

R ile Enerji Analizi – Bölüm 9- Türkiye Aylık Doğal gaz Talep Modeli ve 2025'e Projeksiyonun Yapılması

Barış Sanlı

Özet: Bu bölümde, 8nci bölümde gördüğümüz TBATS yöntemi ile Türkiye aylık doğalgaz talep tahmin modeli oluşturmaya çalışacağız. İlk kısımda tüm ülkelerin aylık petrol ve doğal gaz istatistiklerine ulaşabileceğiniz JODIDB.org websitesini de kullanmayı göstermeye çalışacağım. Bu web sitesinden istediğiniz veriyi seçip, CSV olarak indirdikten sonra dersimizin 8. Bölümünde olduğu gibi önce bileşenlerine ayırıp sonra da 2015-2025 tahminini alacağız. Bu tahmin %95 ve %80 kesinlikli aralıklar ile çıkarılacaktır. Sonra aylık verilerden yıllık toplamlara geçerek onu da grafiklemeye çalışacağız.

Bu Bölümdeki Fonksiyonlar

```
setwd("çalışma dizini")
dir()
read.csv
ts(veri, start=başladığı tarih, frequency= periyodu)
msts(veri$ortalama_tuketim, seasonal.periods = c(periyod1, periyod2))
tbats(zaman serisi)
plot(veri, main="Grafik Başlığı")
x11()
palette(palet_ismi(renksayısı))
forecast(model, h=projeksiyon süresi)
cbind()
rbind()
aggregate(veri, frekans,FUN="fonksiyon")
```

Veri Dosyaları:

Türkiye 2010-2015 aylık doğalgaz tüketimi (JodiDB.org'dan)

<http://www.barissanli.com/calismalar/dersler/r/dogalgaz-jodi.csv>

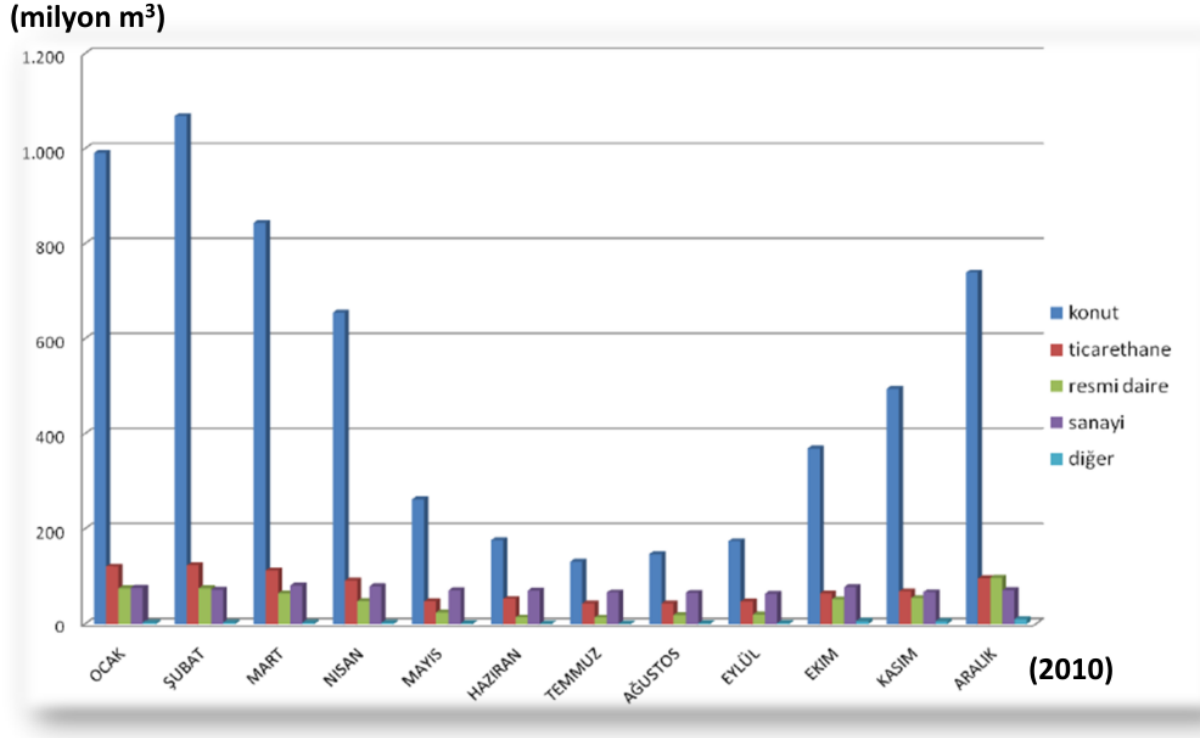
Projeksiyon sonuçları:

