

R ile NOAA üzerinden internet'ten Türkiye Şehirlerinin Sıcaklık verilerini İndirmek

Barış Sanlı, www.barissanli.com , barissanli2@gmail.com

Komutlar:

```
library(noaa)
ncdc_locs
grepl
ncdc_datasets
ncdc_datacats
ncdc_stations
ncdc
ncdc_plot
plot
```

Enerji analizlerindeki en temel parametrelerden biri sıcaklıktır. İnternet'te bir çok kaynakta ücretsiz sıcaklık verisi olmasına rağmen, bunları R üzerinden indirerek, işlemler yapmak, kodlama ve biraz da sabır istemektedir. Bu makalede R ile NOAA (ABD Ulusal Oşinografi ve Atmosfer kurumunun)'ın herkese açık sisteminden verileri indirerek, biraz işlem yapıp, sonra da grafiklemeyi deneyeceğiz.

Amaç: Ankara için 2016 ve 2017 birer yıllık sıcaklık değerlerinin grafiklenmesi

NOAA'dan Giriş Anahtarı Almak

NOAA hizmetini ücretsiz vermesine rağmen sizden bir email isteyerek, bu anahtarı oraya göndermektedir. 2017 Kasım ayı itibari ile bu sayfa

<https://www.ncdc.noaa.gov/cdo-web/token>


adresindedir.



Request Web Services Token

To gain access to [NCDC](#) CDO Web Services, register with your email address. An email will be sent with a unique token which will allow access RESTful services. For more information about CDO Web Services [read the documentation for CDO Web Services guide](#).

Please enter your email address



Submit your email address

- Enter your email address into the field above.
- Click on the SUBMIT button.

Help

General Help

- [Online help](#)
- [Check request status](#)
- [Request assistance](#)
- [Contact Us](#)

Documentation and

- [What is a RESTful web serv](#)
- [View data samples & docu](#)
- [NCDC Web Services](#)
- [CDO Web Services Docum](#)

Sonra yukarıdaki alana adını yazdığınız emaile aşağıdaki gibi bir posta geliyor. Ben kendi mail ve anahtarımı resimden sildim ama email ve Token'ın karşısında veriler var.

You are receiving this email because a request was made using this email address at the NCDC Climate Data Portal website. This email is automatically created and distributed, so please do not reply to this email.



NOAA NATIONAL CLIMAT
NATIONAL OCEANIC AND ATM



NCDC web service key token request

Email:

Token:

"noaa" kütüphanesinin kurulması

Ardından R ekranında, (Rstudio veya R veya jupyter'de) bir sefere mahsus, NOAA verilerine erişim imkanı sağlayacak olan "rnoaa" kütüphanesini sisteme kurmak gerekiyor

```
# Eğer rnoaa kütüphanesini yüklemiyorsanız  
install.packages("rnoaa")
```

Tüm kütüphane kurulduktan sonra, eğer sorun yok ise (error vs), şimdi kütüphaneyi kullanmaya geçebilirsiniz.

rnoaa'nın kullanımı

Kütüphaneyi devreye almak için

library(rnoaa) yazmanız yeterli

```
# verileri ücretsiz olarak noaa.gov'dan indireceğiz.  
# https://github.com/ropensci/rnoaa  
# Station list here = ftp://ftp.ncdc.noaa.gov/pub/data/noaa/isd-history.txt  
library(rnoaa)  
tokenX<- "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```

tokenX diye benim "XXXX..." atadığım değişkene ise size gelen anahtarı yazın. Eğer anahtar yok ise veri de yok. Her komutun sonunda, anahtarımızı eklemek zorundayız yoksa veri çekemeyiz. Ben kendi anahtarına

tokenX

ismini verdim.

Veri Yapısı

rnoaa'yı anlamak biraz zaman alabiliyor. Dünya üzerinde yerleşim yerleri var, şehirler gibi, bu yerleşim yerlerinde ise bazen 1 bazen 10'larca hava durumu istasyonları var. Bu istasyonlardaki verilerin sınıflandırıldığı verisetleri ve bu verisetlerinde veritipleri var.

Mesela Ankara (CITY:TU000005) 'daki (GHCND:TUM00017130) istasyonunda, "GHCND" veri tabanı altında günlük en düşük sıcaklıklar "TMIN" var. Yani hiyerarşiye dikkat. Fakat önce şehirleri bir görelim

Yerleşimleri Listelemek

Eğer herşey yolunda olursa önce alfabetik sıralamaya göre ilk 1000 yerleşim yeri ismini indirelim ve bunu ids isimli değişkene depolayalım.

