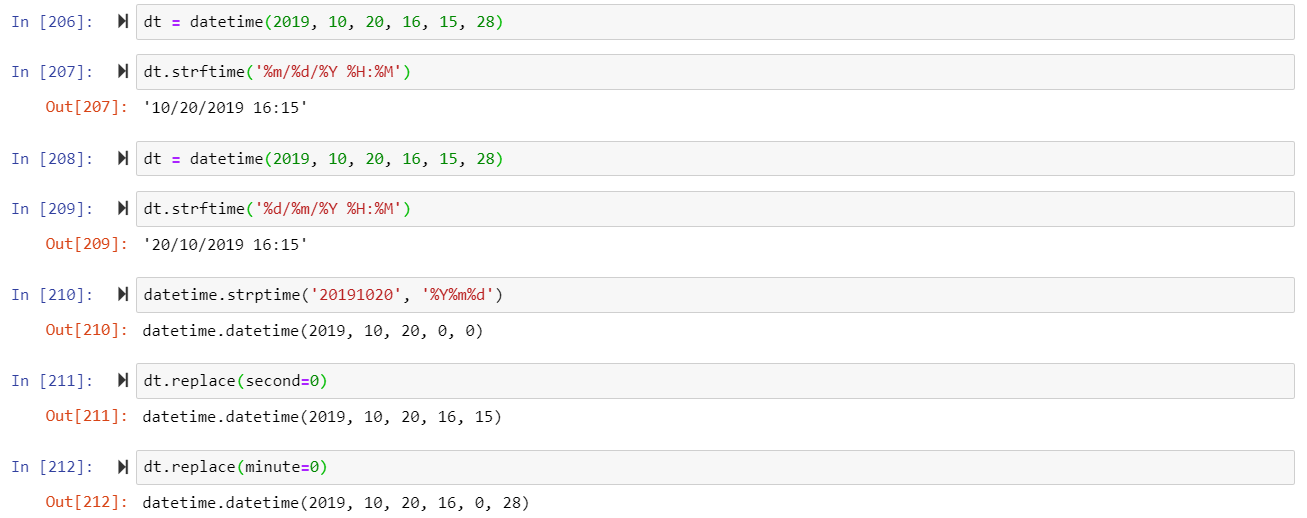


**TABLE OF CONTENTS**

* Chapter 1: Preliminaries
* [Chapter 2: Python Language Basics, IPython, and Jupyter Notebooks](http://nbviewer.ipython.org/github/pydata/pydata-book/blob/2nd-edition/ch02.ipynb)
* [Chapter 3: Built-in Data Structures, Functions, and Files](http://nbviewer.ipython.org/github/pydata/pydata-book/blob/2nd-edition/ch03.ipynb)
* [Chapter 4: NumPy Basics: Arrays and Vectorized Computation](http://nbviewer.ipython.org/github/pydata/pydata-book/blob/2nd-edition/ch04.ipynb)
* [Chapter 5: Getting Started with pandas](http://nbviewer.ipython.org/github/pydata/pydata-book/blob/2nd-edition/ch05.ipynb)
* [Chapter 6: Data Loading, Storage, and File Formats](http://nbviewer.ipython.org/github/pydata/pydata-book/blob/2nd-edition/ch06.ipynb)
* [Chapter 7: Data Cleaning and Preparation](http://nbviewer.ipython.org/github/pydata/pydata-book/blob/2nd-edition/ch07.ipynb)
* [Chapter 8: Data Wrangling: Join, Combine, and Reshape](http://nbviewer.ipython.org/github/pydata/pydata-book/blob/2nd-edition/ch08.ipynb)
* [Chapter 9: Plotting and Visualization](http://nbviewer.ipython.org/github/pydata/pydata-book/blob/2nd-edition/ch09.ipynb)
* [Chapter 10: Data Aggregation and Group Operations](http://nbviewer.ipython.org/github/pydata/pydata-book/blob/2nd-edition/ch10.ipynb)
* [Chapter 11: Time Series](http://nbviewer.ipython.org/github/pydata/pydata-book/blob/2nd-edition/ch11.ipynb)
* [Chapter 12: Advanced pandas](http://nbviewer.ipython.org/github/pydata/pydata-book/blob/2nd-edition/ch12.ipynb)
* [Chapter 13: Introduction to Modeling Libraries in Python](http://nbviewer.ipython.org/github/pydata/pydata-book/blob/2nd-edition/ch13.ipynb)
* [Chapter 14: Data Analysis Examples](http://nbviewer.ipython.org/github/pydata/pydata-book/blob/2nd-edition/ch14.ipynb)
* [Appendix A: Advanced NumPy](http://nbviewer.ipython.org/github/pydata/pydata-book/blob/2nd-edition/appa.ipynb)
* Appendix B: More on the IPython System

