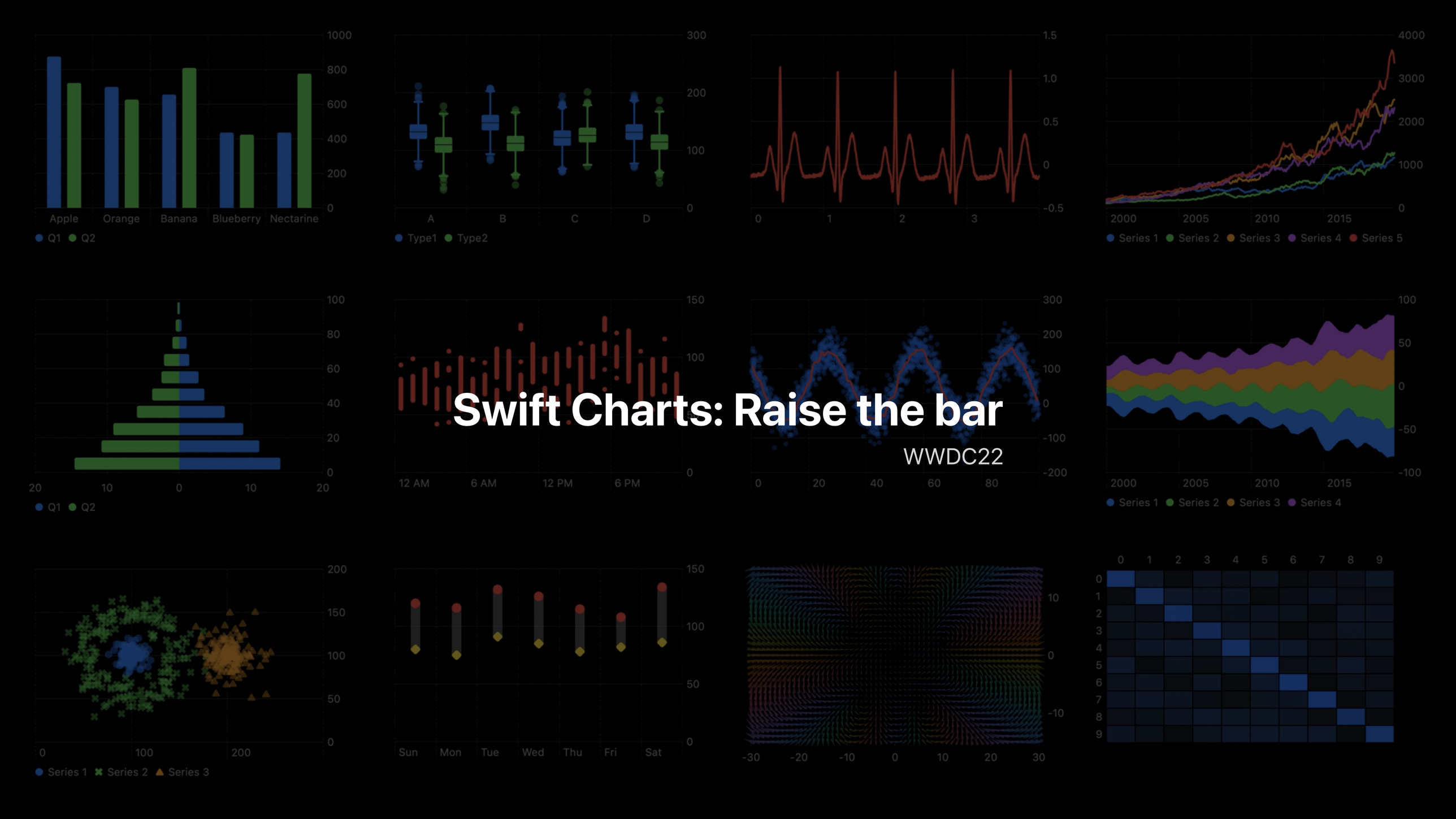
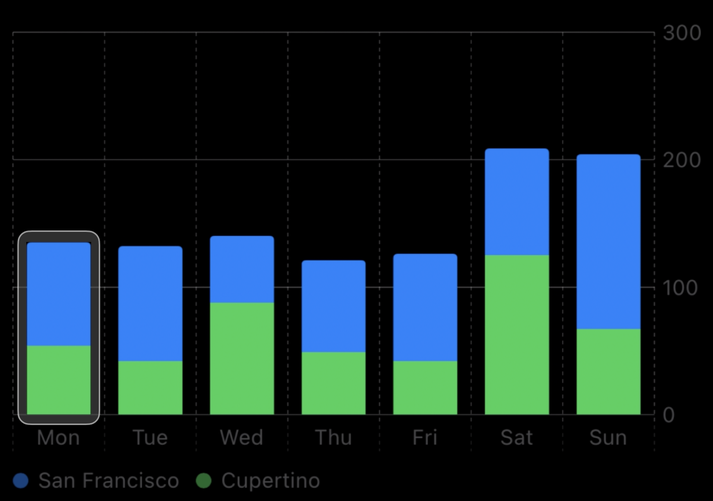
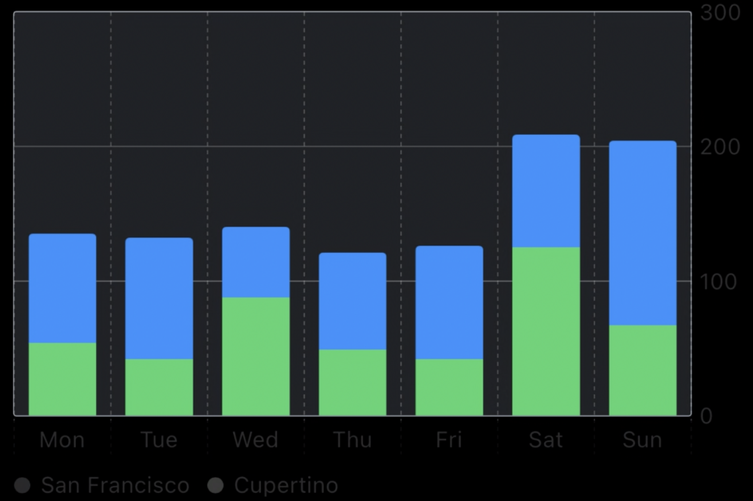

Swift Charts: Raise the bar
WWDC22
Apple
Orange
Banana
Blueberry
Nectarine
Q1
Q2
A
B
C
D
Type1
Type2
Series 1
Series 2
Series 3
Series 4
Series 5
12 AM
6 AM
12 PM
6 PM
Sun
Mon
Tue
Wed
Thu
Fri
Sat



Marks

Axes, Legends



Plot Area

How much can we customize it?

7:56

삼성전자

56,200원

1년 전보다 -23,800원 (29.7%)

1일 1주 3달 1년 5년

호가

판매대기 499,604주

구매대기 1,604,963주

판매하기 구매하기

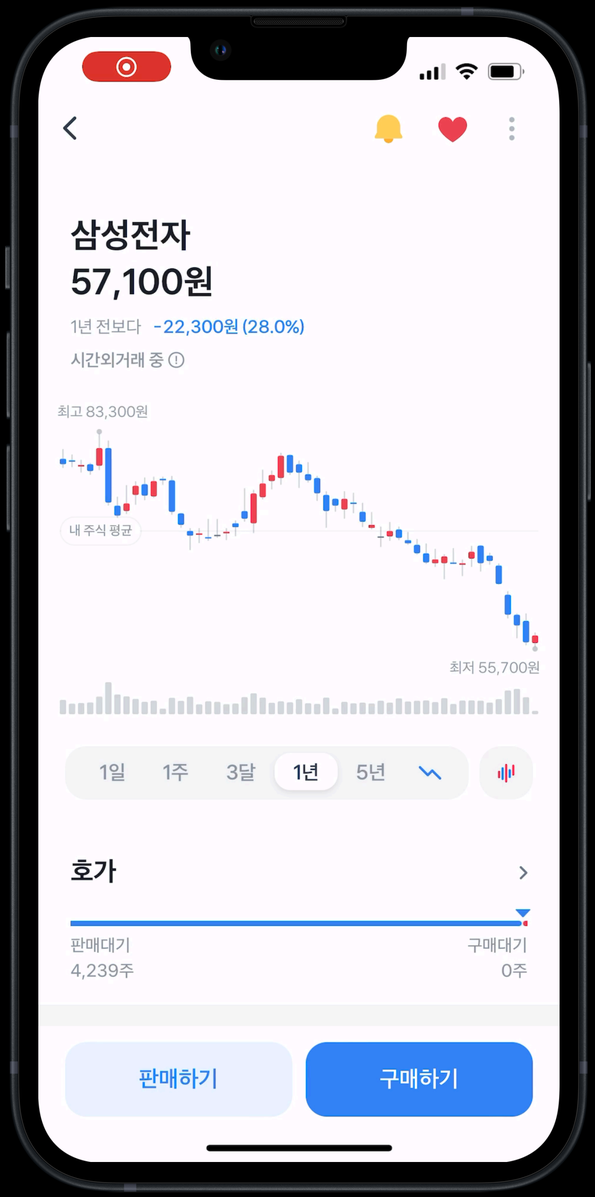
3:01

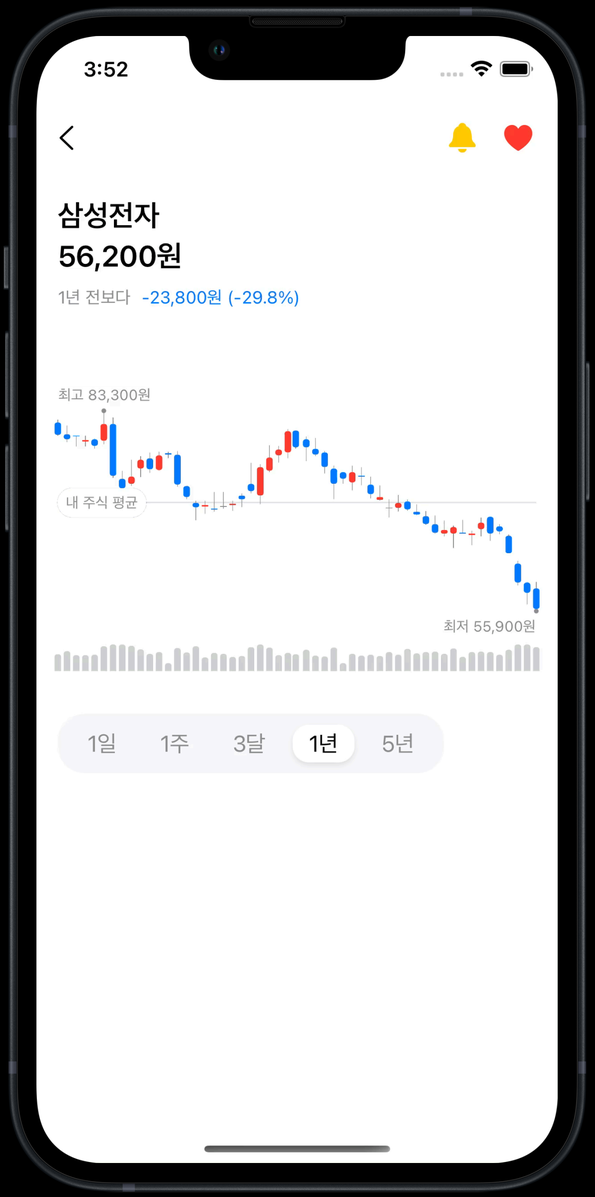
삼성전자

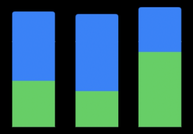
56,200원

1년 전보다 -23,800원 (-29.8%)

1일 1주 3달 1년 5년


삼성전자
57,100원
1년 전보다 -22,300원 (28.0%)
시간외거래 중
최고 83,300원
내 주식 평균
최저 55,700원
1일 1주 3달 1년 5년
호가
판매대기
4,239주
구매대기
0주
판매하기
구매하기


3:52
삼성전자
56,200원
1년 전보다 -23,800원 (-29.8%)
최고 83,300원
내 주식 평균
최저 55,900원
1일 1주 3달 1년 5년

