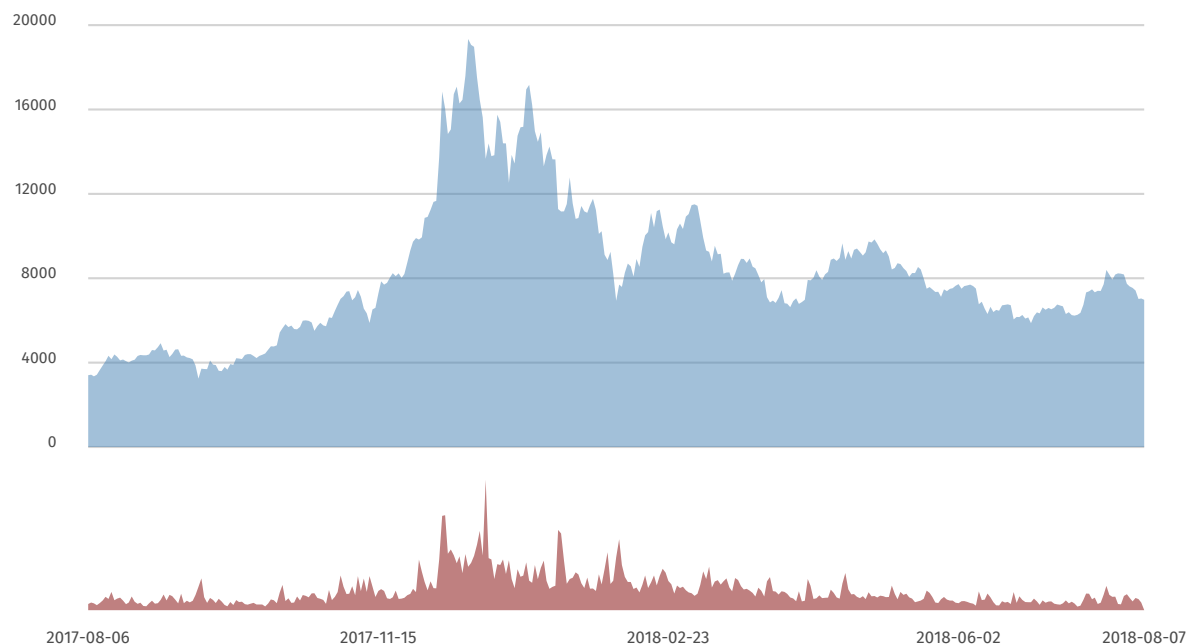
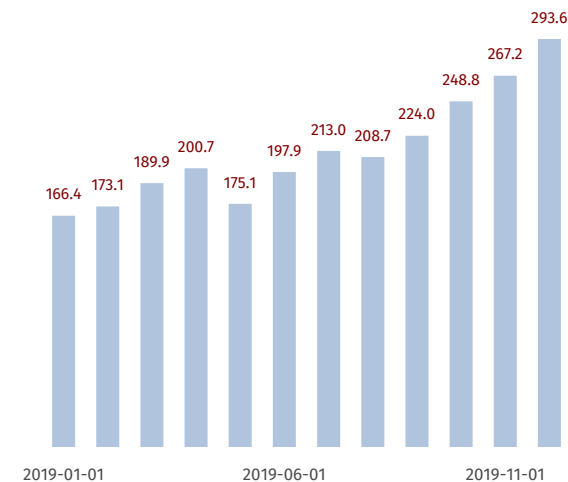


dchart: charts for deck/decksh

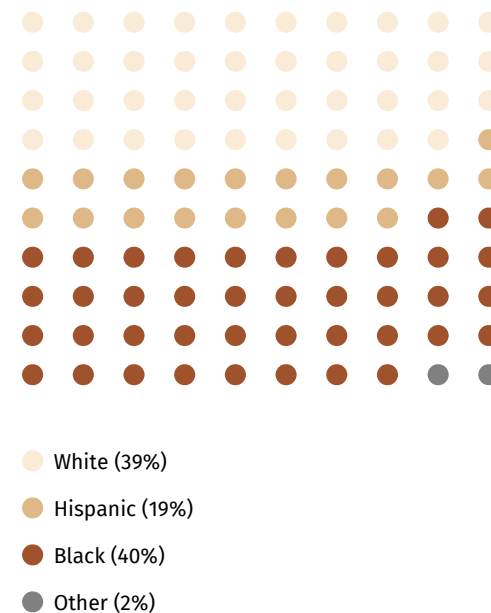
Bitcoin to USD



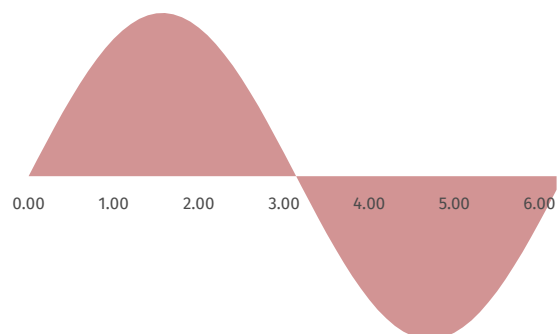
AAPL Closing Price



US Incarceration Rate



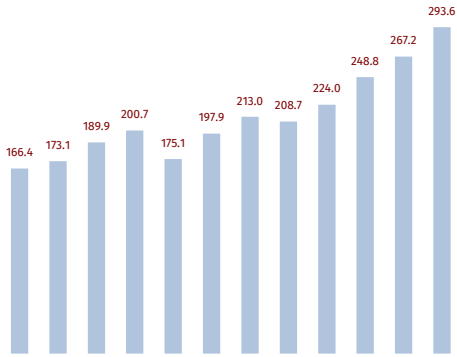
$y=\sin(x)$



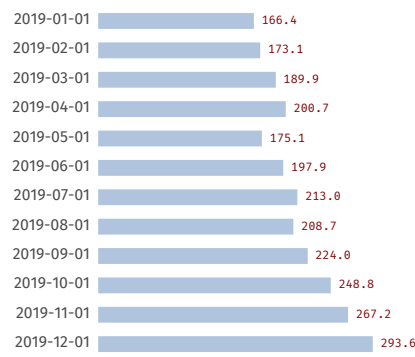
Browser Market Share Dec 2016-Dec 2017



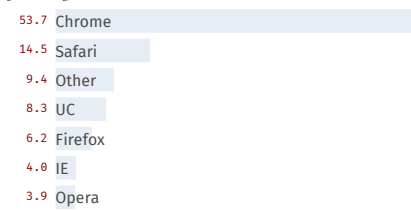
Chart Types



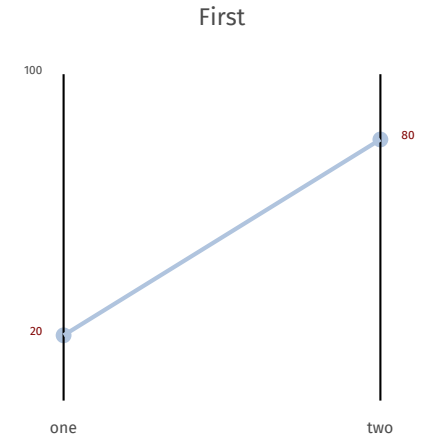
Column



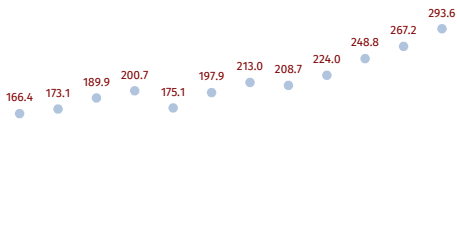
Bar



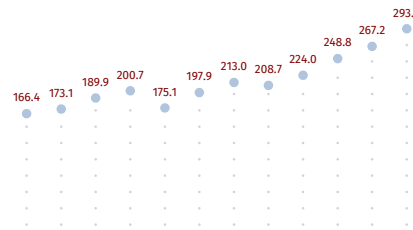
Word Bar



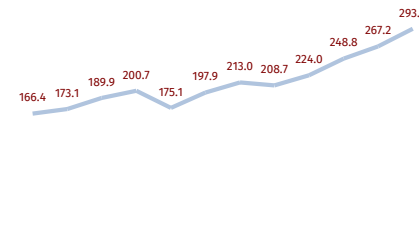
Slope



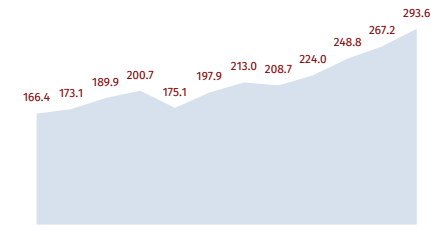
Scatter



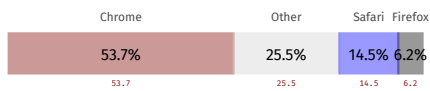
Dot



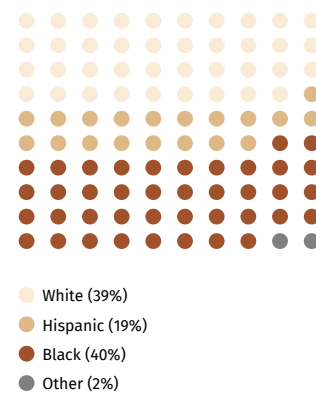
Line



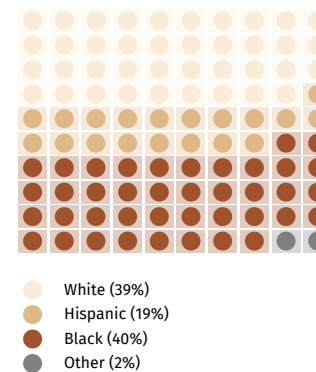
Area



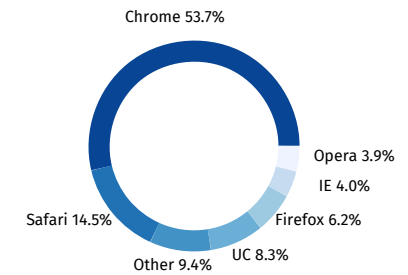
Proportional Map



Proportional Grid

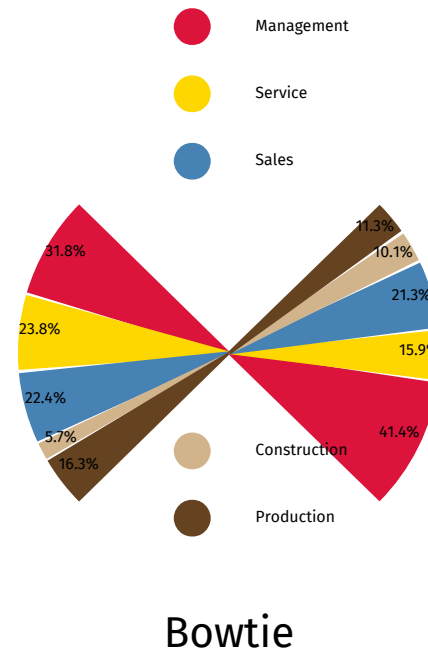
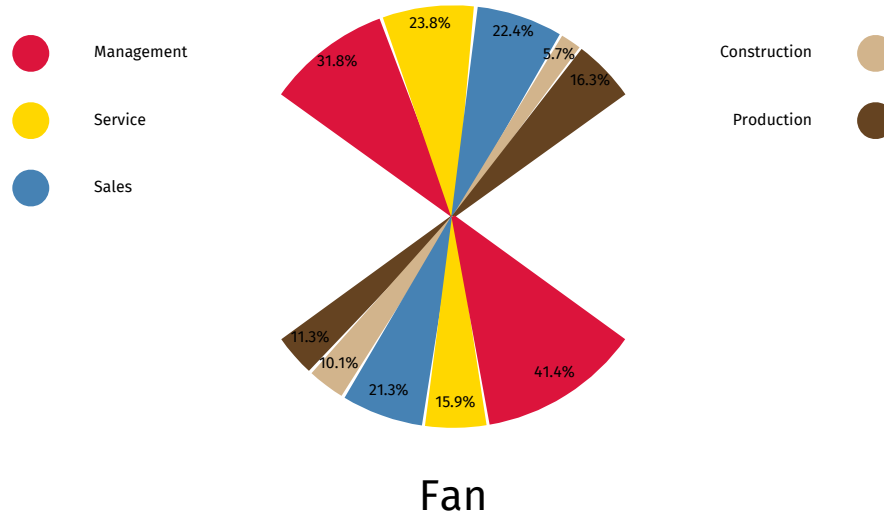
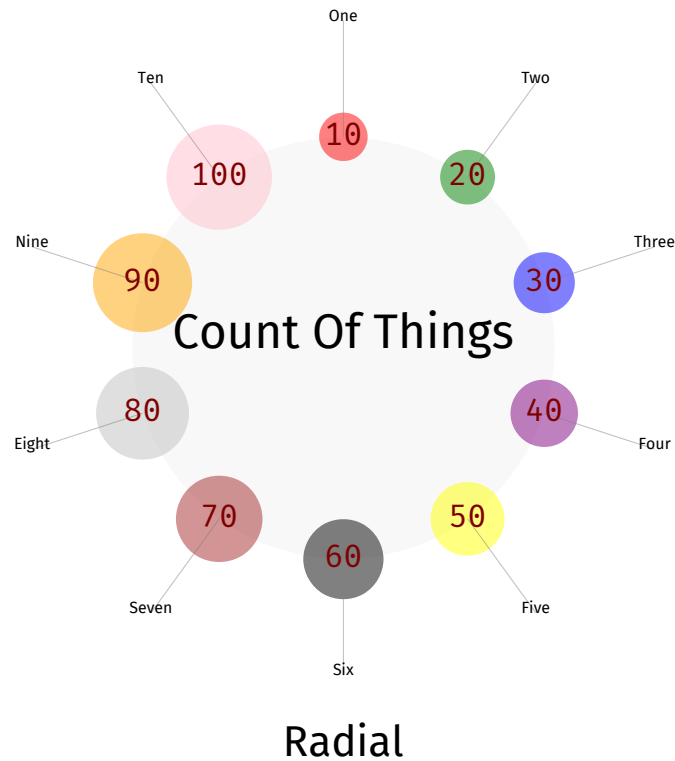


Lego



Donut/Pie

Chart Types (continued)



Data

Tab-Separated

X Label	Y Value
# AAPL Closing Price	
2019-01-01	166.440002
2019-02-01	173.149994
2019-03-01	189.949997
2019-04-01	200.669998
2019-05-01	175.070007
2019-06-01	197.919998
2019-07-01	213.039993
2019-08-01	208.740005
2019-09-01	223.970001
2019-10-01	248.759995
2019-11-01	267.250000
2019-12-01	293.649994

Comma-Separated (CSV)

Columns (pick two)
Date,Open,High,Low,Close,Adj Close,Volume
2019-01-01,154.889999,169.000000,142.000000,166.440002,163.587997,828087400
2019-02-01,166.960007,175.869995,165.929993,173.149994,170.183029,472540600
2019-03-01,174.279999,197.690002,169.500000,189.949997,187.495865,650981400
2019-04-01,191.639999,208.479996,188.380005,200.669998,198.077362,506117700
2019-05-01,209.880005,215.309998,174.990005,175.070007,172.808105,739456600
2019-06-01,175.600006,201.570007,170.270004,197.919998,196.115219,515187300
2019-07-01,203.169998,221.369995,198.410004,213.039993,211.097366,473957000
2019-08-01,213.899994,218.029999,192.580002,208.740005,206.836563,681074600
2019-09-01,206.429993,226.419998,204.220001,223.970001,222.770889,542567100
2019-10-01,225.070007,249.750000,215.130005,248.759995,247.428162,608302700
2019-11-01,249.539993,268.000000,249.160004,267.250000,265.819183,448331500
2019-12-01,267.269989,293.970001,256.290009,293.649994,292.954712,597198700

Data to Chart

Data

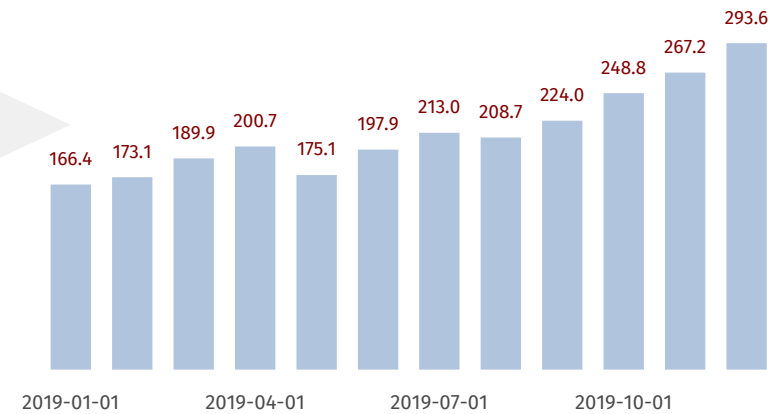
```
# AAPL Closing Price
2019-01-01 166.440002
2019-02-01 173.149994
2019-03-01 189.949997
2019-04-01 200.669998
2019-05-01 175.070007
2019-06-01 197.919998
2019-07-01 213.039993
2019-08-01 208.740005
2019-09-01 223.970001
2019-10-01 248.759995
2019-11-01 267.250000
2019-12-01 293.649994
```

Markup

```
<deck>
  <canvas width="0" height="0" />
  <slide bg="white">
    <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>
```

PDF Rendition

AAPL Closing Price



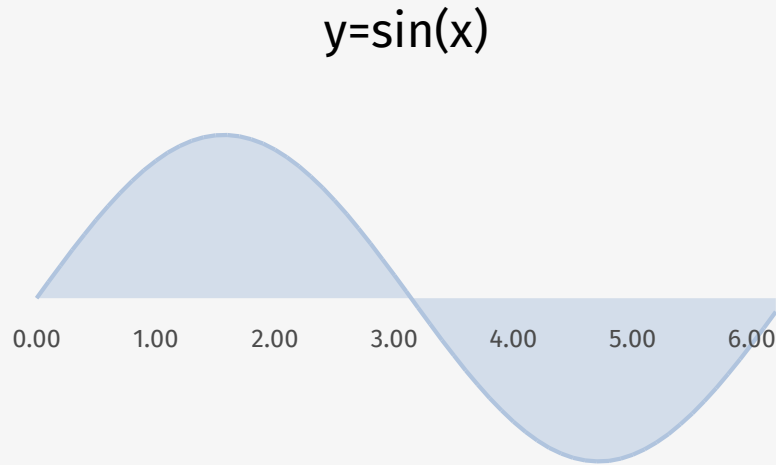
dchart AAPL.d | pdf

Generating data for charts

```
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

```
go run sine.go |  
dchart -bar=f -val=f -xlabel=10 -line -vol -bottom=50 |  
pdfdeck -stdout - > sine.pdf
```

dchart API

```
NewChart(charttype string, top, bottom, left, right float64) => settings
```

```
settings.[thing] = ...
```

```
settings.GenerateChart(deck *generate.Deck, io.ReadCloser)
```

Chart Data	<code>[]ChartData</code>
------------	--------------------------

Chart Settings	<code>Settings</code>
----------------	-----------------------

Read CSV or TSV	<code>Getdata(r io.ReadCloser, readcsv bool, cols string) ([]ChartData, float64, float64, string)</code>
-----------------	--

Read TSV	<code>TSVdata(r io.ReadCloser) ([]ChartData, float64, float64, string)</code>
----------	---

Read CSV	<code>CSVdata(r io.ReadCloser, csvcols string) ([]ChartData, float64, float64, string)</code>
----------	---

Define a Chart	<code>NewChart(chartType string, top, bottom, left, right float64) Settings</code>
----------------	--

Define Standalone	<code>NewFullChart(chartType string, top, bottom, left, right float64) Settings</code>
-------------------	--

Make Chart	<code>(s *Settings) GenerateChart(deck *generate.Deck, r io.ReadCloser)</code>
------------	--

Write the Chart	<code>(s *Settings) Write(w io.Writer, r io.ReadCloser)</code>
-----------------	--

Example Program

```
package main

import (
    "fmt"
    "os"

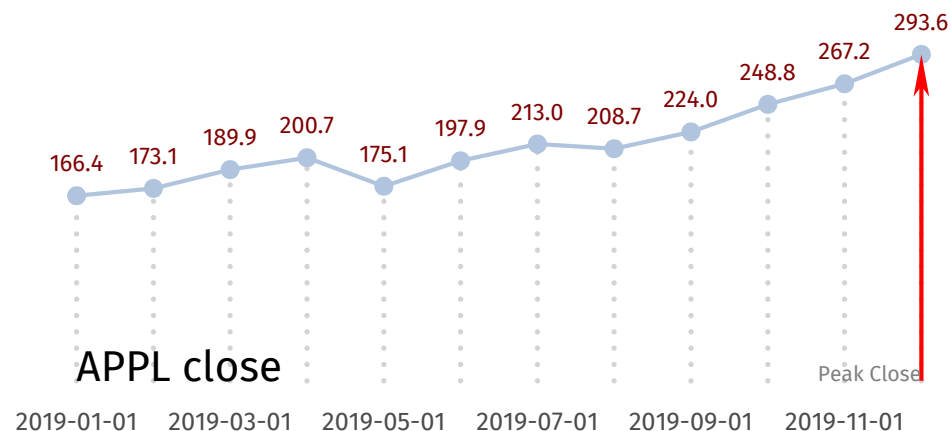
    "github.com/ajstarks/dchart"
    "github.com/ajstarks/deck/generate"
)

func main() {
    chart := dchart.NewFullChart("bar", 0, 0, 0, 0)
    chart.ShowTitle = true
    chart.XLabelInterval = 2
    deck := generate.NewSlides(os.Stdout, 0, 0)
    deck.StartDeck()
    for _, f := range os.Args[1:] {
        r, err := os.Open(f)
        if err != nil {
            fmt.Fprintf(os.Stderr, "%v\n", err)
            continue
        }
        chart.GenerateChart(deck, r)
    }
    deck.EndDeck()
}
```

go run main.go | pdf



Using dchart with decksh

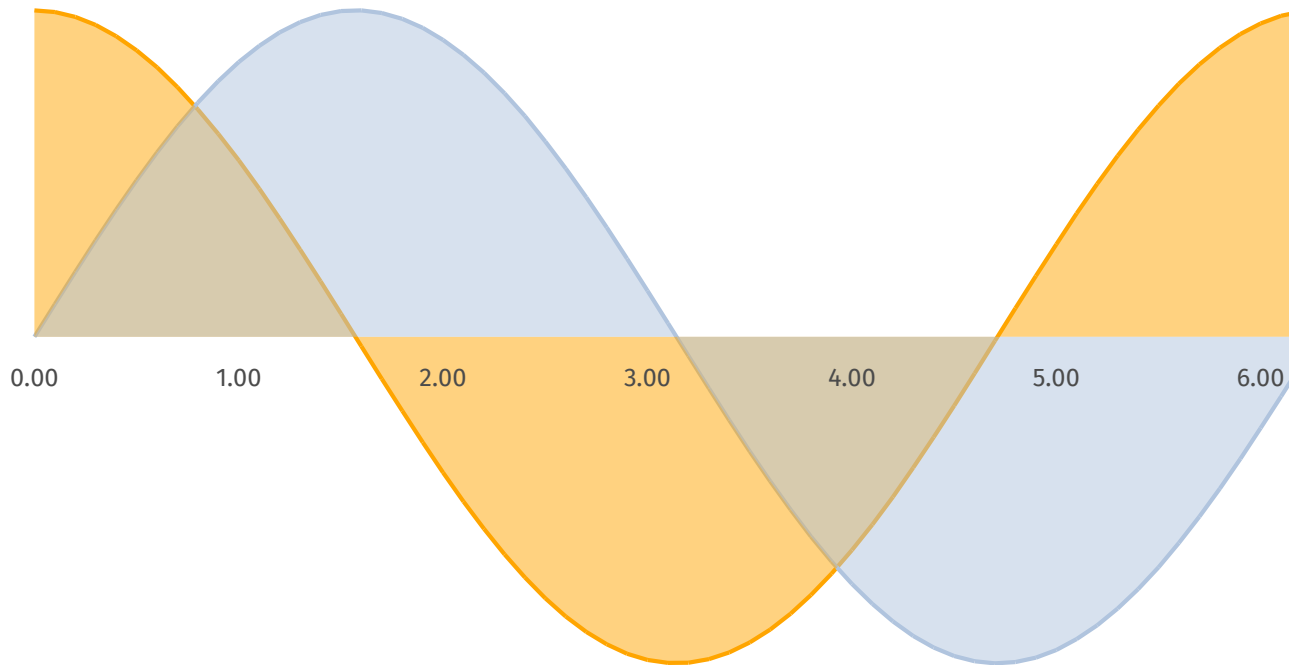