<http://www.barissanli.com/calismalar/dersler/r/sonuc-dg.csv>

9.0. Doğalgaz Talep Tahmini

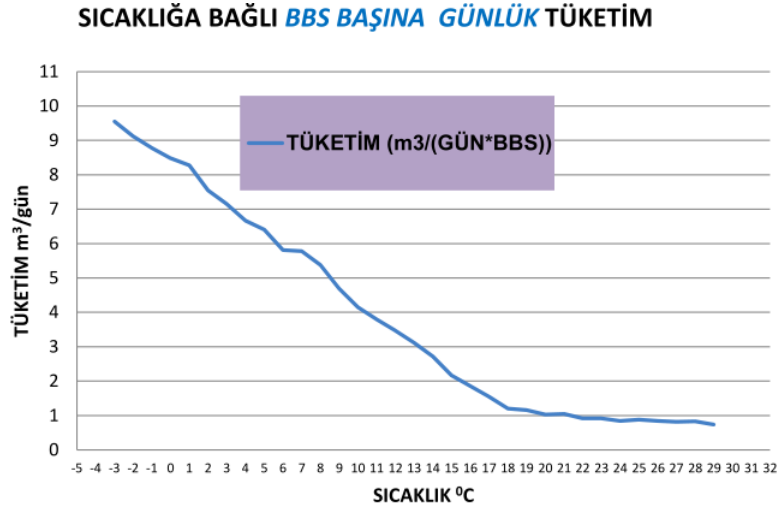
Daha önce elektrikte gördüğümüz gibi, Türkiye doğal gaz talebinde de önemli bir mevsimsellik/sezonsallık vardır. Örneğin kışın en üst tüketimler 240 milyon m³'leri görürken, yazın tüketimler 110 milyon m³'ler civarında olsa da bunun yaklaşık 14-15'i de depoya konulan gaz oluyor. Yani nihai tüketime giden gaz 90 milyon m³ olmaktadır. Bu yaz kış farkı şehirlerin gelişmişlik düzeylerine göre değişmektedir. Örneğin o şehirde sanayi yoğun ise yaz-kış tüketimi birbirine yakın, şehir ne kadar az sanayiye sahipse yaz-kış dengesizliği yüksek olmaktadır.

Elektrikte tüketimin sıcaklıkla değişimi paraboliktir, doğalgaz da ise daha doğrusal ilişkiler görmekteyiz. Bu da doğalgazın nihai tüketiminden gelmektedir. İstanbul özelinde bakarsak, kış aylarındaki yoğun tüketim aşağıdaki grafikte verilmektedir (Kaynak : Uğur Usta, ICCL, http://enerjipostasi.com/haber_resim/files/dosyalar/icci/O4_Ugur_Usta.pdf)



Grafik 1 – İstanbul doğal gaz tüketiminin alt tüketim gruplarına ve aylara göre dağılımı (Uğur Usta)

İGDAŞ'tan Uğur Usta'nın sunumundaki bir diğer grafik var ki, bu grafik bir çok doğalgaz talep tahmini içinde sıklıkla başvurduğumuz bir ilişkiyi empirik olarak göstermektedir. Konut ve ısıtma tüketiminin sıcaklığa göre değişiminin BBS (Bağımsız Birim Sayısı – EPDK birimi – ilk 200 m2 bir birim kabul edilmektedir) başına gösterimi grafiği aşağıdadır.



SICAKLIK (°C)	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
TÜKETİM m ³ /GÜN*BBS)	9,56	9,12	8,78	8,48	8,28	7,55	7,15	6,67	6,41	5,81	5,78	5,38	4,69	4,15	3,8	3,47	3,12	2,72	2,17	1,86	1,55	1,2	1,16	1,03	1,05	0,92	0,92	0,84	0,88	0,84	0,82	0,83	0,74

Grafik 2 – İstanbul için BBS başına gaz tüketimi (Uğur Usta)

Bu grafik İstanbul’da hava sıcaklığı 16 C’nin altına düşükçe doğrusal olarak doğalgaz talebinin arttığını göstermektedir.

Fakat bu daha çok konut tüketimini vermektedir. Bir de dikkat edilmesi gereken nokta sıcaklığın yazın nem ile, kışın rüzgar ile düzeltilmesi gerektiğidir. Yaz için hava sıcaklığı arttıkça konut tüketimi azalmasına rağmen, sanayi ve elektrik santrallerinin doğalgaz tüketimi artmaktadır. Aslında yazın doğal gaz talebi iki sebepten artar. Birincisi artan elektrik tüketiminin doğalgaz santrallerinden karşılanmasından dolayı, ikincisi de doğalgaz depolarına gaz atılmasından dolayıdır.

Bu kadar teknik bilgiden sonra bir de olayın matematiksel boyutuna bakalım.

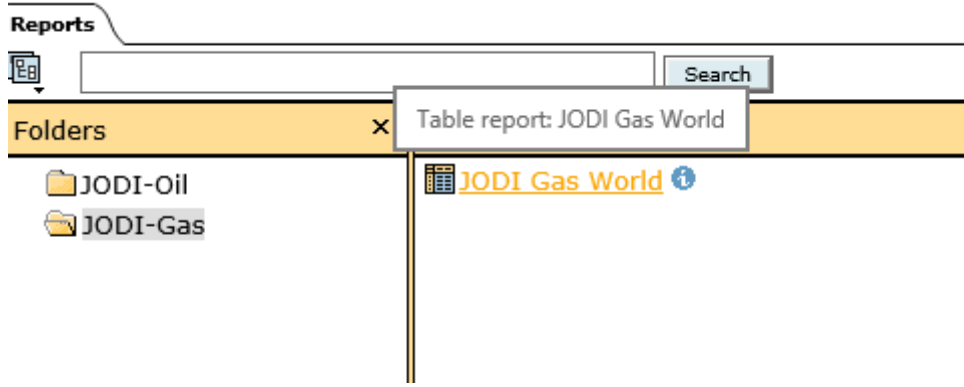
9.1. Veri Kaynağı

JODI (<https://www.jodidata.org>), tüm ülkelerin petrol, petrol ürünleri ve doğal gaz ile ilgili istatistiklerini bir araya getiren bir websitesidir. JODI’nin en güzel özelliklerinden biri ise aylık verilerin bulunması ve bir çok model için yeterli veriyi içermesidir.