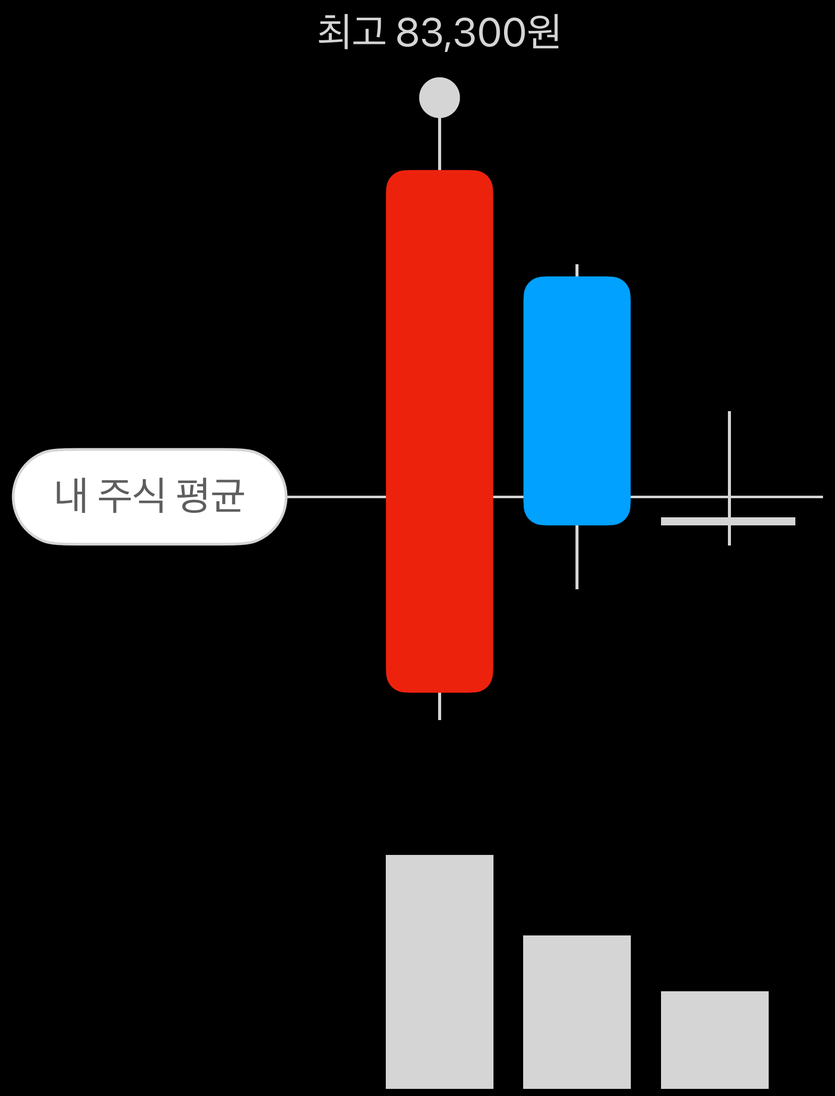


Marks Customization

1. Marks Type
2. Marks Parameters
3. Marks Modifiers

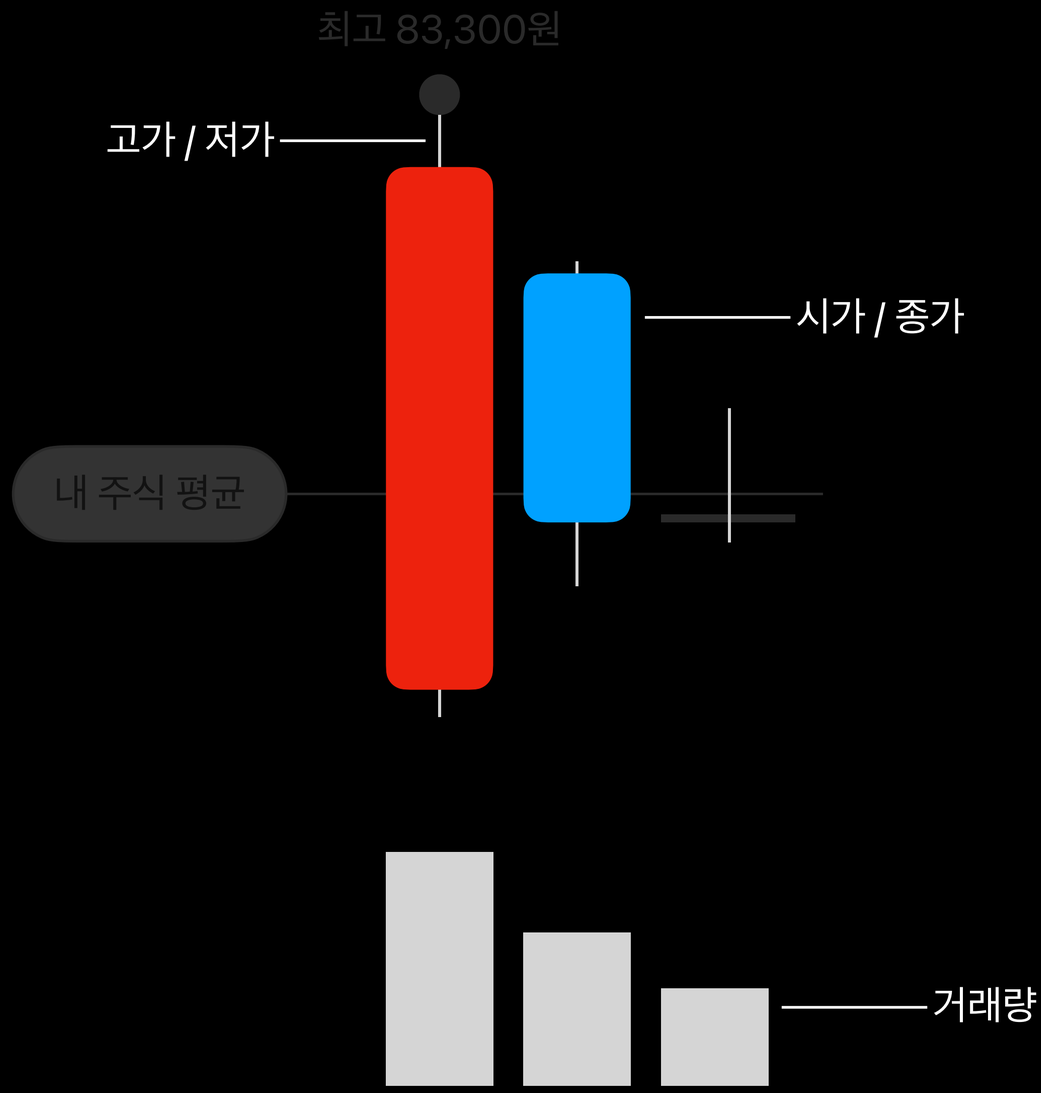
Marks Type

Bar Line Point Area Rule Rectangle

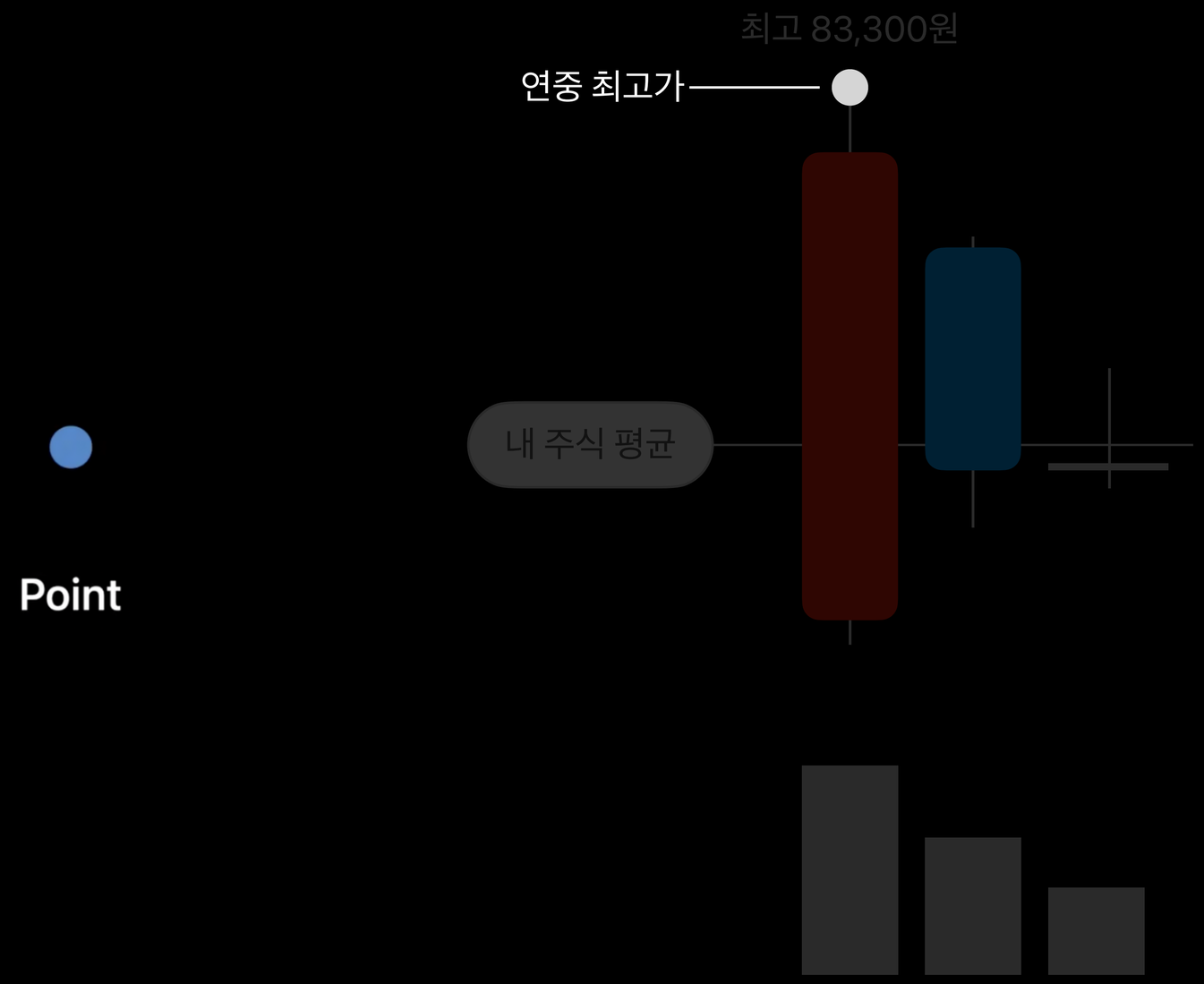

최고 83,300원
내 주식 평균

Marks Type

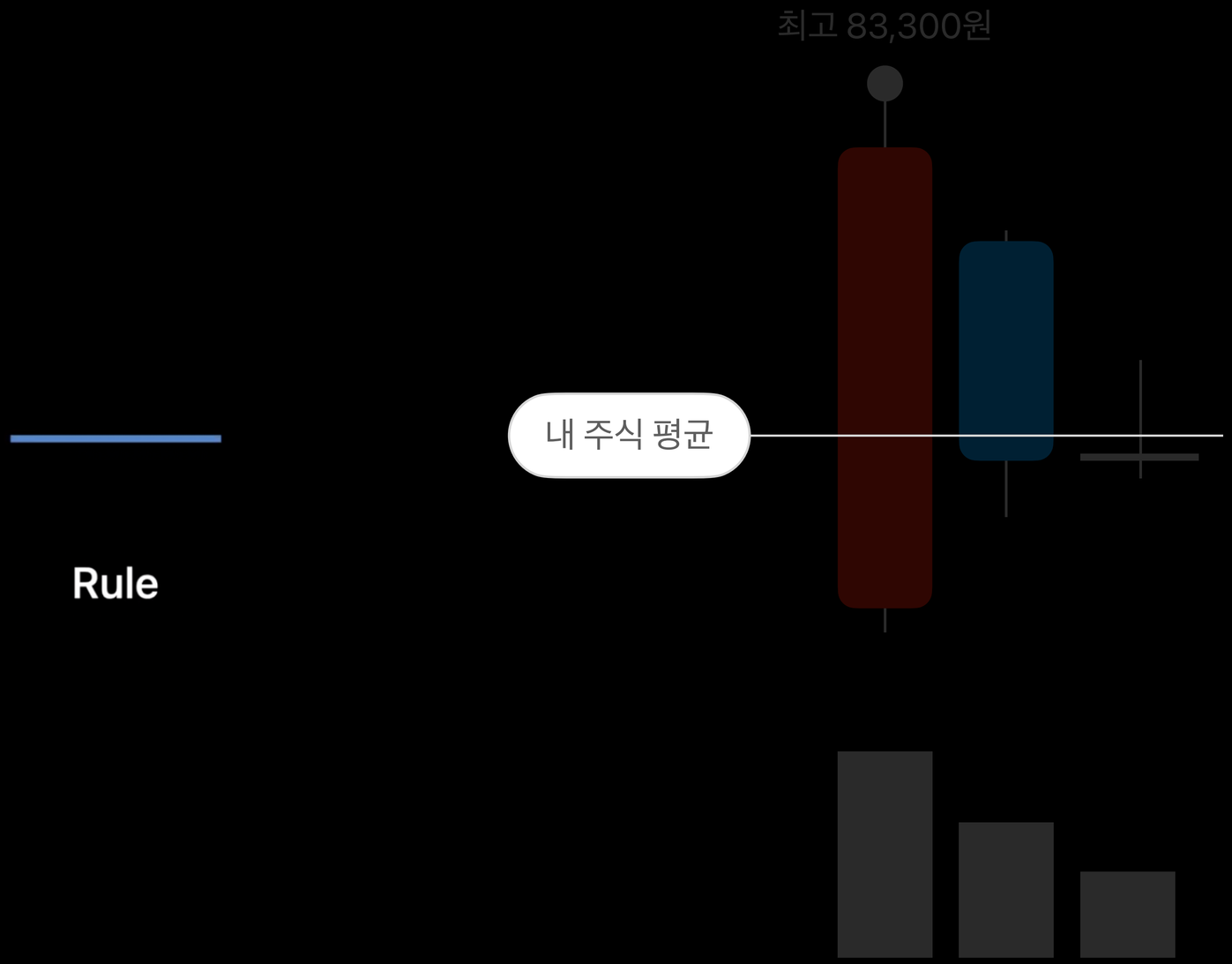
Bar



Marks Type



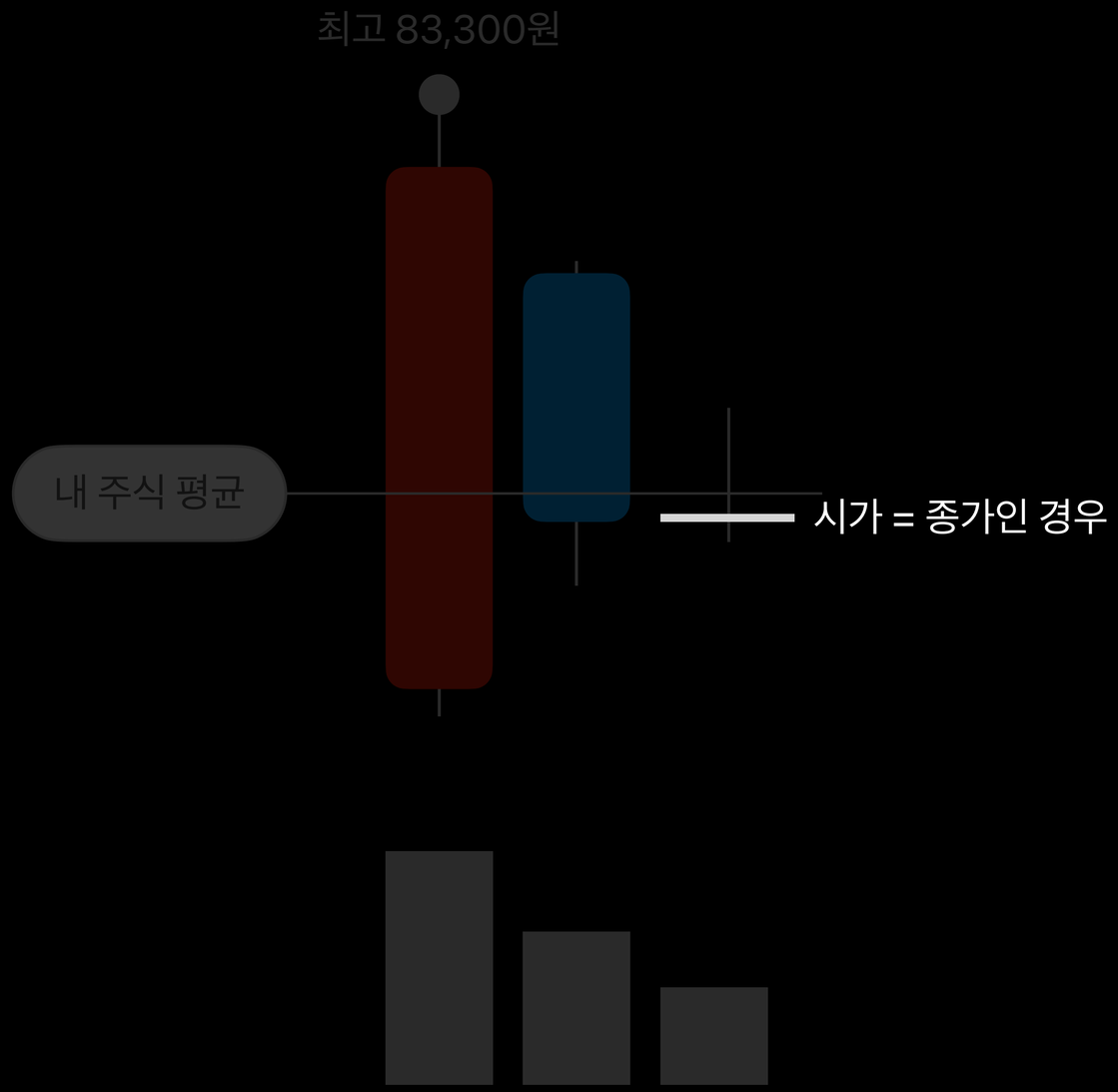
Marks Type



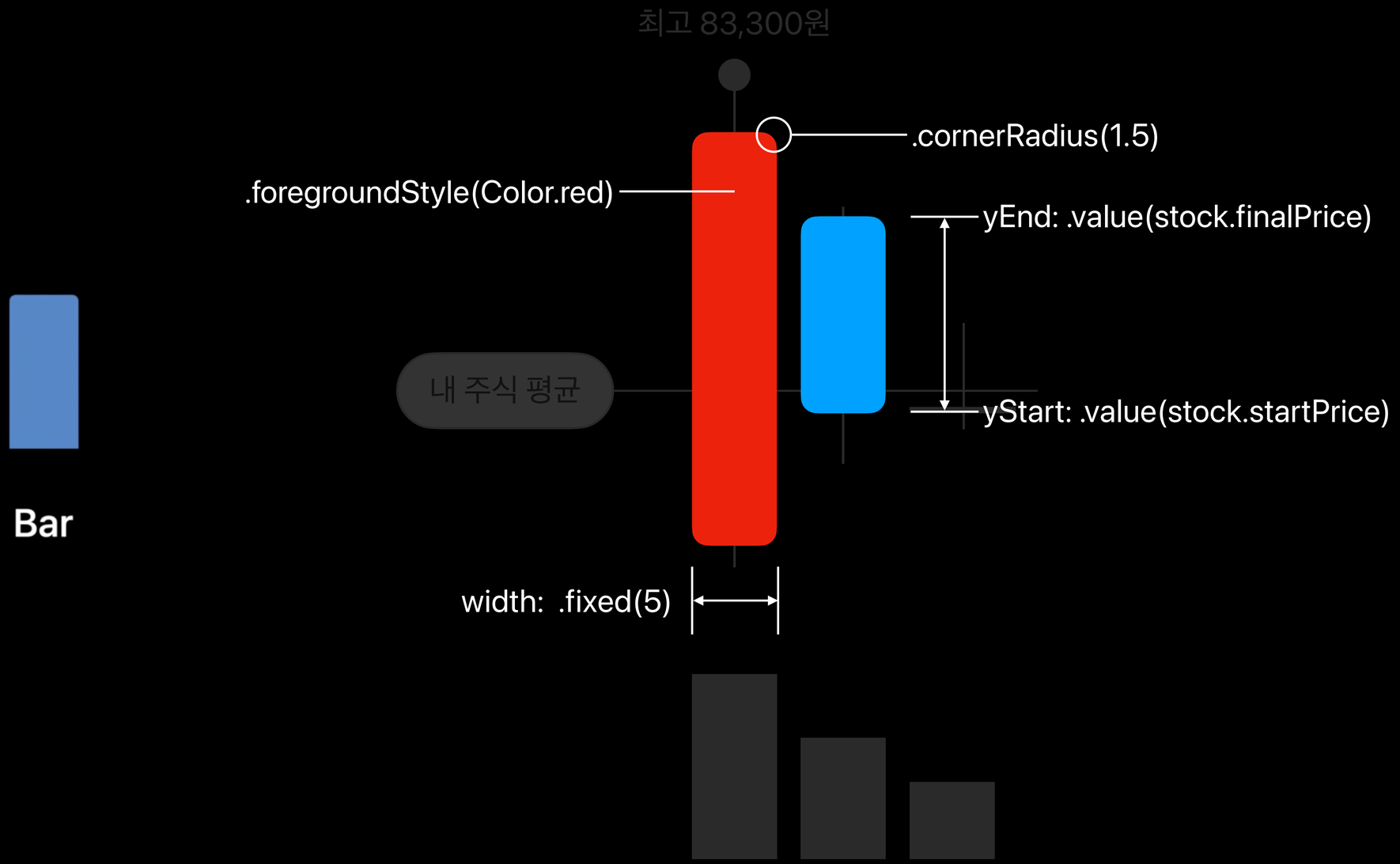
Rule