```
# önce listedeki tüm şehir isimli verileri indirmeye çalışalım  
# dikkat tüm verilerde NOAA veri talebini 1000 ile sınırlamakta, tüm listeyi tek seferde indirmek zor  
ids<-ncdc_locs(locationcategoryid='CITY', sortfield='name', sortorder='asc', token=tokenX, limit=1000)
```

Bir şeye dikkat edin, her komutun arkasından "token=" ile anahtarımızı eklemesek komut çalışmaz.

Yerleşim yeri verileri şimdi ids isimli değişkende, ids değişkeninin başlangıcına bir göz atalım. "**head(ids)**" komutu ile"

```
head(ids)
```

```
$meta  
$totalCount  
1980  
$pageCount  
1000  
$offset  
1  
  
$data  
  mindate  maxdate      name  datacoverage      id  
-----  
1950-12-01 2017-09-30 's-Hertogenbosch, NL      1.0000 CITY:NL000001  
1910-12-01 2017-11-07      Abakan, RS      0.9993 CITY:RS000001  
1950-01-01 2017-11-07      Abeche, CD      0.9253 CITY:CD000001  
1893-01-01 2017-11-11      Aberdeen, SD US      1.0000 CITY:US460001
```

Türkiye'deki Şehirlerin Olduğu Sütunları Bulmak

Listede yok yok, fakat biz daha çok Türkiye'dekiler ile ilgileneceğimizden. Yukarıdaki tabloya dikkat edilirse, "name" sütununda şehrin ismi ve ülke kodu var. Türkiye'nin ülke kodu TU, yani tüm listede isim sütununda ", TU" olan satırları ayıklamak Türkiye'deki yerleşim isimlerini verecektir.

Bunun için de "grep" komutu ile ", TU" olan yerleşimleri ayırıyoruz.

```
# Türkiye'deki istasyonlarda ismin sonrasında , TU var. bu satırları görelim grep ile
ids$data[grep(pattern=" , TU",x=ids$data[,3])==TRUE,]
```

	mindate	maxdate	name	datacoverage	id
16	2007-04-01	2017-11-07	Adiyaman, TU	1.0000	CITY:TU000001
18	2003-11-01	2017-11-07	Agri, TU	1.0000	CITY:TU000002
22	1973-01-01	2017-11-07	Aintab, TU	1.0000	CITY:TU000003
62	2002-12-01	2017-11-07	Ankara, TU	1.0000	CITY:TU000005
67	1951-01-01	2017-11-07	Antalya, TU	0.9819	CITY:TU000006
114	1982-11-01	2017-11-07	Aydin, TU	1.0000	CITY:TU000008
123	1950-01-01	2017-11-07	Balikesir, TU	0.9973	CITY:TU000009
193	1998-05-01	2017-11-07	Bingol, TU	0.9999	CITY:TU000010
221	1973-01-01	2017-11-07	Bolu, TU	0.8286	CITY:TU000011
273	1929-01-01	2017-11-07	Burdur, TU	1.0000	CITY:TU000012
277	1973-01-01	2017-11-07	Bursa, TU	1.0000	CITY:TU000013
299	1963-01-01	2017-11-07	Canakkale, TU	0.9178	CITY:TU000014

İsim listesine bakınca kendi aradığım "Ankara, TU"yu ve onun kimlik kodu olan "id" sütunundaki "CITY:TU000005" görüyoruz.

Lokasyonu belirledikten sonra bu lokasyondaki istasyonları bilmemiz gerekiyor. Yani Ankara'nın kimliği CITY:TU000005 ama buradaki istasyonlar birden fazla olabilir. Bu yerleşimdeki istasyonların isimlerini sormamız gerekiyor. Çünkü sıcaklık verilerini istasyona göre çekeceğiz

Ankara'daki Havadurumu İstasyonunun kodu

```
# Yukarıda Ankara'nın kodunu CITY:TU000005 olarak bulmuştuk
# Şimdi Ankara'daki hava istasyonlarının kodları bulmaya çalışalım
```

```
stations<-ncdc_stations(locationid = 'CITY:TU000005', token=tokenX)
stations
```