Öncelikli olarak

<http://www.jodidb.org>

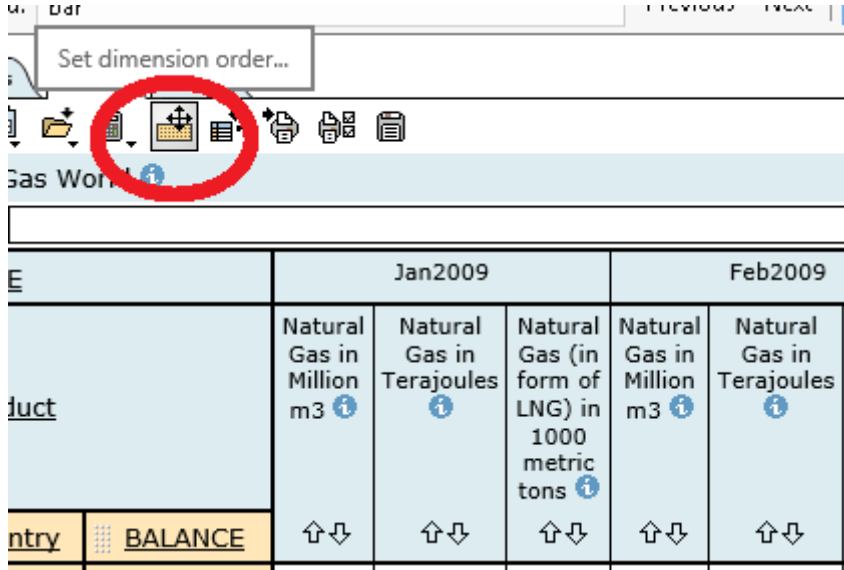
websitesine giriyoruz. Sitede sol taraftaki JODI-Gas dizinine tıkladığımızda “JODI Gas World” veri seti kümesi açılacaktır.



JODI Gas World'e girdiğimiz zaman bir anda önümüzde oldukça karmaşık bir veri ekranı görülecektir. Bu ekrandan istediğimiz, Türkiye için milyon m3 olarak üretim/tüketim vs verileridir.

TIME		Jan2009			Feb2009			Mar2009			Apr2009			May2009		
Country	BALANCE	Natural Gas in Million m3	Natural Gas in Terajoules	Natural Gas (in form of LNG) in 1000 metric tons	Natural Gas in Million m3	Natural Gas in Terajoules	Natural Gas (in form of LNG) in 1000 metric tons	Natural Gas in Million m3	Natural Gas in Terajoules	Natural Gas (in form of LNG) in 1000 metric tons	Natural Gas in Million m3	Natural Gas in Terajoules	Natural Gas (in form of LNG) in 1000 metric tons	Natural Gas in Million m3	Natural Gas in Terajoules	Natural Gas (in form of LNG) in 1000 metric tons
		↑↓	↑↓	↑↓	↑↓	↑↓	↑↓	↑↓	↑↓	↑↓	↑↓	↑↓	↑↓	↑↓	↑↓	↑↓
Argentina	Production	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A
	Receipts from Other Sources	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Imports	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A
	LNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Pipeline	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A
	Exports	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A
	LNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Pipeline	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A
	Stock Change	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A
	Gross Inland Deliveries (Calculated)	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A
Statistical Difference (Calculated)	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A	
Gross Inland																

Yukarıdaki verilere baktığımızda, sol tarafta(turuncu kısım) Arjantin için "Balance" altında, üretim,ithalat, LNG vs gibi satırlarda, ay ve yıl sütunlarında değişik birimlerde (milyon m3, TJ ve 1000 ton LNG) veri hücreleri bulunduğunu görebiliriz. Burada istediğimiz verileri seçmek için ise "Set Dimension order" olan "tablo üzerinde dört yönü gösteren ok" simgesi seçilir. (Aşağıdaki resimde kırmızı daire içinde)

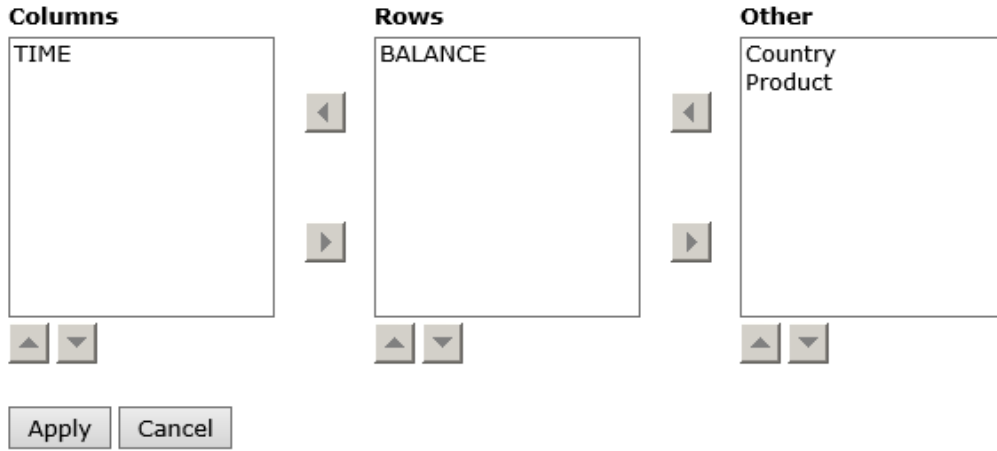


The screenshot shows a software interface with a 'Set dimension order...' dialog box at the top. Below it is a table with the following structure:

