

[2] getSymbols() 함수를 이용한 API 다운로드

앞선 예에서 API 주소를 이용할 경우 매우 간단하게 데이터를 수집할 있음을 살펴 보았습니다. 그러나 해당 방법에는 여러 단점 또한 존재합니다. 먼저, 원하는 항목에 대한 API를 일일이 얻는 것이 힘든 일입니다. 또한 Quandl의 경우 무료로 얻을 수 있는 정보에 제한이 있으며, 다운로드 양에 대한 제한도 있습니다. 한 두 종목의 경우 해당 방법으로 데이터를 수집할 수 있지만, 전 종목의 데이터를 해당 방법으로 구하는 것은 사실상 불가능 합니다.

다행히 야후 파이낸스에서 주가 데이터를 무료로 제공하며, quantmod 패키지의 getSymbols() 함수는 해당 API에 접속하여 데이터를 다운로드 받아옵니다.

주가 다운로드

getSymbols() 함수의 기본적인 사용법은 매우 간단합니다. 괄호 안에 다운로드 받고자 하는 종목의 티커를 입력하면 됩니다. 아래의 코드는 애플의 티커인 "AAPL"을 입력한 경우이며, 티커와 동일한 변수인 AAPL이 생성되며, 주가 데이터가 다운로드 된 후 xts 형태로 입력됩니다.

```
library(quantmod)
getSymbols("AAPL")

head(AAPL)
```

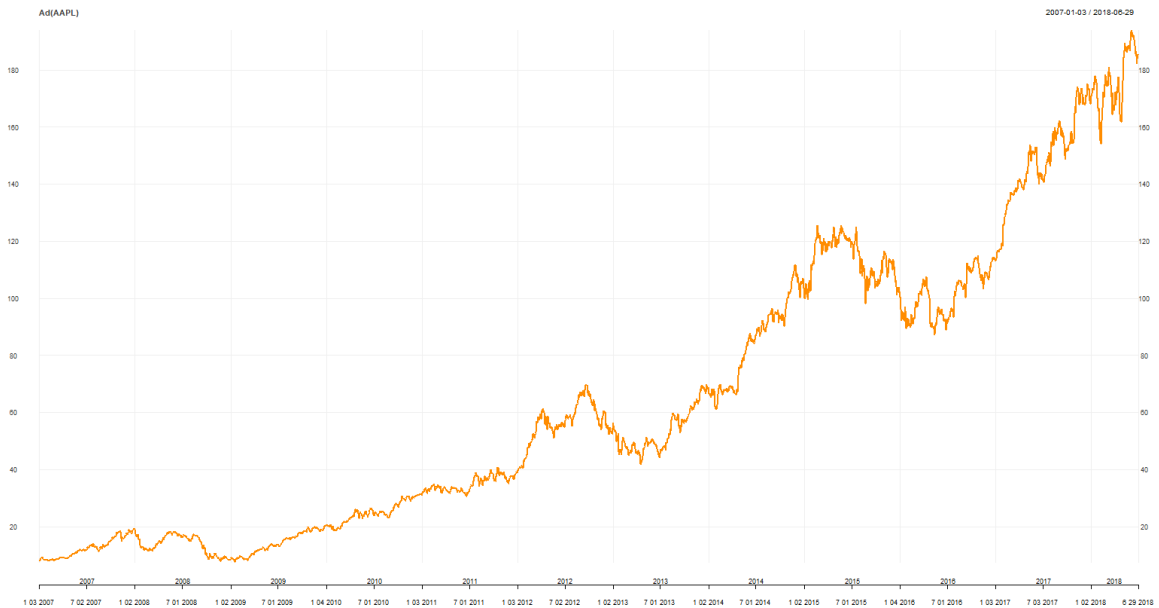
	AAPL.Open	AAPL.High	AAPL.Low	AAPL.Close	AAPL.Volume	AAPL.Adjusted
2007-01-03	12.32714	12.36857	11.70000	11.97143	309579900	8.073009
2007-01-04	12.00714	12.27857	11.97429	12.23714	211815100	8.252192
2007-01-05	12.25286	12.31428	12.05714	12.15000	208685400	8.193427
2007-01-08	12.28000	12.36143	12.18286	12.21000	199276700	8.233888
2007-01-09	12.35000	13.28286	12.16429	13.22429	837324600	8.917878
2007-01-10	13.53571	13.97143	13.35000	13.85714	738220000	9.344647