```
$meta
$meta$totalCount
[1] 1
```

```
$meta$pageCount
[1] 25
```

```
$meta$offset
[1] 1
```

```
$data
  elevation  mindate  maxdate latitude  name datacoverage
1      891 2002-12-01 2017-11-07  39.95 ANKARA CENTRAL, TU          1
  id elevationUnit longitude
1 GHCND:TUM00017130    METERS  32.883
```

```
attr(,"class")
[1] "ncdc_stations"
```

normalde "stations" verisindeki "data" alt değişkeninde (stations\$data)'da benim ekran'da bir alt satıra kaymış olan "id" sütununda GHCND:TUM00017130 isimli istasyon kodunu görebilirsiniz.

Peki Ankara (CITY:TU000005) 'daki (GHCND:TUM00017130) istasyonu ile ilgili hangi verilerin ulaşılabileceğini, depolandığını sorgulayalım.

İstasyona Ait Verisetleri

Bunun için `ncdc_datasets` komutunda `stationid = "GHCND:TUM00017139"` da yazabiliriz.

```
ncdc_datasets(stationid = stations$data$id[1],token=tokenX)
```

```
$meta  
$meta$offset  
[1] 1
```

```
$meta$count  
[1] 3
```

```
$meta$limit  
[1] 25
```

```
$data  
      uid      mindate      maxdate      name  
1 gov.noaa.ncdc:C00861 1763-01-01 2017-11-09      Daily Summaries  
2 gov.noaa.ncdc:C00946 1763-01-01 2017-10-01 Global Summary of the Month  
3 gov.noaa.ncdc:C00947 1763-01-01 2017-01-01 Global Summary of the Year  
  datacoverage      id  
1           1 GHCND  
2           1 GSOM  
3           1 GSOY
```

```
attr(,"class")  
[1] "ncdc_datasets"
```

Sonuç aynı olacaktır.

```
# ncdc_datasets(stationid = stations$data$id[1],token=tokenX) |  
ncdc_datasets(stationid = "GHCND:TUM00017130",token=tokenX)
```

```
$meta  
$meta$offset
```

Bu istasyon için 3 tane veri tabanına erişim imkanı vardır bunlar:

- GHCND : Günlük özetler (tarihi olarak)
- GSOM : Aylık özetler
- GSOY : Yıllık özetler

Bu noktaya gelince elimizde Ankara'da istasyon için 3 veri tabanı verisi olduğunu gördük. Fakat bu veritabanlarındaki veri başlıkları neler?

İstasyondaki Verisetindeki Veri Değişkenleri

Bunun içinde `stationid` de verdiğimiz istasyonun GHCND veri tabanında yer alan veri başlıklarını `ncdc_datatypes` ile sorgulayacağız:

```
# ncdc_datatypes(stationid = stations$data$id[1], datasetid = "GHCND", token=tokenX)|
ncdc_datatypes(stationid = "GHCND:TUM00017130", datasetid = "GHCND", token=tokenX)
```

```
$meta
  offset count limit
1      1      5    25

$data
  mindate  maxdate          name datacoverage  id
1 1781-01-01 2017-11-08    Precipitation          1 PRCP
2 1857-01-18 2017-11-08      Snow depth          1 SNWD
3 1874-10-13 2017-11-09 Average Temperature.          1 TAVG
4 1763-01-01 2017-11-09 Maximum temperature          1 TMAX
5 1763-01-01 2017-11-09 Minimum temperature          1 TMIN

attr(,"class")
[1] "ncdc_datatypes"
```

Görüldüğü üzere bu istasyonda 5 veri var, PRCP (yağış), SNWD(kar kalınlığı), TAVG(ortalama sıcaklık), TMAX(en yüksek sıcaklık), TMIN(minimum sıcaklık)

Veriyi İndirmek

Şimdi ise bu istasyondaki, GHCND verisetindeki TMIN verisini belirtilen tarihler arasında metric olarak okuyalım(Unutmayın tek seferde en fazla 1000 tane indirebilirsiniz):

Bunun için "ncdc" komutunda:

- datasetid ile GHCND verisetini
- stationid'de daha önce Ankara için istasyon verisini değiştikten veya doğrudan el ile "GHCND:TUM00017130"
- datatypeid 'de minimum sıcaklık anlamını taşıyan "TMIN"
- startdate başlangıç tarihi (YIL-AY-GÜN)
- enddate bitiş tarihi
- token anahtarımız tokenX
- units ile metrik birimleri (yoksa fahrenheit gelir)
- limit ile de veri miktarını sınırlıyoruz, ben hep limit=1000 kullanıyorum, maksimum'a kadar veri çekmek için