	Jan2009			Feb2009	
Product	Natural Gas in Million m3	Natural Gas in Terajoules	Natural Gas (in form of LNG) in 1000 metric tons	Natural Gas in Million m3	Natural Gas in Terajoules
Country	BALANCE				

“Set Dimension Order” a tıklayarak, verimizi önce şekle sokalım, öncelikli olarak sütunların tarih olması, satırlarında veri dizisi yani “Balance” olmasını istiyoruz, satır ve sütunlarda yer almayacak seçeneklerde ülke (“Country”) ve birimler (“Product”). Oıkları kullanarak seçeneği aşağıdaki hale getirebilirsiniz.

Set dimension order



The 'Set dimension order' dialog box has three sections: 'Columns', 'Rows', and 'Other'. The 'Columns' section contains 'TIME'. The 'Rows' section contains 'BALANCE'. The 'Other' section contains 'Country' and 'Product'. There are arrows between the sections to move items, and 'Apply' and 'Cancel' buttons at the bottom.

“Apply” a basınca ekranımız aşağıdaki şekle gelecektir. Fakat ülkeyi Türkiye yapmamız gerekiyor.

Reports Table Chart

JODI Gas World

Other: Country - Argentina Product - Natural Gas in Million m3

TIME	Jan2009	Feb2009	Mar2009	Apr2009	May2009	Jun2009	Jul2009	Aug2009	Sep2009	Oct2009	Nov2009	Dec2009	Jan2010	Feb2010
BALANCE	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
Production	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Receipts from Other Sources	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Imports	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pipeline	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Exports	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pipeline	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Stock Change	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gross Inland Deliveries (Calculated)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Statistical Difference (Calculated)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gross Inland Deliveries (Observed)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
of which: Electricity and Heat Generation	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Closing Stocks	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bunun için aşağıda görüldüğü şekilde ya "Country" e tıklayarak, yada aynı çerçevedeki sağ-sol oklarına basarak Türkiye'yi bulmaya çalışabiliriz. Benim önerim "Country" yani ülkeye tıklayın.

JODI Gas World

Other: Country - Argentina Product -

TIME	Jan2009	Feb2009	Mar2009	Apr2009
BALANCE	↕	↕	↕	↕
Production	0	0	0	0
Receipts from Other Sources	0	0	0	0

Burada ülkeye bastığımız anda karşımıza aşağı soldaki ekran açılacak. Önce aşağıdaki kırmızı daire içinde X'e tıklayarak tüm seçenekleri kaldıralım. Sonra da alfabetik sıradaki ülkelerden "Turkey"i seçelim.

JODI Gas World

Table size
14 rows by 74 columns

COLUMNS:
TIME

ROWS:
BALANCE

Other:
Country

Product

Select range

Argentina
Australia
Austria
Azerbaijan
Bahrain
Bangladesh
Barbados
Belarus
Belgium
Belize
Bolivia
Brazil
Brunei Darussalam

Türkiye'yi seçtikten sonra (altta solda) sadece en yukarıdaki "Show Table" a tıklayabiliriz (altta sağda).

Item selection

Show table

JODI Gas World

Table size
14 rows by 74 columns

COLUMNS:
TIME

ROWS:
BALANCE

Sweden
Switzerland
Thailand
Trinidad/Tobago
Tunisia
Turkey
Ukraine
United Kingdom
United States

Karşımıza Türkiye'nin verilerinin olduğu, milyon m³ üzerinden aylara göre alt kırılımlı doğalgaz verileri ekrana gelecektir. Bu veriyi aşağıdaki şekilde "açık dizin-Download" simgesine tıklayarak indirebilirsiniz.

Download

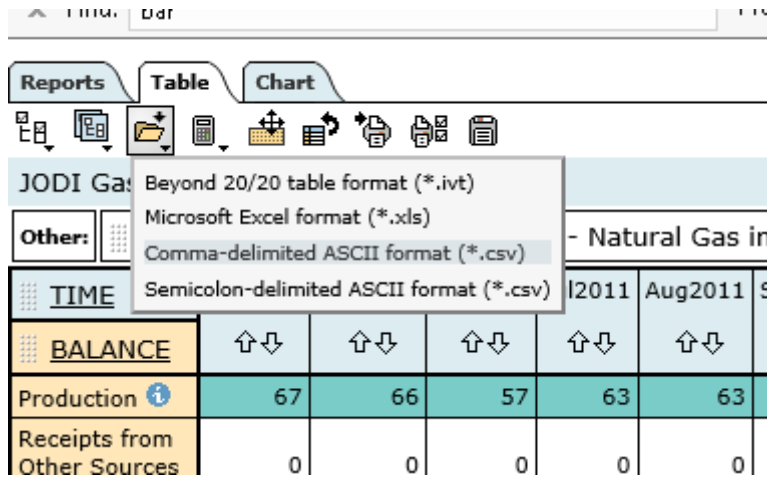
Reports Table Chart

JODI Gas World

Other: Country - Turkey Product - Natural Gas in Million m3

TIME	Apr2011	May2011	Jun2011	Jul2011	Aug2011	Sep2011	Oct2011	Nov2011	Dec2011
BALANCE	↑↓	↑↓	↑↓	↑↓	↑↓	↑↓	↑↓	↑↓	↑↓
Production	67	66	57	63	63	60	67	64	66
Receipts from Other Sources	0	0	0	0	0	0	0	0	0
Imports	3,639	3,069	2,888	3,000	3,215	3,006	3,637	4,356	4,750
LNG	588	325	398	241	495	407	414	748	810
Pipeline	3,051	2,744	2,490	2,759	2,720	2,599	3,223	3,608	3,940
Exports	64	61	50	57	52	57	55	63	70