```

cw=40          // chart width
t=80           // top
b=t-20         // bottom
l1=5           // volume chart left
r1=l1+cw       // volume chart right
l2=r1+10       // close chart left
r2=l2+cw       // close chart right

opts="-fulldeck=f -xlabel=2 -title=f -bar=f"
copts="-dot -line -csv -csvcol Date,Close"
dchart opts -vol -top t -bottom b -left l1 -right r1 code/AAPL-vol.d
dchart opts copts -top t -bottom b -left l2 -right r2 code/AAPL.csv
text "APPL volume" l1 b 2
text "APPL close" l2 b 2
arrow r2 b r2 t 0.2 2 1 "red"
etext "Peak Close" r2 b 1 "sans" "gray"
```

Composite Charts



```
// go run mfunc.go -f cos > code/cos.d
// go run mfunc.go -f sin > code/sin.d
opts="-top=80 -bottom=60 -left=20 -right=80 -fulldeck=f -title=f -val=f -bar=f -line -vol"
dchart opts -xlabel=10 -color orange code/cos.d
dchart opts -xlabel=0 code/sin.d
```

Command Line Options

Chart Types

-bar	true	bar chart
-wbar	false	word bar chart
-hbar	false	horizontal bar chart
-donut	false	donut chart
-dot	false	dot chart
-lego	false	lego chart
-line	false	line chart
-pgrid	false	proportional grid
-pmap	false	proportional map
-bowtie	false	bowtie chart
-fan	false	fan chart
-radial	false	radial chart
-scatter	false	scatter chart
-slope	false	slope chart
-vol	false	volume (area) chart

Chart Elements

-csv	false	read CSV files
-frame	false	show a colored frame
-fulldeck	true	generate full deck markup
-grid	false	show gridlines on the y axis
-note	true	show annotations
-pct	false	show computed percentage
-rline	false	show a regression line
-solidpmap	false	show solid pmap colors
-spokes	false	show spokes in radial chart
-title	true	show the title
-val	true	show values
-xlast	false	show the last x label
-xstagger	false	stagger x axis labels
-yaxis	false	show a y axis
-chartitle	override title in data	specify the title
-datacond	low,high,color	conditional data colors
-hline	value,label	label horizontal line at value
-valpos	t=top, b=bottom, m=middle	value position
-xlabel	default=1, 0 to suppress	x axis label interval
-yrange	min,max,step	specify the y axis label range

Position and Scaling

-top	80	top of the chart
-bottom	30	bottom of the chart
-left	20	left margin
-right	80	right margin
-min	data min	set the minimum data value
-max	data max	set the maximum data value

Measures and Attributes

-bgcolor	white	background color
-barwidth	computed from data size	barwidth
-color	lightsteelblue	data color
-csvcol	labe1,label2	specify csv columns
-datafmt	%.1f	data format for values
-dmin	false	use data minimum, not zero
-framecolor	rgb(127,127,127)	frame color
-lcolor	rgb(75,75,75)	label color
-linewidth	0.2	linewidth
-ls	2.4	linespacing
-noteloc	c=center, r=right, l=left	annotation location
-pmlen	20	pmap label length
-psize	30	diameter of the donut
-pwidth	3	width of the donut or pmap
-rlcolor	rgb(127,0,0)	regression line color
-textsize	1.5	text size
-xlabrot	0	xlabel rotation (deg.)
-vcolor	rgb(127,0,0)	value color
-volop	50	volume opacity %

Command Examples

AAPL Closing Price

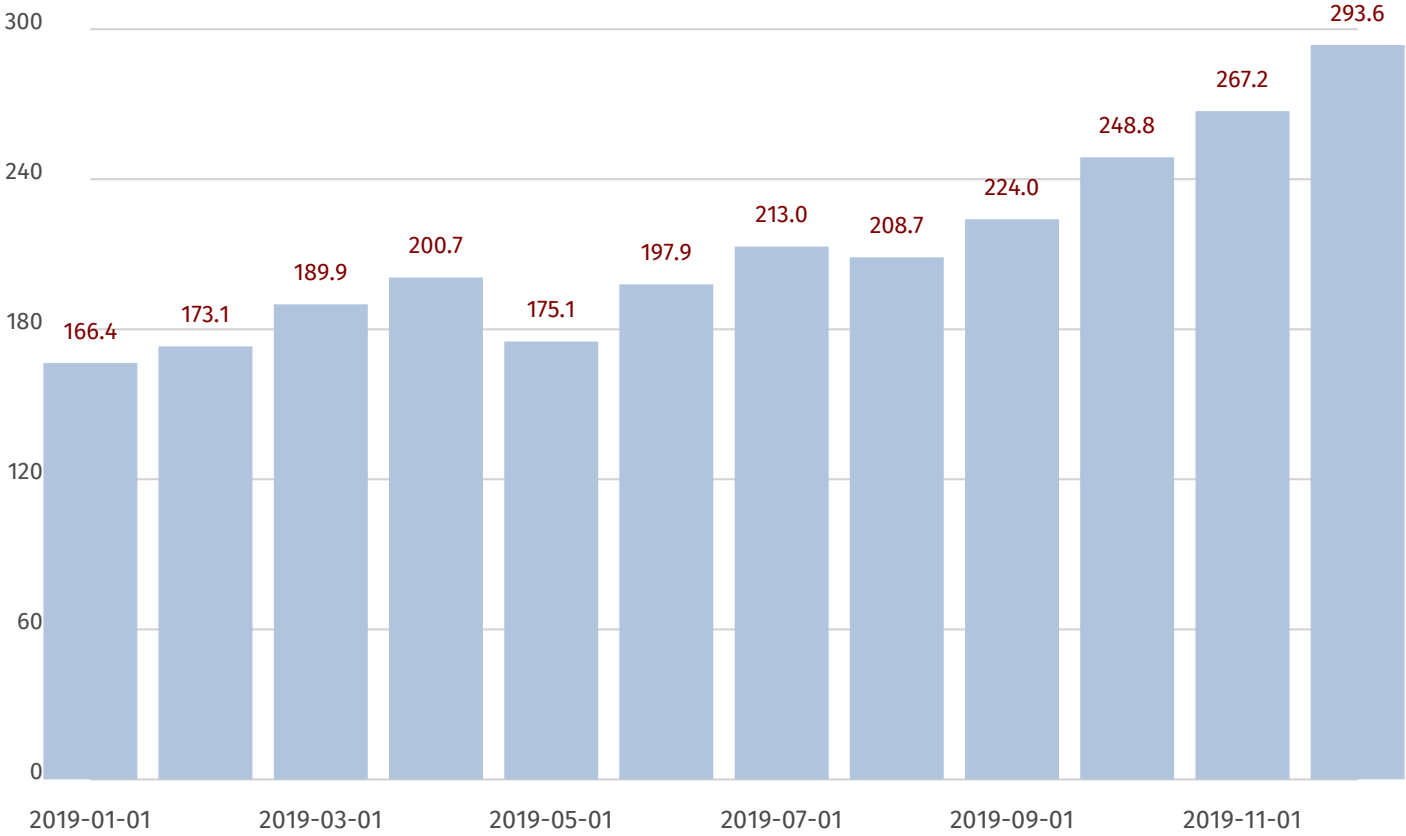
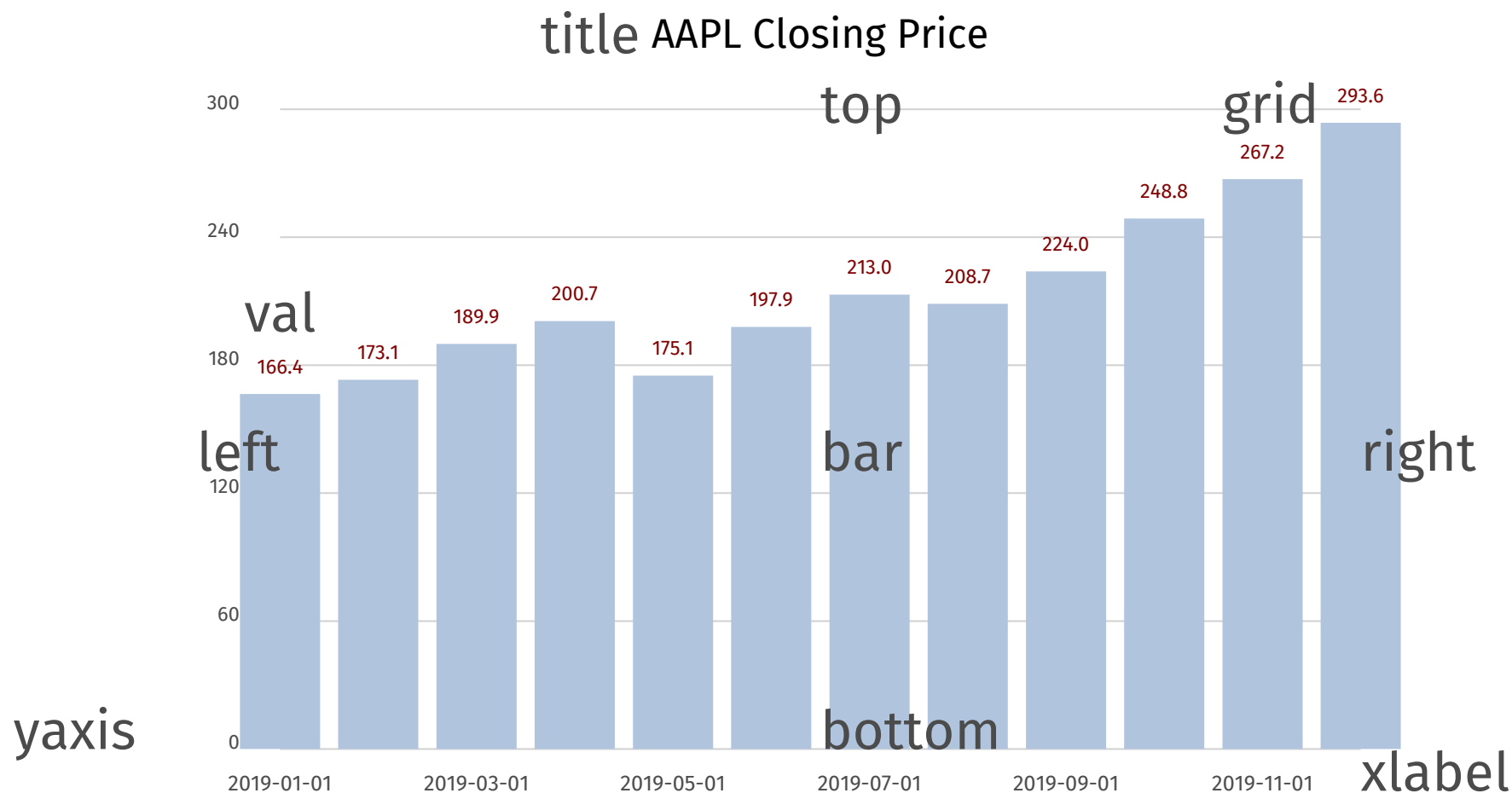
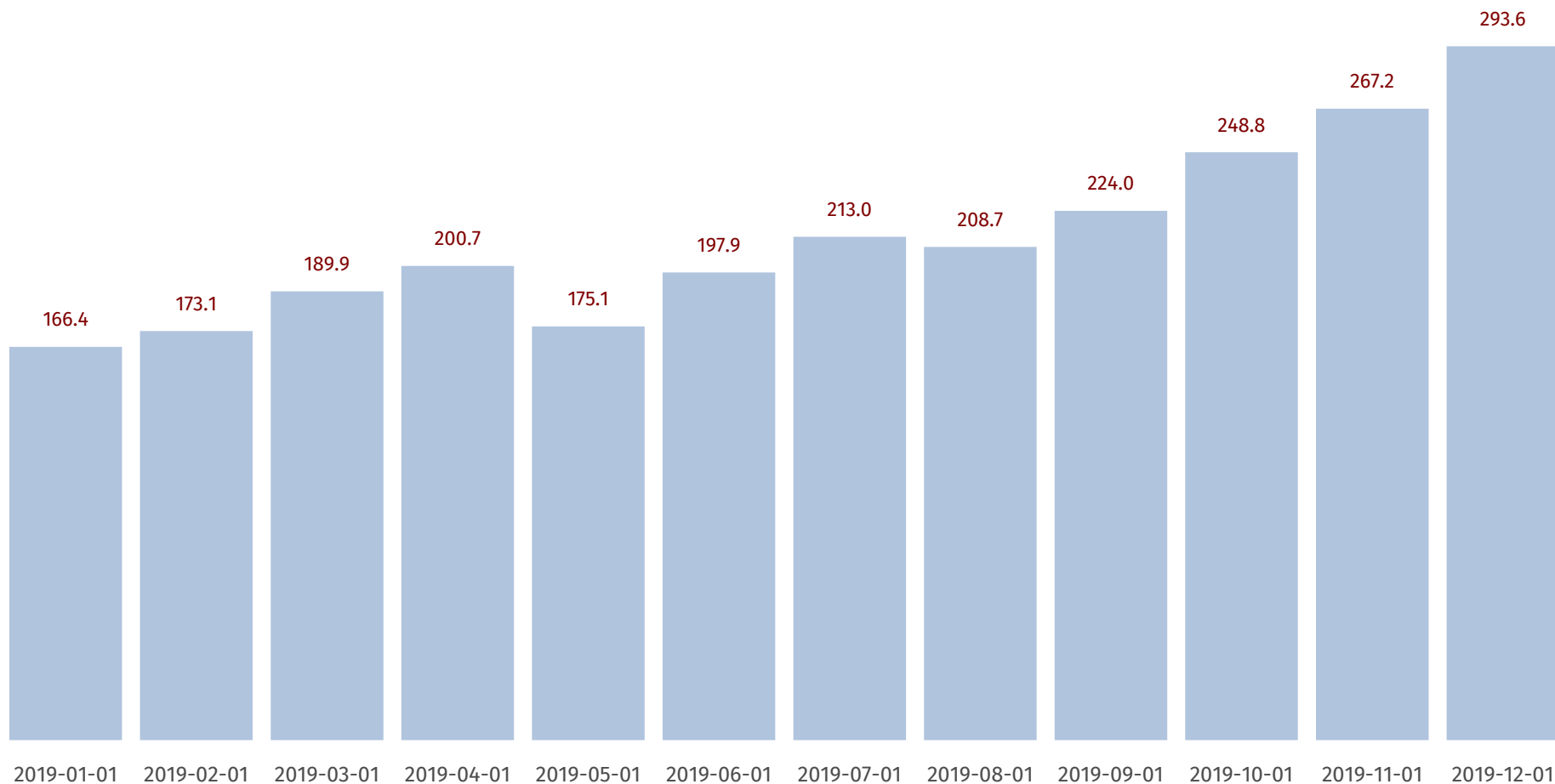


Chart Attributes



```
dchart -left=20 -right=80 -top=75 -bottom=30 -yaxis -grid -xlabel=2 -val AAPL.d
```

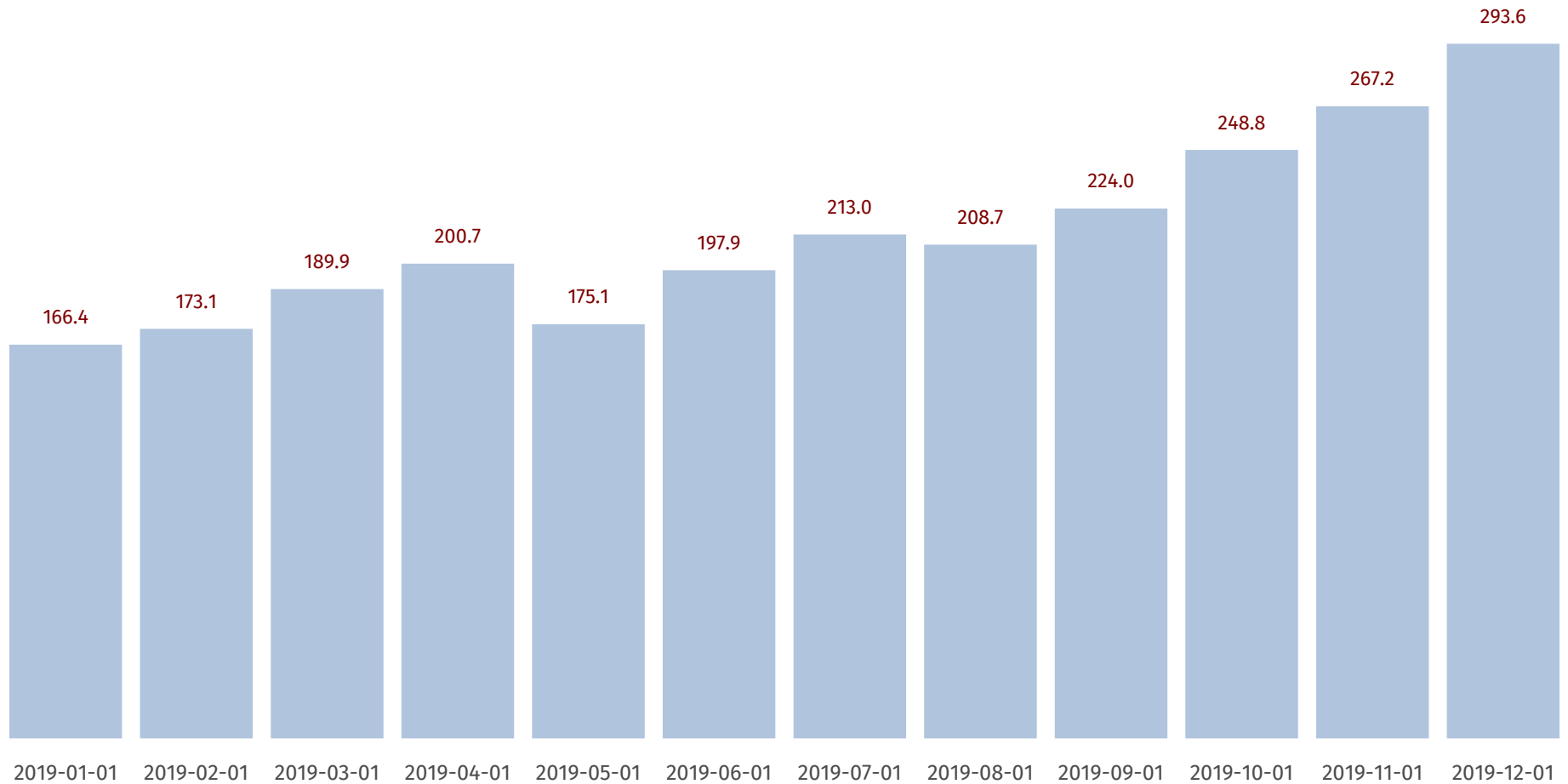
AAPL Closing Price



Default Bar Chart

dchart AAPL.d

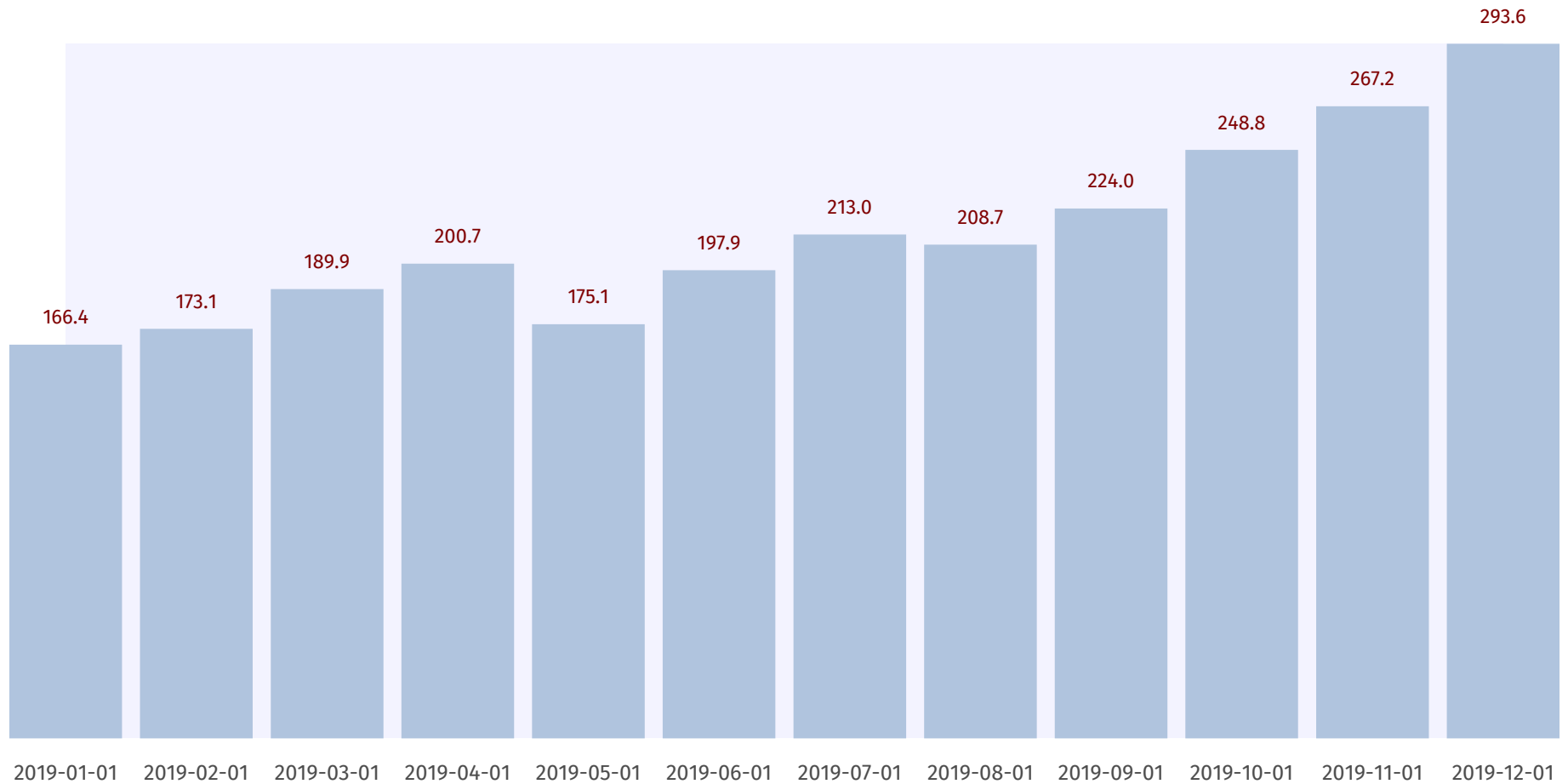
Close



Reading CSV files

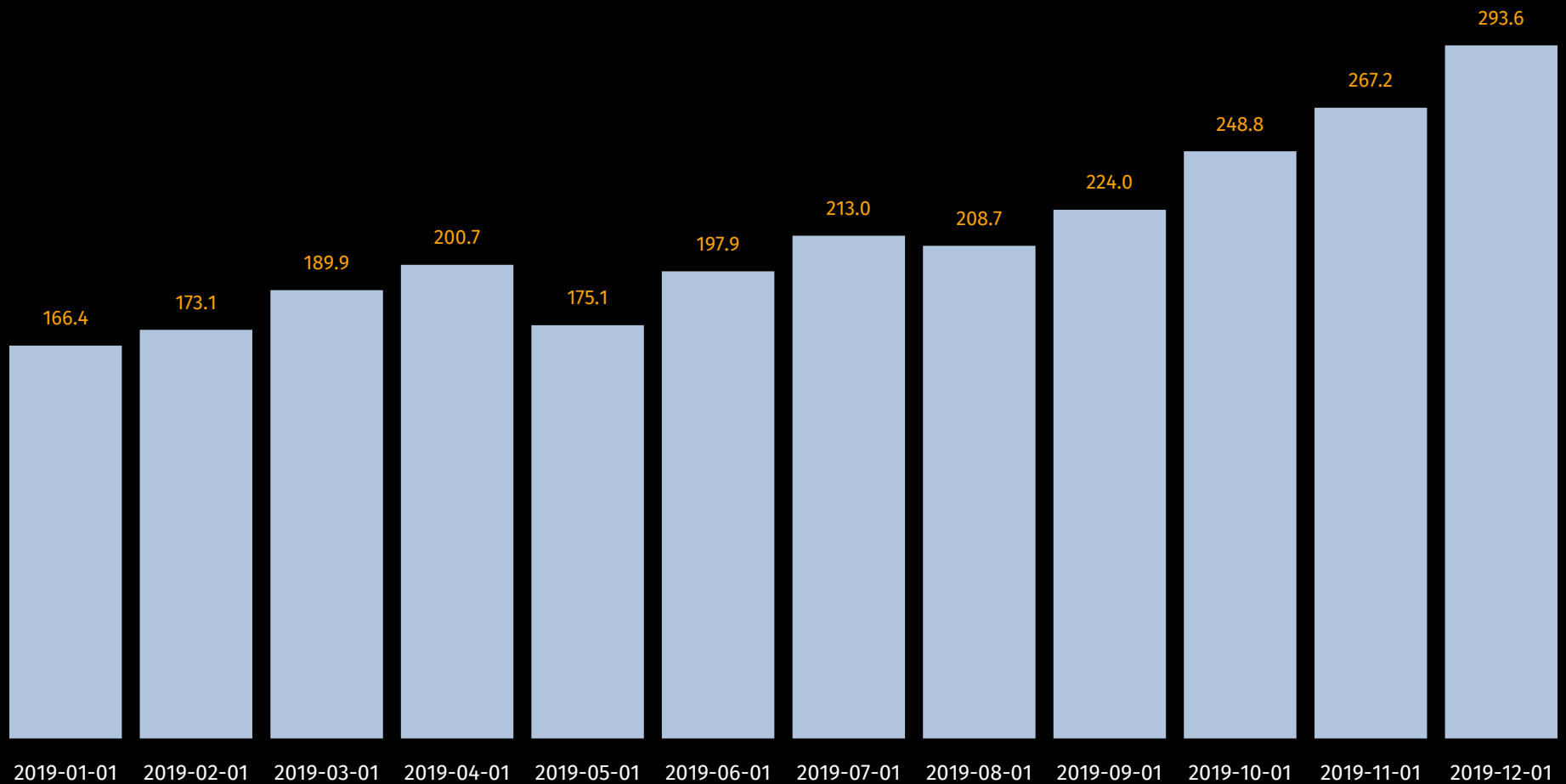
```
dchart -csv -csvcol=Date,Close AAPL.csv
```

AAPL Closing Price



Frame, Frame Color