Marks Type

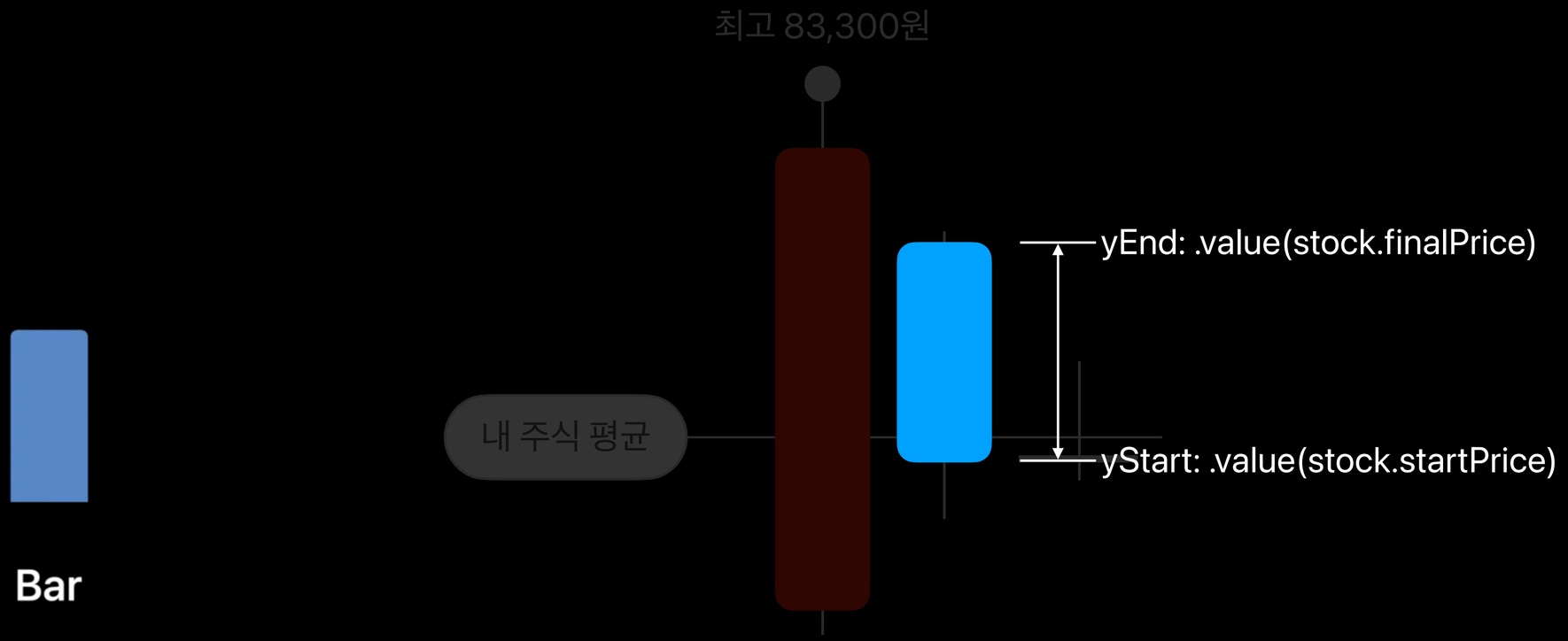
Rectangle



Marks Custom

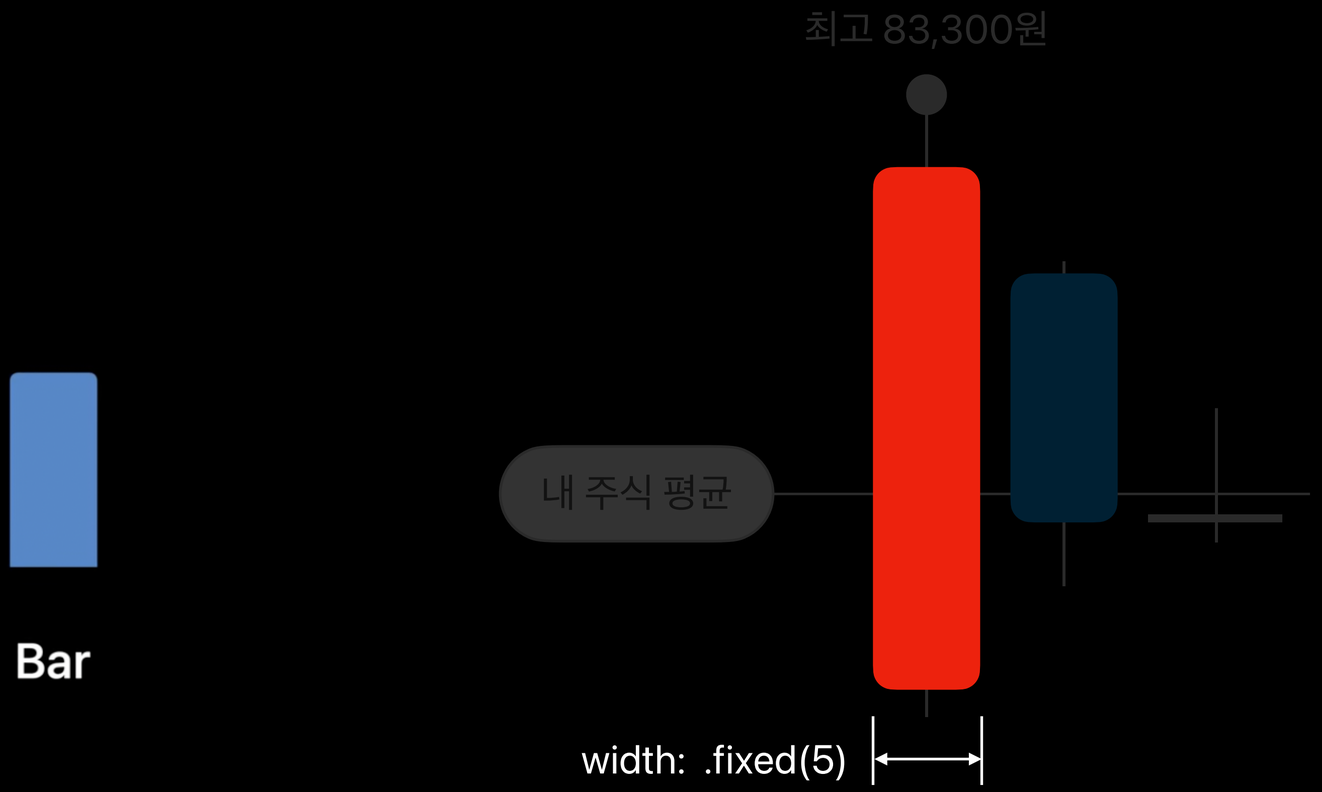


Marks Parameters



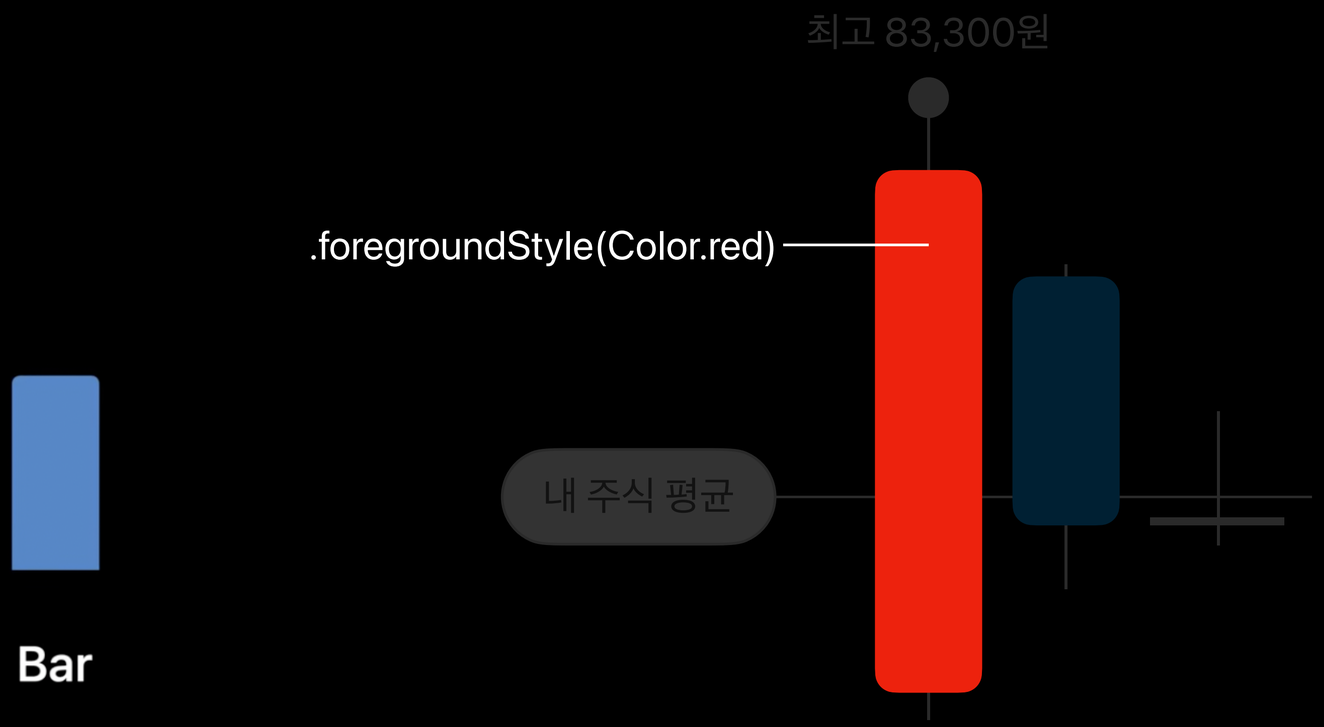

```
BarMark(
    x: .value("Day", stock.date),
    yStart: .value("Price", stock.startPrice),
    yEnd: .value("Price", stock.finalPrice),
    width: .fixed(5)
)
.foregroundStyle(stock.startPrice < stock.finalPrice ? Color.red : Color.blue)
.cornerRadius(1.5)
```