다운로드 결과로써 총 6개의 열이 생성됩니다. Open은 시가, High는 고가, Low는 저가, Close는 종가를 의미합니다. 또한, Volume은 거래량을 의미하며, Adjusted는 배당이 반영된 수정주가를 의미합니다. 이 중 가장 많이 사용되는 데이터는 Adjusted, 즉 배당이 반영된 수정주가입니다. Ad() 함수는 getSymbols() 함수를 통해 다운로드 받은 데이터에서 수정주가만을 선택하여 줍니다. 시계

열 그래프를 그려주는 `chart_Series()`와 함께 수정주가를 그리면 다음과 같습니다.

```
chart_Series(Ad(AAPL))
```

[그림] 애플의 수정주가 그래프



시계열 기간을 입력하지 않을 경우 2007년 1월부터 현재까지의 데이터가 다운로드 되며, 추가적인 입력 변수를 통해 원하는 기간의 데이터를 다운로드 받을 수도 있습니다.

```
getSymbols("AAPL", from = "2000-01-01", to = "2017-12-31")
```

```
head(AAPL)
```

	AAPL.Open	AAPL.High	AAPL.Low	AAPL.Close	AAPL.Volume	AAPL.Adjusted
2000-01-03	3.745536	4.017857	3.631696	3.997768	133949200	2.695920
2000-01-04	3.866071	3.950893	3.613839	3.660714	128094400	2.468626
2000-01-05	3.705357	3.948661	3.678571	3.714286	194580400	2.504751
2000-01-06	3.790179	3.821429	3.392857	3.392857	191993200	2.287994
2000-01-07	3.446429	3.607143	3.410714	3.553571	115183600	2.396373
2000-01-10	3.642857	3.651786	3.383929	3.491071	126266000	2.354226

```
tail(AAPL)
```

	AAPL.Open	AAPL.High	AAPL.Low	AAPL.Close	AAPL.Volume	AAPL.Adjusted
2017-12-21	174.17	176.02	174.10	175.01	20949900	173.6298
2017-12-22	174.68	175.42	174.50	175.01	16349400	173.6298
2017-12-26	170.80	171.47	169.68	170.57	33185500	169.2248
2017-12-27	170.10	170.78	169.71	170.60	21498200	169.2546
2017-12-28	171.00	171.85	170.48	171.08	16480200	169.7308
2017-12-29	170.52	170.59	169.22	169.23	25999900	167.8954

from에는 시작 기간을, to에는 종료 기간을 입력하여 주면, 해당 시점의 데이터가 다운로드 받아짐이 확인됩니다.

```
data = getSymbols("AAPL", from = "2000-01-01", to = "2017-12-31", auto.assign = FALSE)
```

```
head(data)
```

	AAPL.Open	AAPL.High	AAPL.Low	AAPL.Close	AAPL.Volume	AAPL.Adjusted
2000-01-03	3.745536	4.017857	3.631696	3.997768	133949200	2.695920
2000-01-04	3.866071	3.950893	3.613839	3.660714	128094400	2.468626
2000-01-05	3.705357	3.948661	3.678571	3.714286	194580400	2.504751
2000-01-06	3.790179	3.821429	3.392857	3.392857	191993200	2.287994
2000-01-07	3.446429	3.607143	3.410714	3.553571	115183600	2.396373
2000-01-10	3.642857	3.651786	3.383929	3.491071	126266000	2.354226

```
tail(data)
```

	AAPL.Open	AAPL.High	AAPL.Low	AAPL.Close	AAPL.Volume	AAPL.Adjusted
2017-12-21	174.17	176.02	174.10	175.01	20949900	173.6298
2017-12-22	174.68	175.42	174.50	175.01	16349400	173.6298
2017-12-26	170.80	171.47	169.68	170.57	33185500	169.2248
2017-12-27	170.10	170.78	169.71	170.60	21498200	169.2546
2017-12-28	171.00	171.85	170.48	171.08	16480200	169.7308
2017-12-29	170.52	170.59	169.22	169.23	25999900	167.8954

getSymbols() 함수를 통해 다운로드 받은 데이터는 자동으로 티커와 동일한 변수명에 저장됩니다. 만일 티커명이 아닌 원하는 변수명에 데이터를 저장하고 싶을 경우, 추가적인 입력 변수가 필요합니다. 아래 코드처럼 데이터를 저장하고자 하는 변수를 지정한 후, auto.assign = FALSE 변수를 추가적으로 입력하면 AAPL 데이터가 data 변수에 데이터가 저장됩니다.

```
ticker = c("FB", "NVDA")
getSymbols(ticker)
```

```
head(FB)
```