```
dchart -frame=t -framecolor=blue AAPL.d
```



Background, Label, Value Color

```
dchart -bgcolor=black -lcolor=white -vcolor=orange AAPL.d
```

Close:2019

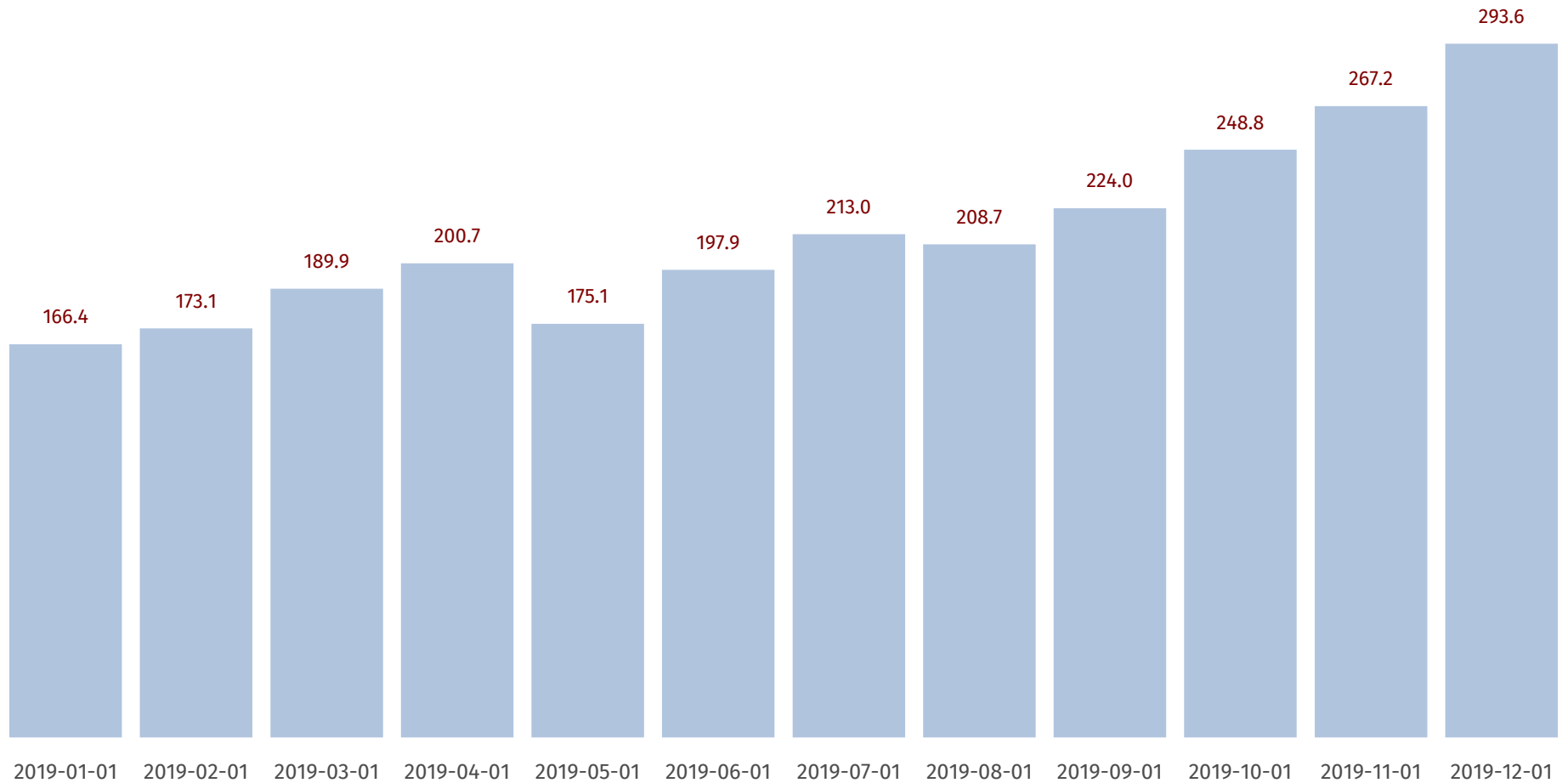
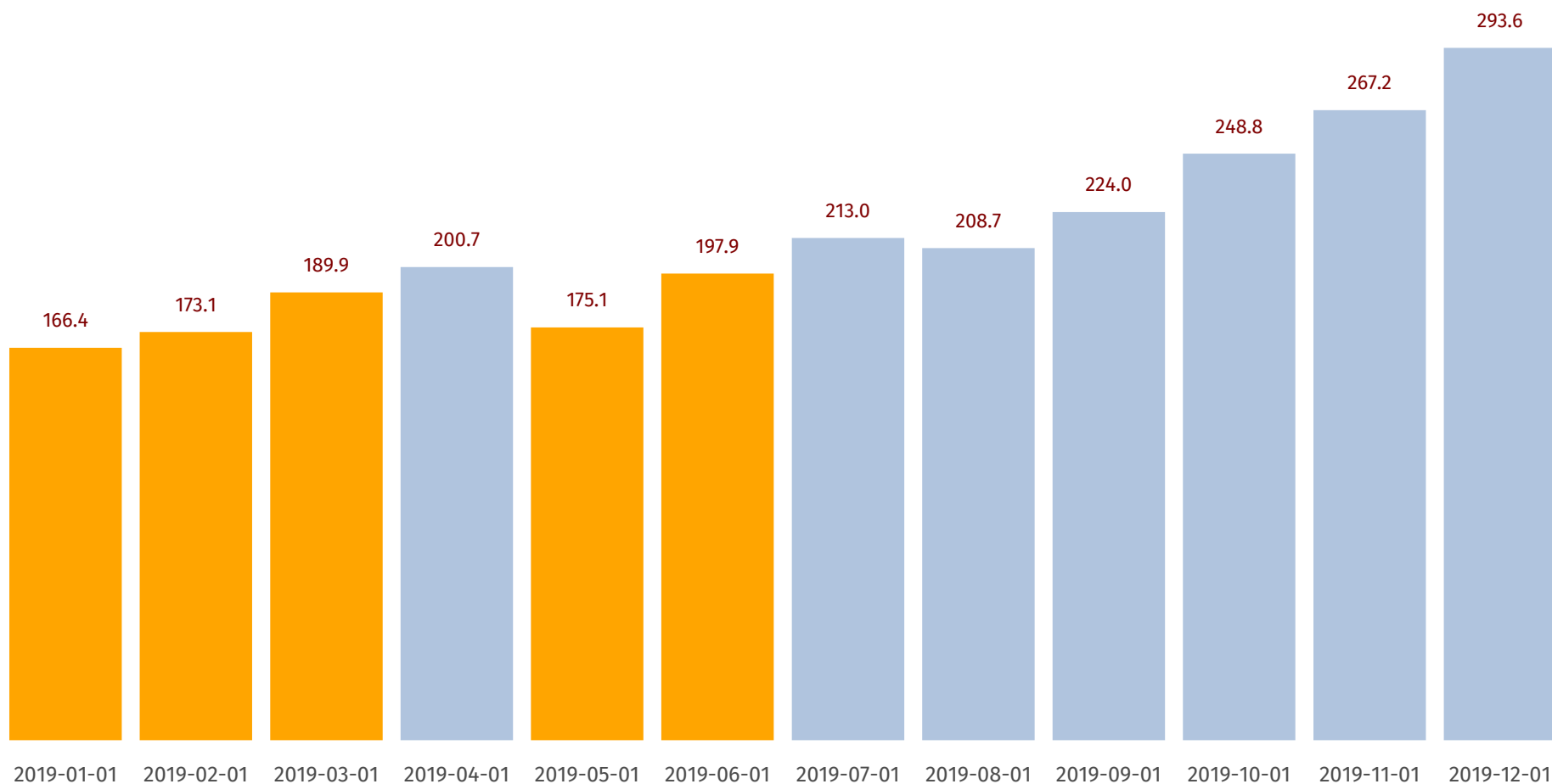


Chart Title

dchart -charttitle="Close:2019" AAPL.d

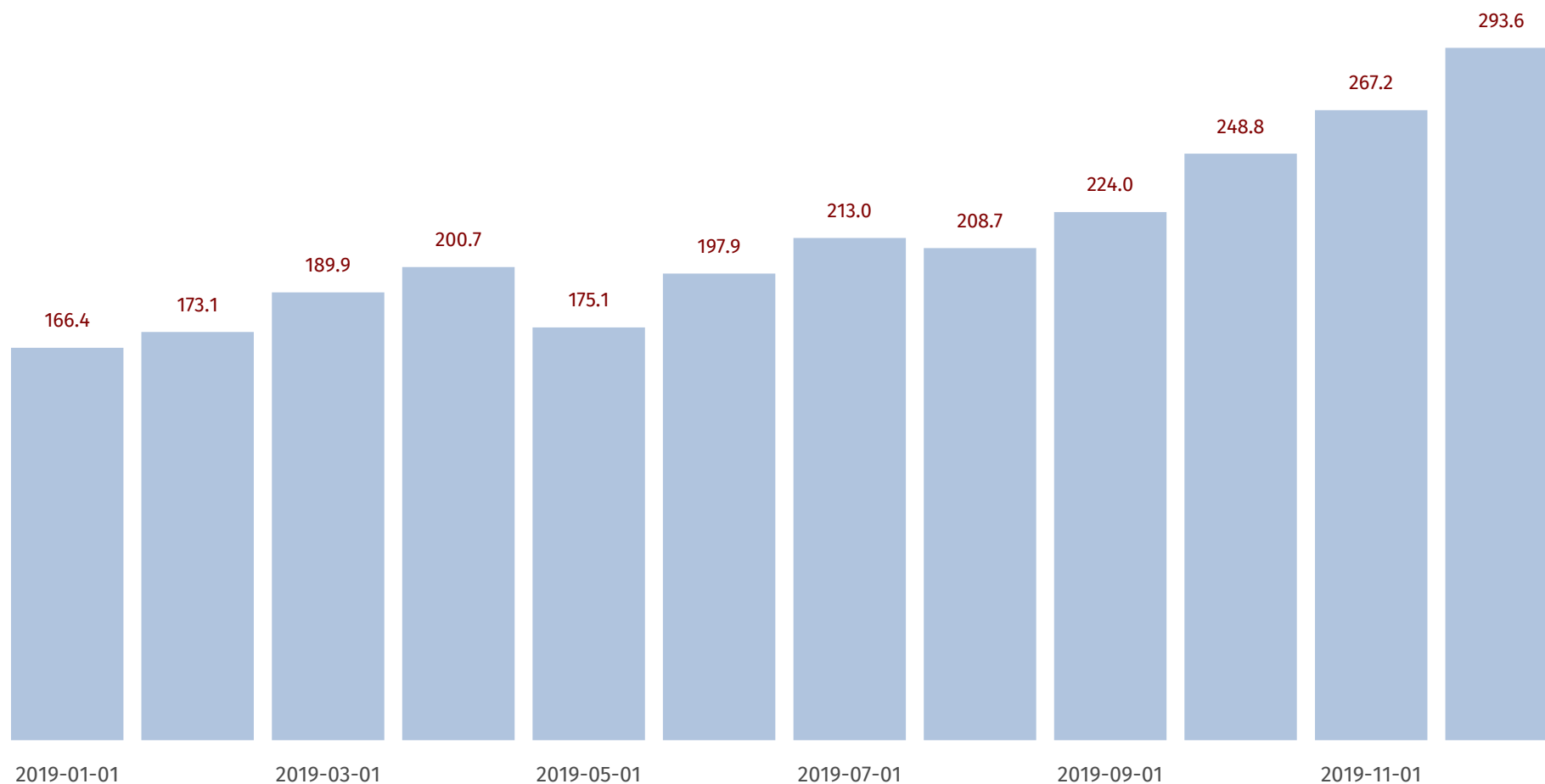
AAPL Closing Price



Data Conditions

dchart -datacond=150,200,orange AAPL.d

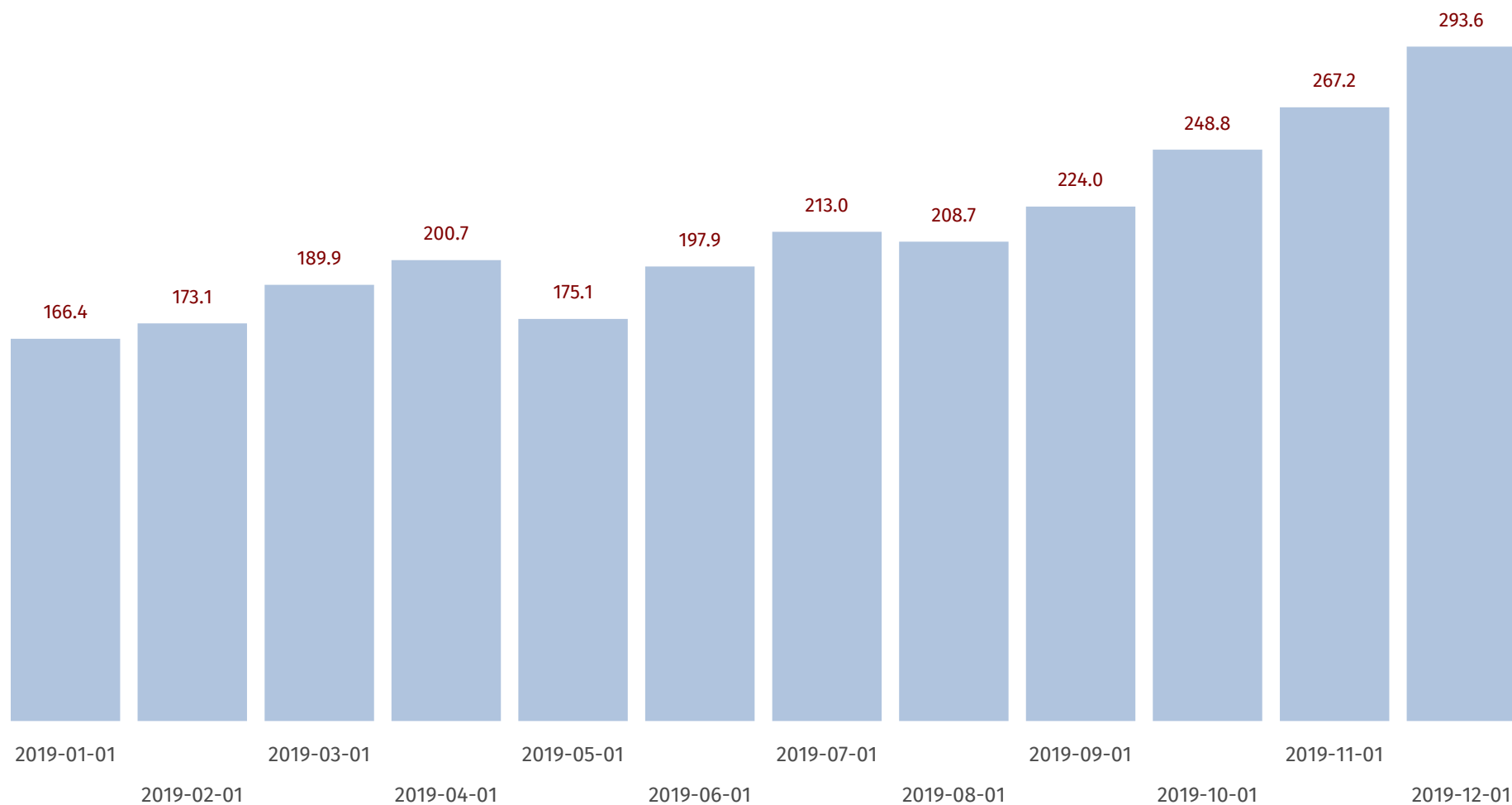
AAPL Closing Price



X-Axis Label Interval

```
dchart -xlabel=2 AAPL.d
```

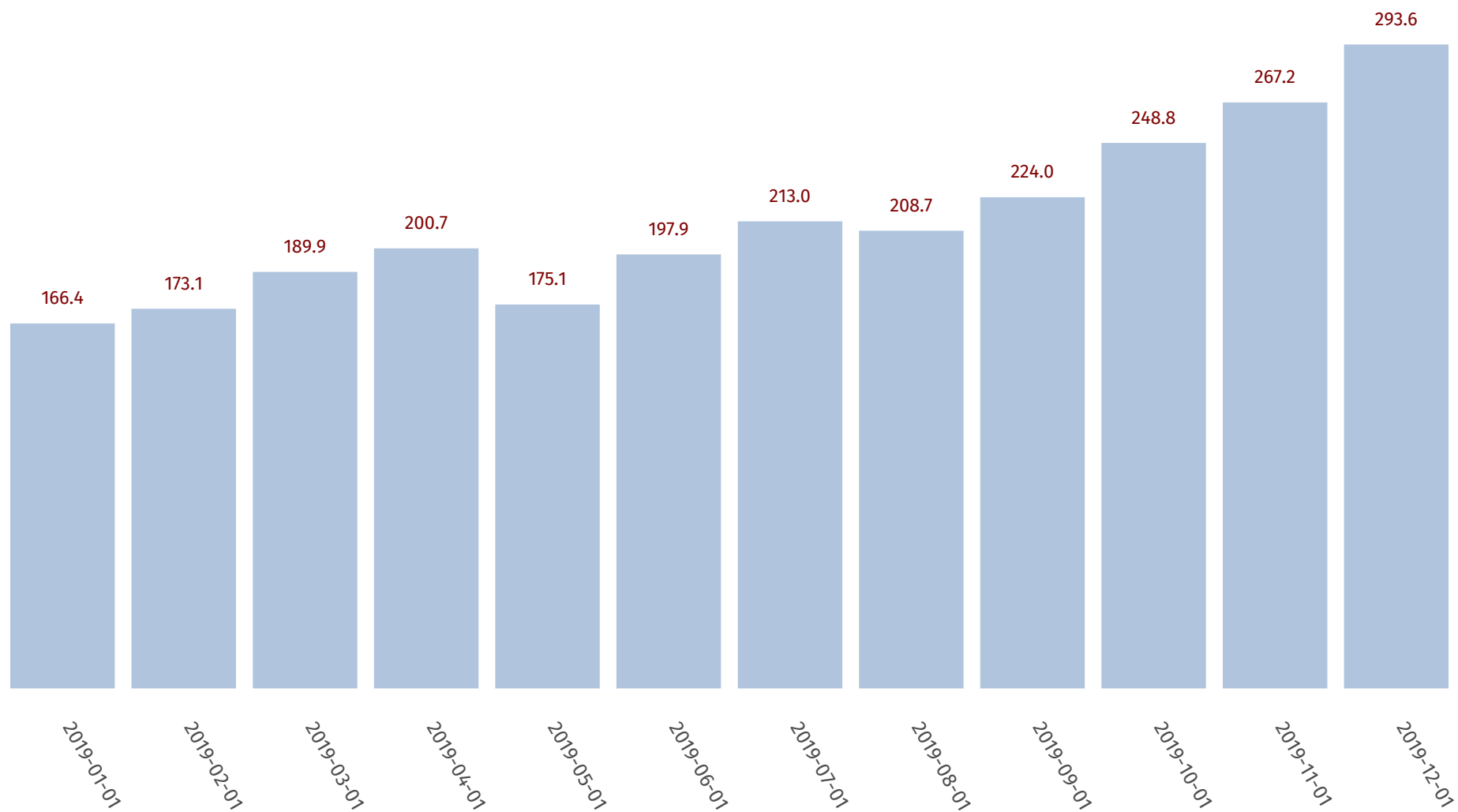
AAPL Closing Price



Stagger X-Axis Labels

`dchart -xstagger AAPL.d`

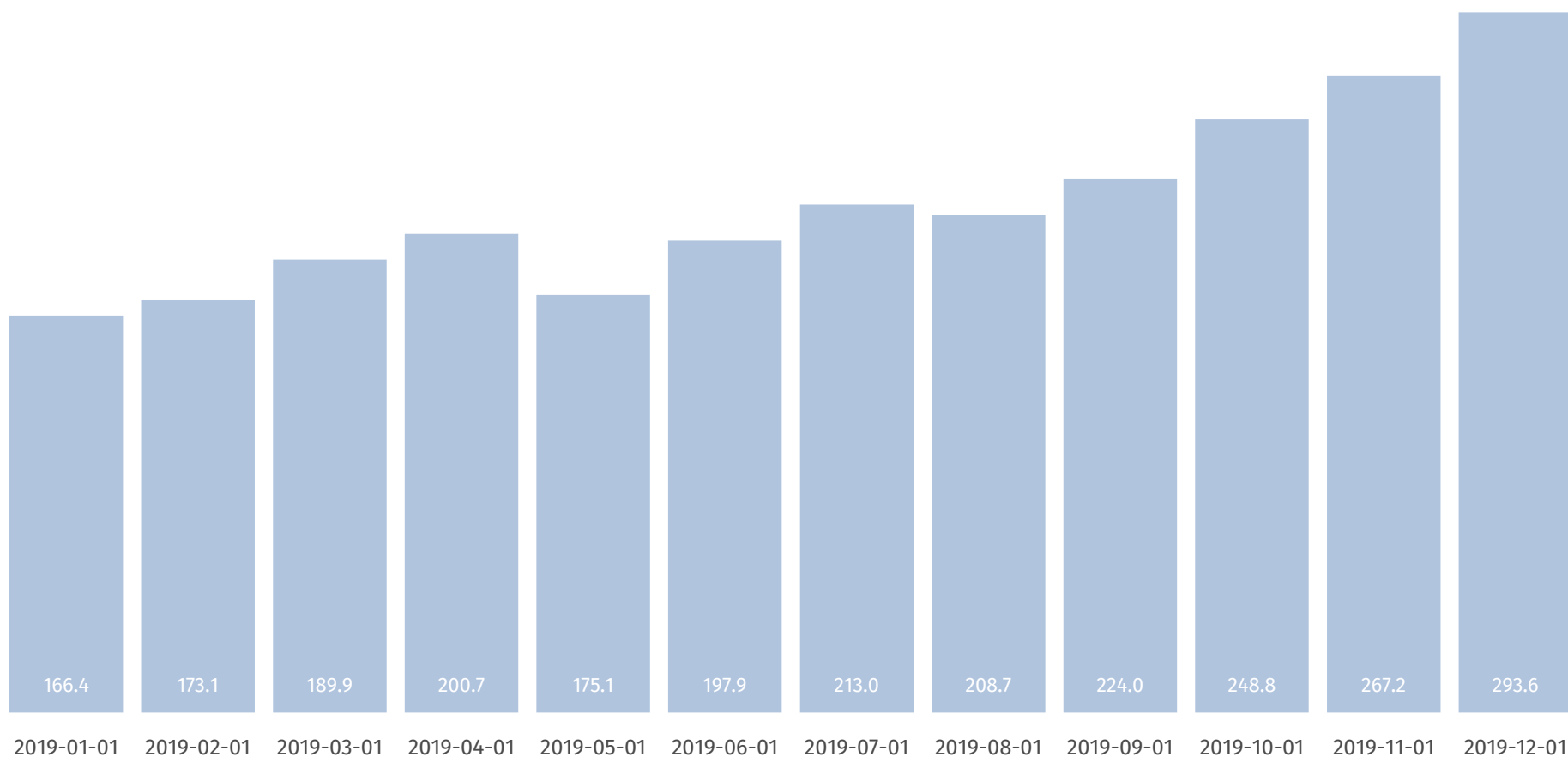
AAPL Closing Price



X-Axis Label Rotation