Marks Parameters



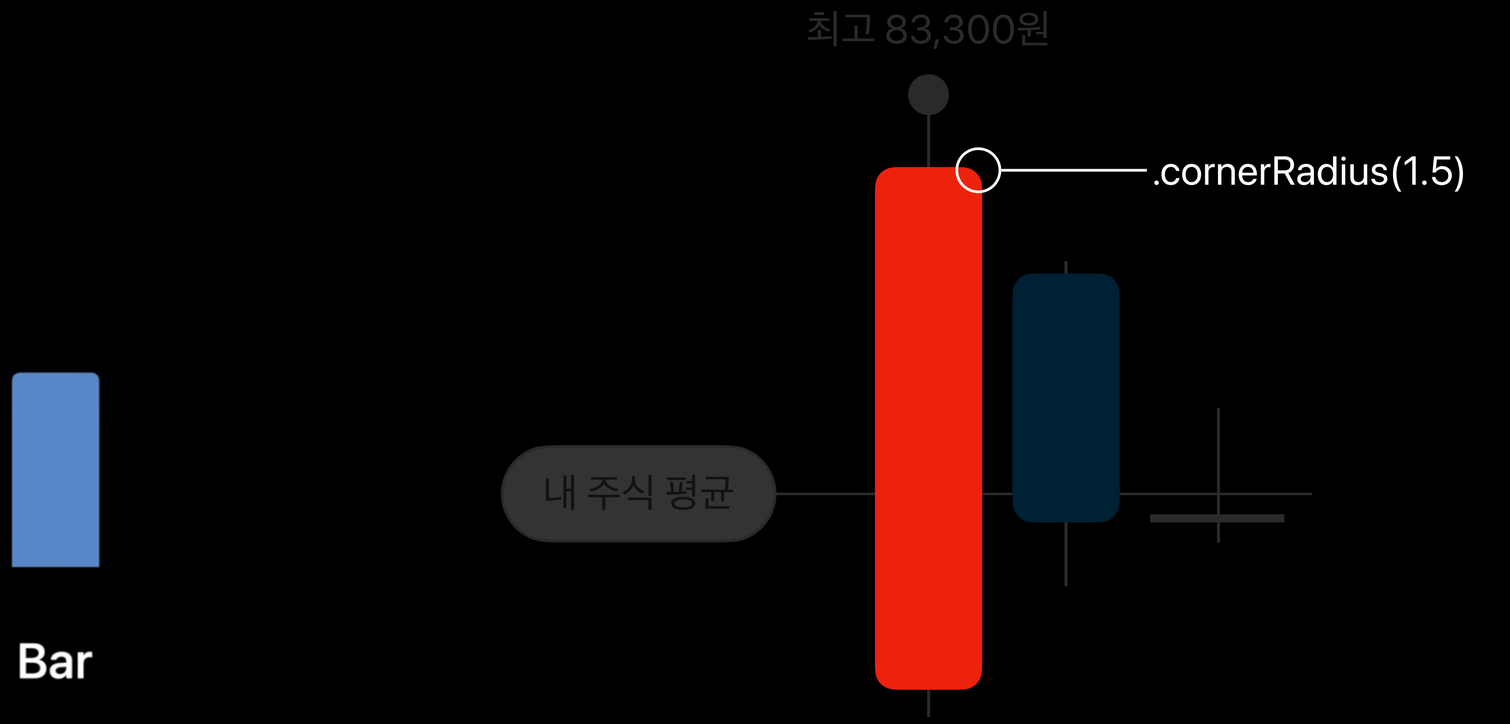

```
BarMark(
    x: .value("Day", stock.date),
    yStart: .value("Price", stock.startPrice),
    yEnd: .value("Price", stock.finalPrice),
    width: .fixed(5)
)
.foregroundStyle(stock.startPrice < stock.finalPrice ? Color.red : Color.blue)
.cornerRadius(1.5)
```

Marks Modifiers




```
BarMark(
    x: .value("Day", stock.date),
    yStart: .value("Price", stock.startPrice),
    yEnd: .value("Price", stock.finalPrice),
    width: .fixed(5)
)
.foregroundStyle(stock.startPrice < stock.finalPrice ? Color.red : Color.blue)
.cornerRadius(1.5)
```

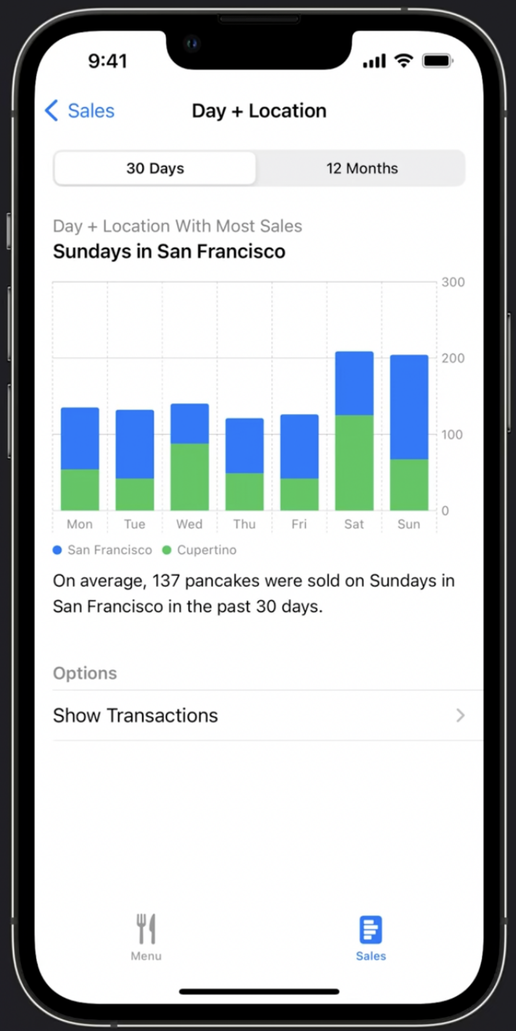
Marks Modifiers

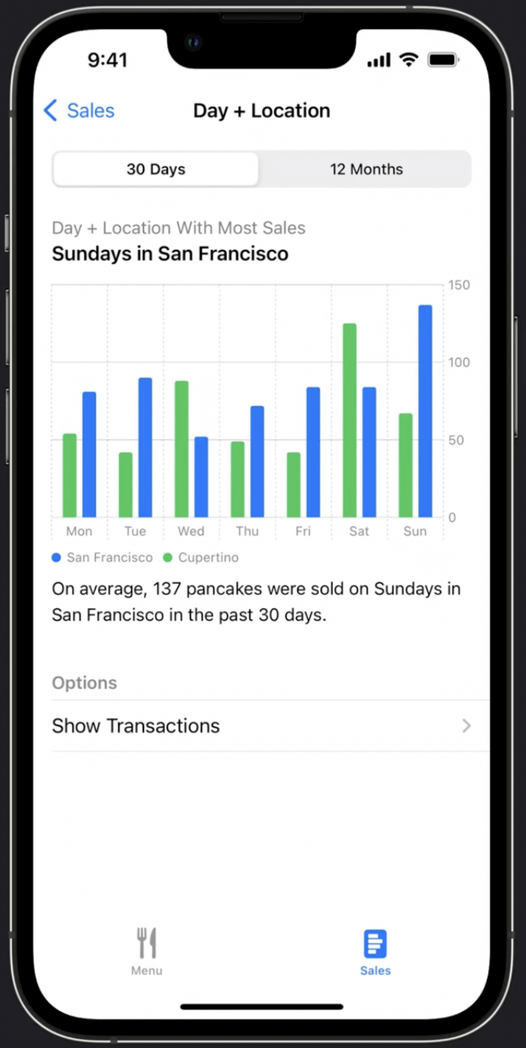



```
BarMark(
    x: .value("Day", stock.date),
    yStart: .value("Price", stock.startPrice),
    yEnd: .value("Price", stock.finalPrice),
    width: .fixed(5)
)
.foregroundStyle(stock.startPrice < stock.finalPrice ? Color.red : Color.blue)
.cornerRadius(1.5)
```

Marks Modifiers

```
Chart {
    ForEach(seriesData, id: \.city) { series in
        ForEach(series.data, id: \.month) {
            BarMark(
                x: .value("Weekday", $0.weekday, unit: .day),
                y: .value("Sales", $0.sales)
            )
        }
        .foregroundStyle(by: .value("City", series.city))
        .position(by: .value("City", series.city))
    }
}
```

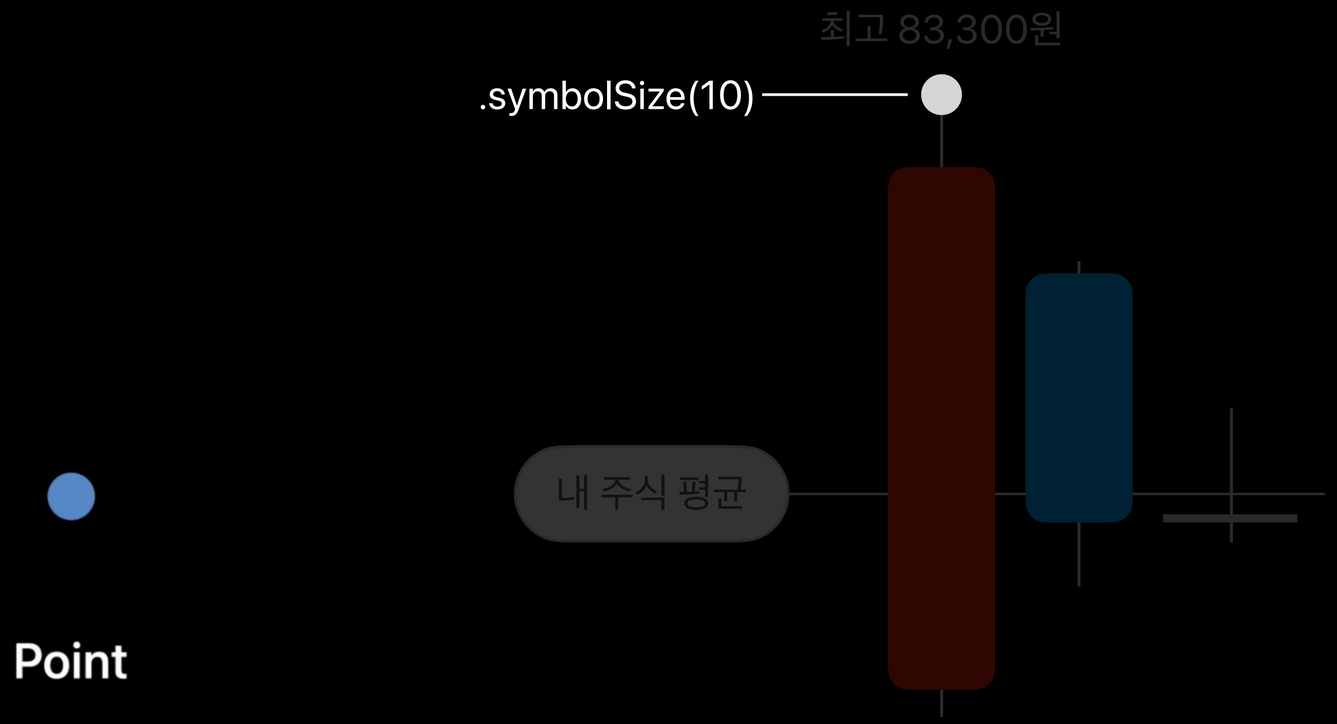




Marks Modifiers

.offset
.opacity
.blendMode
.clipShape
...

Marks Modifiers




```
PointMark(
    x: .value("Day", stock.date),
    y: .value("MaxPrice", stock.maxPrice)
)
.foregroundStyle(Color.gray)
.symbolSize(10)
.opacity(selectedElement == nil ? 1 : 0)
```

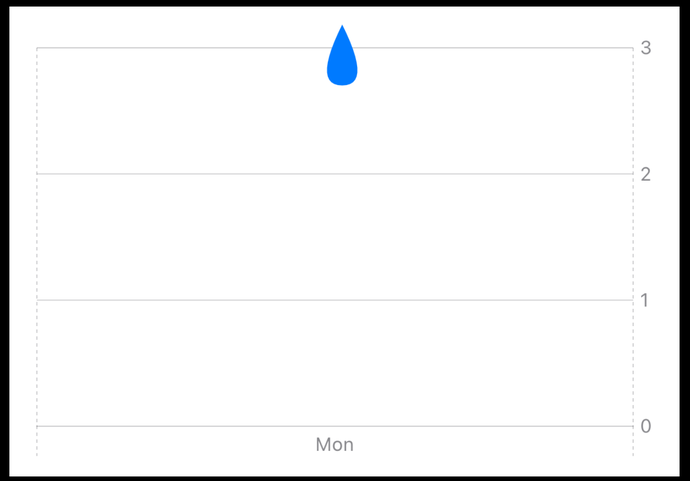
Marks Modifiers

```
struct RainDropSymbol: ChartSymbolShape {
    func path(in rect: CGRect) -> Path {
        var path = Path()
        path.move(to: CGPoint(x: rect.size.width/2, y: 0))
        path.addQuadCurve(
            to: CGPoint(x: rect.size.width/2, y: rect.size.height),
            control: CGPoint(x: rect.size.width, y: rect.size.height)
        )
        path.addQuadCurve(
            to: CGPoint(x: rect.size.width/2, y: 0),
            control: CGPoint(x: 0, y: rect.size.height))
        return path.applying(.init(translationX: rect.minX - 10, y: rect.minY - 10))
    }

    var perceptualUnitRect: CGRect {
        return CGRect(x: 0, y: 0, width: 5, height: 5)
    }
}
```

