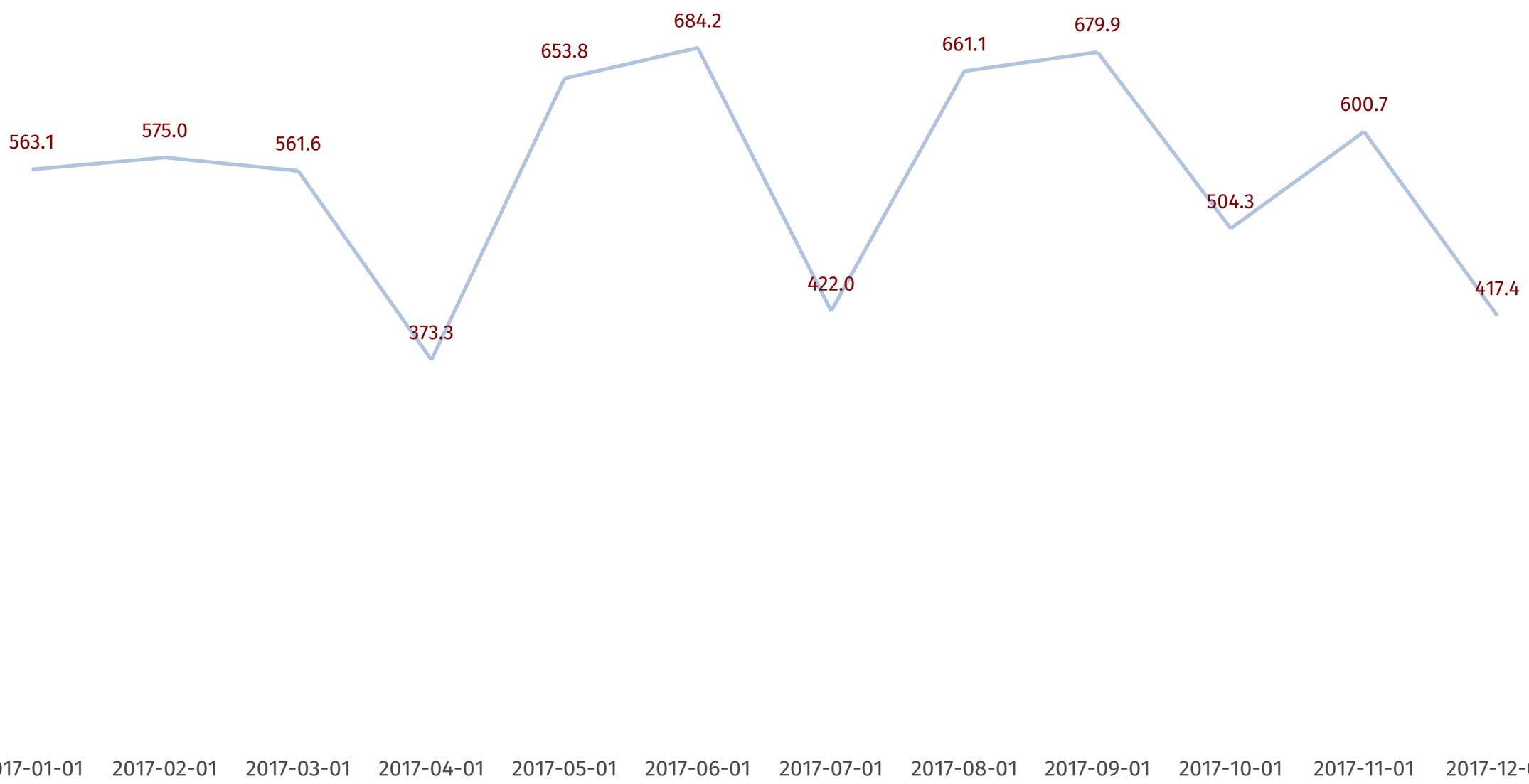
volume




# dot + line + value

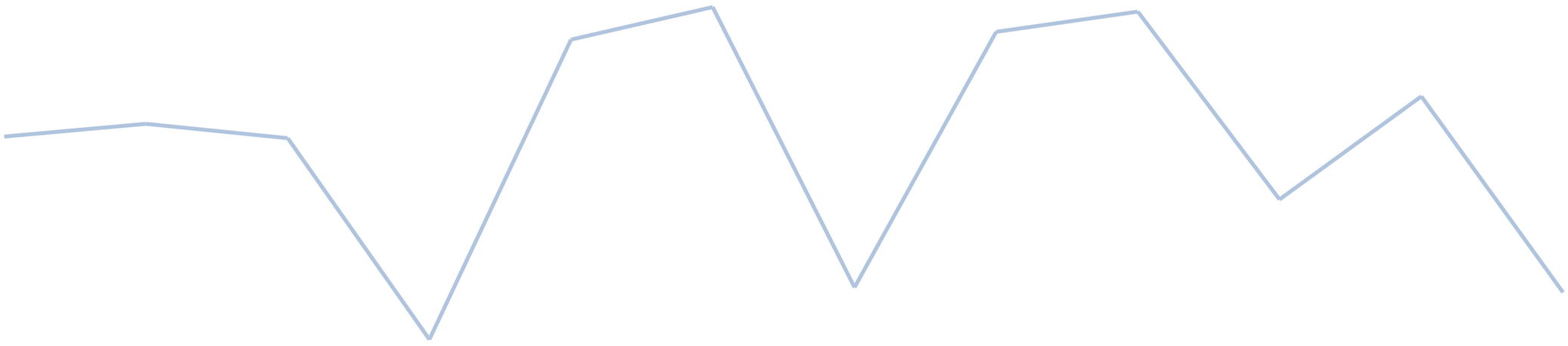



# dot + line + value + volume


AAPL Volume
563.1
575.0
561.6
373.3
653.8
684.2
422.0
661.1
679.9
504.3
600.7
417.4
2017-01-01
2017-02-01
2017-03-01
2017-04-01
2017-05-01