```
dchart -xlabrot=300 AAPL.d
```

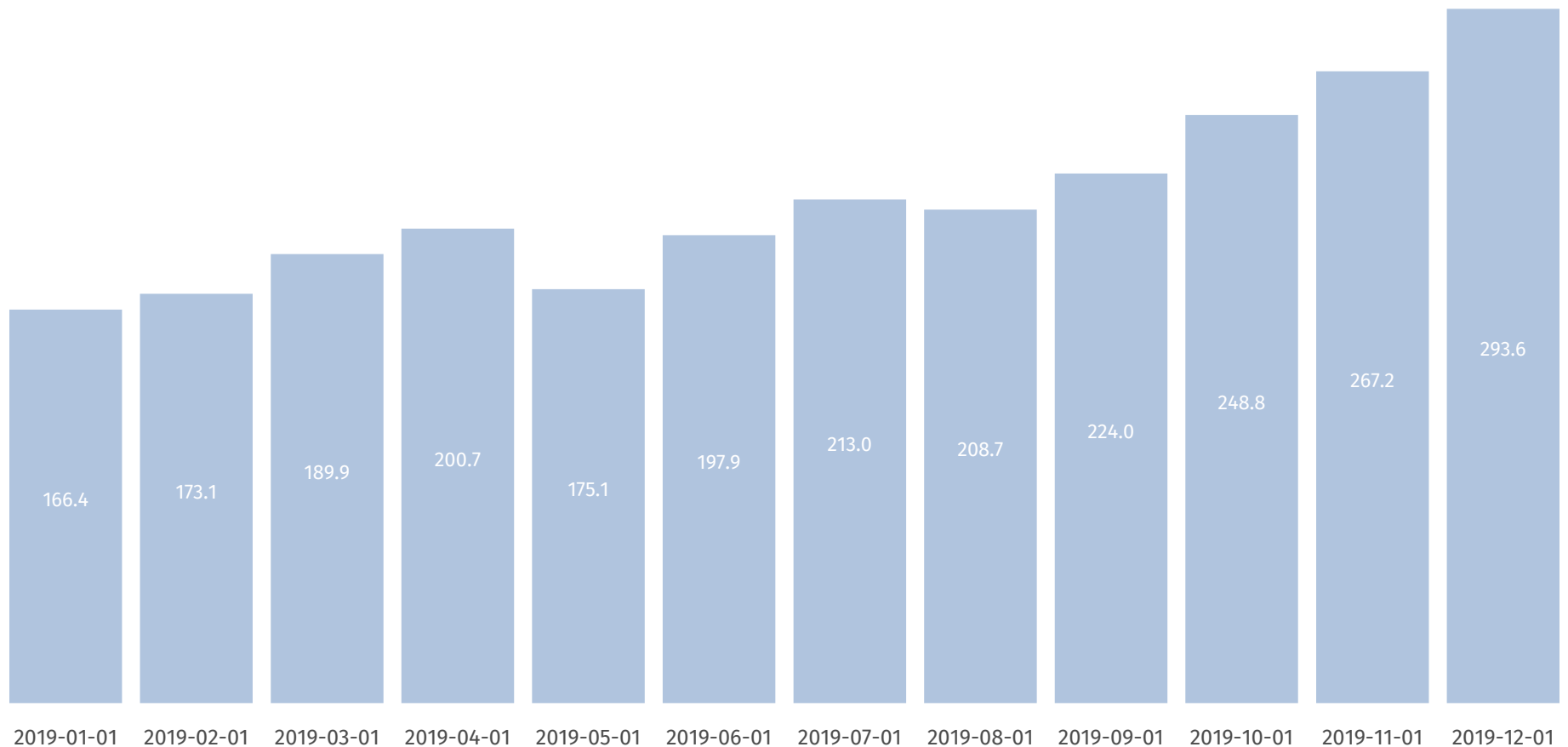

AAPL Closing Price



Value Color, Value Position Bottom

dchart -vcolor=white -valpos=b AAPL.d

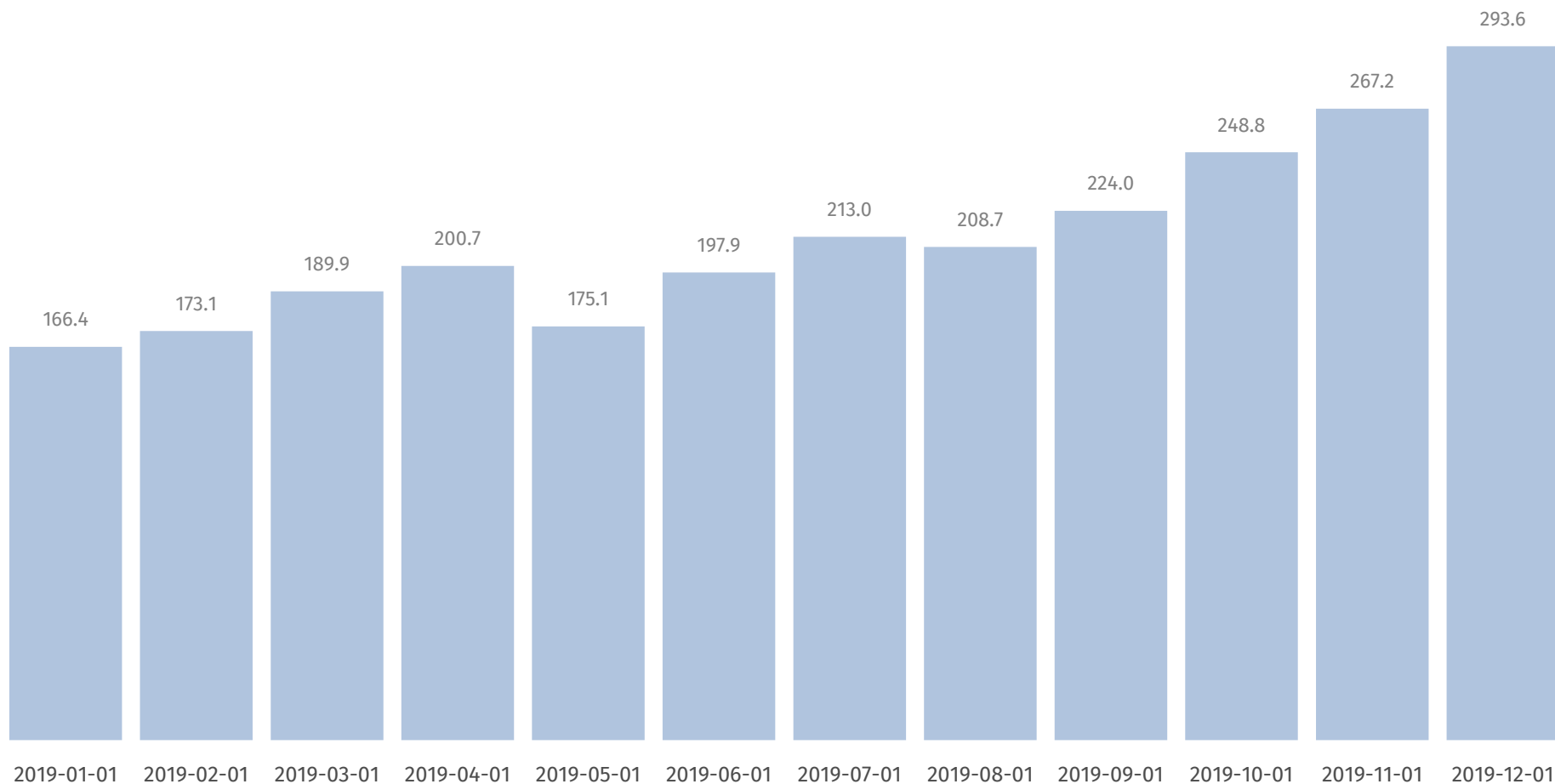
AAPL Closing Price



Value Color, Value Position Middle

dchart -vcolor=white -valpos=m AAPL.d

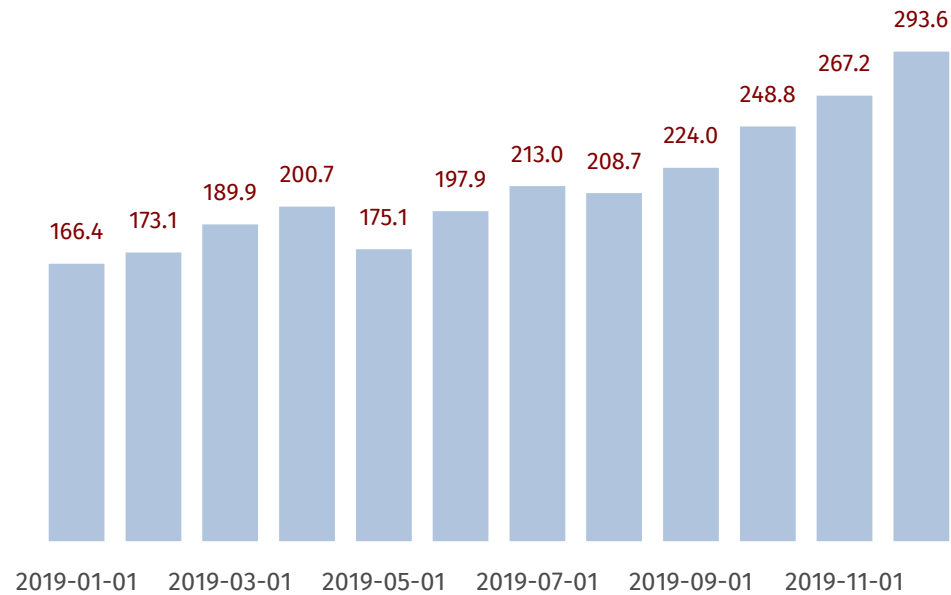
AAPL Closing Price



Value Color, Value Position Top

dchart -vcolor=gray AAPL.d

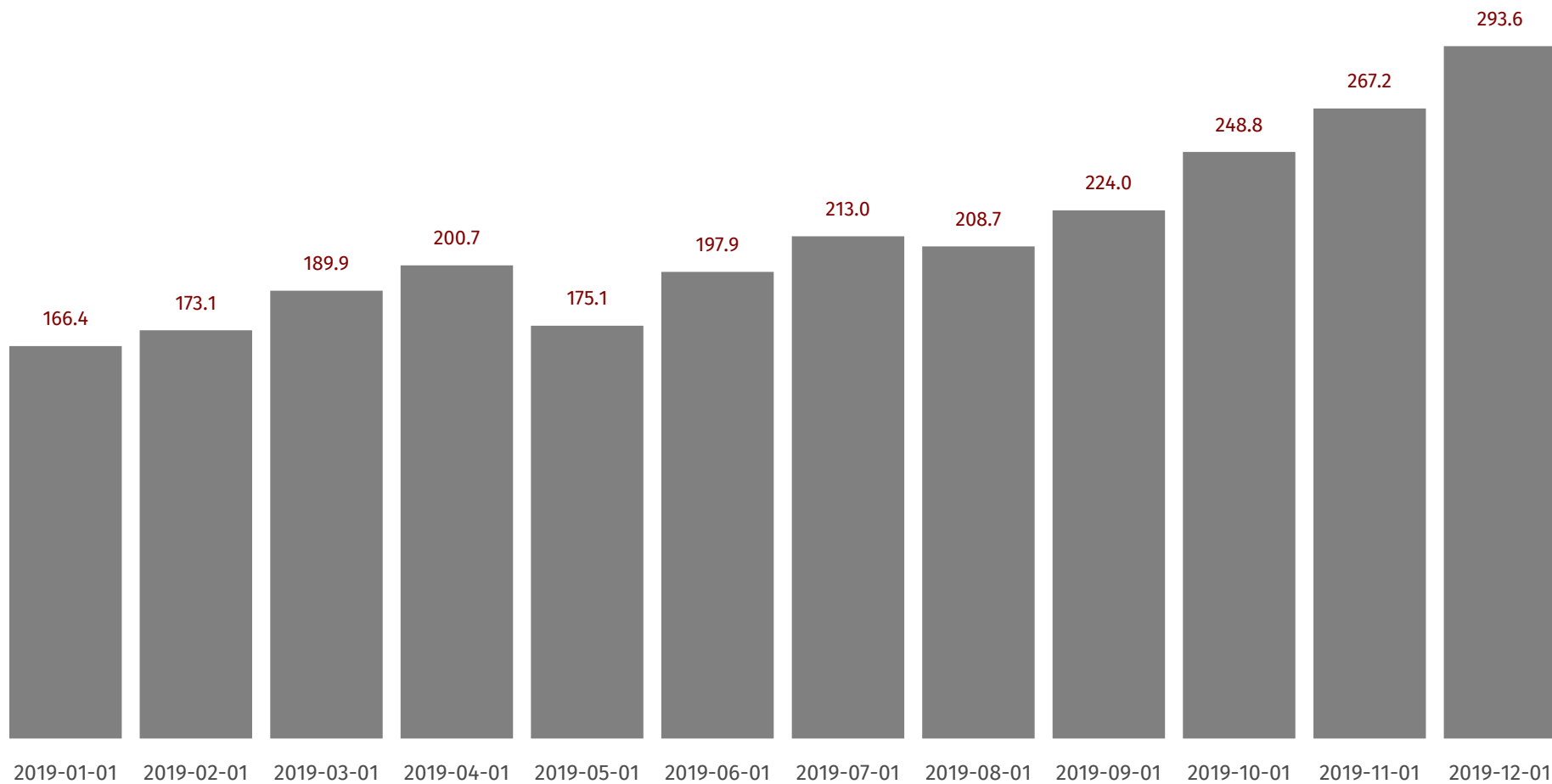
AAPL Closing Price



Scaling