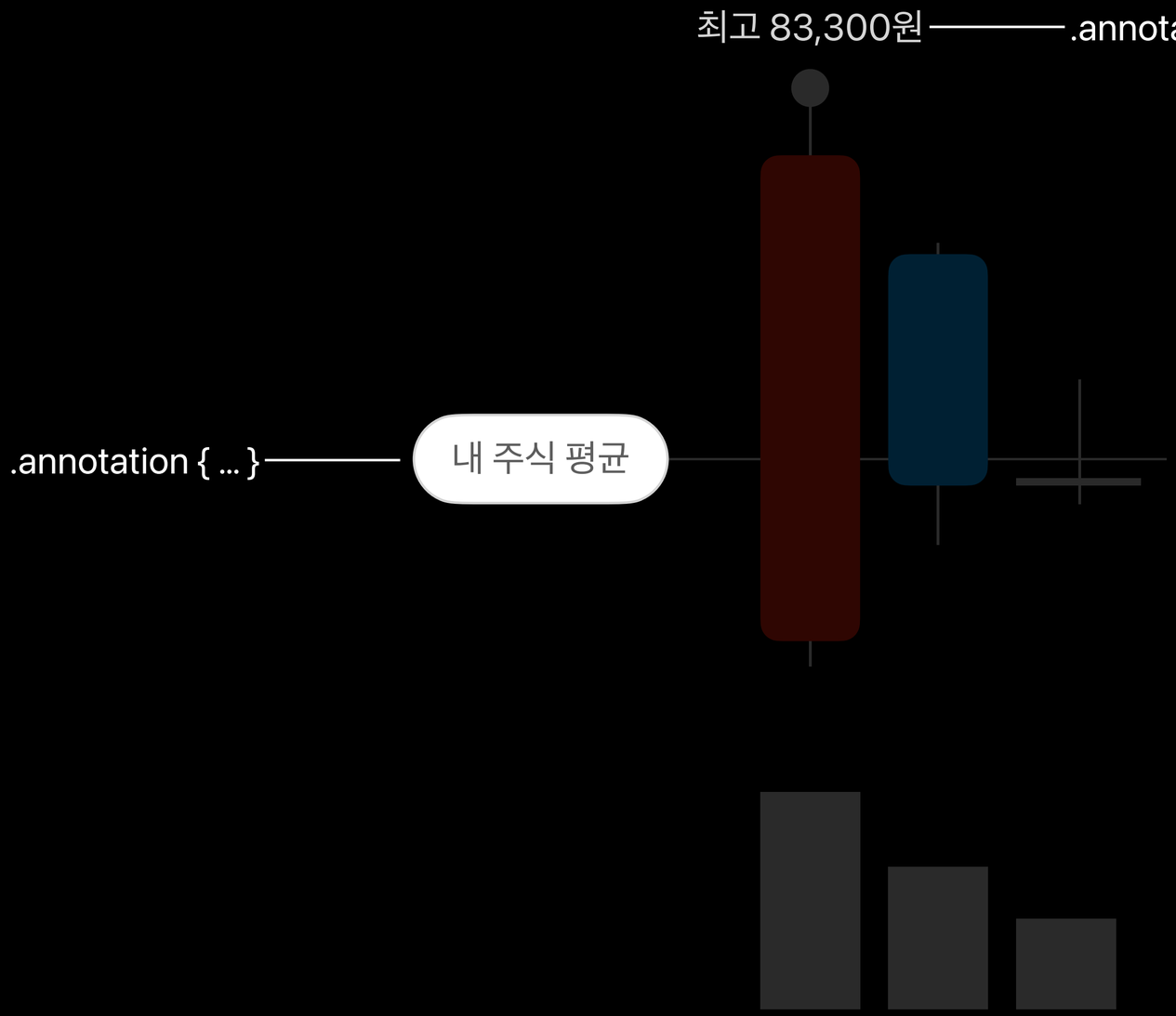
Marks Modifiers

Point

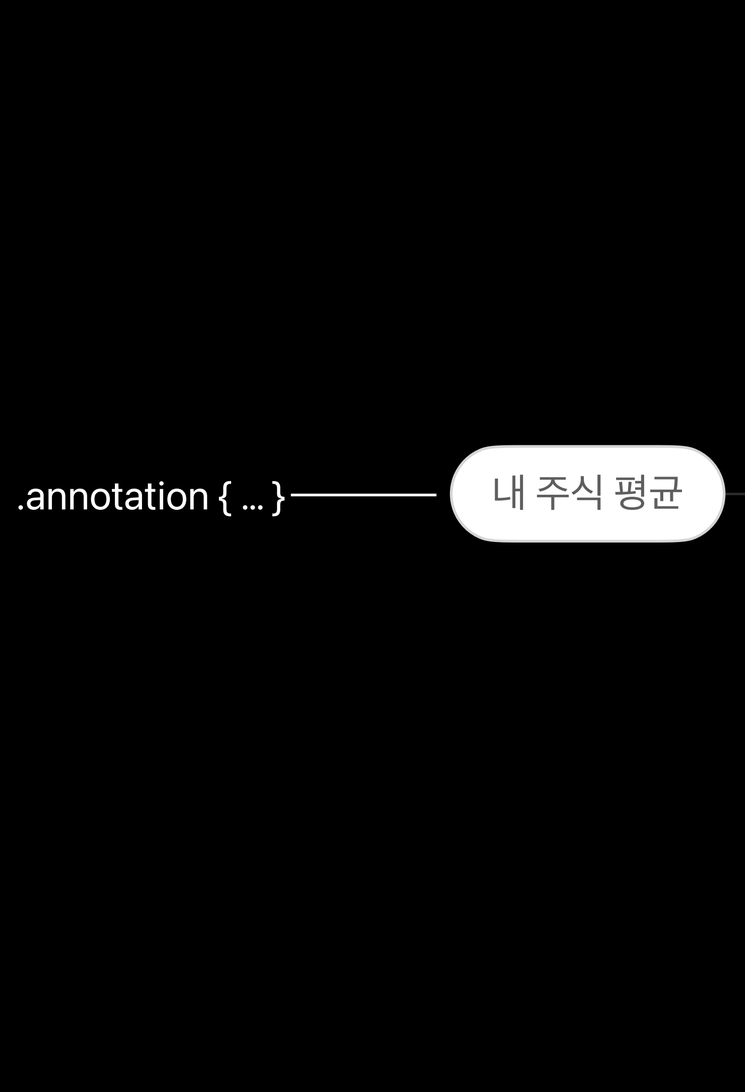
```
PointMark(
    x: .value("Day", "Mon"),
    y: .value("Rain", 3)
)
.symbol(RainDropSymbol())
```



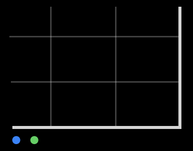
Marks Modifiers



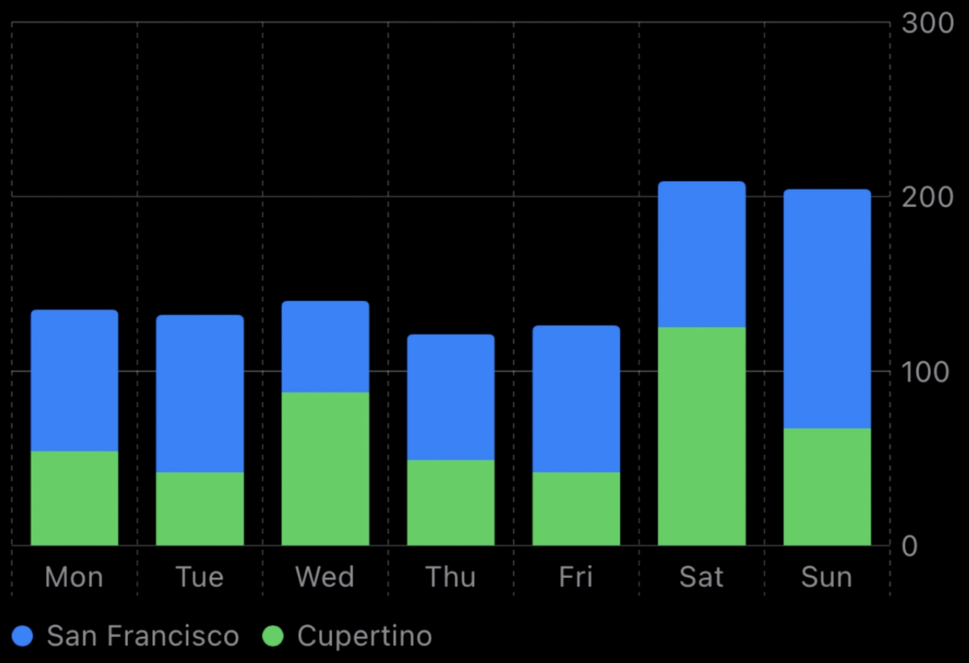
Marks Modifiers



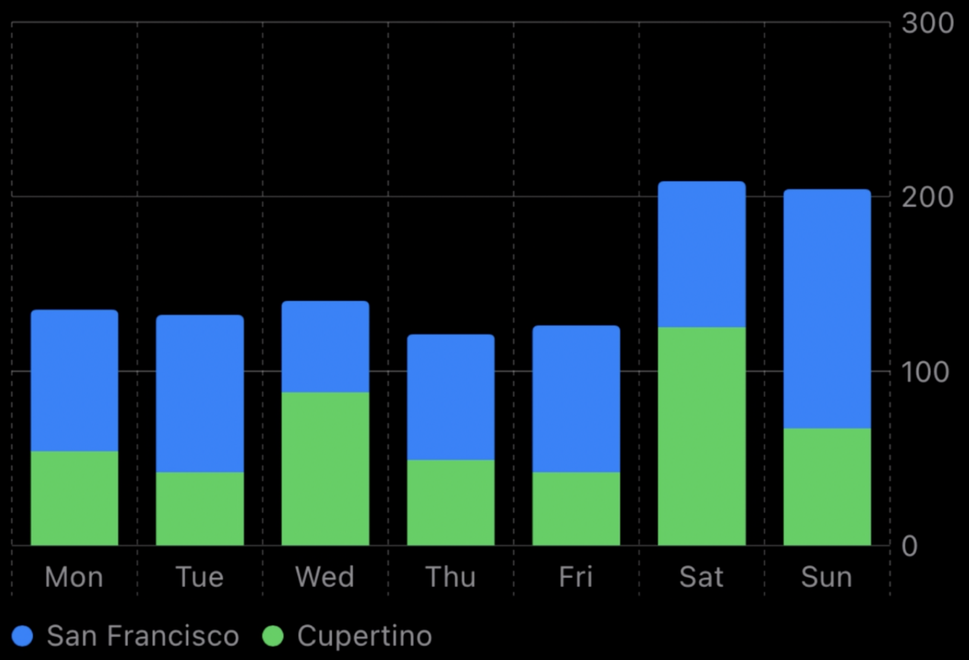

```
.annotation(alignment: .leading) {
    Text("내 주식 평균")
        .font(.caption2)
        .foregroundColor(.gray)
        .padding(.horizontal, 6)
        .padding(.vertical, 4)
        .background {
            ZStack {
                Capsule()
                    .stroke()
                    .foregroundColor(.gray)
                    .opacity(0.005)
                Capsule().fill(.white)
            }
        }
        .offset(y: 15)
        .opacity(selectedElement == nil ? 1 : 0)
}
```



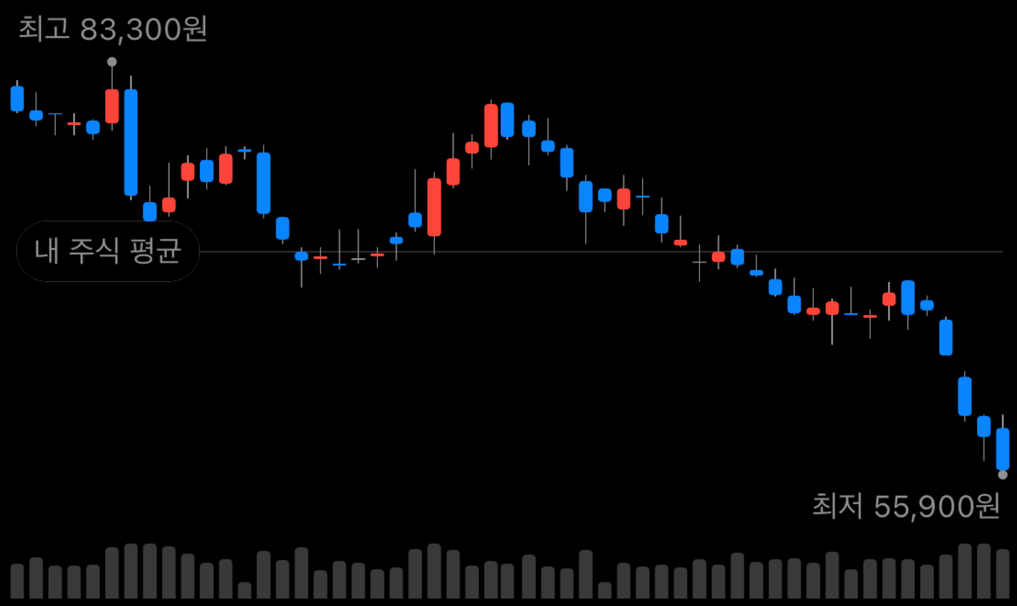
Axes Customization


300
200
100
0
Mon
Tue
Wed
Thu
Fri
Sat
Sun
San Francisco
Cupertino


최고 83,300원
내 주식 평균
최저 55,900원

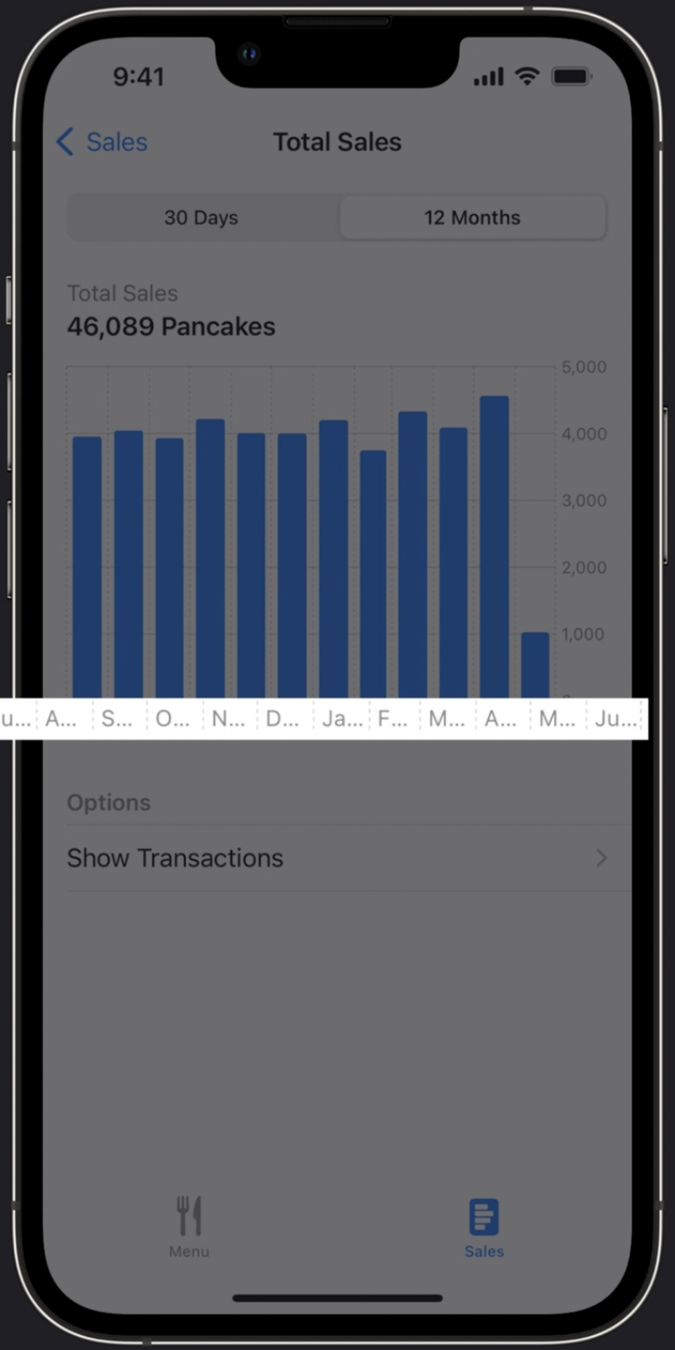



```
.chartXAxis(.hidden)
.chartYAxis(.hidden)
```