2017-06-01
2017-07-01
2017-08-01
2017-09-01
2017-10-01
2017-11-01
2017-12-01

# line + value

AAPL Volume



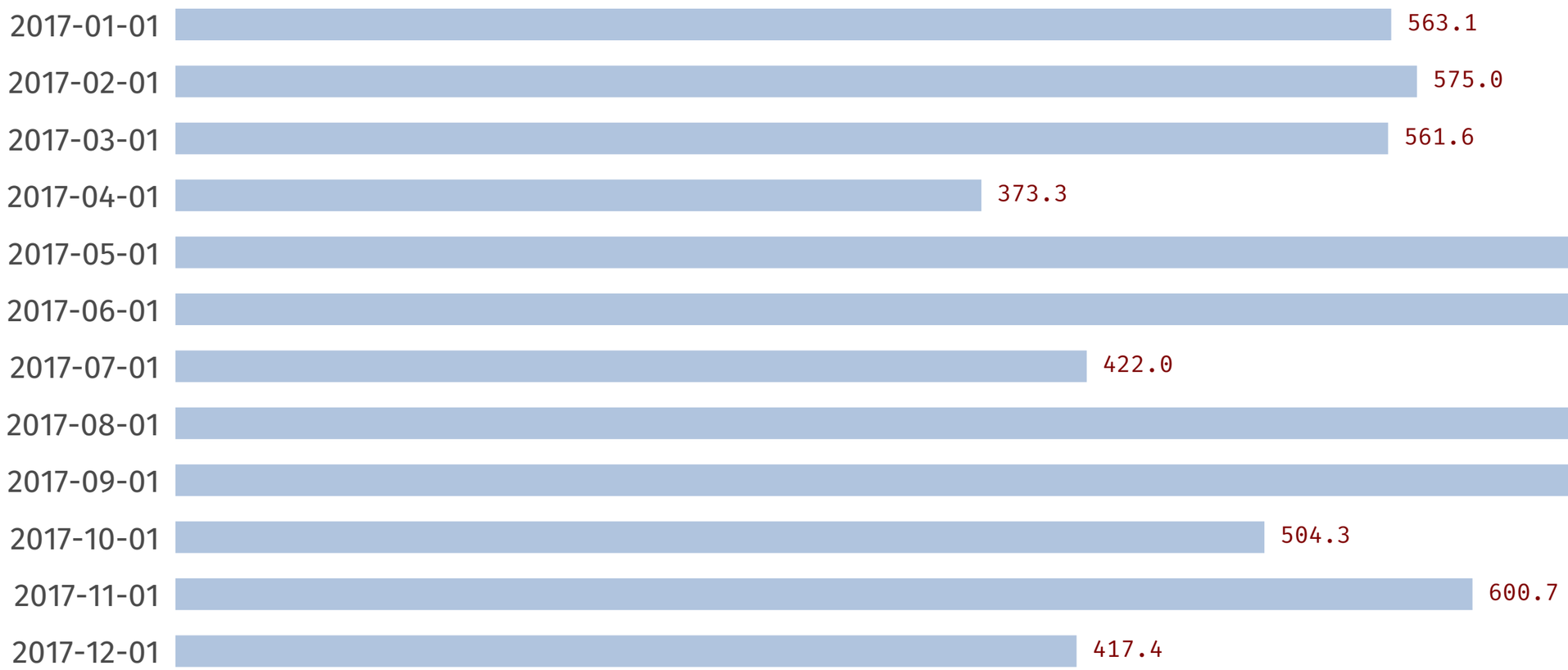
2017-01-01 2017-02-01 2017-03-01 2017-04-01 2017-05-01 2017-06-01 2017-07-01 2017-08-01 2017-09-01 2017-10-01 2017-11-01 2017-12-01