```
dchart -xlabel=2 -left 30 -right 70 -top 70 -bottom 40 AAPL.d
```

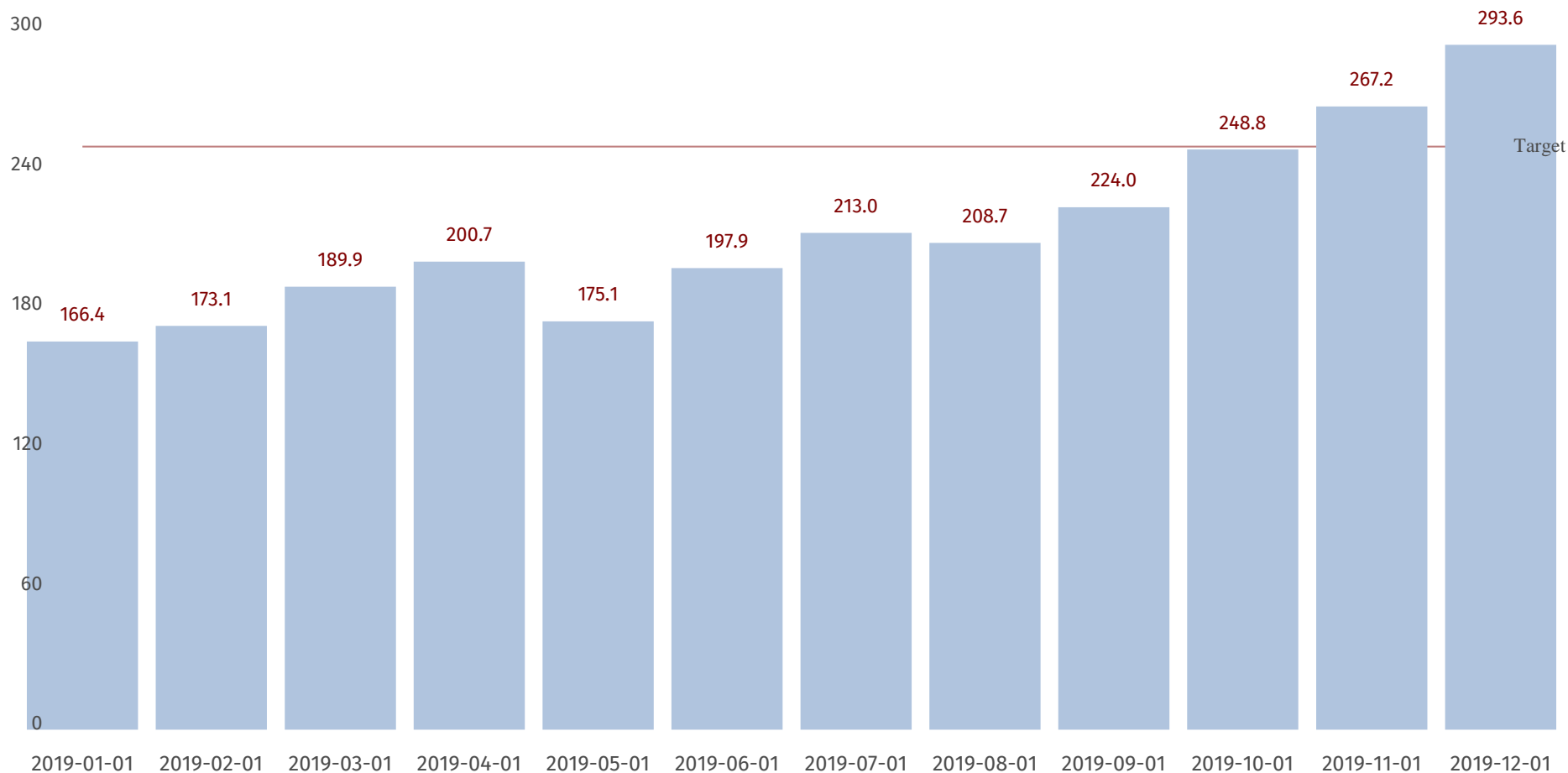
AAPL Closing Price



Color

`dchart -color gray AAPL.d`

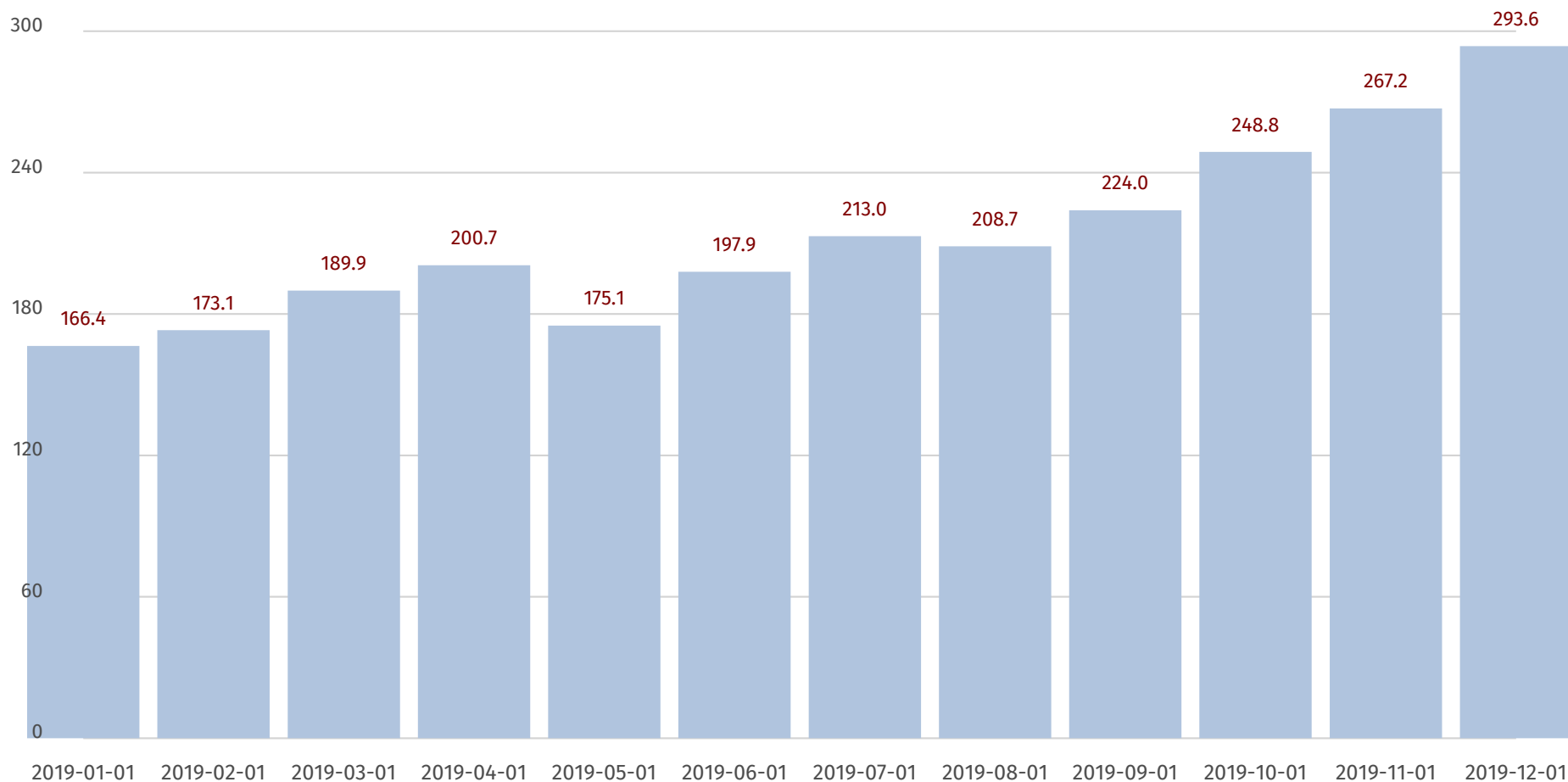
AAPL Closing Price



Target Line, Y-Axis

dchart -hline=250,Target -yaxis AAPL.d

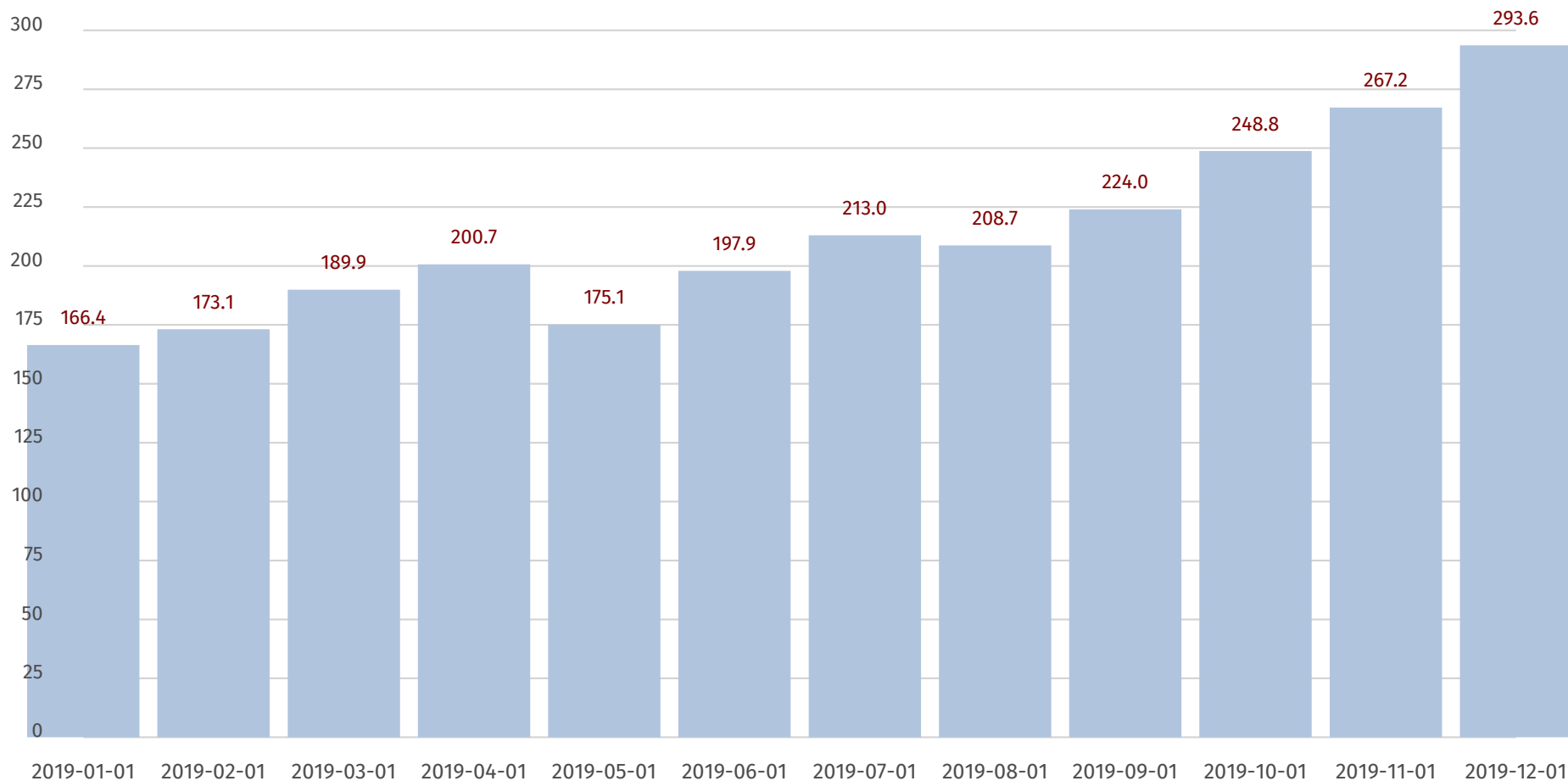
AAPL Closing Price



Y-Axis, Grid

`dchart -grid -yaxis AAPL.d`

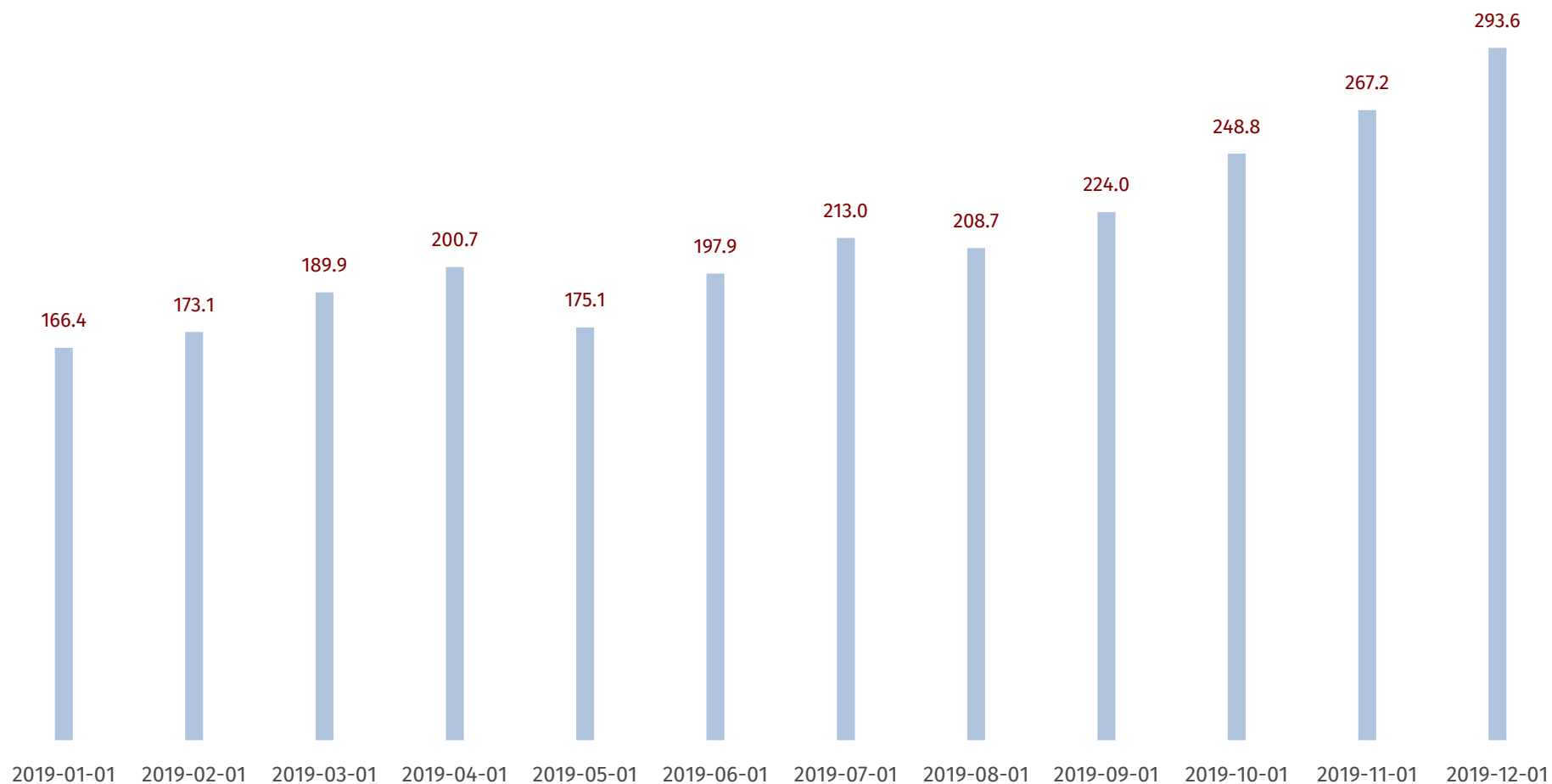
AAPL Closing Price



Y-Range

`dchart -yrange=0,300,25 -grid -yaxis AAPL.d`

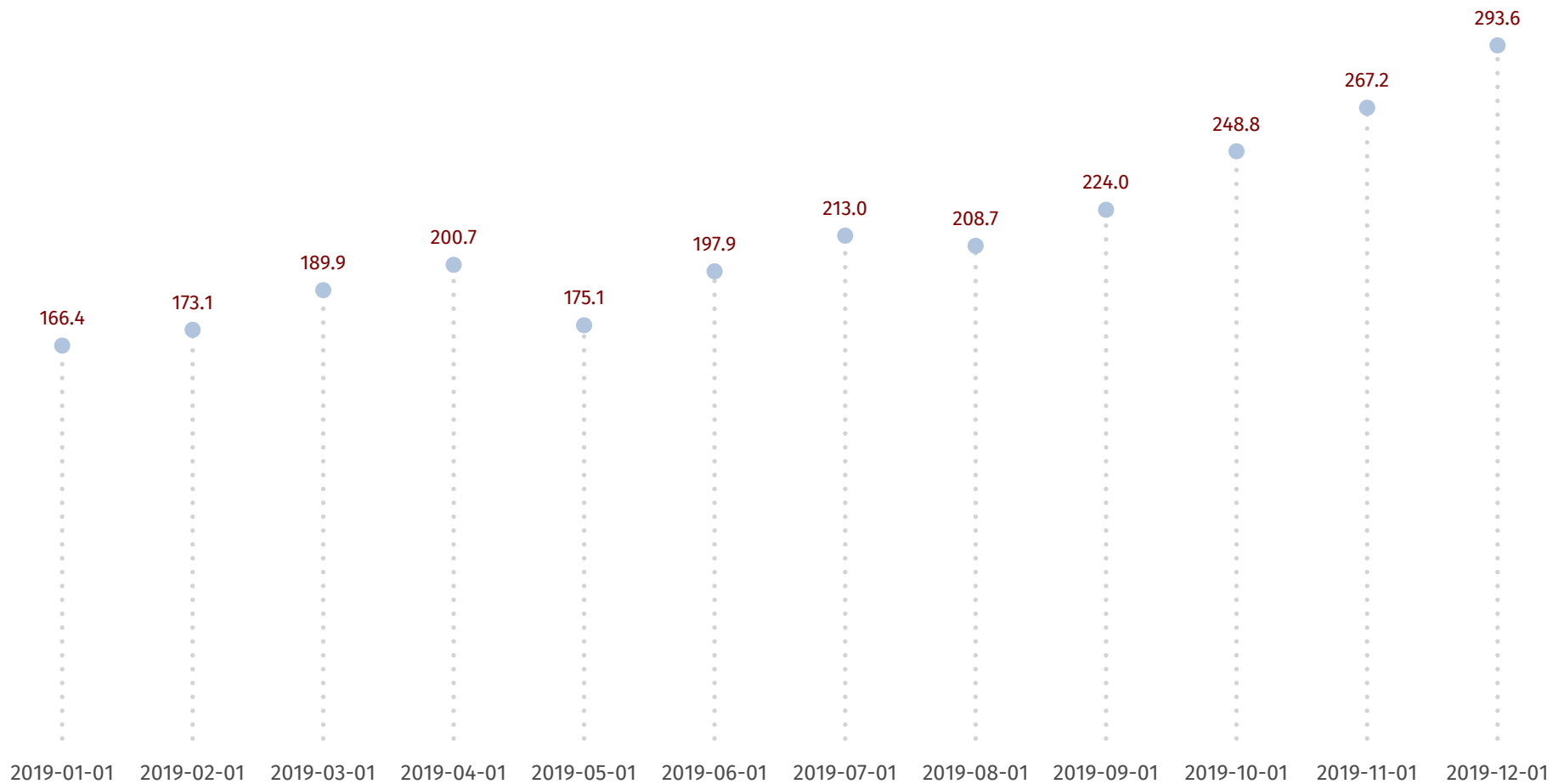
AAPL Closing Price



Adjusting Bar Width

`dchart -barwidth=1 AAPL.d`

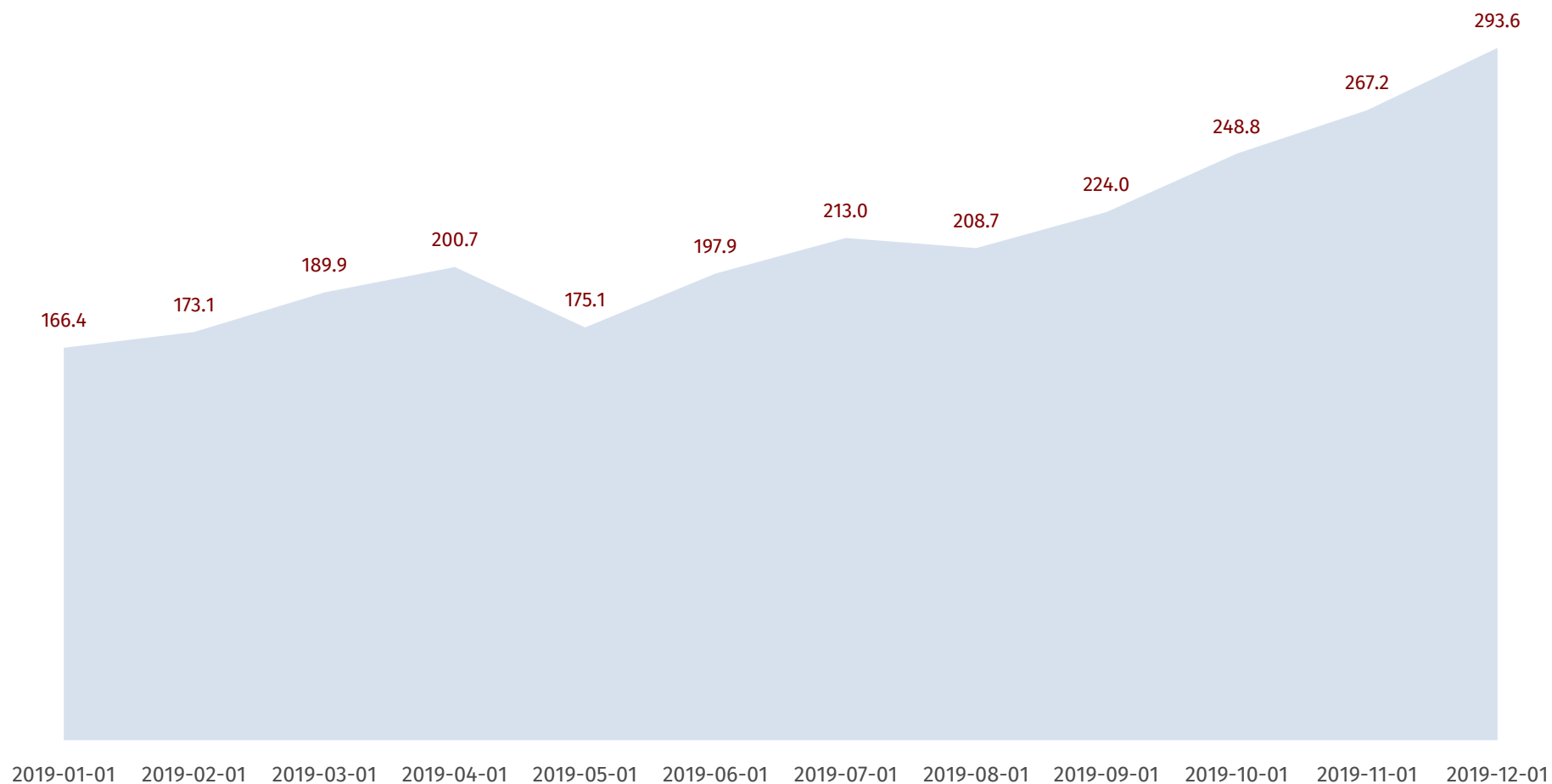
AAPL Closing Price



Dot Chart

`dchart -bar=f -dot AAPL.d`

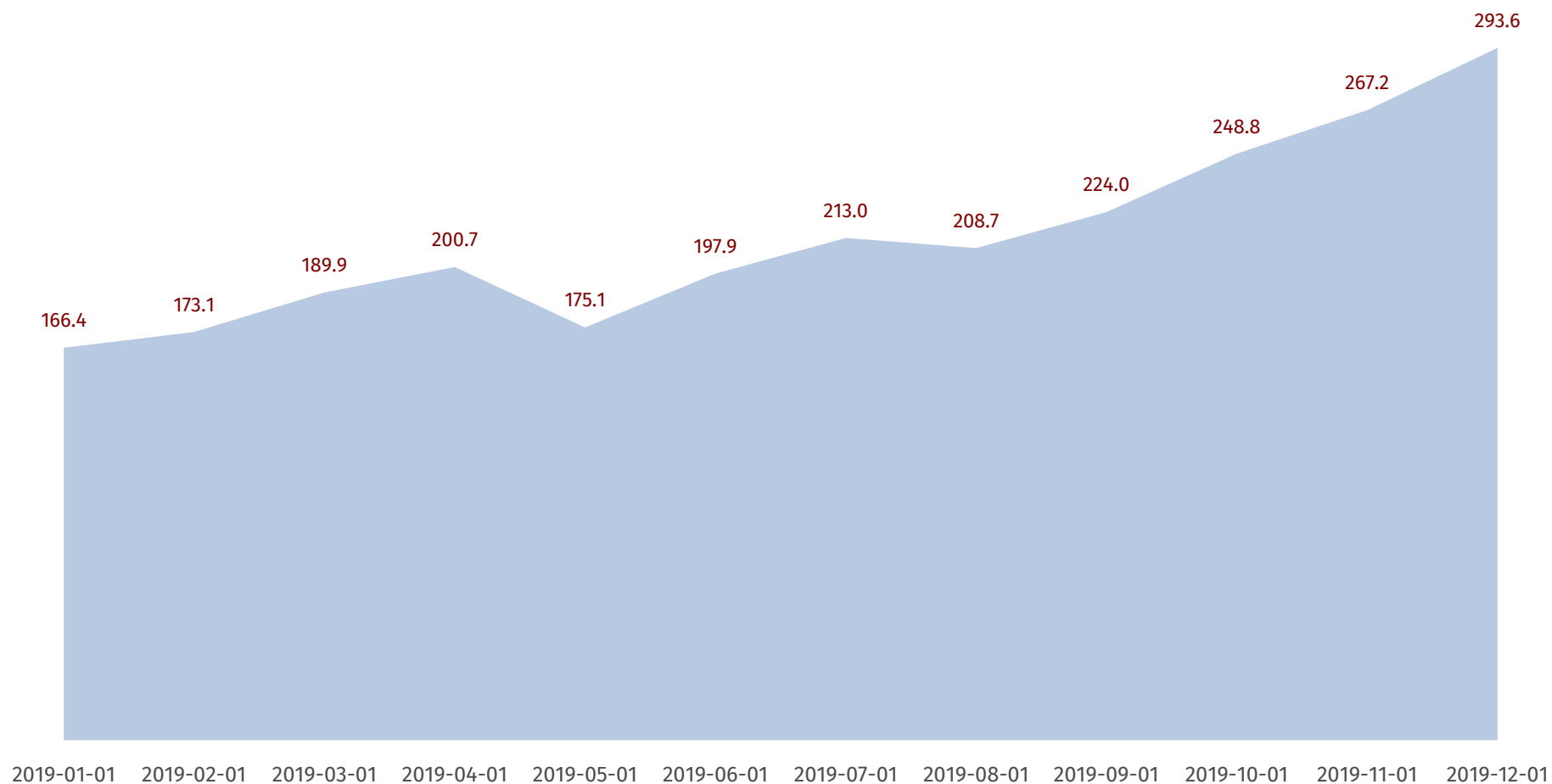
AAPL Closing Price



Area Chart

`dchart -bar=f -vol AAPL.d`

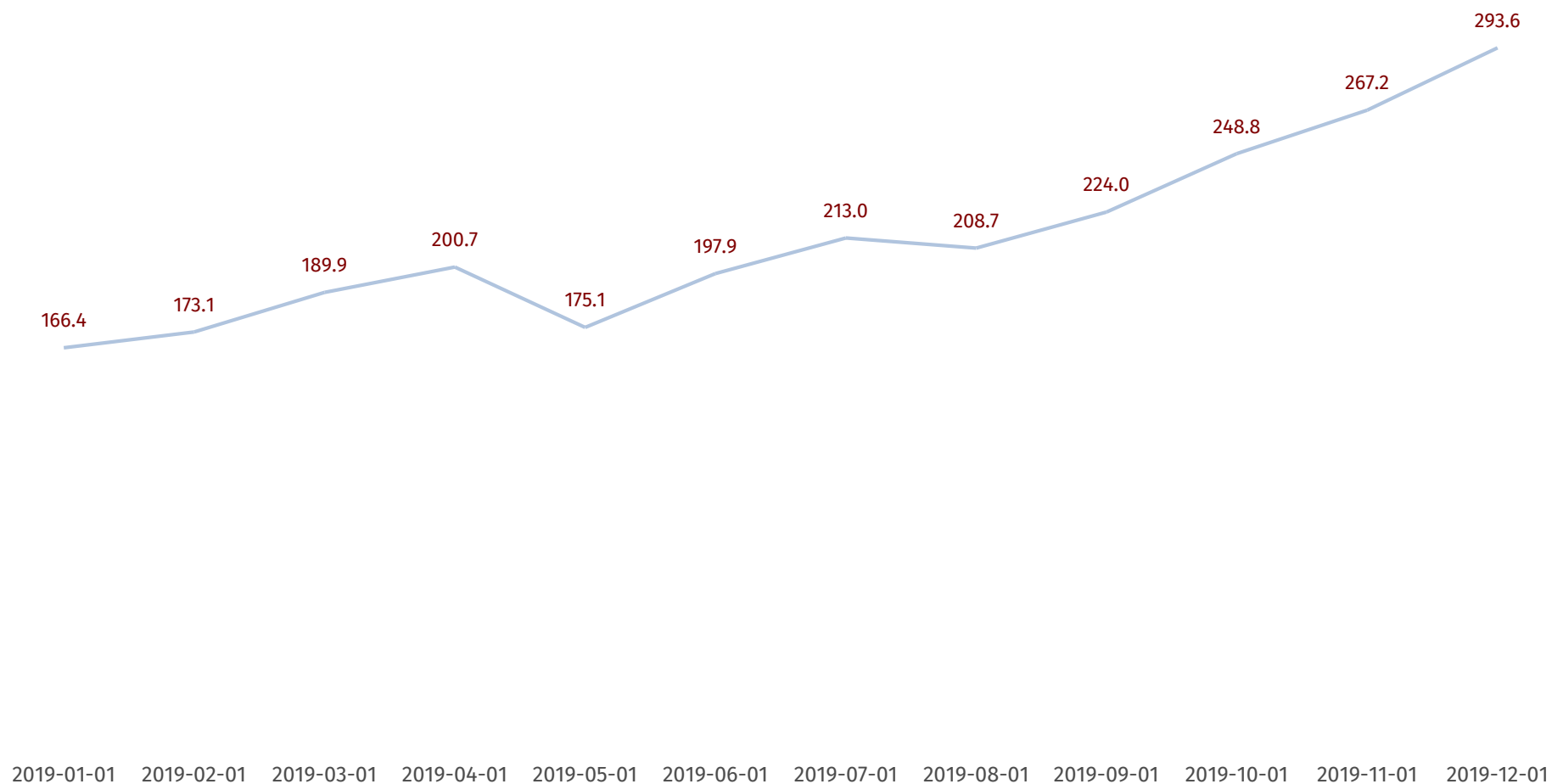
AAPL Closing Price



Area Chart, Opacity