"Download" a tıklayınca önümüze birkaç indirme seçeneği geliyor. Benim önerim Excel indirin, dosyayı düzenleyip sonra csv olarak saklayın. Ama direkt indirip CSV kullanacağım dersiniz de CSV olarak indirme seçeneği mevcut. Türkçe excel kullanıyorsanız en altta yer alan "Semicolon-delimited ASCII" yi seçin çünkü Türkçe'de ondalık sayılarda "." Veya "," kullanımı konusundaki farklılık dosya okumalarında soruna yol açabiliyor.



TIME	BALANCE	↑↓	↑↓	↑↓	↑↓	↑↓
Production	67	66	57	63	63	
Receipts from Other Sources	0	0	0	0	0	

Ben kendi indirdiğim dosyayı bu linke yerleştirdim.

<http://www.barissanli.com/calismalar/dersler/r/dogalgaz-jodi.csv>

Şimdi dosya üzerinde işlemlere başlayalım.

9.2. İlk Analizler

Herşeyden önce csv dosyasının olduğu dizinde olduğumuzdan emin olalım. Bence csv dosyalarını harddiskte doğalgaz diye bir dizin açıp oraya yerleştirin. Programınızda

```
setwd("C:/dogalgaz-projeeksiyon/")  
# dir komutu dizindeki dosyaları ekrana yazar  
dir()
```

komutları ile csv dosyasının dizinine gidip." dir() " komutu ile bulunduğunuz dizindeki dosyaları kontrol edin. "dogalgaz-jodi.csv" dosyasını görmeye çalışın. Dosyayı gördükten sonra dosyayı veri değişkenine yükleyelim.

```
dg=read.csv("dogalgaz-jodi.csv",sep=";")
```

```
> dir()  
[1] "baris6.csv" "baris6.xls" "dg-tahmin.R"  
[4] "dogalgaz-jodi.csv" "dogalgaz-projeeksiyon.xls" "dogalgaz-sonuc.csv"  
[7] "file.csv" "jodi-gas_wdb_64434428690000.xls" "jodi-gas_wdb_64438038950000.xls"  
> dg=read.csv("dogalgaz-jodi.csv",sep=";")  
> str(dg)  
'data.frame': 62 obs. of 15 variables:  
 $ Year : Factor w/ 62 levels "Apr2010","Apr2011",...: 22 16 38 1 43 33 28 6 58 53  
 ...  
 $ Production : int 53 50 59 54 54 57 58 58 50 63 ...  
 $ Receipts.from.Other.Sources : int 0 0 0 0 0 0 0 0 0 0 ...  
 $ Imports : int 3641 3046 3278 2912 2502 2788 2798 3142 2526 3717 ...  
 $ LNG : int 904 744 902 573 583 489 419 667 509 774 ...  
 $ Pipeline : int 2737 2302 2376 2339 1919 2299 2379 2475 2017 2943 ...  
 $ Exports : int 75 69 53 42 37 32 69 75 72 35 ...  
 $ LNG.1 : int 0 0 0 0 0 0 0 0 0 0 ...  
 $ Pipeline.1 : int 75 69 53 42 37 32 69 75 72 35 ...  
 $ Stock.Change : int -366 -259 -157 -6 -47 334 -39 123 -239 359 ...  
 $ Gross.Inland.Deliveries..Calculated. : int 3985 3286 3441 2930 2566 2479 2826 3002 2743 3386 ...  
 $ Statistical.Difference..Calculated. : int 0 0 0 0 0 0 0 0 0 0 ...  
 $ Gross.Inland.Deliveries..Observed. : int 3985 3286 3441 2930 2566 2479 2826 3002 2743 3386 ...  
 $ of.which..Electricity.and.Heat.Generation : int 0 0 0 0 0 0 0 0 0 0 ...  
 $ Closing.Stocks : int 1746 1487 1330 1324 1277 1611 1572 1695 1456 1815 ...
```

Dosyayı dizinde gördükten sonra


```
str(dg)
```

ile dosyadaki yapıya bakalım. Çünkü mesela ondalık ayırıcı “.” Veya “,” dosyamızda farklı kullanılmış ise, veri yapısında “int”leri göremeyiz. Yukarıdaki doğru şekil... Verilerimiz tamsayı olduğundan okunan veriler de “int” yani tamsayı olarak gösterilmiş olması önemlidir.