line only

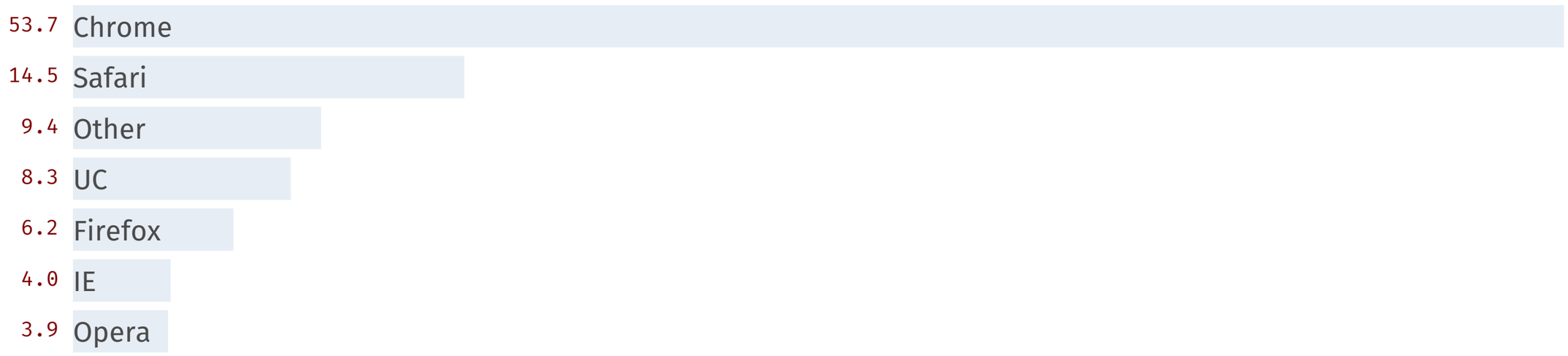
AAPL Volume

2017-01-01 2017-02-01 2017-03-01 2017-04-01 2017-05-01 2017-06-01 2017-07-01 2017-08-01 2017-09-01 2017-10-01 2017-11-01 2017-12-01