```
dchart -bar=f -vol -volop=90 AAPL.d
```

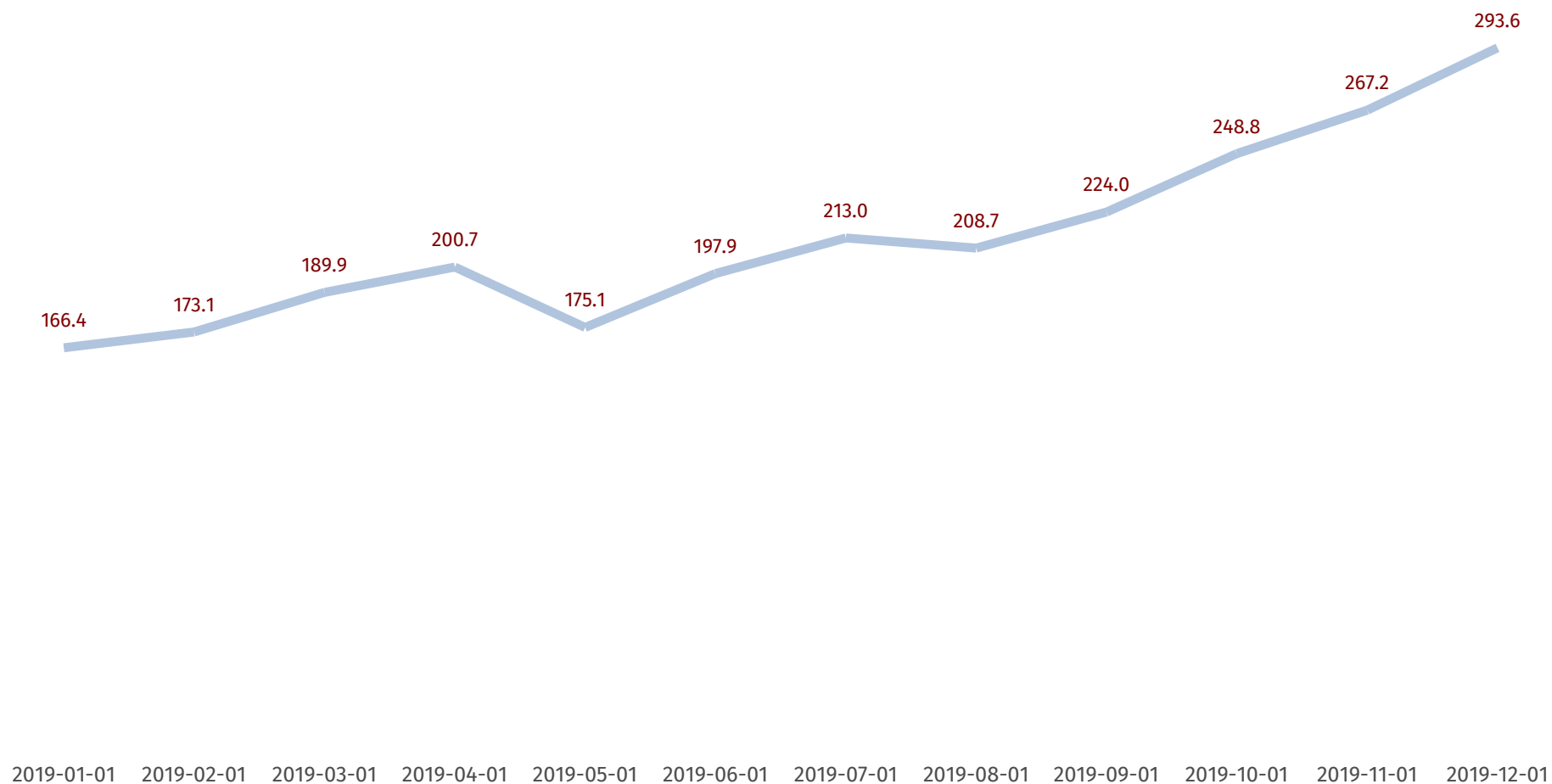
AAPL Closing Price



Line Chart

`dchart -bar=f -line AAPL.d`

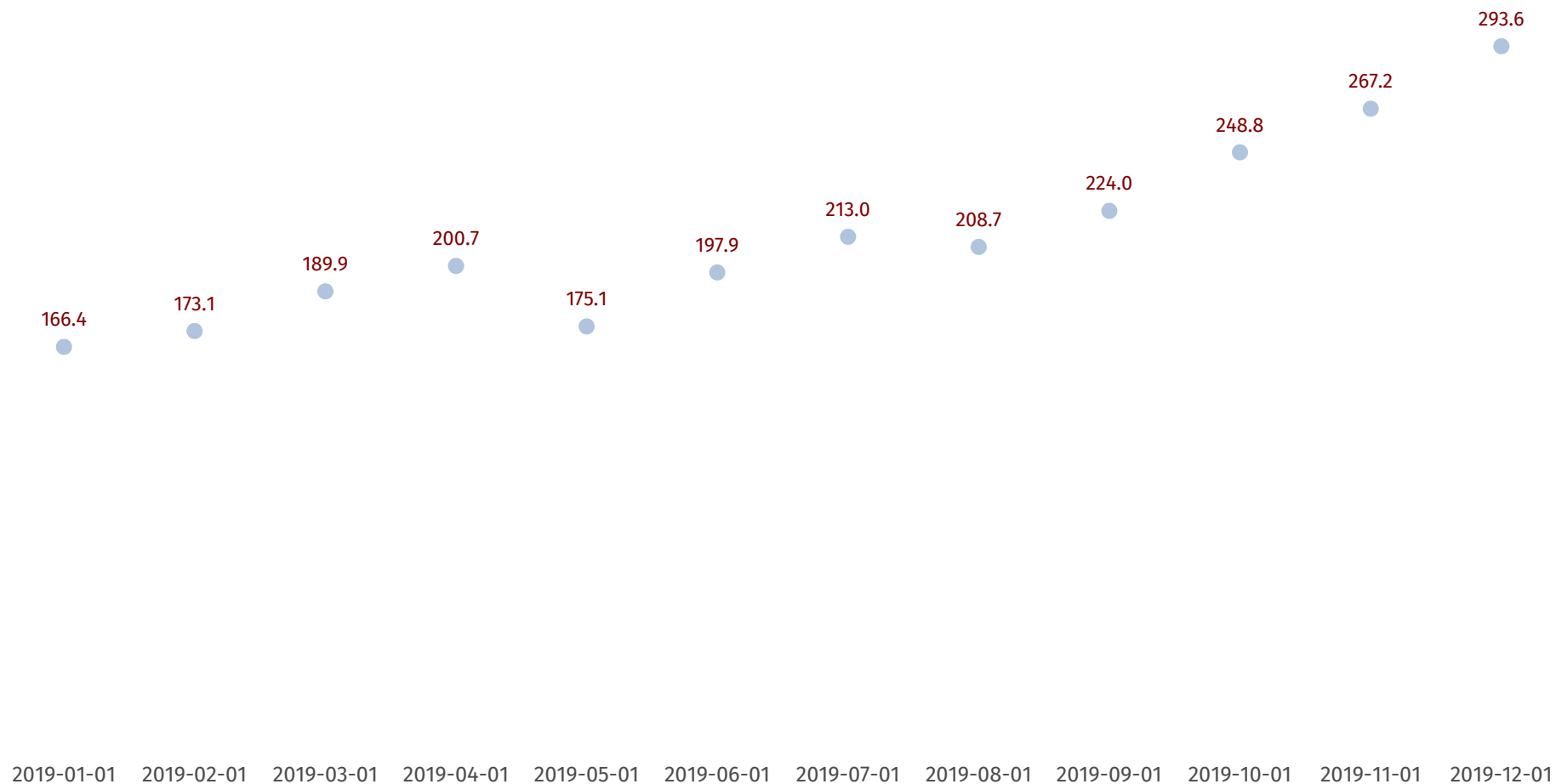
AAPL Closing Price



Line Chart, Line Width

```
dchart -bar=f -line -linewidth=0.5 AAPL.d
```

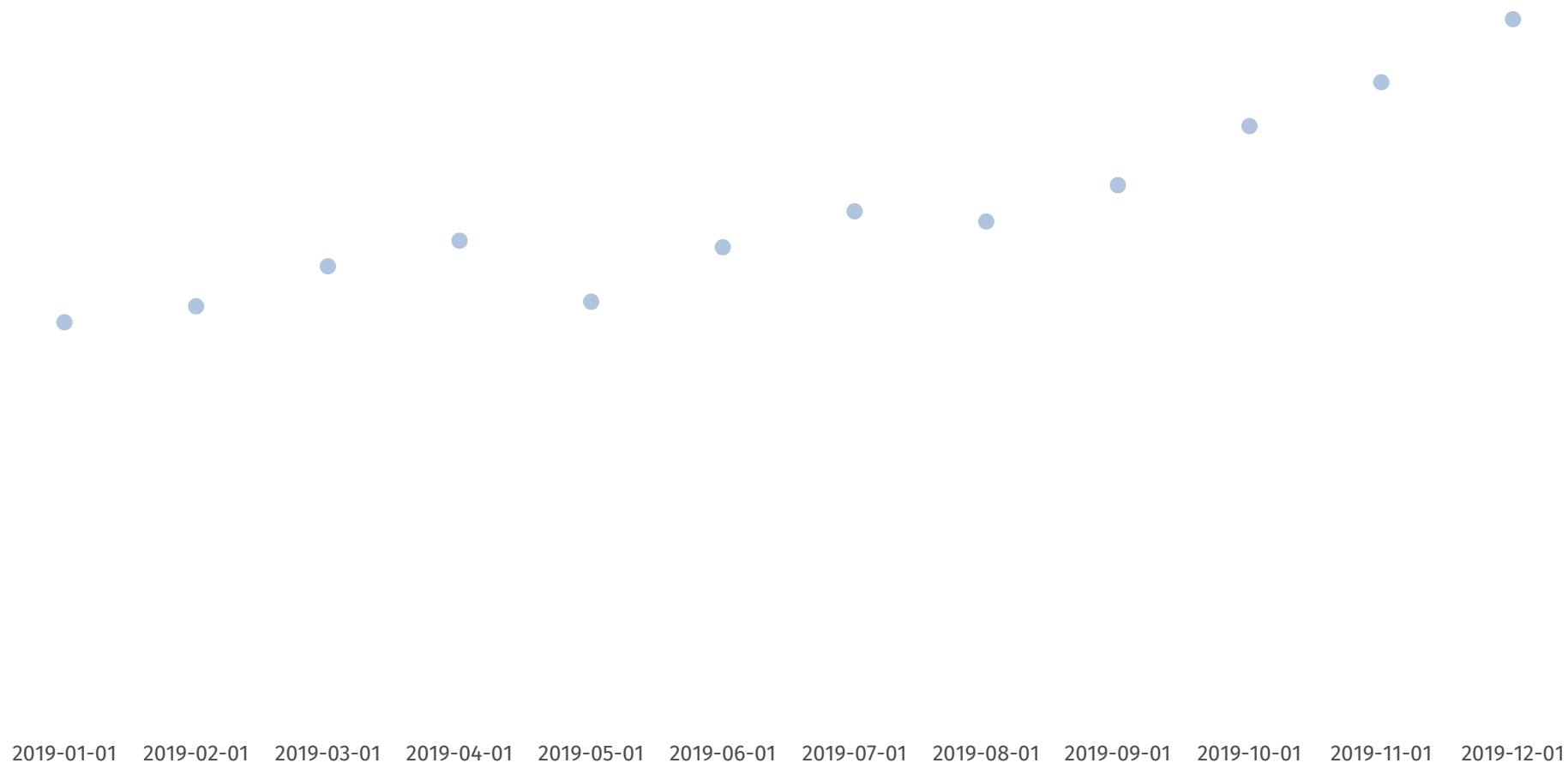
AAPL Closing Price



Scatter Chart

`dchart -bar=f -scatter AAPL.d`

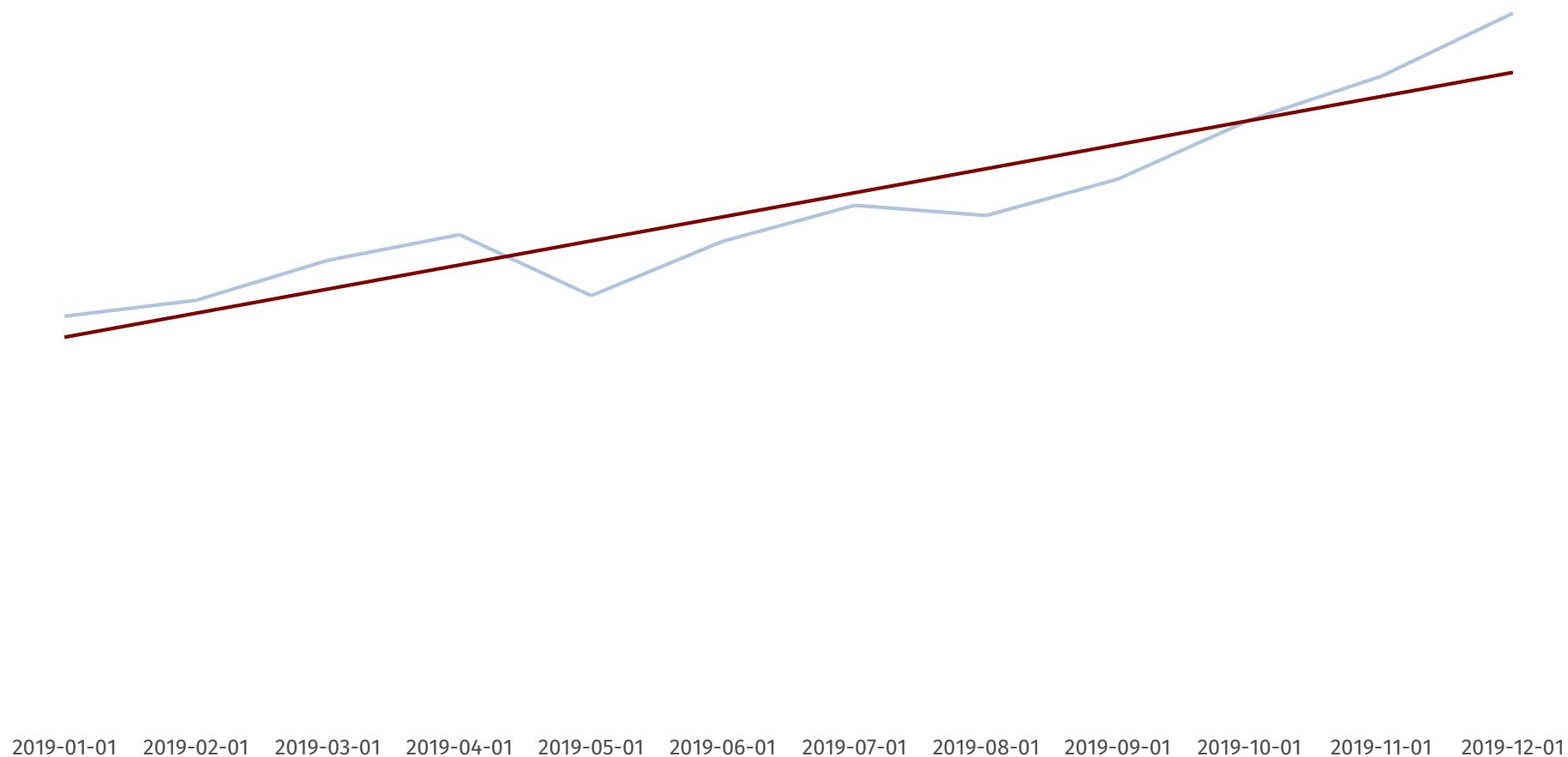
AAPL Closing Price



Scatter Chart, No Values

```
dchart -bar=f -scatter -val=f AAPL.d
```

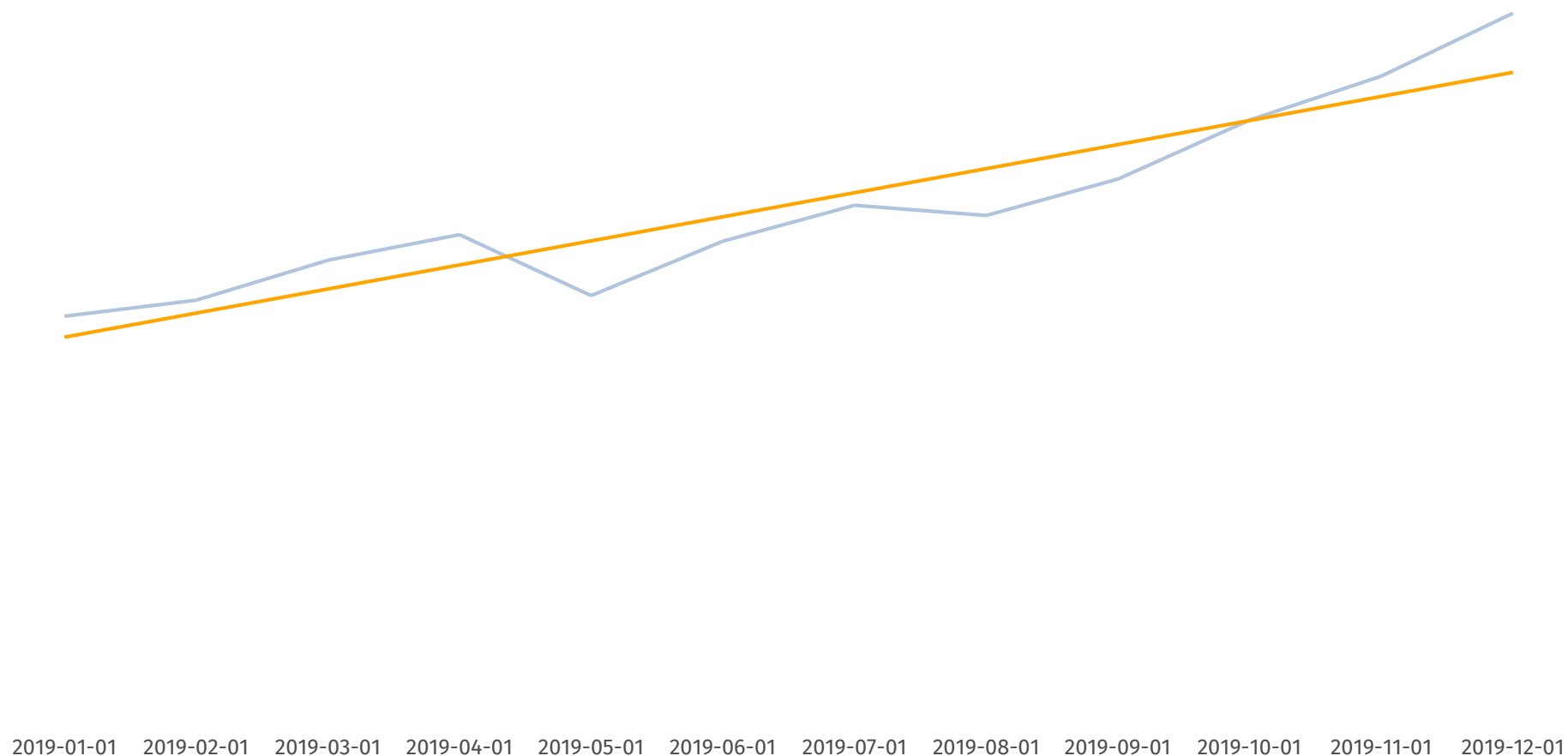

AAPL Closing Price



Line Chart, No Values, Regression Line

```
dchart -bar=f -line -val=f -rline AAPL.d
```

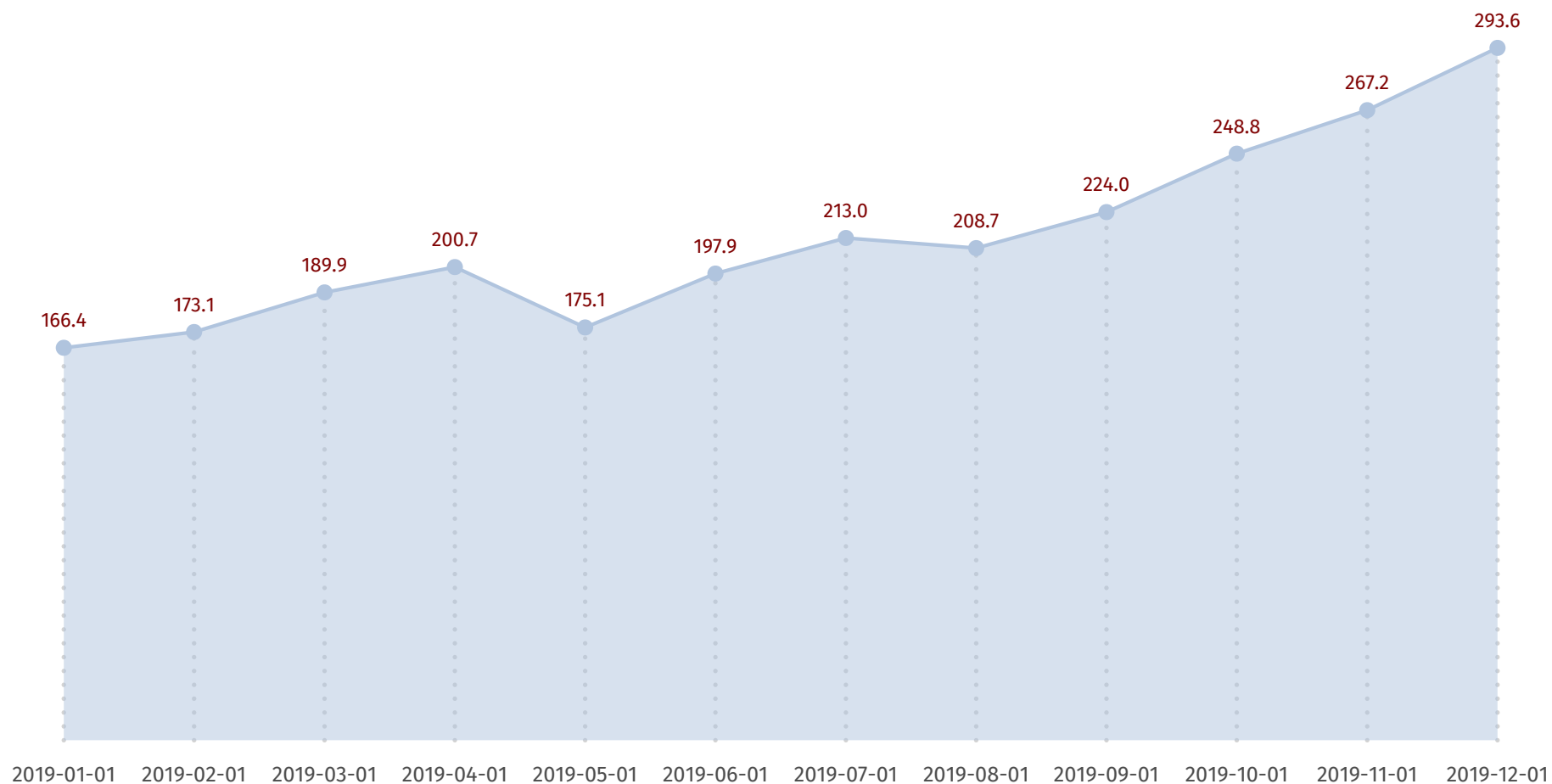
AAPL Closing Price



Line Chart, No Values, Regression Line Color