```
mintemp<-ncdc(datasetid = "GHCND",
  stationid = stations$data$id[1],
  datatypeid = "TMIN",
  startdate = "2016-11-10",
  enddate = "2017-11-09",
  token=tokenX,
  units="metric",
  |limit=1000)
```

bu çektiğimiz veriyi "mintemp" değişkenine atalım. Eğer sorun yok ise "mintemp" değişkenine kısaca bakalım

Bunun için "head(mintemp)" yazmamız yeterli.

```
head(mintemp)

$meta
$totalCount
268
$pageCount
1000
$offset
1

$data
      date  datatype      station  value  fl_m  fl_q  fl_so  fl_t
-----
2016-11-12T00:00:00  TMIN  GHCND:TUM00017130    40      S
2016-11-13T00:00:00  TMIN  GHCND:TUM00017130    43      S
2016-11-18T00:00:00  TMIN  GHCND:TUM00017130   -48      S
2016-11-19T00:00:00  TMIN  GHCND:TUM00017130   -30      S
```

Görüldüğü üzere mintemp içinde data alt kümesinde 8 sütun veri var.

mintemp\$data[,4] "value" sütunundaki tüm verilere eşittir. **Burada dikkat edilmesi gereken birşey var. İstasyon verisinde ondalık değer kullanılmadığından söz konusu değer 10'a bölünmesi gerekir.**

O yüzden 4.sütundaki tüm verileri 10'da 1'i ile değiştireceğiz:

Bunun için "mintemp\$data[,4]<-(mintemp\$data[,4])/10" komutu yeterli olacaktır.

```
stations$data$id[1]
mintemp$data[,4]<-(mintemp$data[,4])/10

'GHCND:TUM00017130'
```

Şimdi bu gerçek değişkenin bir de grafiğini görelim. Yani 2016 Kasım'dan 2017 Kasım'a Ankara'daki istasyonun ölçtüğü minimum sıcaklıklar nasıl değişmiş.

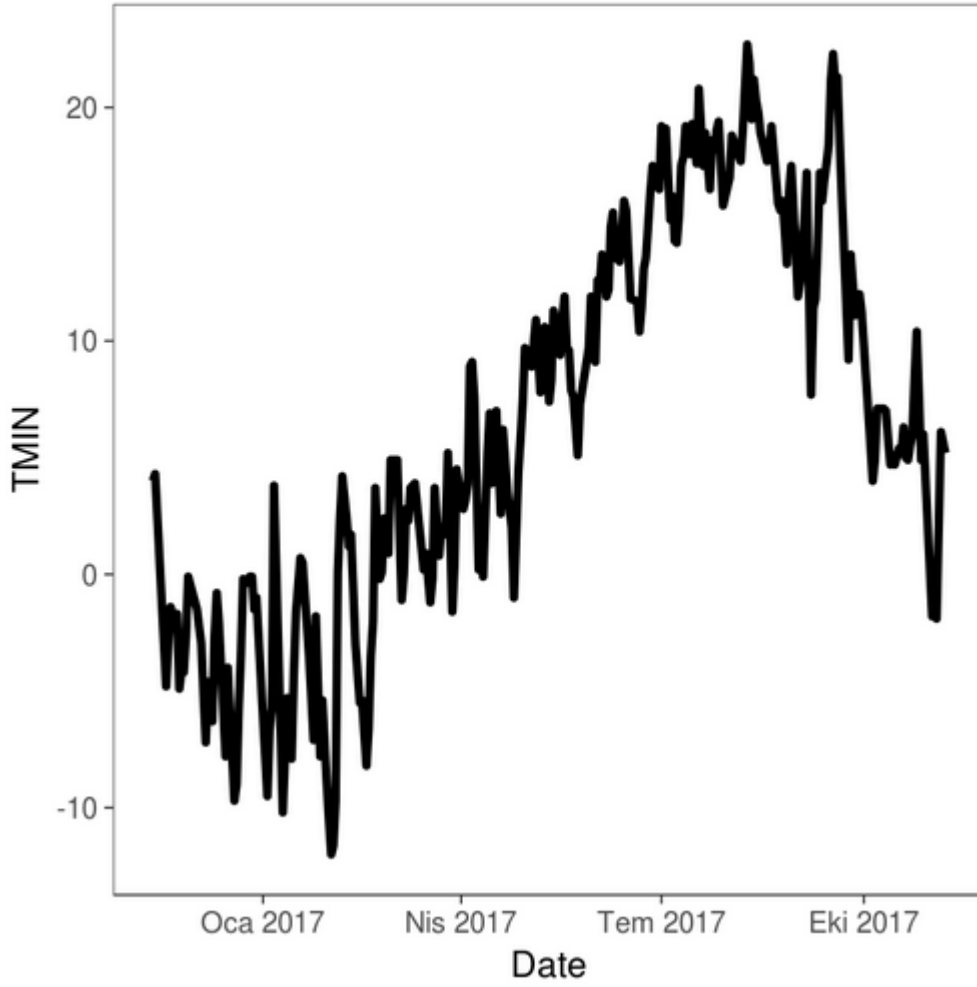
Grafiklemek

Bunun için rnoaa kütüphanesindeki:

- ncdc_plot komutuna mintemp verisini girerek

istediğimiz verinin grafiğini elde edebiliriz.