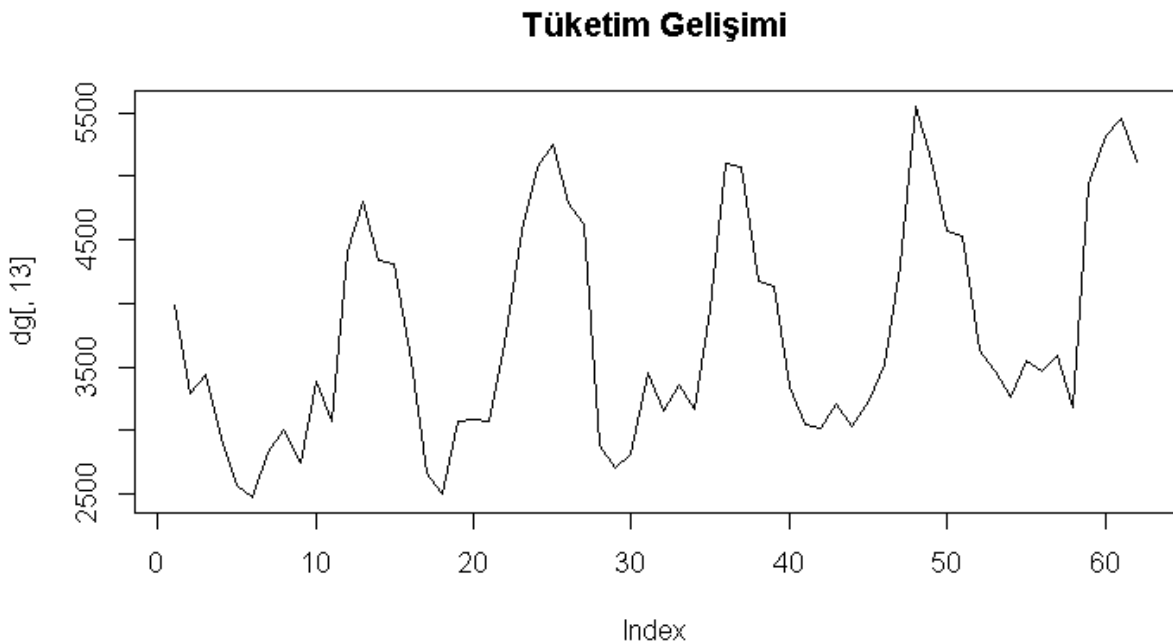
Verimiz bir “data.frame” ve burada 15 sütun veri var (en üstte “62 obs of 15 variables”). 15 sütunda 62 veri satırı var. Sütunlardan biri “Gross. Inland.Deliveries..Observed.”. Bu kavramı iç tüketim olarak düşünelim. Çünkü anlamı : “Brüt iç piyasaya arz edilen” – kısaca “tüketim” denilecek bundan sonra doğalgaz verisi olarak düşünebiliriz. Verileri görmek için

```
head(dg)
```

yazarız. Şimdi de 13. Sırada olan tüketim verisini çizdirip, “Tüketim Gelişimi” başlığı atalım.

```
plot(dg[,13],type="l",main="Tüketim Gelişimi")
```

```
> plot(dg[,13],type="l",main="Tüketim Gelişimi")  
>
```



Yukarıdaki grafiği çizdirdik ama veride zaman serisi yani tarih verileri gözüküyor. Oysa verilerimiz 2010-2015 arası verileri içermekteydi.

9.3. Zaman Serisi

Yukarıdaki grafikte herşey doğru olmasına rağmen, x ekseninden tarihler yer almamaktadır. Bunun için önce zaman serilerini kullanmak için “forecast” kütüphanesi yüklüyoruz. (“forecast” yüklemeyen de “ts” komutu ile yapabiliriz, ilerde göreceğiz) . Eğer daha önceki derslere bakmadıysanız komut satırından `install.packages("forecast")` ile “forecast” paketini yükleyebilirsiniz.

```
library(forecast)
```

```
> library("forecast")
Zorunlu paket yükleniyor: zoo

Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric

Zorunlu paket yükleniyor: timeDate
This is forecast 5.6

Warning messages:
1: package 'forecast' was built under R version 3.1.1
2: package 'zoo' was built under R version 3.1.1
3: package 'timeDate' was built under R version 3.1.1
```

Forecast kütüphanesini işaret edince otomatik olarak zoo kütüphanesini de ekliyor. O da yoksa gene

```
install.packages("zoo")
```

ile yükleyebilirsiniz.

Verimizi seçerken 2010 Ocak'tan başlayarak(2009 Ocak-2009 Aralık arasında Jodidb'de veri yoktu, hücre değerleri 0 idi) , 12 aylık verileri seçmiştik. Ama R'da tarih ve zaman işleri bazen arapsaçı olabildiği için sütundan tarih okumadık. Fakat bunu da okumak mümkündür.

Bunun yerine

```
if (package "timeDate" was built under R version 3.1.1)
> t_dg=msts(dg[,13],start=2010,seasonal.periods=c(6,12*354.37/365,12))
> plot(t_dg,main="Zaman Eklenmiş Tüketim Gelişimi")

t_dg=msts(dg[,13],start=2010,seasonal.periods=c(6,12*354.37/365,12))
plot(t_dg,main="Zaman Eklenmiş Tüketim Gelişimi")
```

şeklindeki komutu kullanabiliriz.