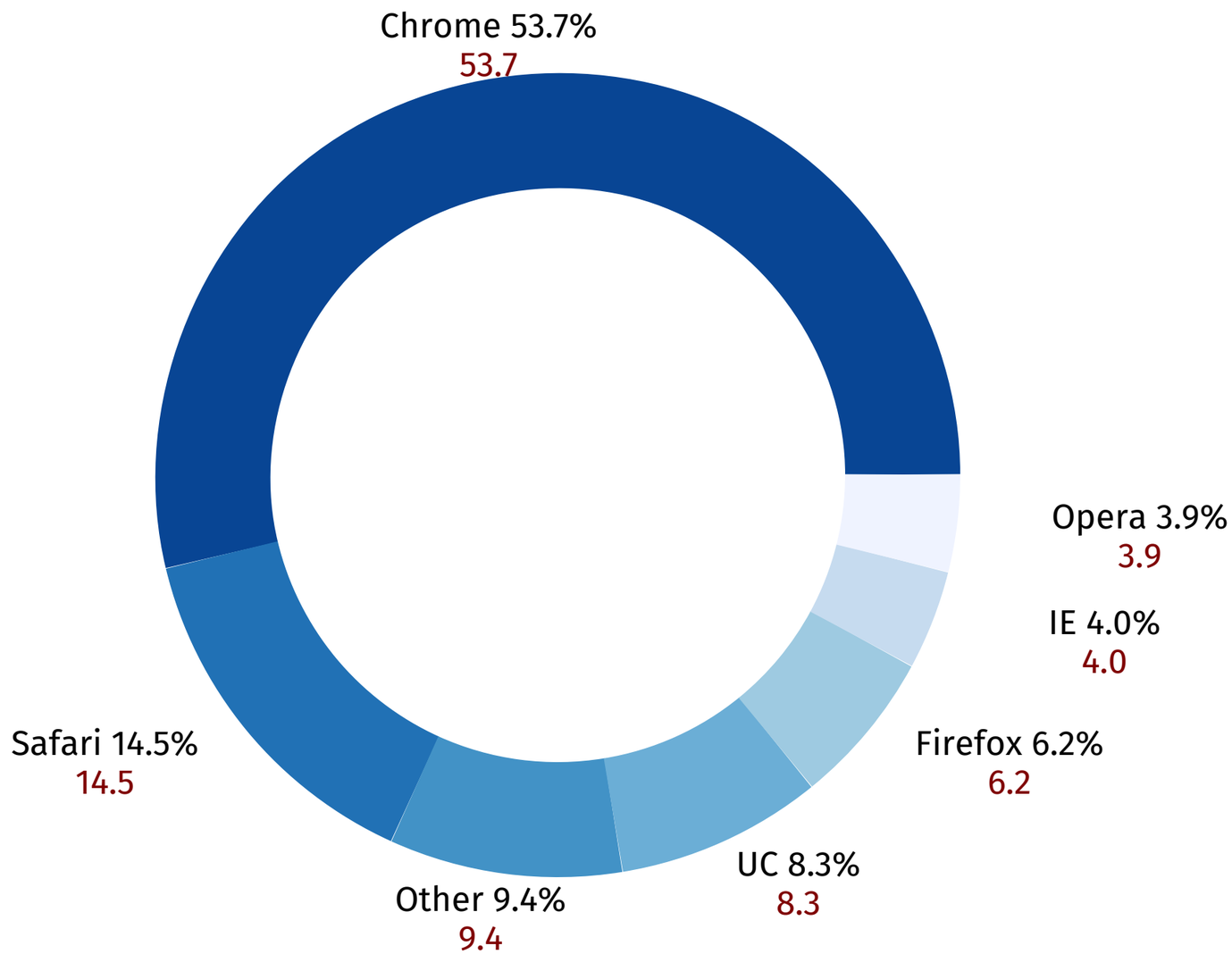
scatter



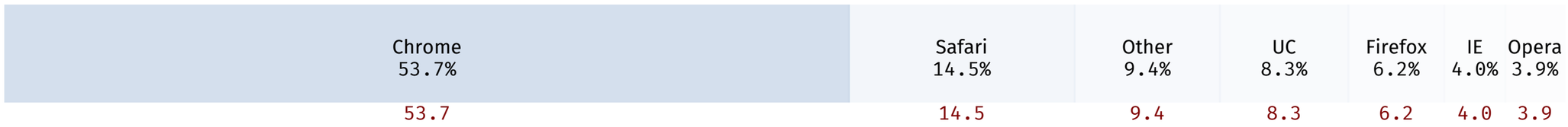

# horizontal bar


Browser Market Share Dec 2016-Dec 2017
53.7
Chrome
14.5
Safari
9.4
Other
8.3
UC
6.2
Firefox
4.0
IE
3.9
Opera

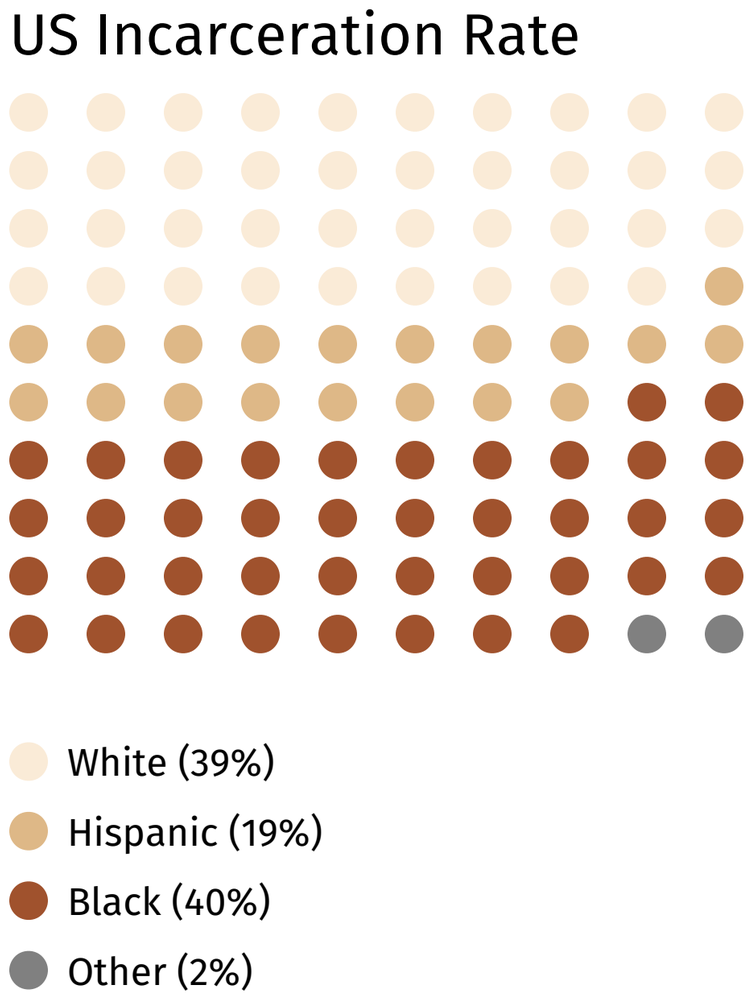
word bar


Chrome 53.7%
53.7
Opera 3.9%
3.9
IE 4.0%
4.0
Firefox 6.2%
6.2
UC 8.3%
8.3
Other 9.4%
9.4
Safari 14.5%
14.5

donut


Browser Market Share Dec 2016-Dec 2017
Chrome
53.7%
Safari
14.5%
Other
9.4%
UC
8.3%
Firefox
6.2%
IE
4.0%
Opera
3.9%
53.7
14.5
9.4
8.3
6.2
4.0
3.9