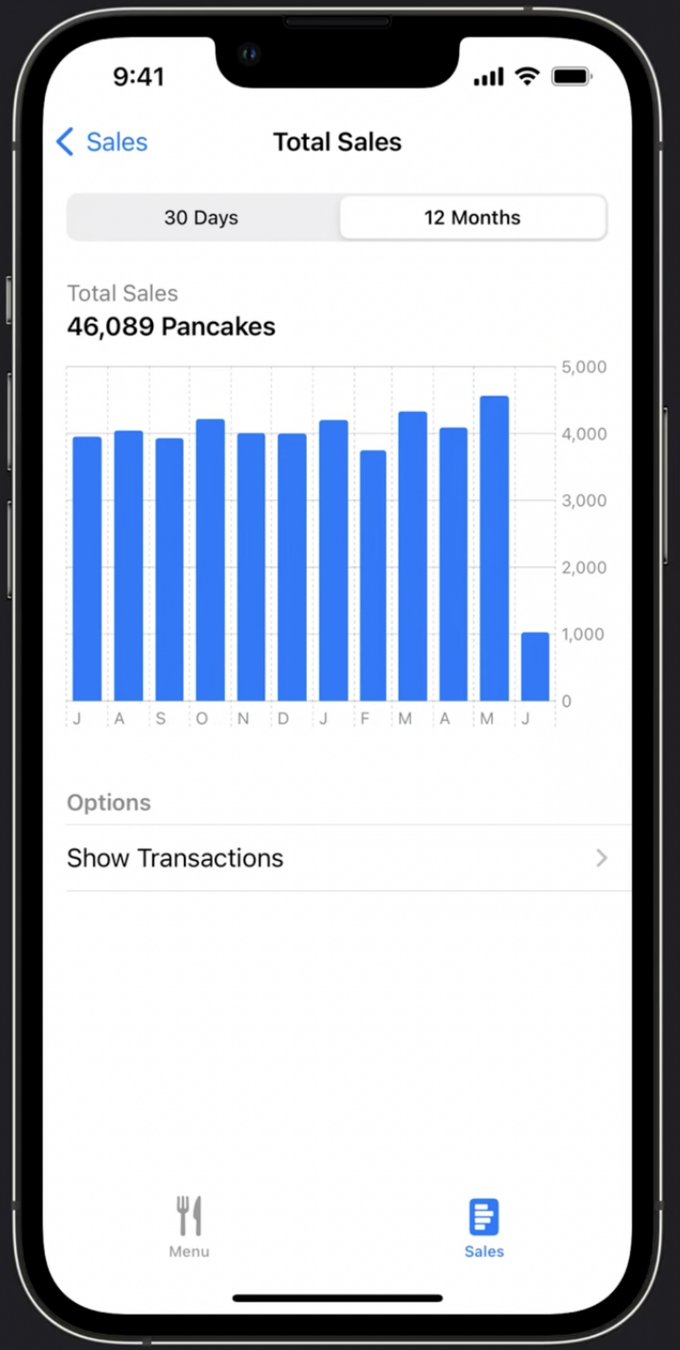


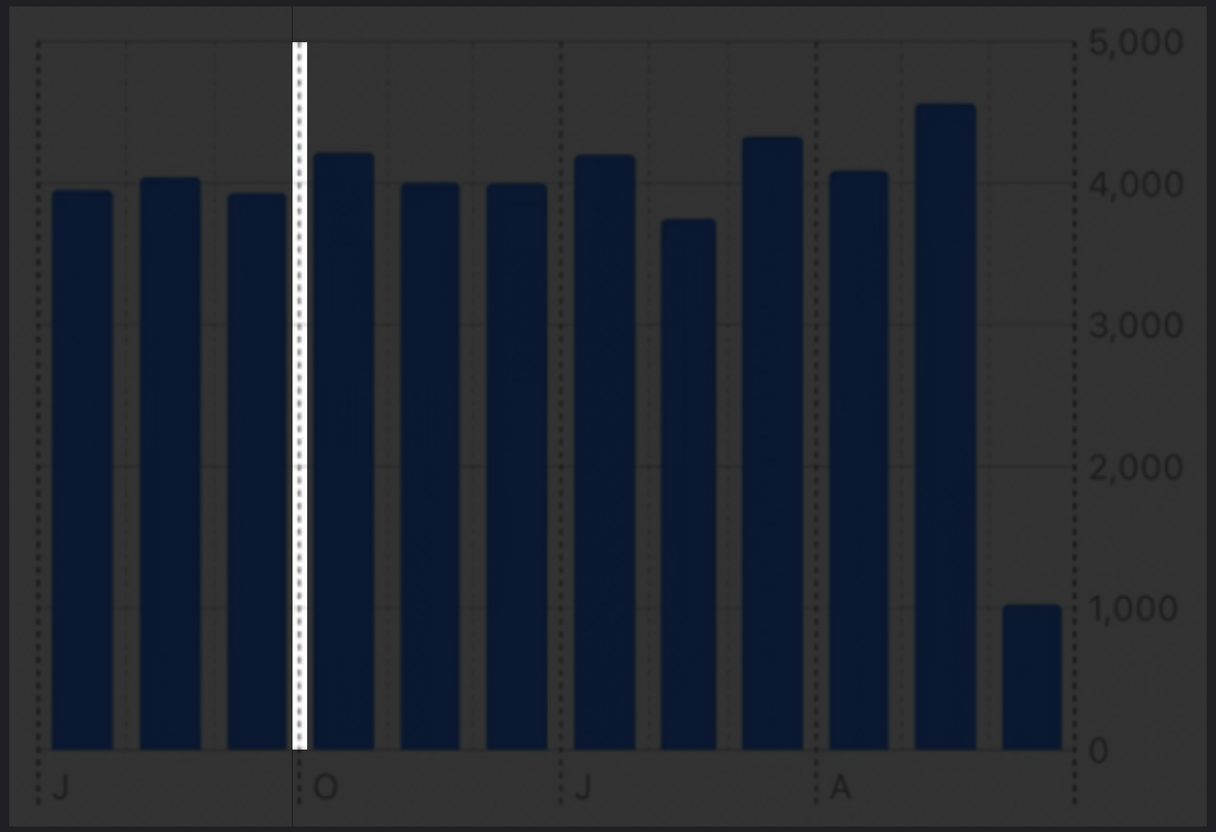

```
.chartYScale(domain: minPriceSortedData[0].minPrice ... maxPriceSortedData[0].maxPrice)
```

```
let data: [(month: Date, sales: Int)] = [...]

Chart(data, id: \.month) {
    BarMark(
        x: .value("Month", $0.month, unit: .month),
        y: .value("Sales", $0.sales)
    )
}
.chartXAxis {
    AxisMarks(values: .stride(by: .month))
}
```

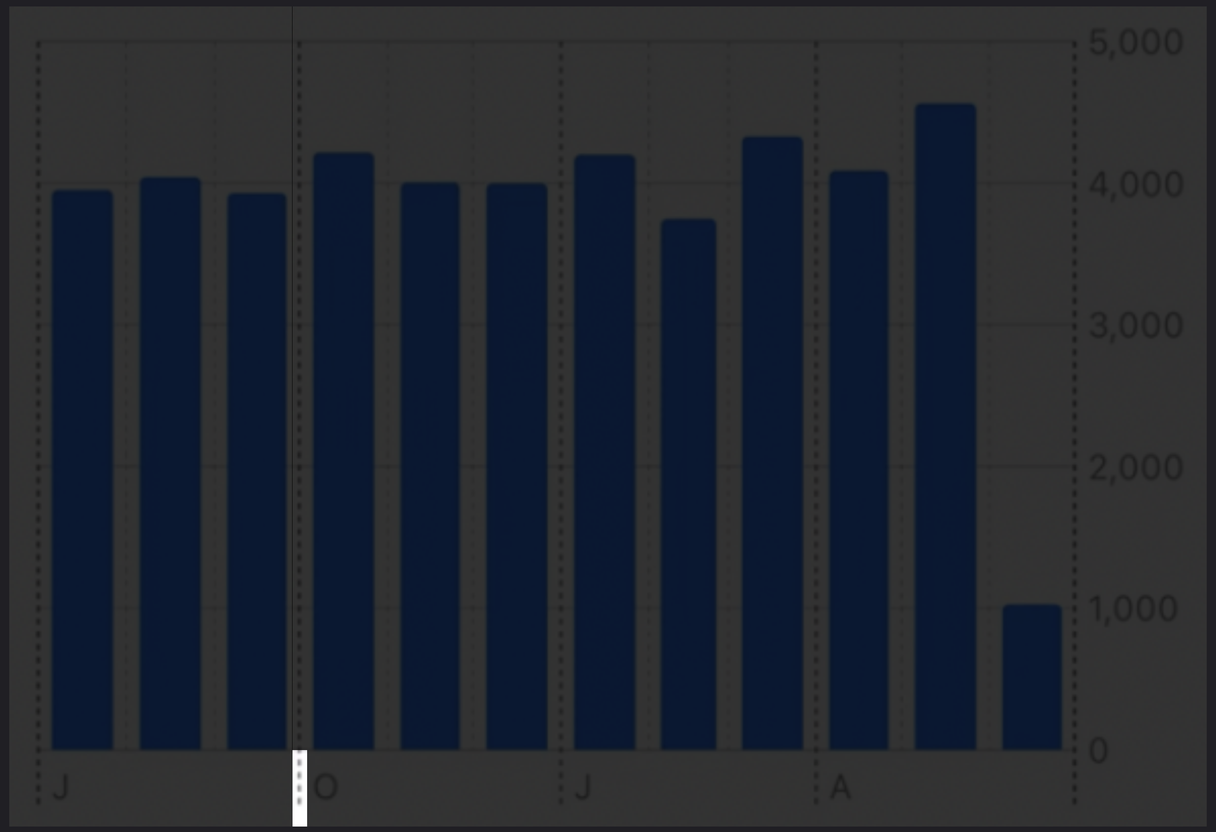


```
let data: [(month: Date, sales: Int)] = [...]

Chart(data, id: \.month) {
    BarMark(
        x: .value("Month", $0.month, unit: .month),
        y: .value("Sales", $0.sales)
    )
}
.chartXAxis {
    AxisMarks(values: .stride(by: .month)) { value in
        AxisGridLine()
        AxisTick()
        AxisValueLabel(
            format: .dateTime.month(.narrow)
        )
    }
}
```