Burada önce t_dg diye bir değişken tanımlıyoruz. Yani doğal gaz (dg)'nin t(time series) hali. Bu t_dg değişkeni ise, daha önceki tüketim verimiz olan dg[,13] yani 13.sütundaki tüm verilerin başlangıcı 2010 olmak üzere zaman serisi şekline dizilmiş halidir. Aynı şekilde

ts.frequency = 12

parametresini ekleyerek de, verinin frekansının 12 olduğunu bildirebiliriz. O zaman komutumuz aşağıdaki gibi olur.

```
t_dg=msts(dg[,13],start=2010,seasonal.periods=c(6,12*354.37/365,12),ts.frequency = 12)
```

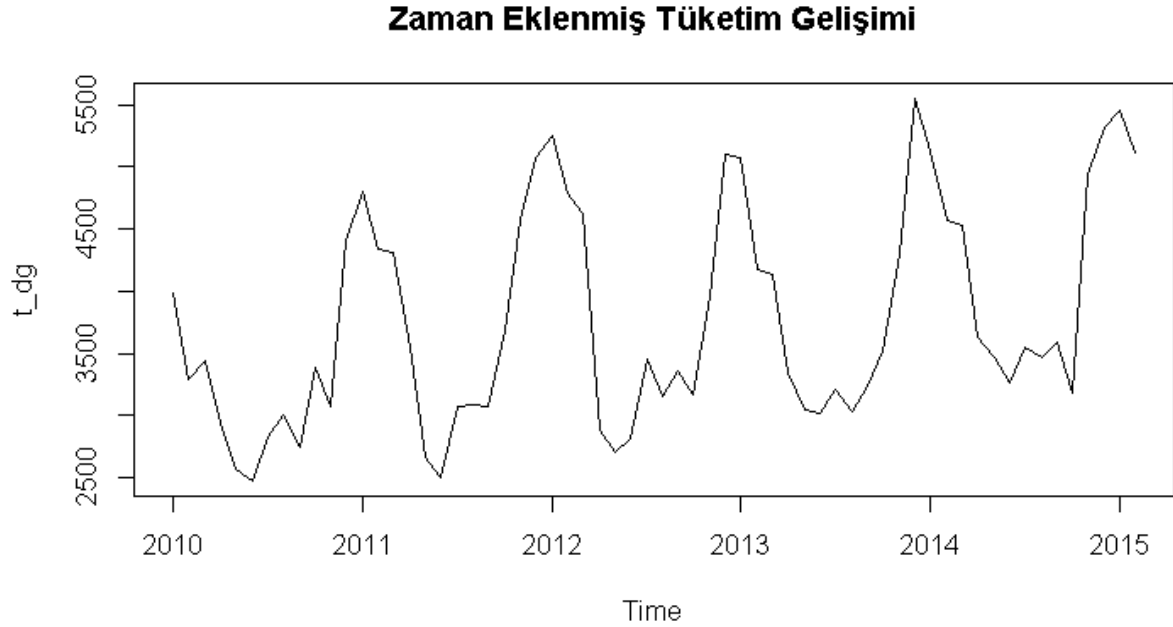
msts komutunda dikkat çeken en önemli noktalardan biri ise daha önce elektrikte yaptığımız gibi, veride tahmin ettiğimiz "mevsimsel periyotları" eklemek olacak. Bu da seasonal.periods parametresine "c(" ile vektör şeklinde üç periyodu ekleyerek elde ediyoruz.

- İlk periyot 6, çünkü doğalgaz talebinde elektrik üretimi santralleri 6şar aylık periyotlara sebep vermektedir. Bu tam 6 olmayabilir, internette optimum dönemsel periyotları bulmak için kodlar mevcut. Elektrik santralleri örneğin Şubat ve Ağustos'ta çok doğalgaz tüketiyor, Nisan ve Ekim de az tüketiyor olabilir.
- İkinci periyot $12 * 354.37 / 365.12$ yani 11.64 gibi bir şey ki, bu da Hyndmann'ın Türkiye elektrik talebindeki hicri takvim (şeker-kurban bayramları) etkilerini analiz etmek için kullanılan periyod.

- Üçüncü ve son periyod ise 12 yani yıllık döngüdür. Zaten grafiğe baktığınızda yıllık bir döngünün olduğu çok açıktır.

Şimdi geriye kalan tek şey, grafiğimizi çizdirmek:

```
plot(t_dg,main="Zaman Eklenmiş Tüketim Gelişimi")
```



9.4. Veriyi Mevsimsel Bileşenlerine Ayırma

Verimizi msts komutu ile alt sezonları ile tanımladık. Ardından daha önce elektrikte olduğu gibi “tbats” komutunu kullanarak parçalarına ve modele ayıracağız.

```
> library("forecast")  
> ?tbats
```

?tbats

Komutu yazdığımızda zaten eğer “forecast” paketini yükledi iseniz, bir yardım ekranı gözükecektir. R Studio’da ise ekranın sağ alttaki kısmına tbats’i açar. Tbat’s’da modelleme için veri serimizi mutlaka msts ile tanımlamalıyız.

TBATS model (Exponential smoothing state space model with Box-Cox transformation, ARMA errors, Trend and Seasonal components)