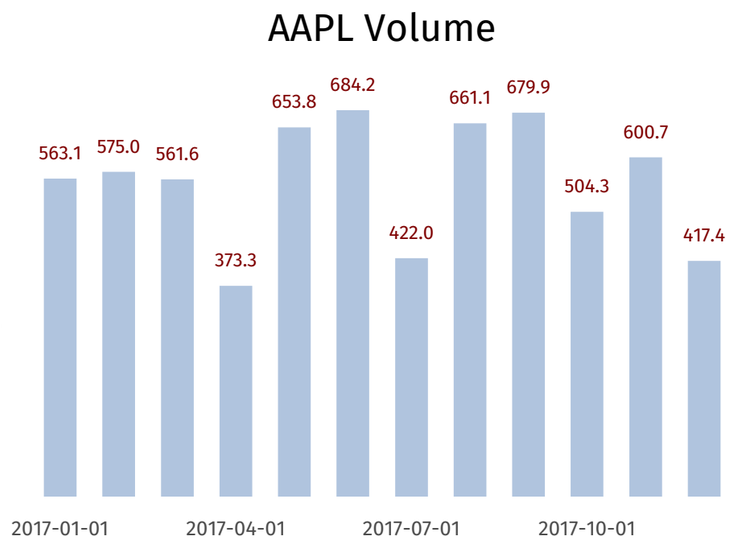
pmap




# pgrid

```
# AAPL Volume
2017-01-01    563.122
2017-02-01    574.969
2017-03-01    561.628
2017-04-01    373.304
2017-05-01    653.755
2017-06-01    684.178
2017-07-01    421.992
2017-08-01    661.069
2017-09-01    679.879
2017-10-01    504.291
2017-11-01    600.663
2017-12-01    417.354
```

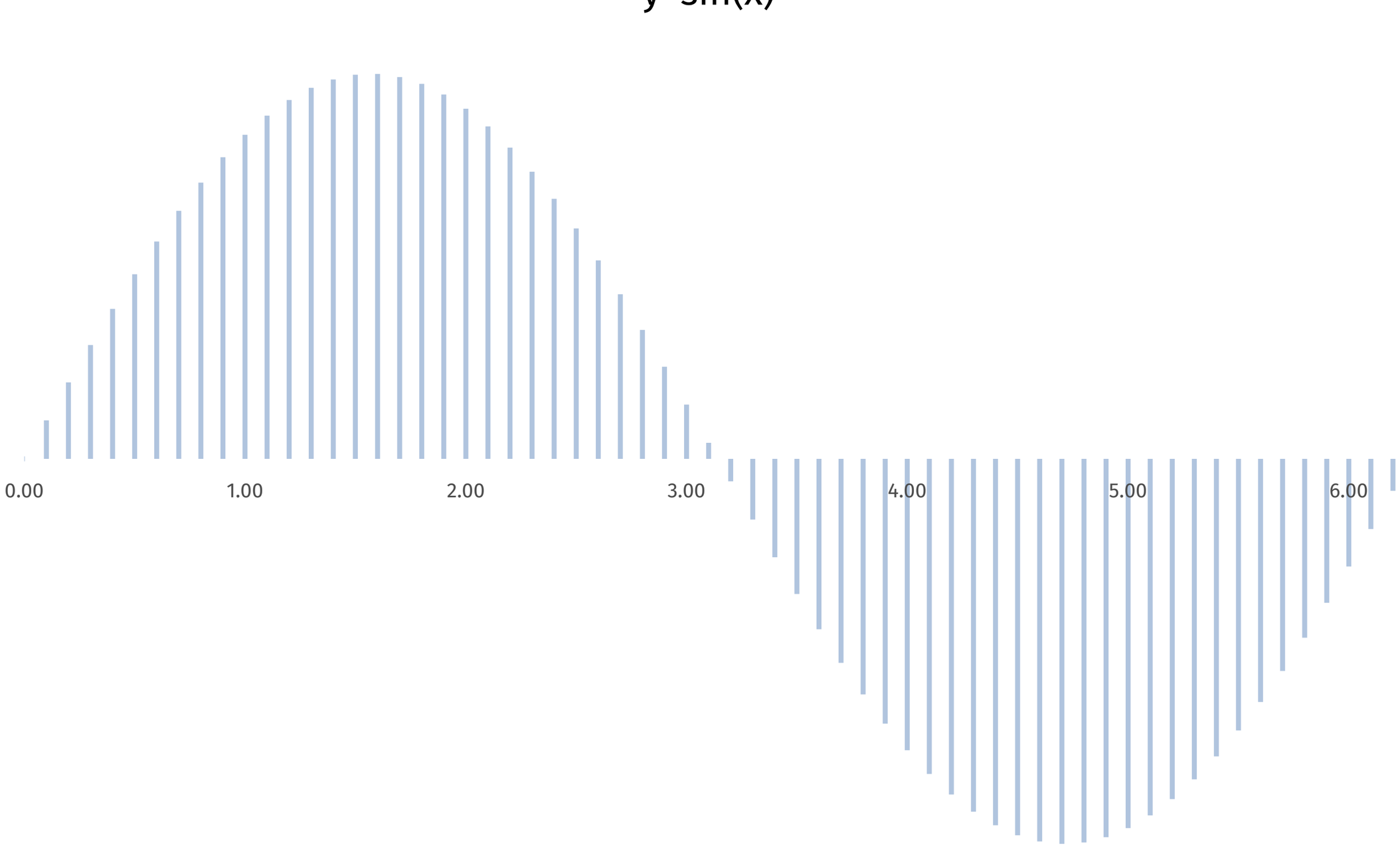
```
<deck>
    <slide>
        <text ...>AAPL Volume</text>
        <line ... color="lightsteelblue"/>
        <text ... color="rgb(127,0,0)">563.1</text>
        <text ... color="rgb(75,75,75)">2017-01-01</text>
    </slide>
</deck>
```