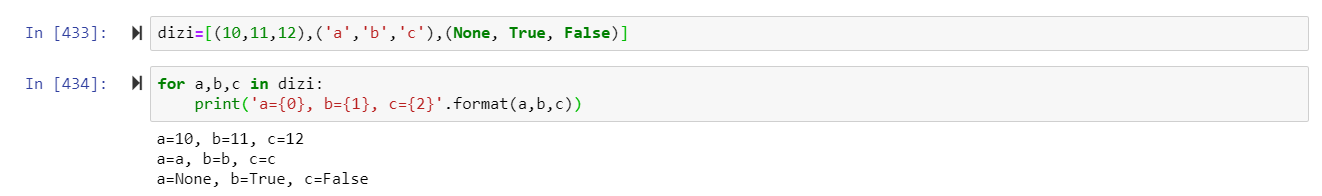


* Burada tarihin ve zamanın gösterim biçimini değiştirmek için strftime() fonksiyonu, birleşik yazılmış tarihler için strptime() fonksiyonuyla ayırabileceğimizi, replace() fonksiyonuyla değerleri değiştirebileceğimiz gösteriliyor.

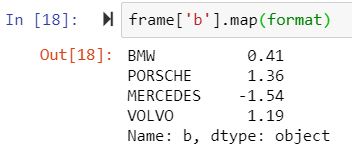
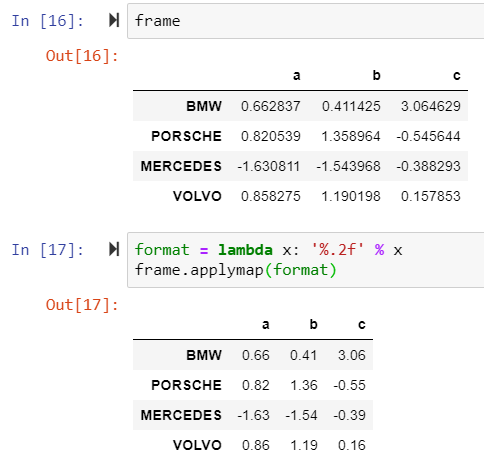


* deger değişkeni None olduğunda o değişkeni atlayıp bir sonraki elemana geçmesi için ‘continue’ komutu kullanılıyor. Birinci kod bloğunda none olmayan değişkenleri toplarken, ikinci kod bloğunda sayaç artırılıp sayacın son değeri gösteriliyor.



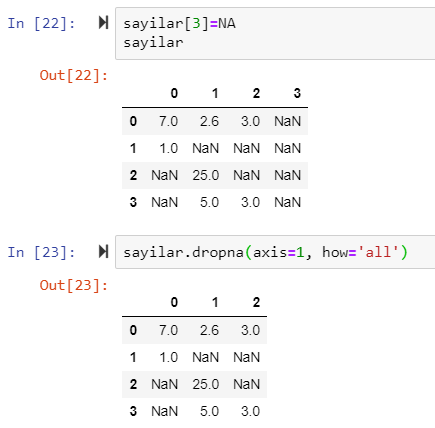
* Tuple türü, içerisindeki değerlerin değiştirilmesine izin vermeyen bir koleksiyon çeşididir.  
  
* Dizi içerisindeki tuple verileri, tek tek incelemek için format fonksiyonunu kullanarak, for döngüsüyle diziden tek tek okuyabiliriz.

## Fonksiyon Uygulaması ve Haritalama



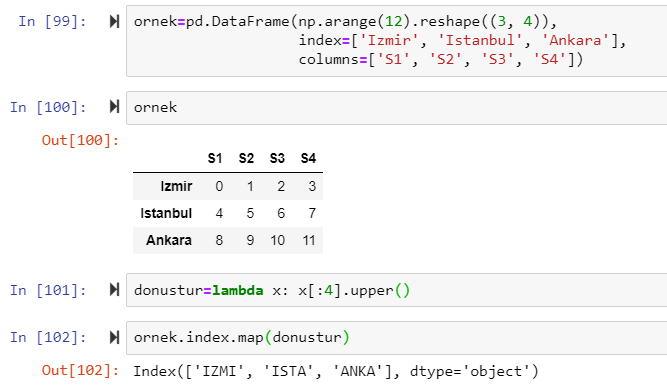
* Applymap() fonksiyonunu, frame tablosunun değerlerini yeni değerlere dönüştürmek istediğimizde kullanabiliriz. Bu örnekte tablonun değerlerini virgülden sonra 2 basamak gösterecek şekilde yuvarlanmış sayılar halinde yeniden güncelledim. Eğer sadece bir sütunu görmek istersem yukarıdaki gibi frame['b'].map(format) koduyla, b sütununa uygulanan map işlemini görebilirim.

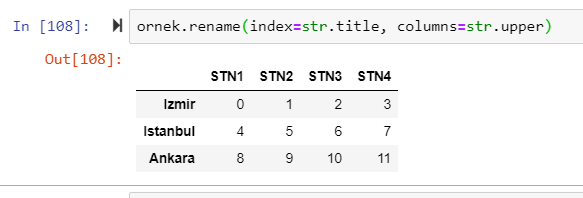
## Eksik Verileri Filtreleme



* Bu örnekte de sayilar serisine tamamı NaN değerlerden oluşan yeni bir sütun ekledim. Axis=1 ifadesiyle sütun bazında işlem yapmasını, how=’all’ ifadesiyle de tamamı NaN olan sütunları bulmasını, dropna() metoduyla da bulduğu sütunu kaldırmasını istemiş oluyorum.