```
ncdc_plot(mintemp)
```



İşi biraz daha geliştirelim hem 2015-2016 hem 2016-2017 Kasım'dan Kasıma sıcaklık grafiğini çıkaralım.

Bunun için gerekli verileri de "mintemp2016" değişkenine atıp, yine değerleri 10'a bölelim.

```
mintemp2016<-ncdc(datasetid = "GHCND", stationid = stations$data$id[1], datatypeid = "TMIN",  
startdate = "2015-11-10", enddate = "2016-11-09", token=tokenX, units="metric",limit=1000)
```

```
mintemp2016$data[,4]<- (mintemp2016$data[,4])/10  
head(mintemp2016)
```

\$meta

\$totalCount

271

\$pageCount

1000

\$offset

1

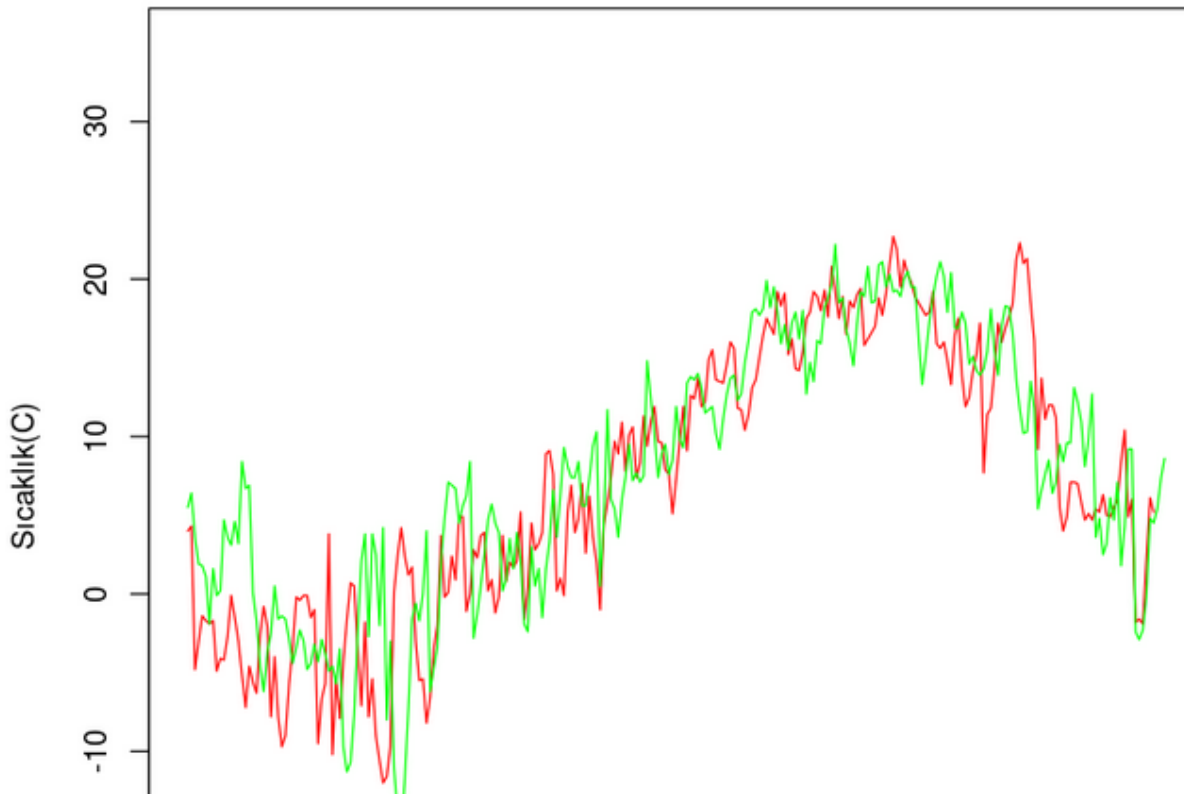
\$data

	date	datatype	station	value	fl_m	fl_q	fl_so	fl_t
	2015-11-10T00:00:00	TMIN	GHCND:TUM00017130	5.5				S
	2015-11-11T00:00:00	TMIN	GHCND:TUM00017130	6.4				S
	2015-11-13T00:00:00	TMIN	GHCND:TUM00017130	3.6				S
	2015-11-14T00:00:00	TMIN	GHCND:TUM00017130	1.9				S

Şimdi hem son 1 yıldaki hem de ondan önceki yılın verisini birlikte çizdirelim