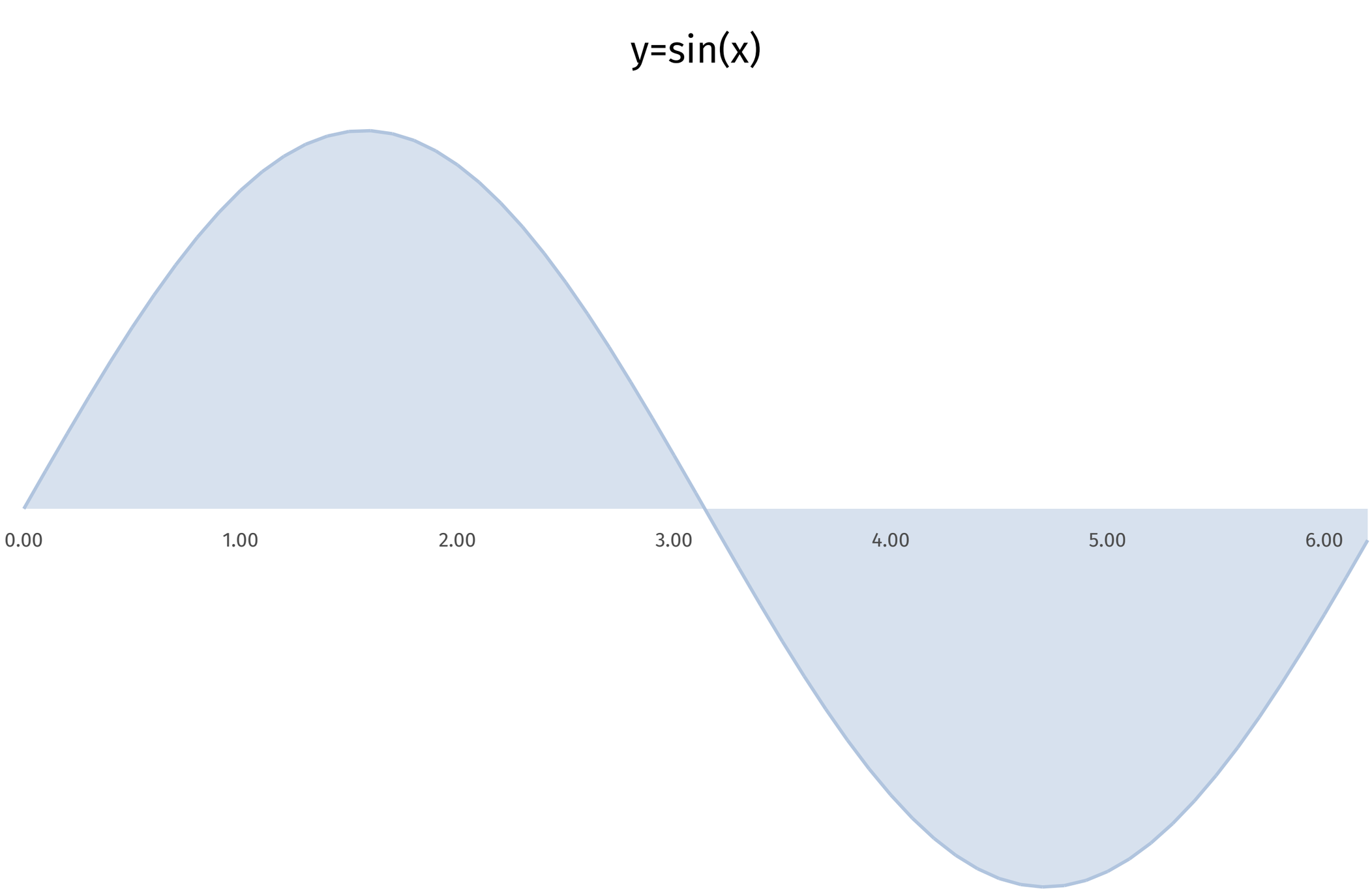

# data | dchart | pdf

```go
package main

import (
    "fmt"
    "math"
)

func main() {
    fmt.Println("# y=sin(x)")
    for x := 0.0; x < math.Pi*2; x += 0.1 {
        fmt.Printf("%.2f\t%.4f\n", x, math.Sin(x))
    }
}
```

```
# y=sin(x)
0.00    0.0000
0.10    0.0998
0.20    0.1987
0.30    0.2955
0.40    0.3894
0.50    0.4794
0.60    0.5646
0.70    0.6442
0.80    0.7174
...
5.80    -0.4646
5.90    -0.3739
6.00    -0.2794
6.10    -0.1822
6.20    -0.0831
```




```
sine | dchart -val=f -bottom=50 -xlabel=10
```