**Eksen Dizinlerini Yeniden Adlandırma**

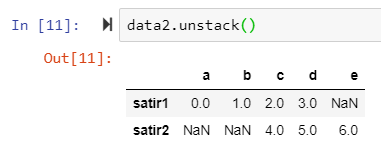


* Burada veri dönüştürme işlemlerine örnek gösteriliyor. donuştur değişkenine indekslerin ilk 4 karakterlerinin alınmasını ve upper() fonksiyonu ile büyük harfe dönüştürülmesini, ornek.index.map(donuştur) koduyla da eşleştirmenin sağlanıp indekslere istediğim işlemin uygulanmasını sağlamış oldum. DataFrame’in orjinalini bozmadan güncelleme yapmak istediğimizde ise rename() metodunu kullanabiliriz. Örnekte görüldüğü gibi sütun başlıklarının tamamını büyük harfe dönüştürdüm. Bu işlem DataFrame’in orjinalini bozmadan gerçekleştiriliyor.

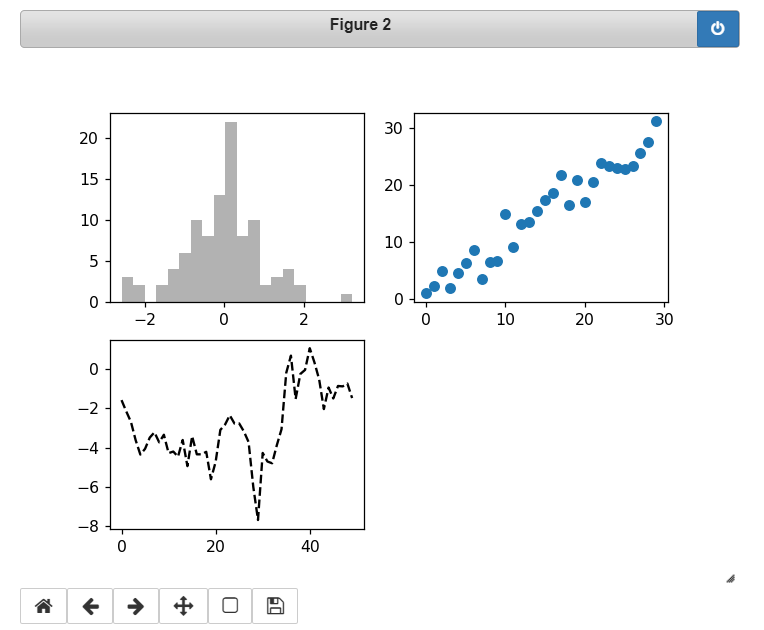
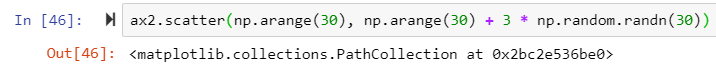
## Hiyerarşik Endeksleme ile Yeniden Şekillendirme



* s1 ve s2 adında iki seri oluşturdum. Concat() metoduyla bu iki seriyi birleştirdim. Serilerin başlangıç değerlerinden itibaren satir1 ve satir2 olarak başlık verdim.

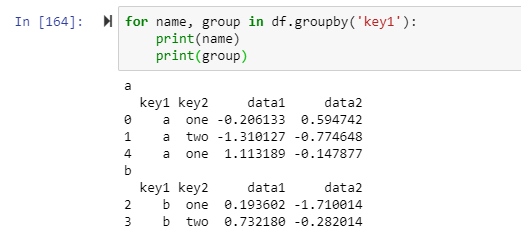
data2’ye unstack() metodunu uyguladığımızda yandaki formata dönüşüyor. Burada dikkat etmemiz gereken nokta s1’de e elemanı olmamasına rağmen, s2’de olduğu için s1’de o elemana NaN değer oluşturarak yer verdi. Düzeydeki değerlerin tümü alt gruplarda bulunmazsa, ayrılmamış veriler eksik veriye yol açabilir.

## Çizim ve Görselleştirme



* Burada ise ax3 grafiğine nokta modelli grafik çizimi yapıldı. 30x30luk düzlemde 30 tane noktanın dağılımı grafiğe dökülmüştür.
* En başta bulunan plt.close(‘all’) koduyla grafik işlemlerinin karışmaması için diğer grafikleri temizlemiş oluyoruz. Devamındaki yatay sütun grafiğinde ise tablodaki verilere bakarak gün içindeki her değer için birden fazla gözlem olduğundan, çubuklar tip\_pct'nin ortalama değeridir. Çubuklara çizilen siyah çizgiler %95 güven aralığını temsil eder. Ayrıca bu isteğe bağlı değişkenlerle yapılandırılabilir.

## Gruplar Üzerinde İterasyon



* GroupBy nesnesi, veri yığınıyla birlikte grup adını içeren 2 tuples dizisi oluşturan yinelemeyi destekler. Yandaki ekran görüntüsünde örneği mevcuttur.