	FB.Open	FB.High	FB.Low	FB.Close	FB.Volume	FB.Adjusted
2012-05-18	42.05	45.00	38.00	38.23	573576400	38.23
2012-05-21	36.53	36.66	33.00	34.03	168192700	34.03
2012-05-22	32.61	33.59	30.94	31.00	101786600	31.00
2012-05-23	31.37	32.50	31.36	32.00	73600000	32.00
2012-05-24	32.95	33.21	31.77	33.03	50237200	33.03
2012-05-25	32.90	32.95	31.11	31.91	37149800	31.91

```
head(NVDA)
```

	NVDA.Open	NVDA.High	NVDA.Low	NVDA.Close	NVDA.Volume	NVDA.Adjusted
2007-01-03	24.71333	25.01333	23.19333	24.05333	28870500	22.27390
2007-01-04	23.96667	24.05333	23.35333	23.94000	19932400	22.16896
2007-01-05	23.37333	23.46667	22.28000	22.44000	31083600	20.77992
2007-01-08	22.52000	23.04000	22.13333	22.60667	16431700	20.93426
2007-01-09	22.64000	22.79333	22.14000	22.16667	19104100	20.52681
2007-01-10	21.93333	23.46667	21.60000	23.26000	27718600	21.53926

한번에 여러 종목의 주가를 다운로드 받을 수도 있습니다. 아래 예제와 같이 페이스북과 엔비디아의 티커인 "FB"와 "NVDA"를 ticker 변수에 입력하여 주고, getSymbols() 함수에 티커들을 입력한 변수를 넣어주면 두 종목의 주가가 동시에 다운로드 됩니다.

국내 종목 주가 다운로드

getSymbols() 함수를 이용하면 미국뿐 아니라 국내 종목의 주가를 다운로드 받을 수도 있습니다. 국내 종목의 티커는 총 6자리로 구성되어 있으며, 해당 함수에 입력되는 티커는 코스피 상장 종목의 경우 "티커.KS", 코스닥 상장 종목의 경우 "티커.KQ"의 형태로 입력해 주어야 합니다.

```
getSymbols("005930.KS", from = "2000-01-01", to = "2017-12-31")
```

```
tail(Ad('005930.KS'))
      005930.KS.Adjusted
2017-12-19      18180.67
2017-12-21      17327.35
2017-12-22      17524.81
2017-12-26      16995.89
2017-12-27      31420.12
2017-12-28      32438.60
```

먼저 코스피 상장종목의 예시입니다. 삼성전자의 티커인 005930에 .KS를 붙여 함수에 입력할 경우, 티커명에 해당하는 005930.KS 변수명에 데이터가 저장됩니다. 변수명에 콤마(,)가 있는 관계로, Ad 함수를 통해 수정주가를 확인하고자 할 때는 변수명의 앞뒤에 역음부호(')를 붙여주어야 합니다.

해당 데이터를 살펴보면 17년 12월 26일 수정주가는 16995.89인 반면, 다음날인 12월 27일의 수정주가는 31420.12로써 하루 동안 무려 2배 가까이 상승한 것처럼 보이며, 이는 잘못된 데이터가 서버에 저장되어 있기 때문입니다. 물론 시간이 지나면서 오류가 있는 데이터는 수정이 되지만, 신뢰성이 떨어지는 데이터를 사용하는 것은 권장하지 않습니다. 따라서 국내 종목의 경우 배당이 반영된 값 보다는 단순 증가를 사용하기를 권장합니다.

```
> tail(CI('005930.KS'))
      005930.KS.Close
2017-12-19      51560
2017-12-21      49140
2017-12-22      49700
2017-12-26      48200
2017-12-27      49360
2017-12-28      50960
```

CI() 함수는 Close, 즉 종가만을 선택하여 주며, 사용 방법은 기존 Ad() 함수와 동일합니다. 비록 배당을 고려할 수는 없지만, 전반적으로 오류가 없는 데이터를 사용할 수 있습니다.

```
getSymbols("068760.KQ", from = "2000-01-01", to = "2017-12-31")

> tail(CI('068760.KQ'))
      068760.KQ.Close
```

2017-12-19	58921.6
2017-12-21	56372.6
2017-12-22	59215.7
2017-12-26	60686.3
2017-12-27	62000.0
2017-12-28	61200.0

이번엔 코스닥 상장종목의 예시입니다. 셀트리온제약의 티커인 068670에 .KQ를 붙여 함수에 입력합니다. 역시나 데이터가 다운로드 되어 티커명의 변수에 저장됩니다.