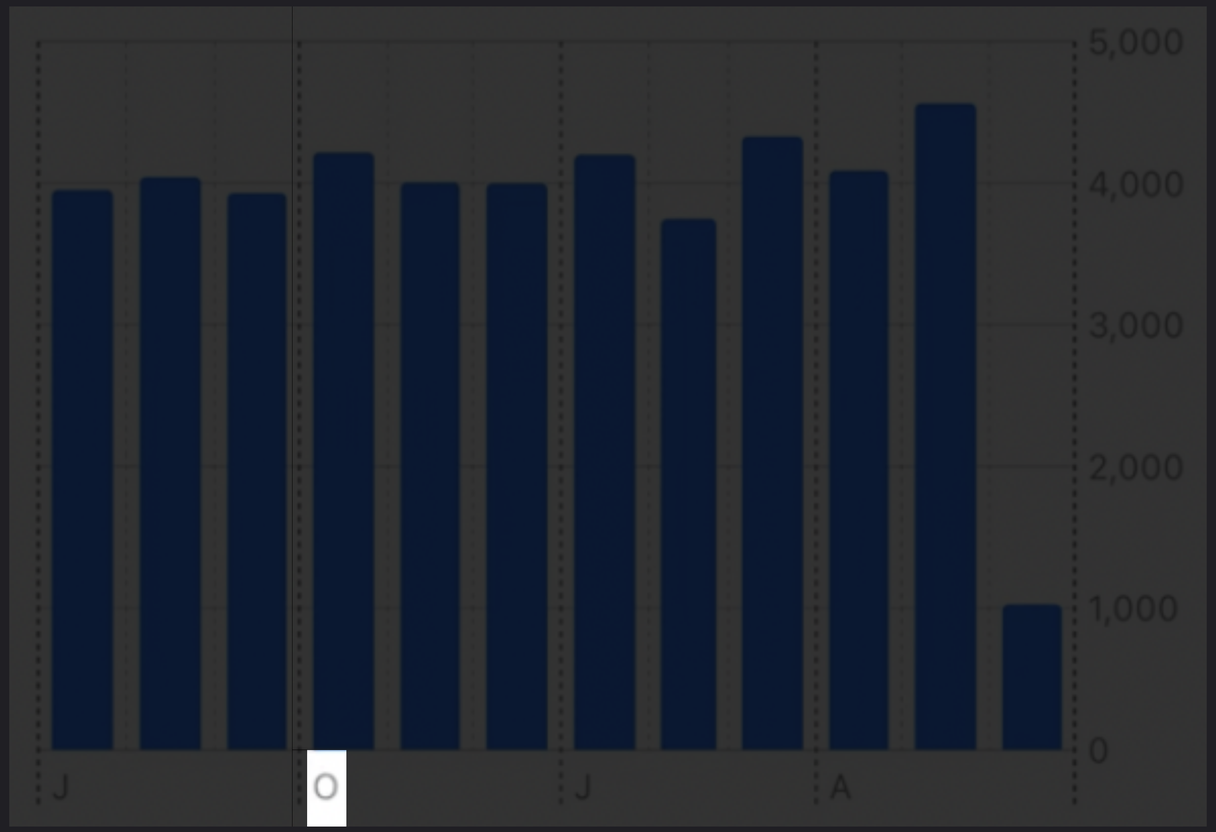




AxisGridLine: 차트 내부 선

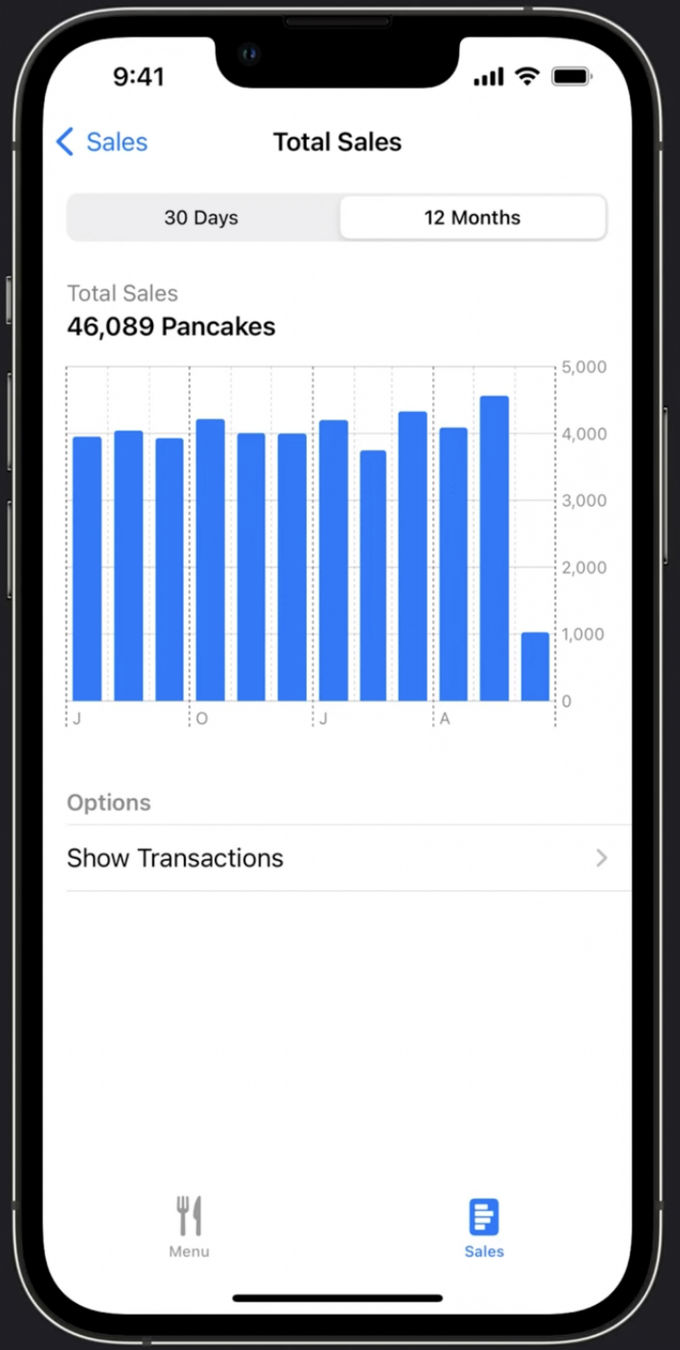


AxisTick: 차트 밖으로 삐져나오는 선

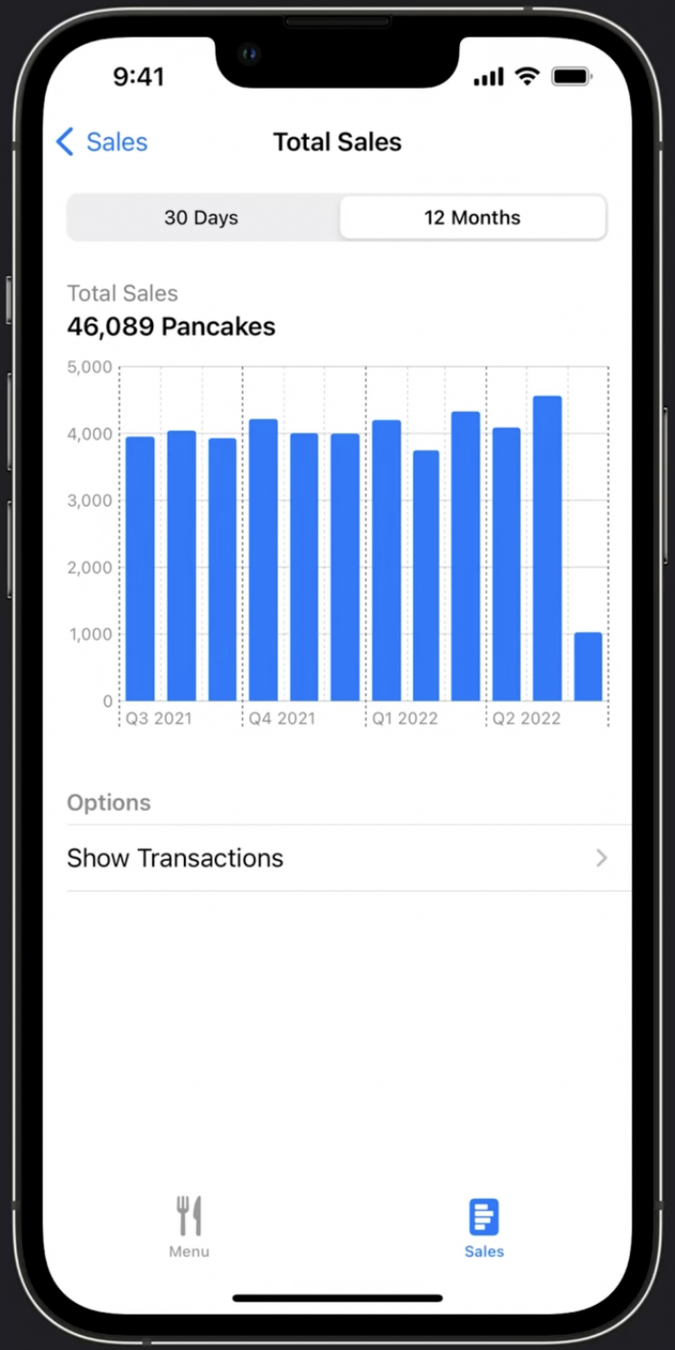


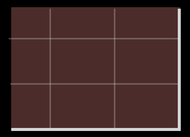
AxisValueLabel: 축 라벨

```
let data: [(month: Date, sales: Int)] = [...]

Chart {
    ...
}
.chartXAxis {
    AxisMarks(values: .stride(by: .month)) { value in
        if value.as(Date.self)!.isFirstMonthOfQuarter {
            AxisGridLine().foregroundStyle(.black)
            AxisTick().foregroundStyle(.black)
            AxisValueLabel(
                format: .dateTime.month(.narrow)
            )
        } else {
            AxisGridLine()
        }
    }
}
```



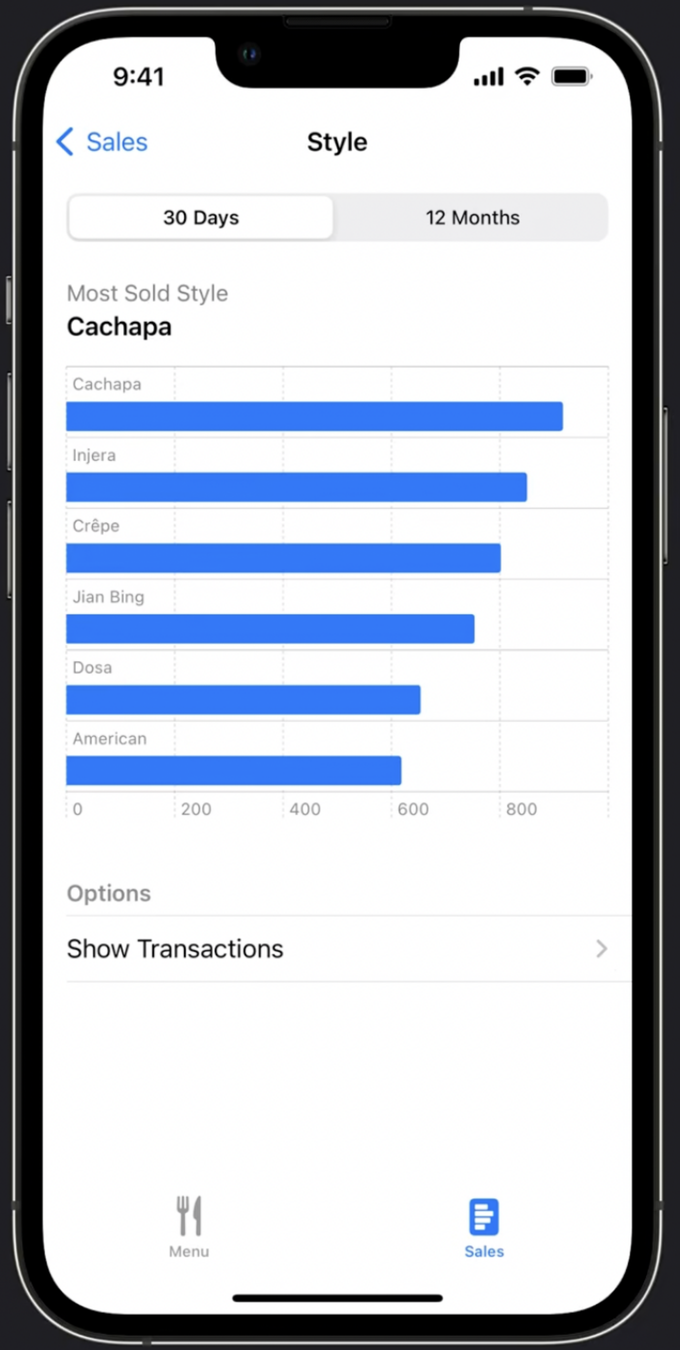
```
let data: [(month: Date, sales: Int)] = [...]

Chart {
    ...
}
.chartYAxis {
    AxisMarks(position: .leading)
}
```



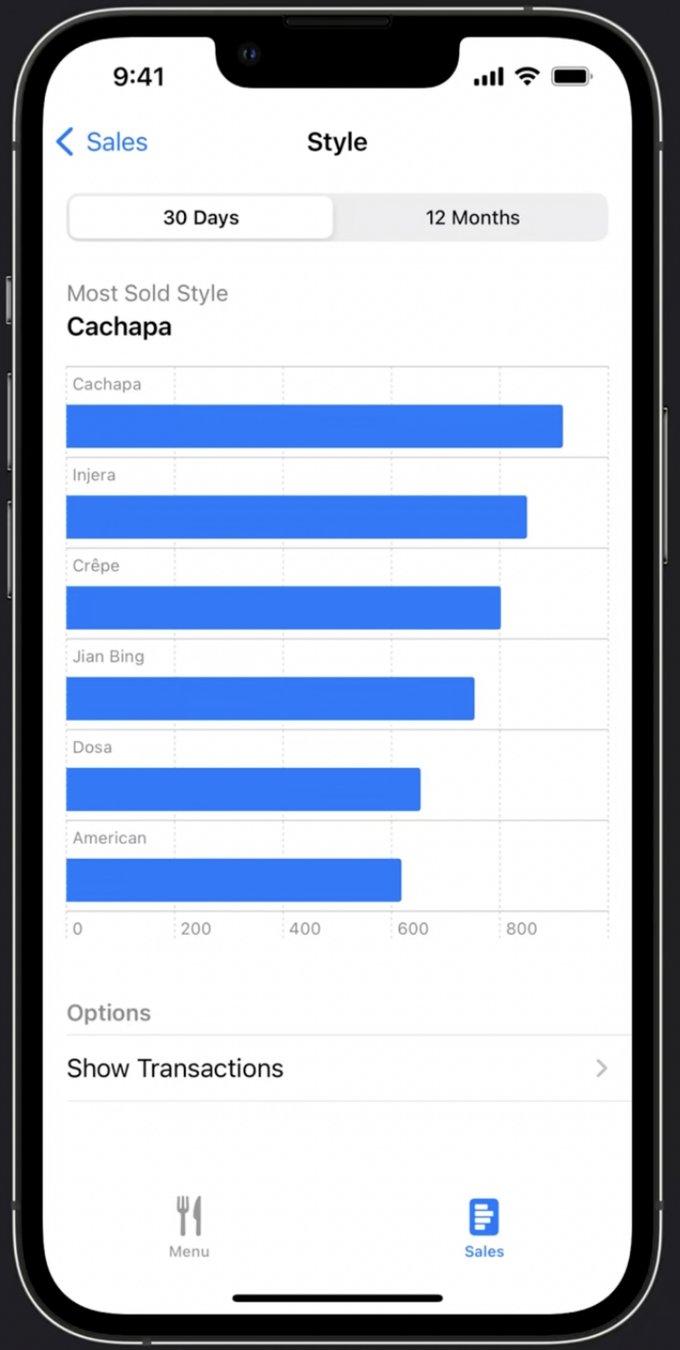


Plot Area Customization

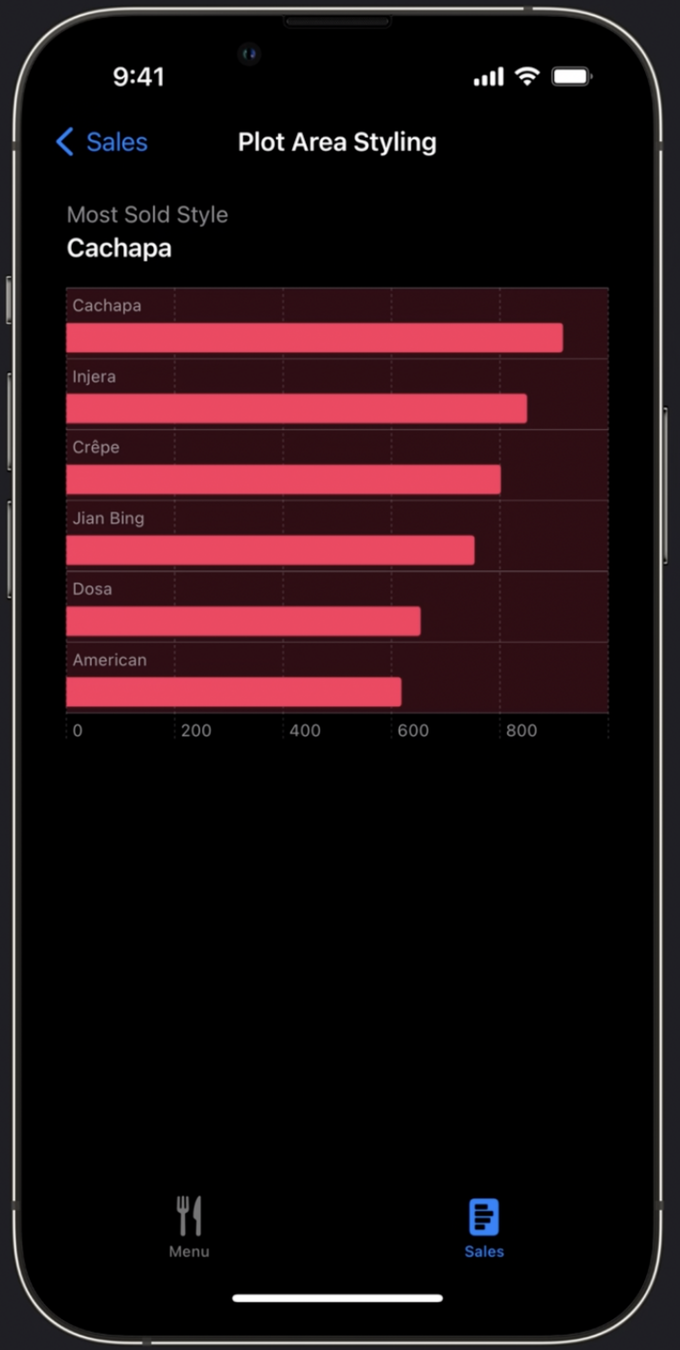
```
Chart {
    ...
}
.chartPlotStyle { plotArea in
    // TODO: return the content of the plot area.
    return plotArea
}
```



```
Chart {
    ...
}
.chartPlotStyle { plotArea in
    plotArea.frame(height: 60 * numberOfCategories)
}
```



```
Chart {
    ...
}
.chartPlotStyle { plotArea in
    plotArea.background(.pink.opacity(0.2))
}
```