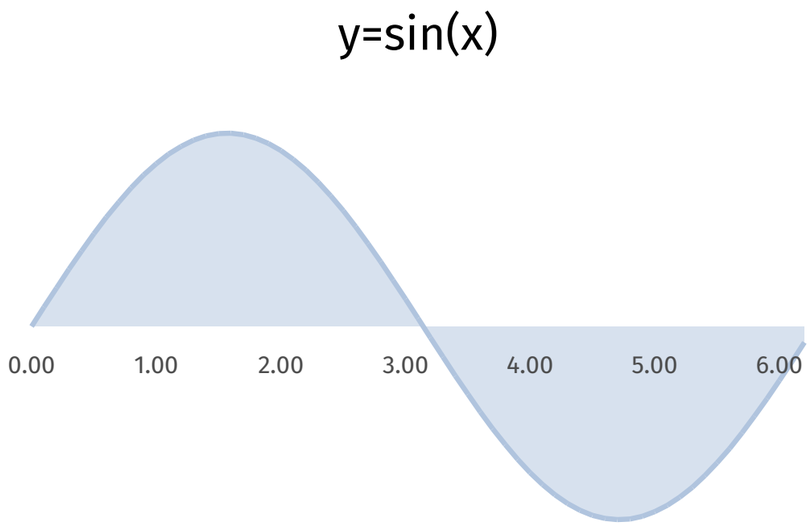



```
sine | dchart -val=f -bottom=50 -xlabel=10 -bar=f -line -vol
```

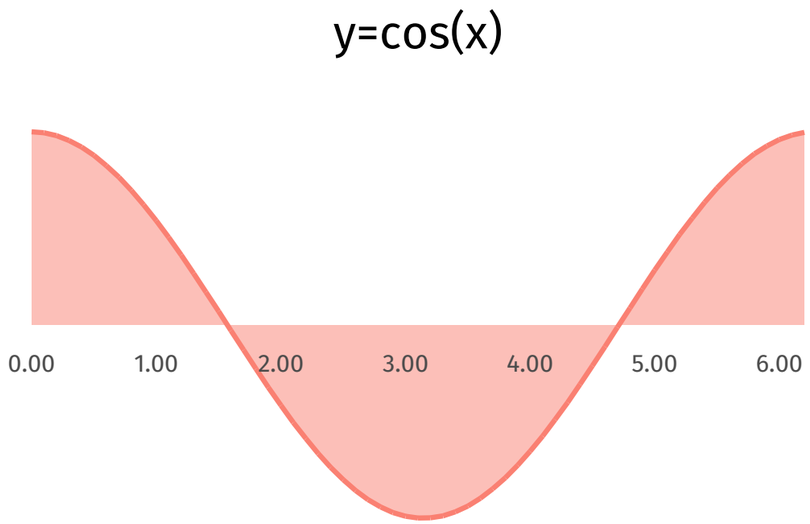
# mfunc.go

```
package main

import (
    "flag"
    "fmt"
    "math"
)

type tfunc struct {
    label    string
    function func(float64) float64
}

func power(a, b int) int {
    p := 1
    for b > 0 {
        if b&1 != 0 {
            p *= a
        }
        b >>= 1
        a *= a
    }
    return p
}

func pow2(n float64) float64 {
    return float64(power(2, int(n)))
}

func linear(x float64) float64 {
    return x
}

func squared(x float64) float64 {
    return x*x
}

func main() {
    fname := flag.String("f", "sine", "function name")
    xrange := flag.String("x", "0,6.283185,0.1", "x range")
    flag.Parse()

    var (
        f     tfunc
        xmin  = 0.0
        xmax  = 2 * math.Pi
        xstep = 0.1
    )
    fmt.Sscanf(*xrange, "%f,%f,%f", &xmin, &xmax, &xstep)
    switch *fname {
    case "square":
        f = tfunc{"y=x*x", squared}
```