```
dchart -bar=f -line -val=f -rlcolor=orange AAPL.d
```

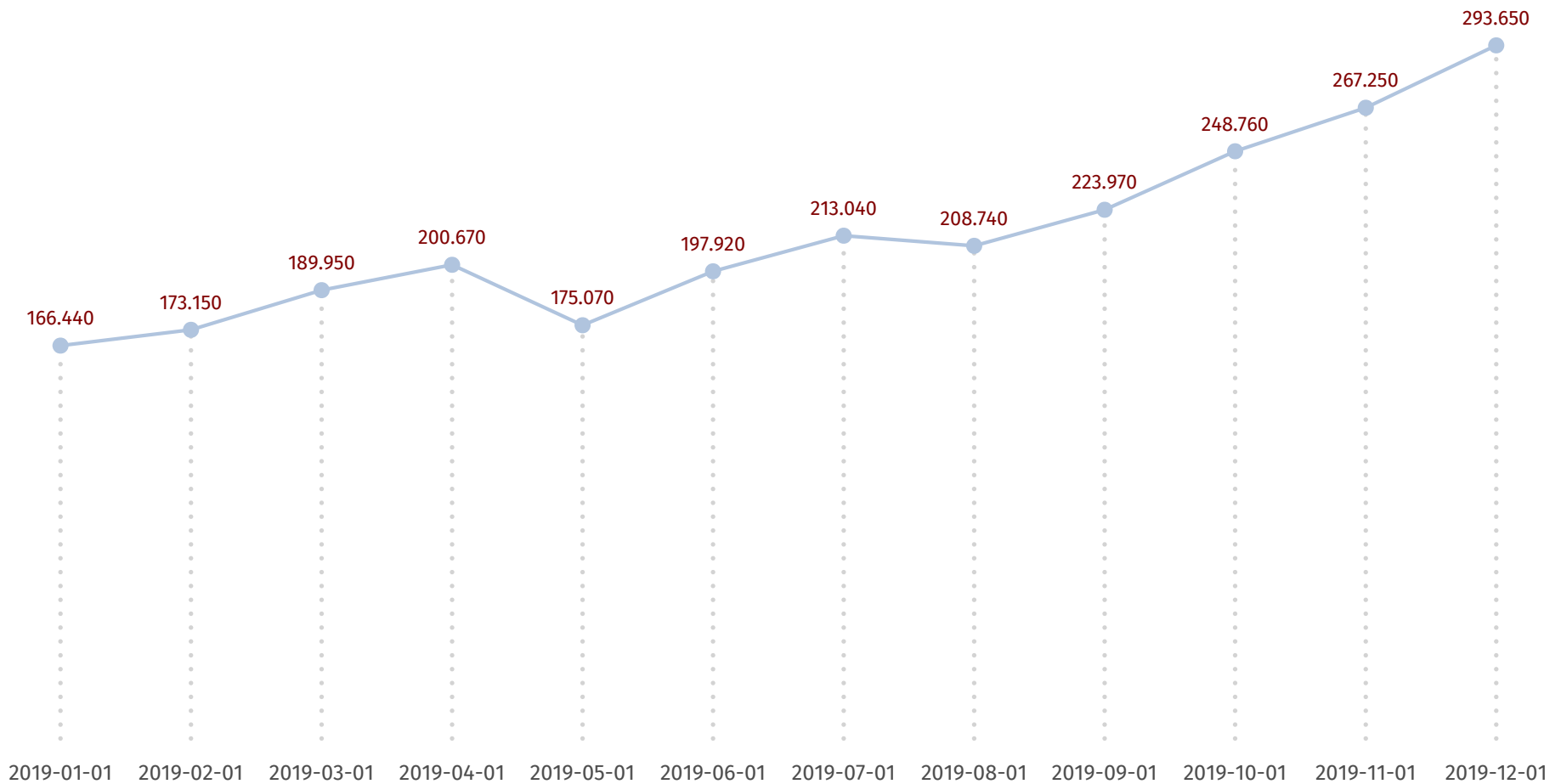
AAPL Closing Price



Volume, Line, Dot

```
dchart -bar=f -line -vol -dot AAPL.d
```

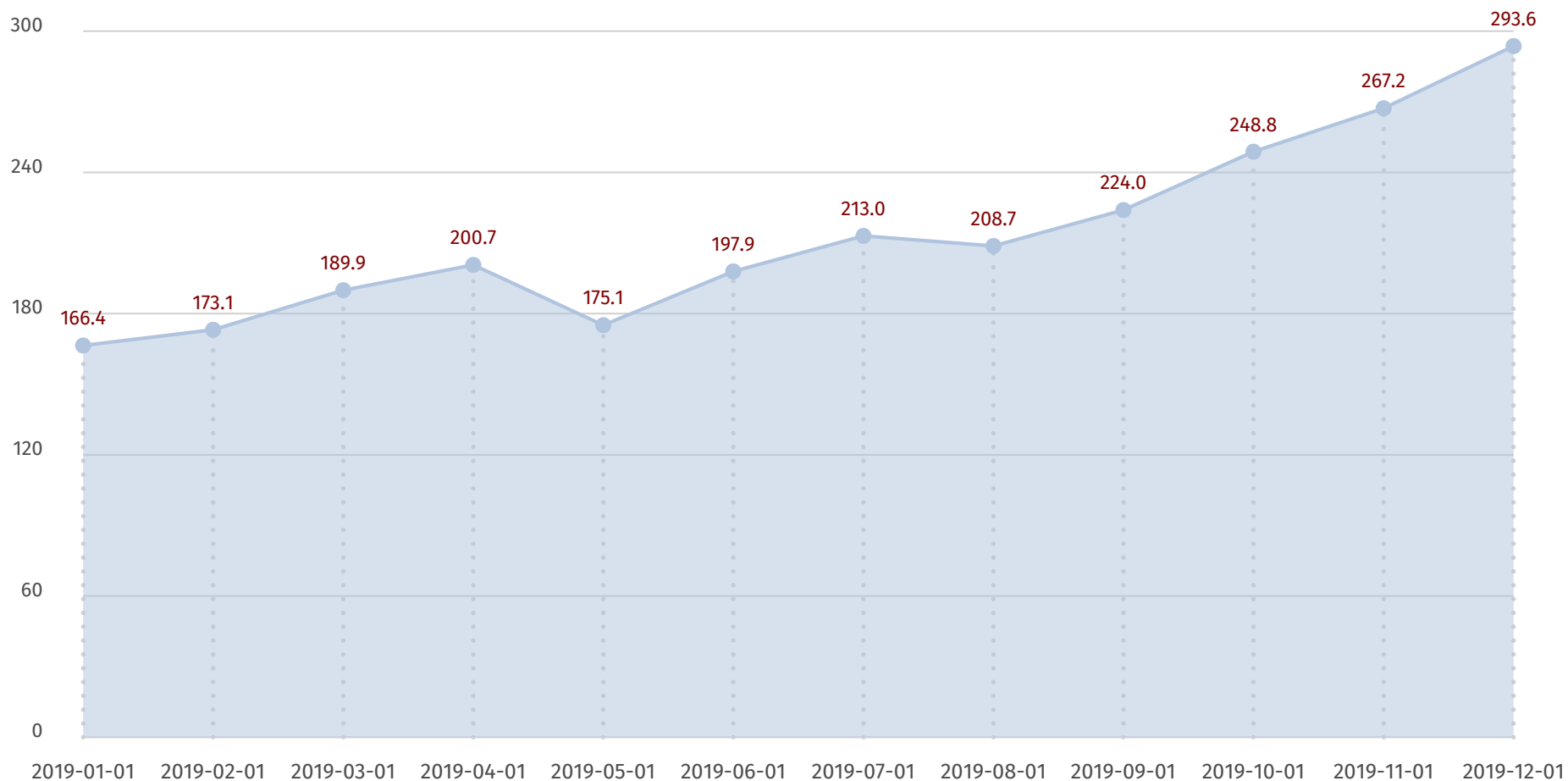
AAPL Closing Price



Dot, Line, Data Format

```
dchart -datafmt %0.3f -bar=f -dot -line AAPL.d
```

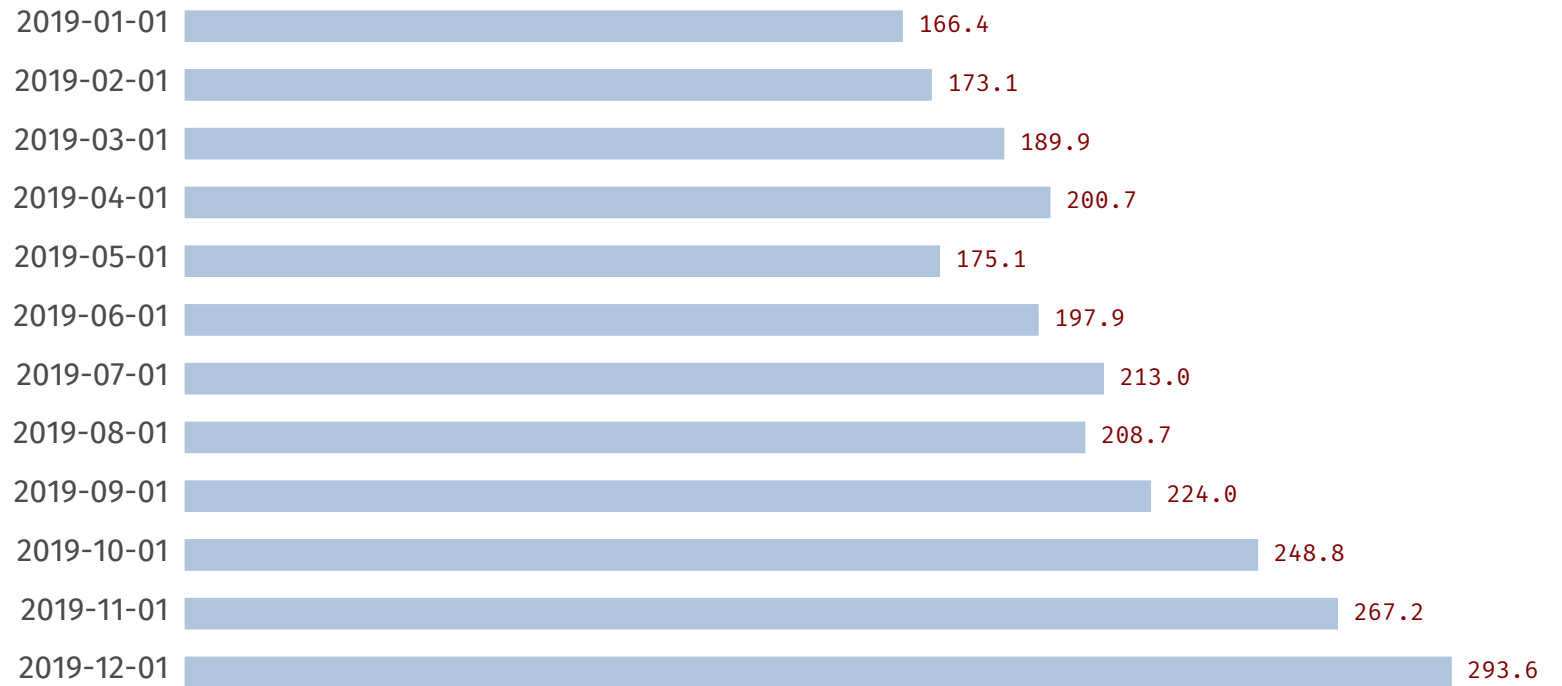
AAPL Closing Price



Line, Area, Dot, Y-Axis, Grid

dchart -bar=f -line -vol -dot -grid -yaxis AAPL.d

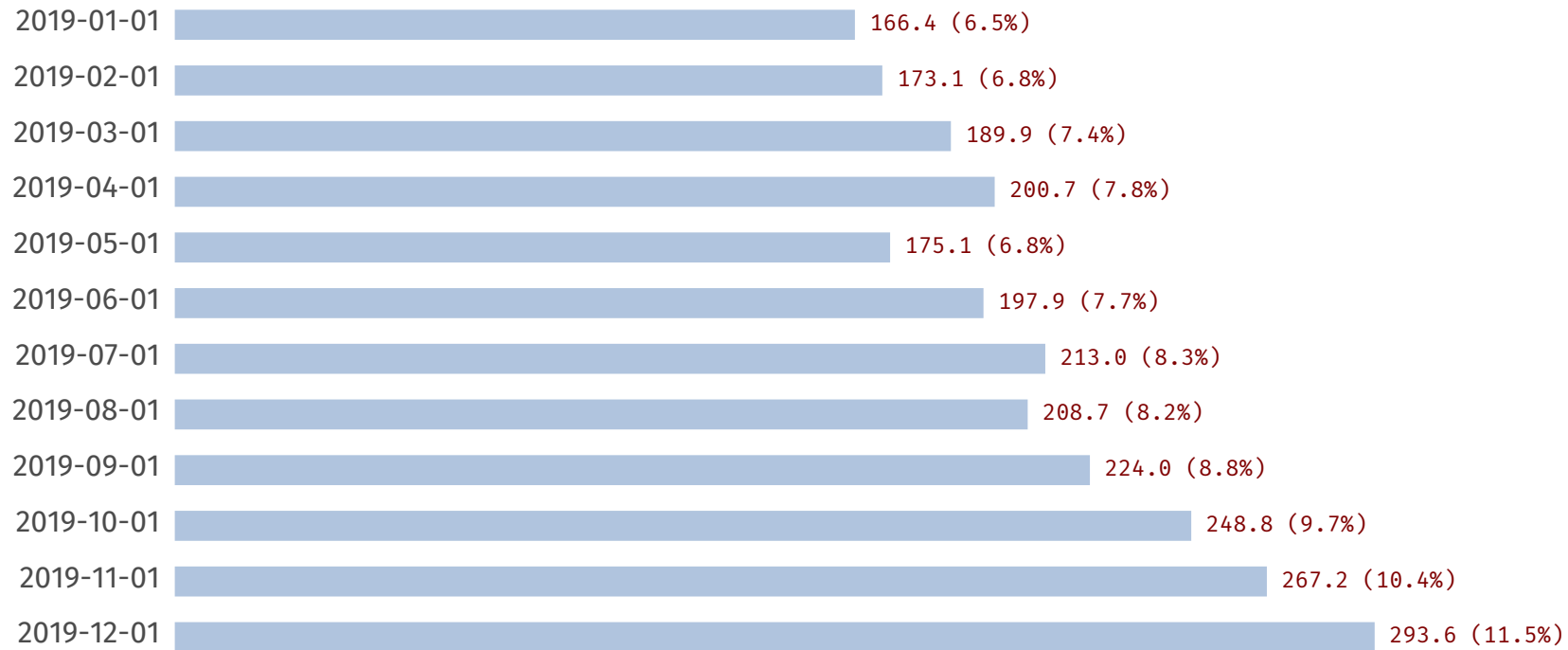
AAPL Closing Price



Horizontal Bar

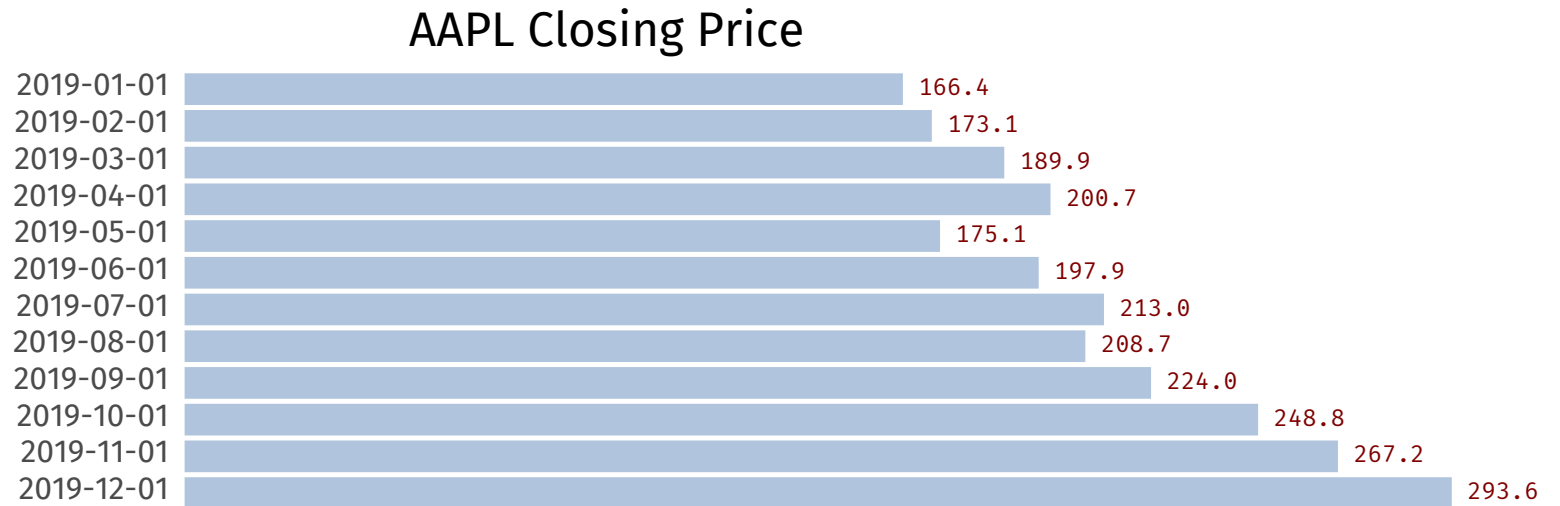
`dchart -hbar AAPL.d`

AAPL Closing Price



Horizontal Bar, Show Percentages

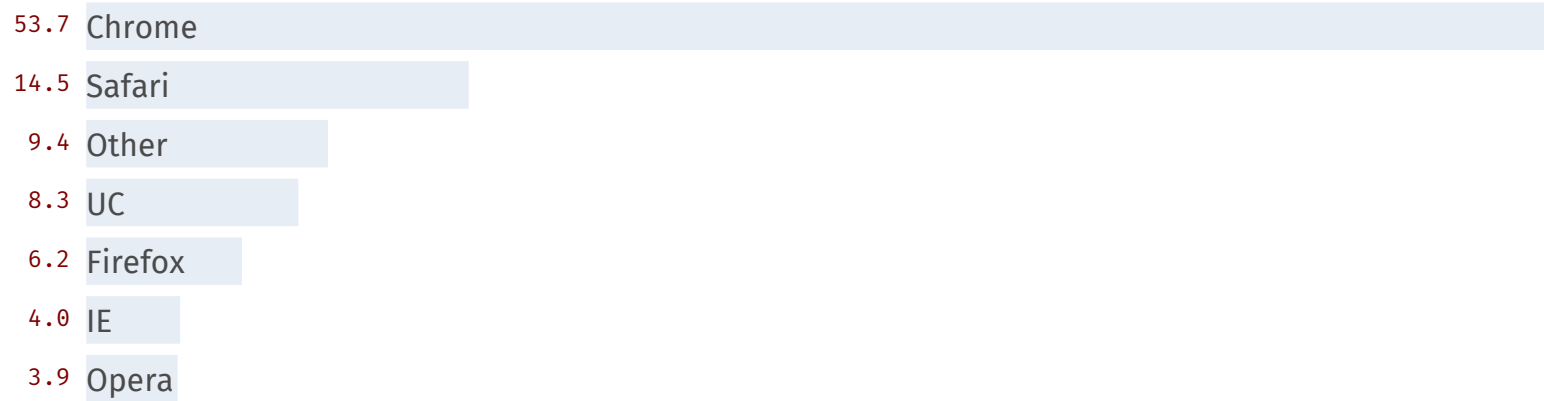
`dchart -hbar -pct AAPL.d`



Horizontal Bar, Line Spacing