```
plot(mintemp$data$value,type="l",col="red", ylim=c(-20,35),xlab="gün",ylab="Sıcaklık(C)")  
lines(mintemp2016$data$value,col="green", ylim=c(-20,35))  
title(main="2015-2016 Kasım(yeşil) ve 2016-2017 Kasım(kırmızı) min sıcaklık")
```

2015-2016 Kasım(yeşil) ve 2016-2017 Kasım(kırmızı) min sıcaklık



Veriler elimizde olduğu için ister histogram yapar, ister matematiksel olacak işlem yapabiliriz.

İyi çalışmalar, önerilerinizi bekliyorum

barissanli2@gmail.com

Tüm KOD

```
#Tüm kod
library(rnoaa)
tokenX<-"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
ids<-ncdc_locs(locationcategoryid='CITY', sortfield='name', sortorder='asc',token=tokenX, limit=1000)
head(ids)
# Türkiye'deki istasyonlarda ismin sonrasında , TU var. bu satırları görelim grep ile
ids$data[grepl(pattern=" TU",x=ids$data[,3])==TRUE,]
ncdc_datasets( token=tokenX, locationid = 'CITY:TU000005' )
# ncdc_datatypes(stationid = stations$data$id[1], datasetid = "GHCND", token=tokenX)
ncdc_datatypes(stationid = "GHCND:TUM00017130", datasetid = "GHCND", token=tokenX)
mintemp<-ncdc(datasetid = "GHCND",
              stationid = stations$data$id[1],
              datatypeid = "TMIN",
              startdate = "2016-11-10",
              enddate = "2017-11-09",
              token=tokenX,
              units="metric",
              limit=1000)
mintemp$data[,4]<-(mintemp$data[,4])/10
ncdc_plot(mintemp)
```

#Tüm kod

```
library(rnoaa)
```

```
tokenX<-"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```

```
ids<-ncdc_locs(locationcategoryid='CITY', sortfield='name',
sortorder='asc',token=tokenX, limit=1000)
```

```
head(ids)
```

```
# Türkiye'deki istasyonlarda ismin sonrasında , TU var. bu satırları görelim grep ile
```

```
ids$data[grepl(pattern=" TU",x=ids$data[,3])==TRUE,]
```

```
ncdc_datasets( token=tokenX, locationid = 'CITY:TU000005' )
```

```
# ncdc_datatypes(stationid = stations$data$id[1], datasetid = "GHCND",
token=tokenX)
```

```
ncdc_datatypes(stationid = "GHCND:TUM00017130", datasetid = "GHCND",
token=tokenX)
```

```
mintemp<-ncdc(datasetid = "GHCND",
              stationid = stations$data$id[1],
```

```
datatypid = "TMIN",  
startdate = "2016-11-10",  
enddate = "2017-11-09",  
token=tokenX,  
units="metric",  
limit=1000)  
mintemp$data[,4]<-(mintemp$data[,4])/10  
ncdc_plot(mintemp)
```