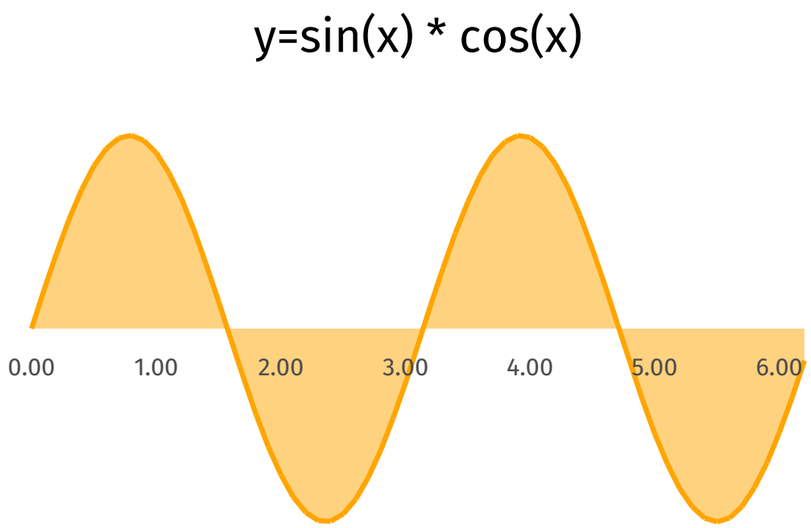
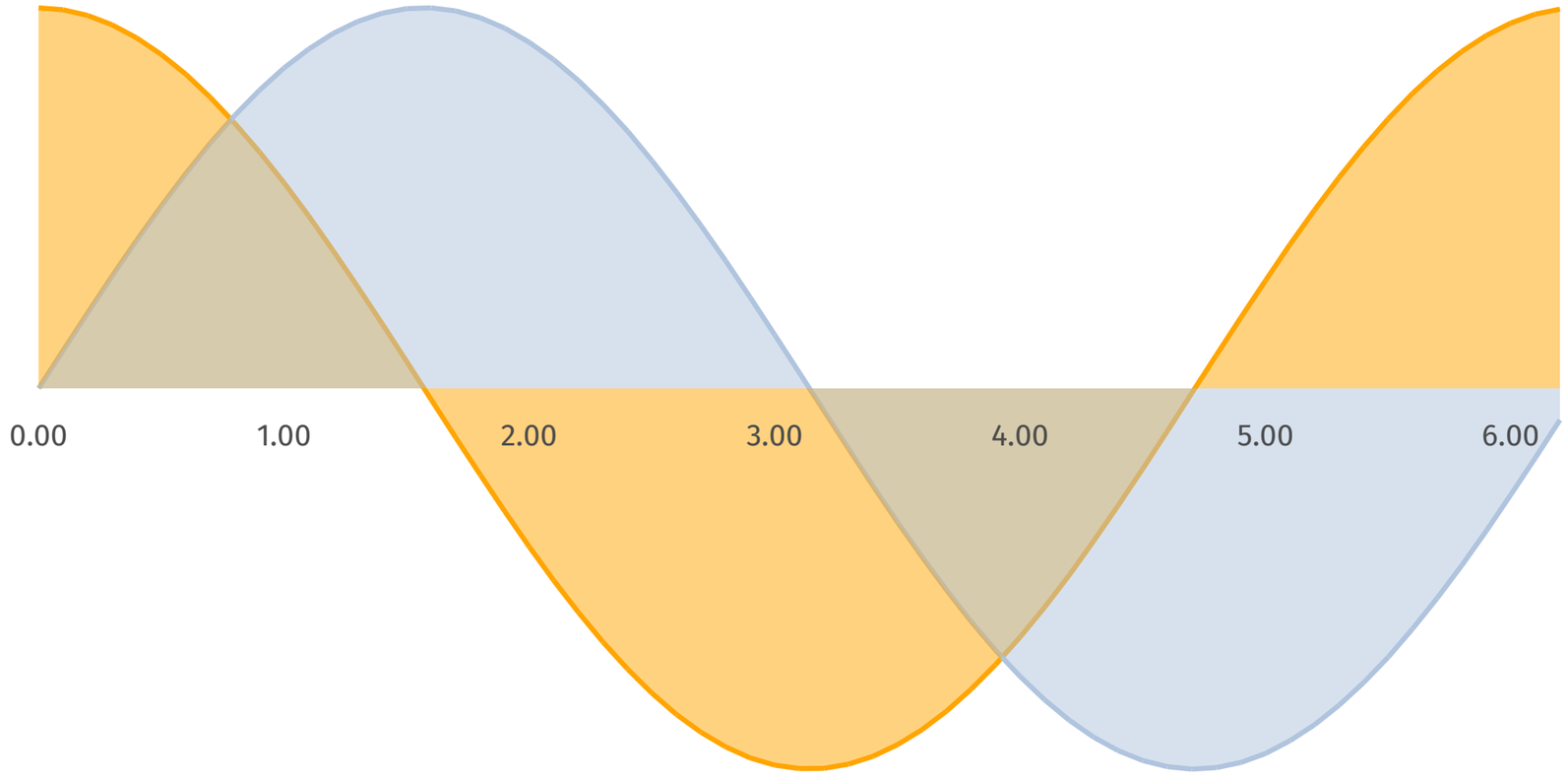
```
mfunc -f sine | dchart ... -top=90 -bottom=80
```




```
mfunc -f cosine | dchart ... -top=60 -bottom=50
```




```
mfunc -f sincos | dchart ... -top=30 -bottom=20
```



```
#!/bin/sh
mopts="-fulldeck=f -bar=f -vol -top=90 -bottom 60 -left=20 -right=8
mfunc -f sine   | dchart $mopts -xlabel=10
mfunc -f cosine | dchart $mopts -xlabel=0 -color=orange
```