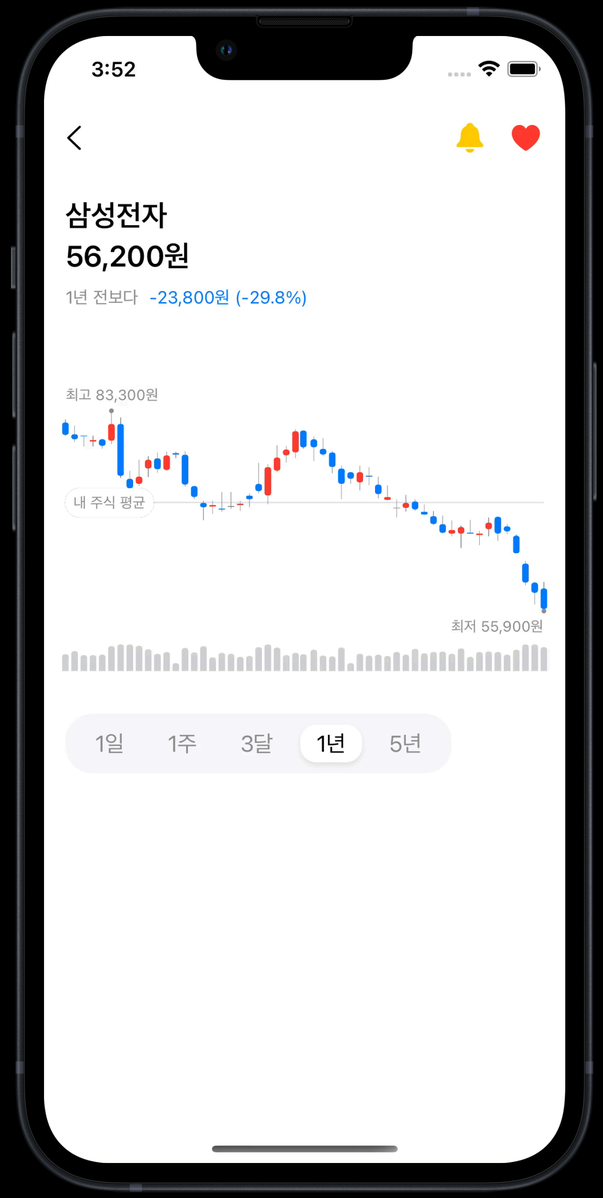
ChartProxy

Access the X and Y scales in a chart with `ChartProxy`.

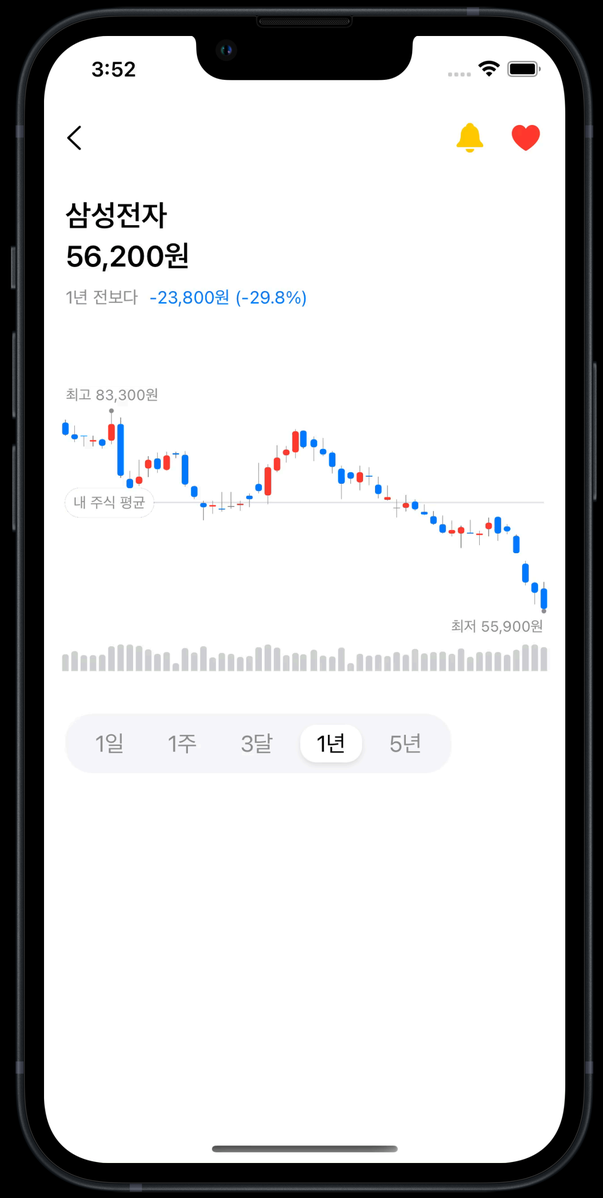
```
let proxy: ChartProxy

proxy.position(forX: 123.0) // get the X position for value 123.0.
proxy.value(atX: 100)       // get the data value at X position 100pt.
```

```
Chart {
    ...
}
.chartOverlay { proxy in
    // Define the overlay view as a function
    // of the chart proxy.
}
.chartBackground { proxy in
    // Define the background view as a function
    // of the chart proxy.
}
```

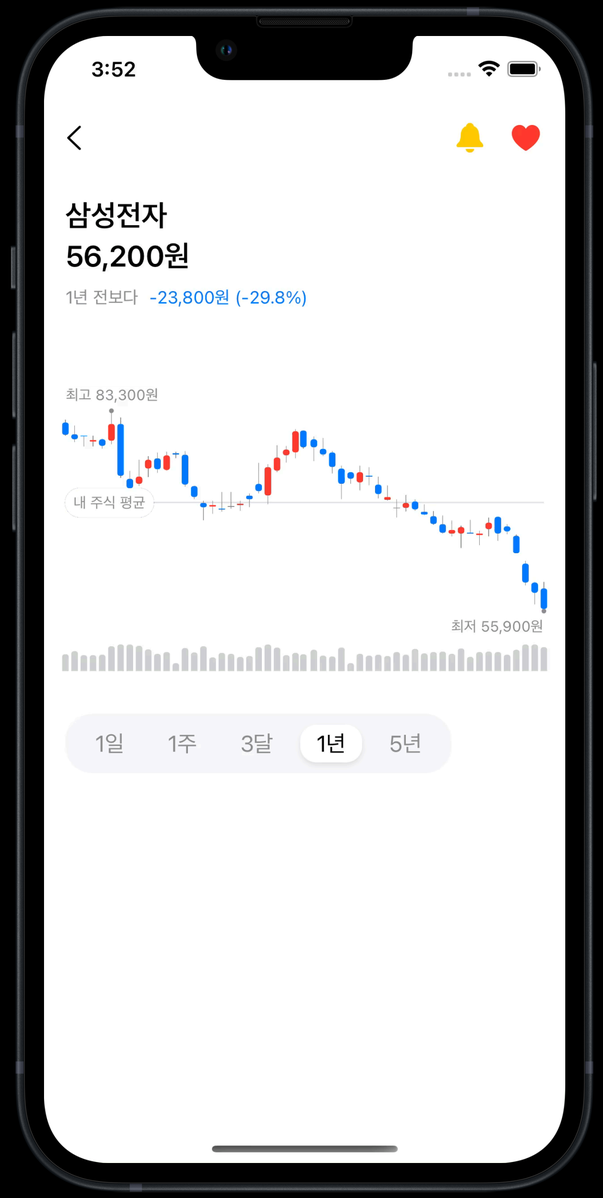
```swift
.chartOverlay { proxy in
    ZStack {
        GeometryReader { nthGeoItem in
            if let selectedElement {
                let dateInterval = Calendar.current.dateInterval(of: .day, for: selectedElement.date)!
                let startPositionX1 = proxy.position(forX: dateInterval.start) ?? 0
                let startPositionX2 = proxy.position(forX: dateInterval.end) ?? 0
                let midStartPositionX = (startPositionX1 + startPositionX2) / 2

                let lineX = midStartPositionX - 0.5 // 선이 놓일 X 좌표
                let lineY = nthGeoItem[proxy.plotAreaFrame].midY // 선이 놓일 Y 좌표

                let textOffset = max(0, min(nthGeoItem.size.width - selectedDateTextWidth, lineX -
                    selectedDateTextWidth / 2))

                // 선
                Rectangle()
                    .fill(.gray)
                    .frame(width: 1, height: proxy.plotAreaSize.height + 70)
                    .offset(y: 10)
                    .position(x: lineX, y: lineY)

                // 선택된 날짜
                Text("\(dateConverting(date: selectedElement.date))")
                    .font(.caption)
                    .foregroundColor(.gray)
                    .offset(x: textOffset, y: -45)
                    .readSize { size in
                        selectedDateTextWidth = size.width
                    }
            }
        }
    }
}
```



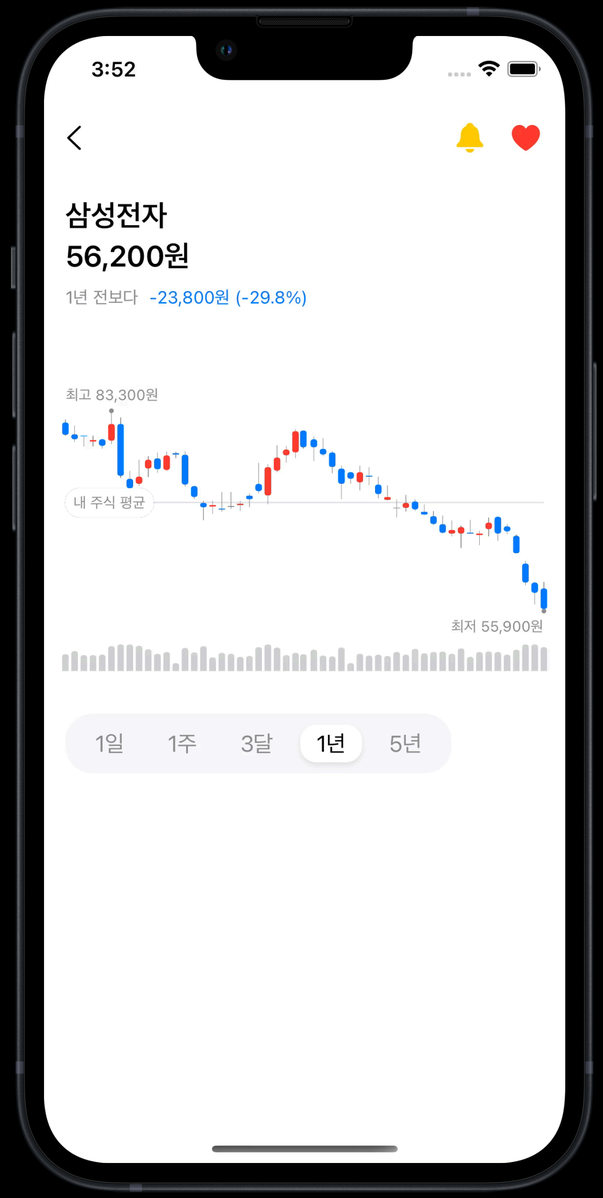
```swift
.chartOverlay { proxy in
    ZStack {
        GeometryReader { nthGeoItem in
            if let selectedElement {
                let dateInterval = Calendar.current.dateInterval(of: .day, for: selectedElement.date)!
                let startPositionX1 = proxy.position(forX: dateInterval.start) ?? 0
                let startPositionX2 = proxy.position(forX: dateInterval.end) ?? 0
                let midStartPositionX = (startPositionX1 + startPositionX2) / 2

                let lineX = midStartPositionX - 0.5 // 선이 놓일 X 좌표
                let lineY = nthGeoItem[proxy.plotAreaFrame].midY // 선이 놓일 Y 좌표

                let textOffset = max(0, min(nthGeoItem.size.width - selectedDateTextWidth, lineX -
                    selectedDateTextWidth / 2))

                // 선
                Rectangle()
                    .fill(.gray)
                    .frame(width: 1, height: proxy.plotAreaSize.height + 70)
                    .offset(y: 10)
                    .position(x: lineX, y: lineY)

                // 선택된 날짜
                Text("\(dateConverting(date: selectedElement.date))")
                    .font(.caption)
                    .foregroundColor(.gray)
                    .offset(x: textOffset, y: -45)
                    .readSize { size in
                        selectedDateTextWidth = size.width
                    }
            }
        }
    }
}
```



```swift
.chartOverlay { proxy in
    ZStack {
        GeometryReader { nthGeoItem in
            if let selectedElement {
                let dateInterval = Calendar.current.dateInterval(of: .day, for: selectedElement.date)!
                let startPositionX1 = proxy.position(forX: dateInterval.start) ?? 0
                let startPositionX2 = proxy.position(forX: dateInterval.end) ?? 0
                let midStartPositionX = (startPositionX1 + startPositionX2) / 2

                let lineX = midStartPositionX - 0.5 // 선이 놓일 X 좌표
                let lineY = nthGeoItem[proxy.plotAreaFrame].midY // 선이 놓일 Y 좌표

                let textOffset = max(0, min(nthGeoItem.size.width - selectedDateTextWidth, lineX -
                    selectedDateTextWidth / 2))

                // 선
                Rectangle()
                    .fill(.gray)
                    .frame(width: 1, height: proxy.plotAreaSize.height + 70)
                    .offset(y: 10)
                    .position(x: lineX, y: lineY)

                // 선택된 날짜
                Text("\(dateConverting(date: selectedElement.date))")
                    .font(.caption)
                    .foregroundColor(.gray)
                    .offset(x: textOffset, y: -45)
                    .readSize { size in
                        selectedDateTextWidth = size.width
                    }
            }
        }
    }
}
```



```swift
.chartOverlay { proxy in
    ZStack {
        GeometryReader { nthGeoItem in
            if let selectedElement {
                let dateInterval = Calendar.current.dateInterval(of: .day, for: selectedElement.date)!
                let startPositionX1 = proxy.position(forX: dateInterval.start) ?? 0
                let startPositionX2 = proxy.position(forX: dateInterval.end) ?? 0
                let midStartPositionX = (startPositionX1 + startPositionX2) / 2

                let lineX = midStartPositionX - 0.5 // 선이 놓일 X 좌표
                let lineY = nthGeoItem[proxy.plotAreaFrame].midY // 선이 놓일 Y 좌표

                let textOffset = max(0, min(nthGeoItem.size.width - selectedDateTextWidth, lineX -
                    selectedDateTextWidth / 2))

                // 선
                Rectangle()
                    .fill(.gray)
                    .frame(width: 1, height: proxy.plotAreaSize.height + 70)
                    .offset(y: 10)
                    .position(x: lineX, y: lineY)

                // 선택된 날짜
                Text("\(dateConverting(date: selectedElement.date))")
                    .font(.caption)
                    .foregroundColor(.gray)
                    .offset(x: textOffset, y: -45)
                    .readSize { size in
                        selectedDateTextWidth = size.width
                    }
            }
        }
    }
}
```



Thank You.