FRED 데이터 다운로드

미국 및 각국의 중요 경제지표 데이터를 살펴볼 때, 가장 많이 참조되는 곳 중 하나가 미 연방준비 은행에서 관리하는 Fred Economic Date 입니다. getSymbols() 함수를 통해 FRED 데이터를 다운로드 받을 수 있습니다. 먼저 미 국채 10년물 금리를 다운로드 받는 예제를 살펴보도록 하겠습니다.

```
getSymbols("DGS10", src="FRED")
chart_Series(DGS10)
```

먼저 미 국채 10년물 금리에 해당하는 티커인 "DGS10"을 입력해 줍니다. 그 후, 데이터 소스에 해당하는 src에 "FRED"를 입력해 주면, 해당 데이터가 다운로드 됩니다. chart_Series를 통해 해당 데이터를 그래프로 나타내면 다음과 같습니다.

[그림] 미 국채 10년물 금리 그래프



각 항목 별 티커를 찾는 방법은 매우 간단합니다. 먼저 FRED의 홈페이지ⁱⁱ에 접속하여, 원하는 데이터를 검색합니다. 예시로써 원/달러 환율에 해당하는 **South Korea / U.S. Foreign Exchange Rate** 를 검색하여 원하는 페이지에 접속합니다. 이 중 페이지 주소에서 /series/ 다음에 위치하는 **DEXKOUS** 가 해당 항목의 티커입니다.

[그림] FRED 사이트 내 원/달러 환율의 티커 확인



해당 티커를 입력하면, 홈페이지와 동일한 데이터가 다운로드 됨이 확인됩니다. 이 외에도 509,000 여개의 방대한 FRED 데이터를 해당 함수를 통해 손쉽게 R에서 다운로드 받을 수 있습니다.

```
getSymbols("DEXKOUS", src="FRED")
```

```
tail(DEXKOUS)
```

DEXKOUS

2018-06-15 1101.89

2018-06-18 1105.07

2018-06-19 1110.64

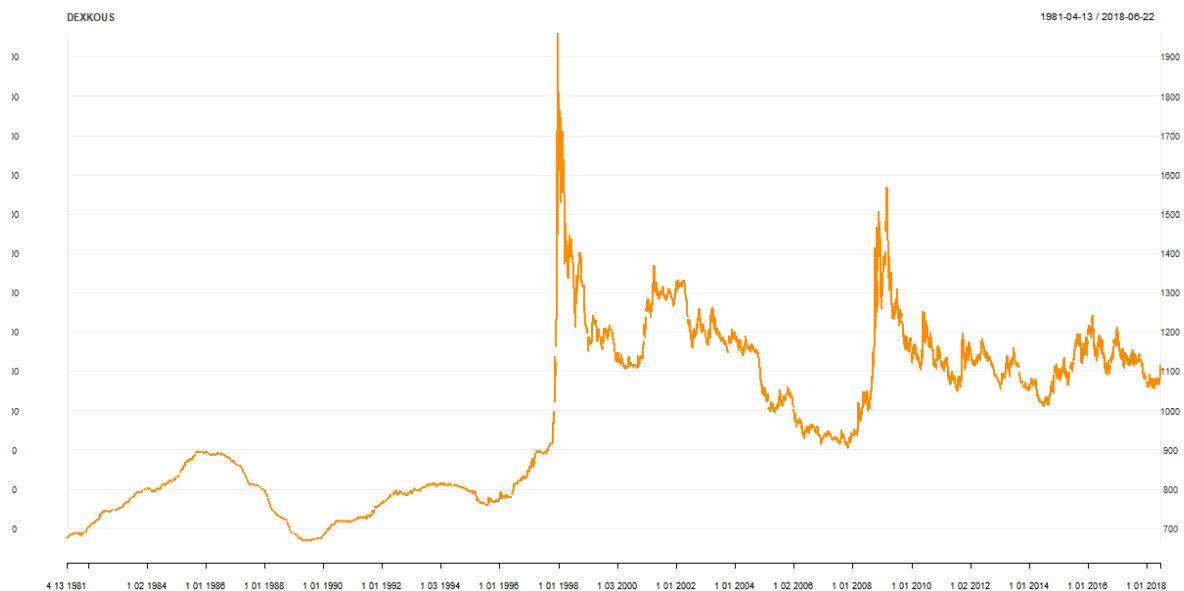
2018-06-20 1107.42

2018-06-21 1110.01

2018-06-22 1114.49

```
chart_Series(DEXKOUS)
```

[그림] 원/달러 환율 그래프



ⁱ 야후 파이낸스에서 제공하는 데이터의 경우 기본적으로 시가, 고가, 저가, 종가 모두 주식 유·무
상증자나 액면분할 등을 고려한 수정주가를 제공합니다.

ⁱⁱ <https://fred.stlouisfed.org/>