```
dchart -hbar -ls 1.5 AAPL.d
```

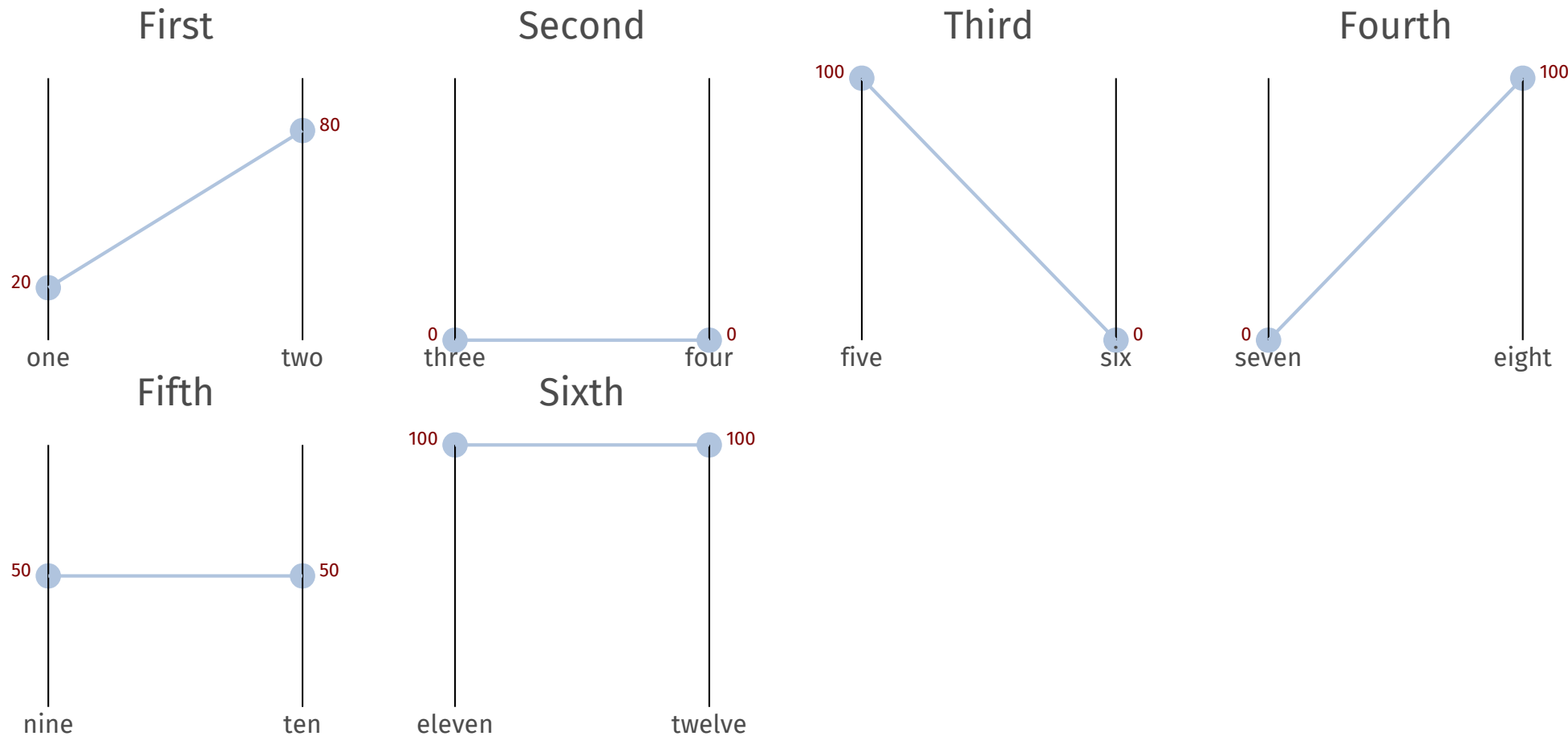

Browser Market Share Dec 2016-Dec 2017



Word Bar

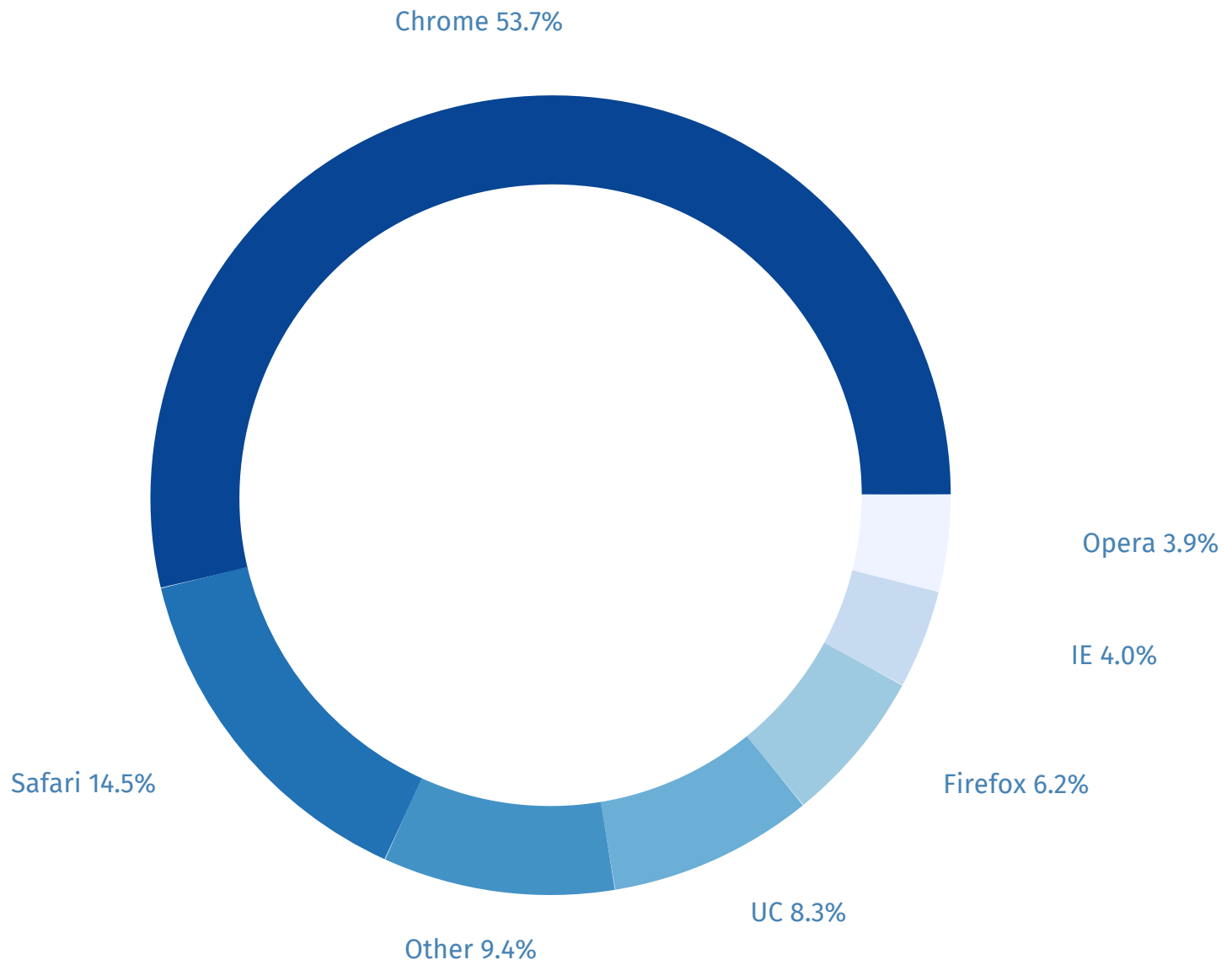
dchart -wbar AAPL.d

Test Slope Graphs



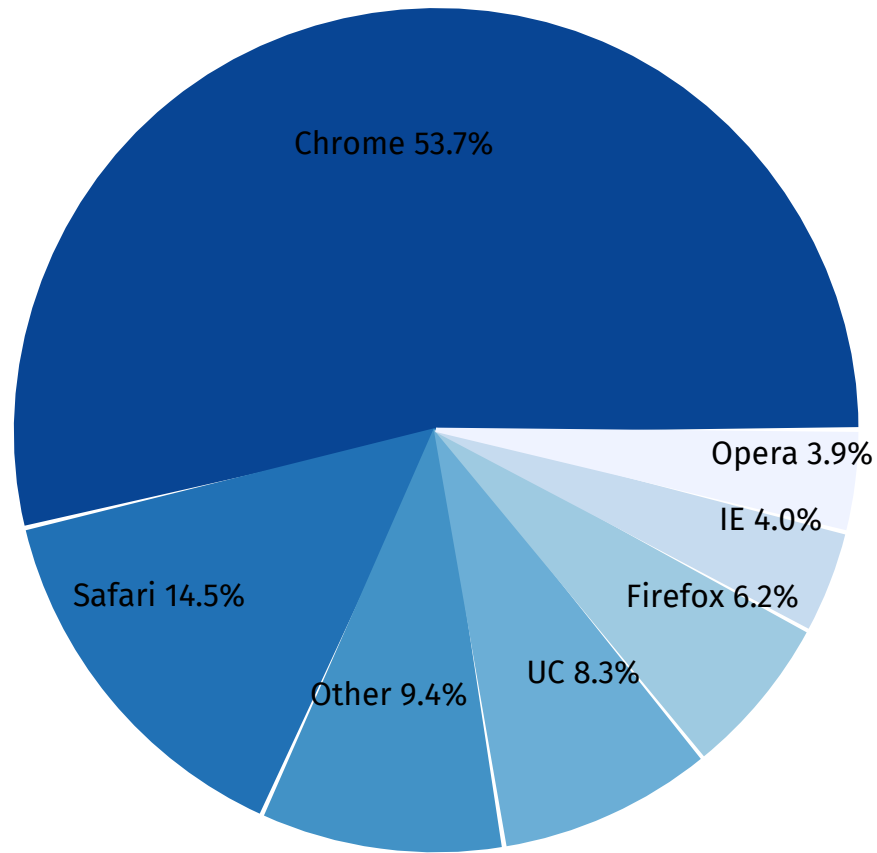
Slope Chart

```
dchart -left=10 -right=25 -top=80 -bottom=60 -slope slope.d
```



Donut

```
dchart -donut -color=std -pwidth=5 browser.d
```



Pie

```
dchart -donut -color=std -title=f -top=70 -pwidth=20 -psize=20 browser.d
```

Browser Market Share Dec 2016-Dec 2017



Pmap

```
dchart -pmap -pwidth=5 -textsize=1 browser.d
```

Browser Market Share Dec 2016-Dec 2017



Pmap with Solid Colors

```
dchart -pmap -pwidth=5 -textsize=1 -solidpmap browser.d
```

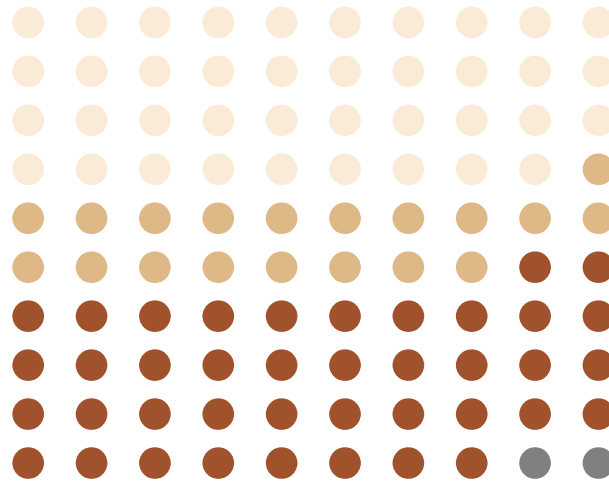
Browser Market Share Dec 2016-Dec 2017



Pmap with Solid Colors, Length Threshold

```
dchart -pmap -pwidth=5 -textsize=1 -solidpmap -pmlen=30 browser.d
```

US Incarceration Rate

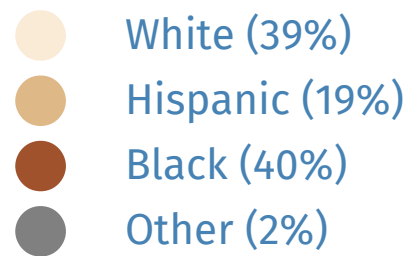
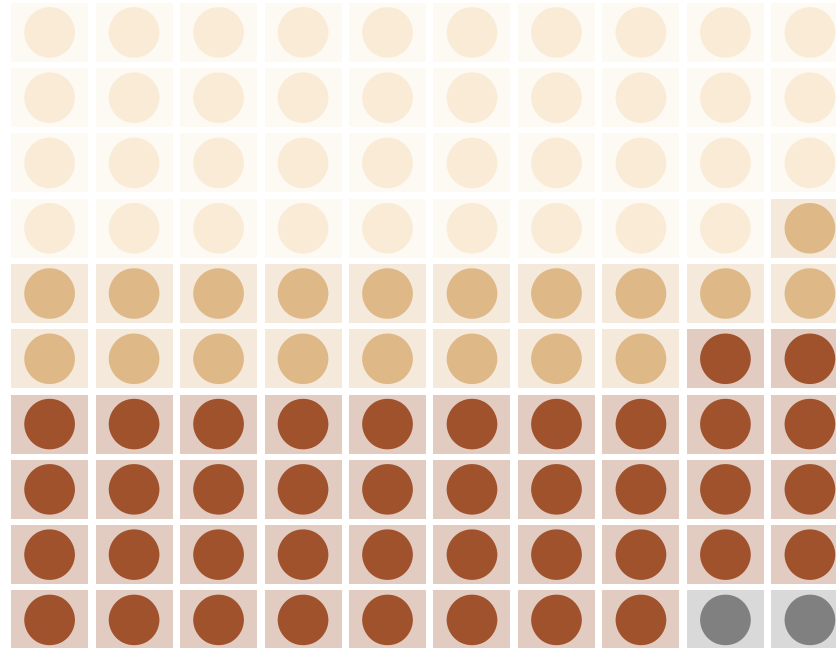


- White (39%)
- Hispanic (19%)
- Black (40%)
- Other (2%)

Pgrid

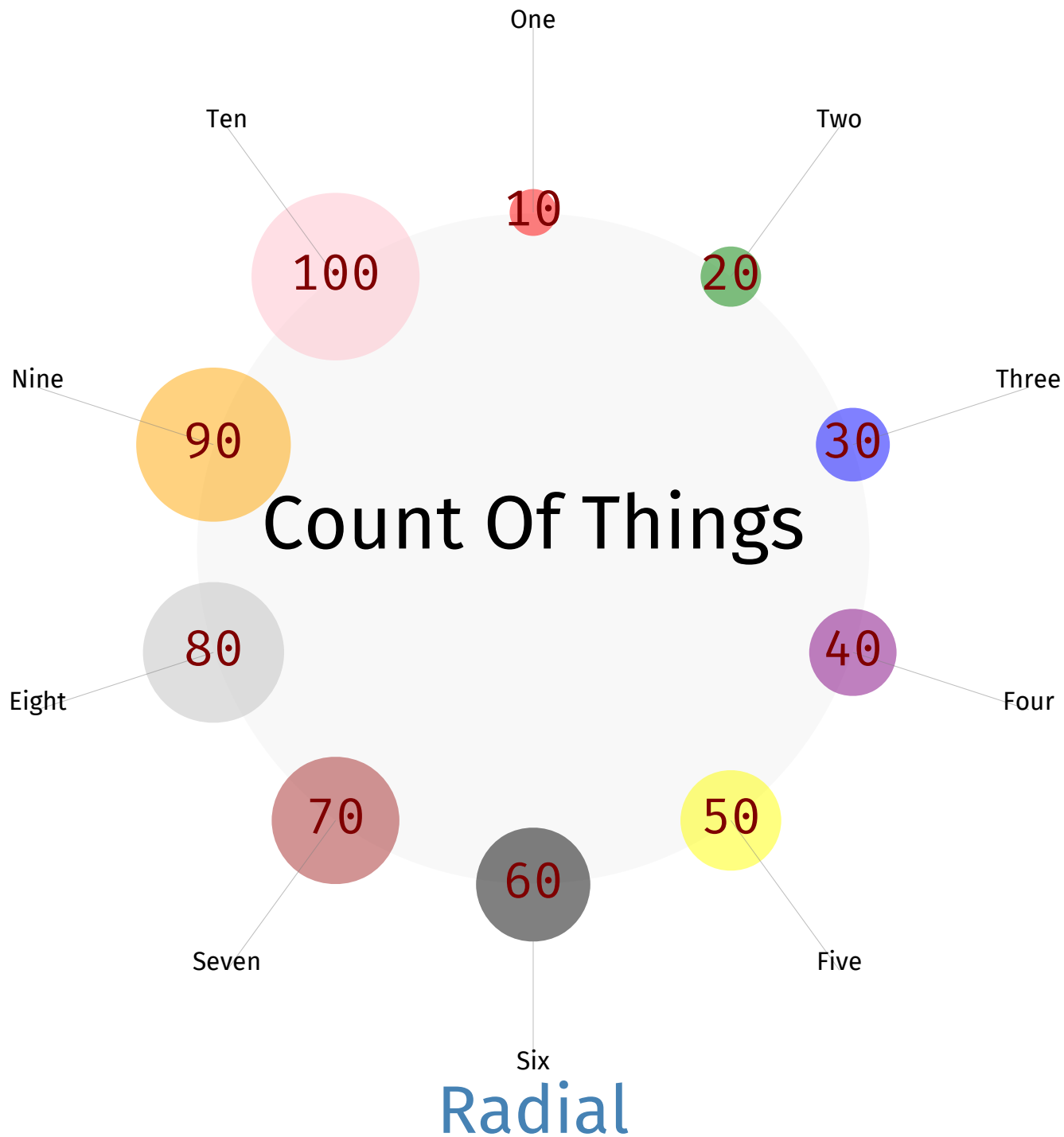
```
dchart -left 35 -top 80 -ls 3 -pgrid -val=f incar.d
```


US Incarceration Rate

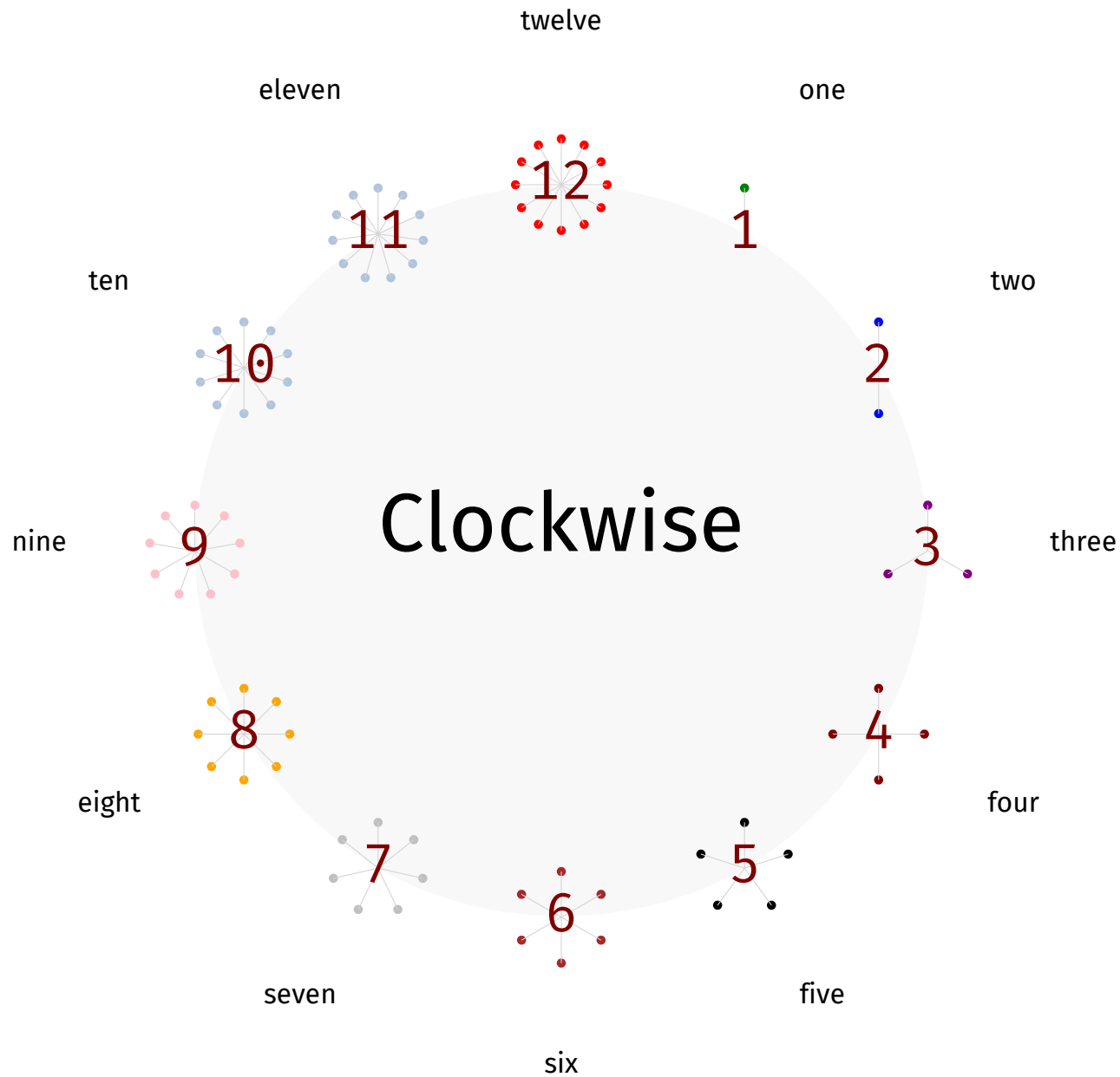


Lego

```
dchart -left 30 -top 80 -textsize 4 -lego incar.d
```



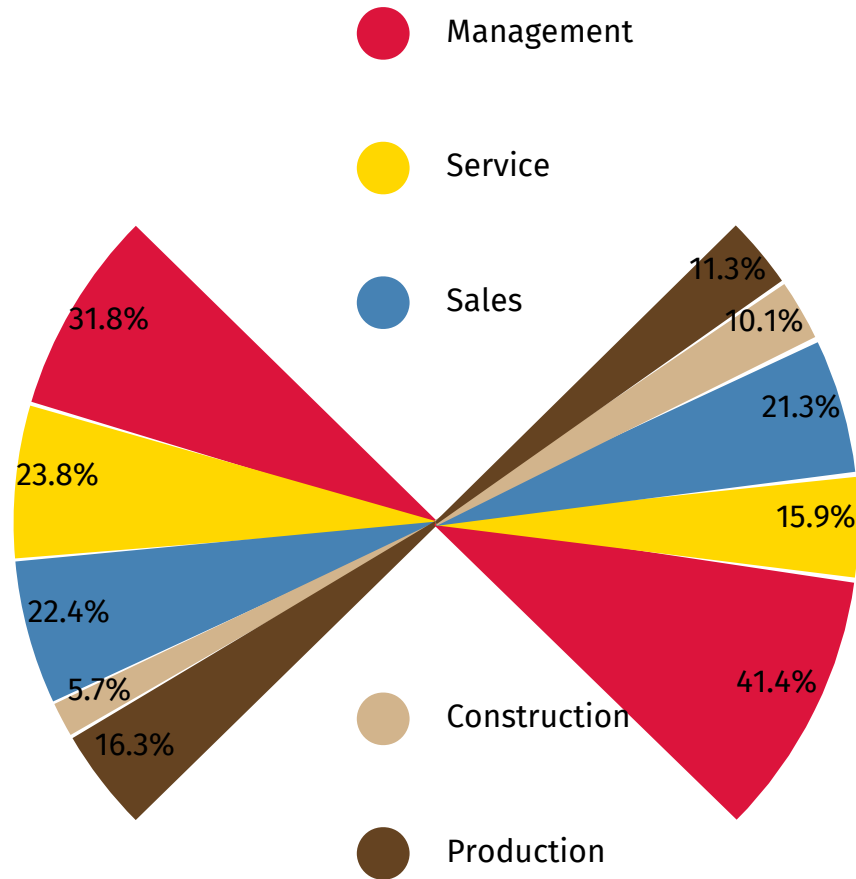
```
dchart -radial -psize=10 -pwidth=20 -top=55 -textsize=3 count.d
```



Radial with Spokes

```
dchart -radial -psize=5 -pwidth=20 -top=55 -textsize=3 -spokes clock.d
```

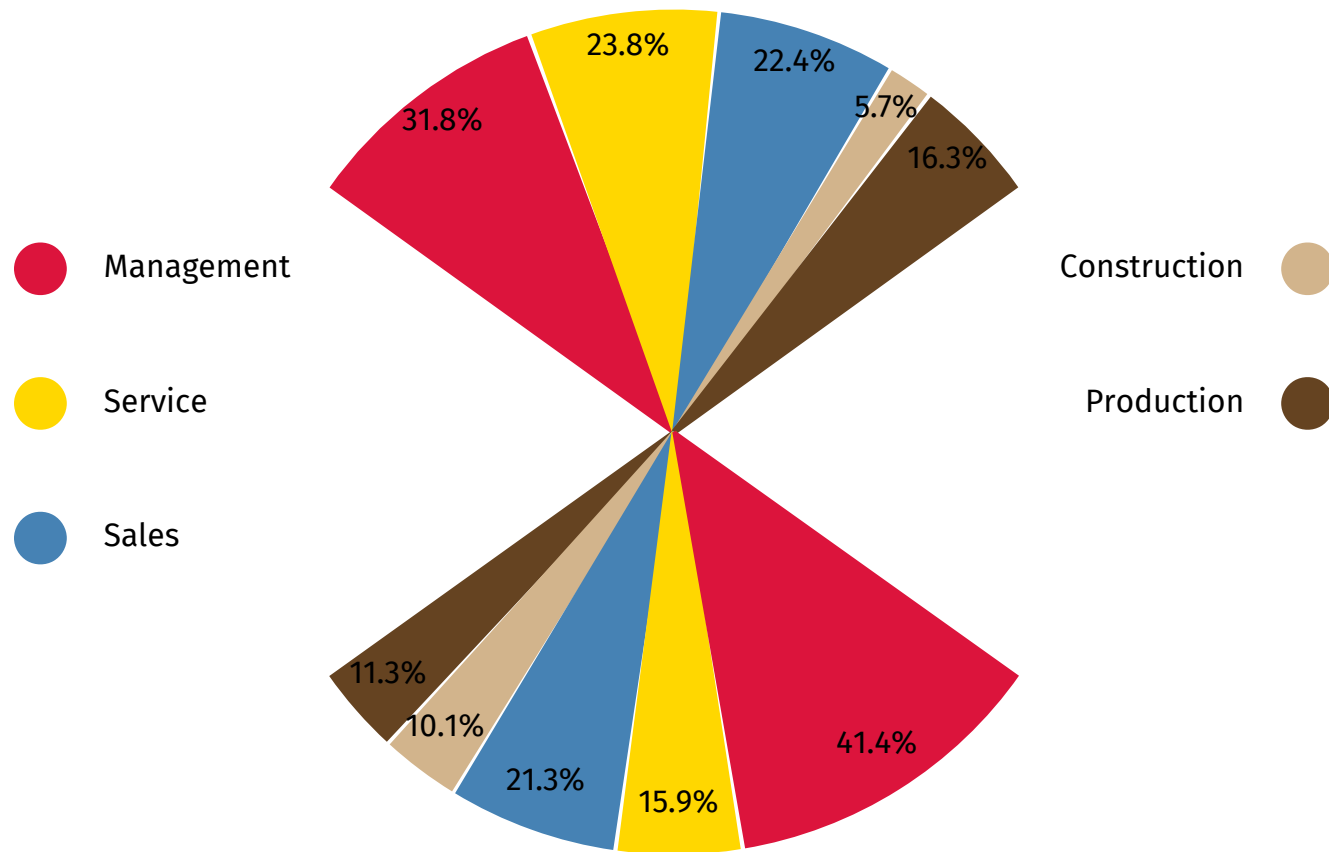
Occupations of African-Americans and Whites (2019)



Bowtie chart

```
dchart -val=t -psize=20 -top=60 -bowtie occupation.d
```

Occupations of African-Americans and Whites (2019)



Fan chart

```
dchart -val=t -psize=20 -top=60 -fan occupation.d
```