Description

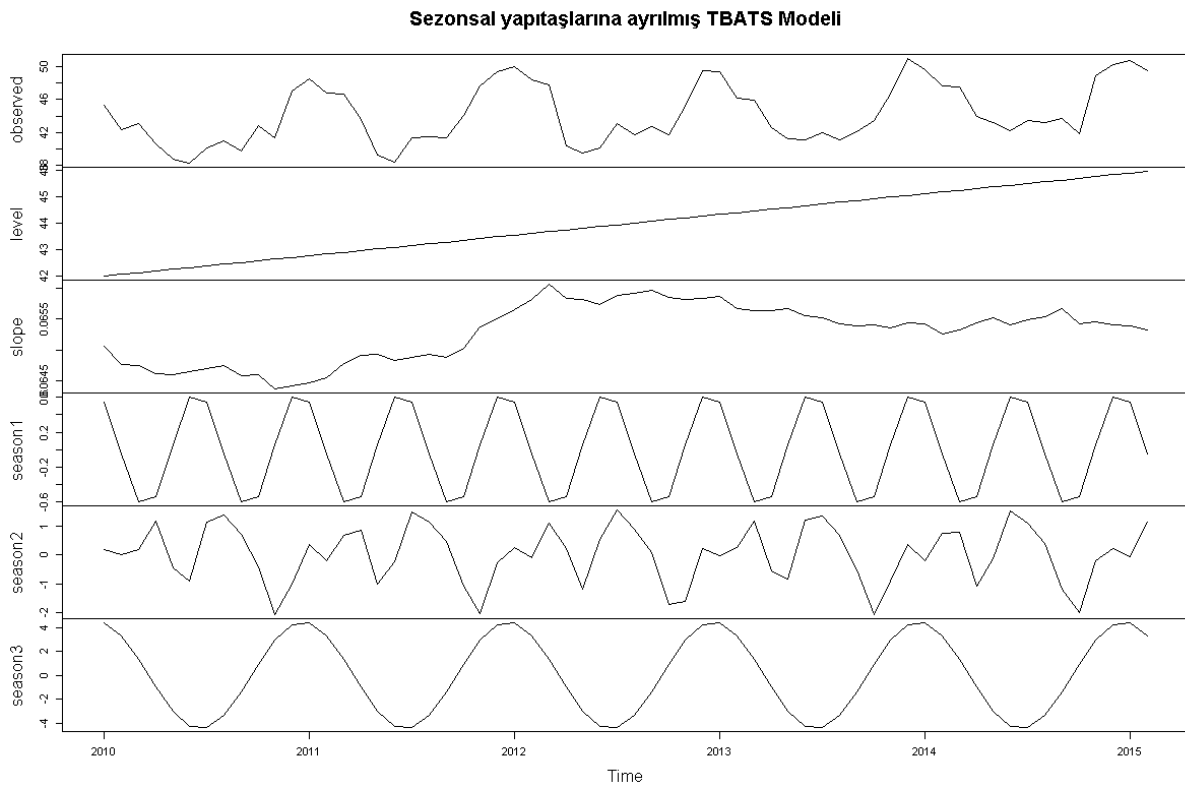
Fits a TBATS model applied to y , as described in De Livera, Hyndman & Snyder (2011). Parallel processing is used by default to speed up the computations.

TBATS aslında bir çok modelleme sistemini bir arada kullanan ve genelde model parametrelerini test ederek otomatik seçen bir algoritmadır. En başarılısı mı? Tabii ki hayır, sayılar sadece başka sayılar ile

ilişki kurular, altında yatan fiziksel nedenleri bilemezler, o yüzden daima iyi uzmanların sayılarının işlenmesinden gelen sonuçları yorumlamasını isteyin.

Dolayısıyla hiçbir parametre girmeden (zaten msts'te bir grup sezon parametresi girmiştik) komutu doğrudan çalıştırıyoruz. Komutun çalışması biraz zaman alıyor (15sn-1.5 dk). Sabırla bekleyin, sonra ekrana imleç gelince de sonucu ekrana çizdirin. Ekrana çizdirirken, grafiğimizin başlığını da "Sezonsal yapıtaşlarına ayrılmış TBATS Modeli" olarak belirlemek için "main" parametresini kullanalım.

```
model_dg<-tbats(t_dg)
plot(model_dg,main="Sezonsal yapıtaşlarına ayrılmış TBATS Modeli")
```



Yukarıdan aşağı baktığımızda, sezonsal bileşenlerine ayrılmış ana tüketim verimizi görebiliriz. Bu verilerin analizi kısaca

- Level: verimizde bir yükseliş trendi var. Yani Türkiye'nin doğalgaz tüketimi artışını sürdürüyor.
- Season1, 6 aylık veriydi, kış ve yaz aylarındaki elektrik üretimindeki doğalgaz'dan kaynaklanan bileşendi
- Season 2, hicri takvimden kaynaklanan bileşendi, burada bazı dik tepeler ve plato tepeler var, bunları da ancak sıcaklık ile ayırabilirsiniz.
- Season 3 ise bildiğimiz senelik döngü ki gördüğümüz gibi kışın en üst tüketimi yapıyor

9.5 Doğal Gaz Projeksiyonu Yapmak

Teknik olarak doğalgaz talep tahmin modelleri yapmak, biraz daha fazla veri ve bakış ister, mesela alt kırılımlar ne, nelerden etkileniyor, ekonomik büyüme ne, sıcaklık ne, rüzgar, nem vs... Fakat modelinizde ne kadar çok parametre olursa o parametreleri de ileri taşımak zorunda kalacağınızdan onları da tahmin etmeniz gerekir. Mesela her parametre de %5 hata yaparsanız, ilk yıl için hata marjınız $1.05 * 1.05$ 'ten %10'lara ulaşır. Eğer 10 parametreniz olsa ve her birinde %5 hatanız olsa sadece ilk yıl

için %60'ın üzerinde hata yapma ihtimaliniz olabilir. O yüzden basit ve anlaşılabilir bir modeliniz olsun, bu modeli T cetveli gibi kullanarak daha karmaşık modellere şablon hazırlayın.

O yüzden basit tutalım, forecast kütüphanesinin çok basit bir komutu var: "forecast".

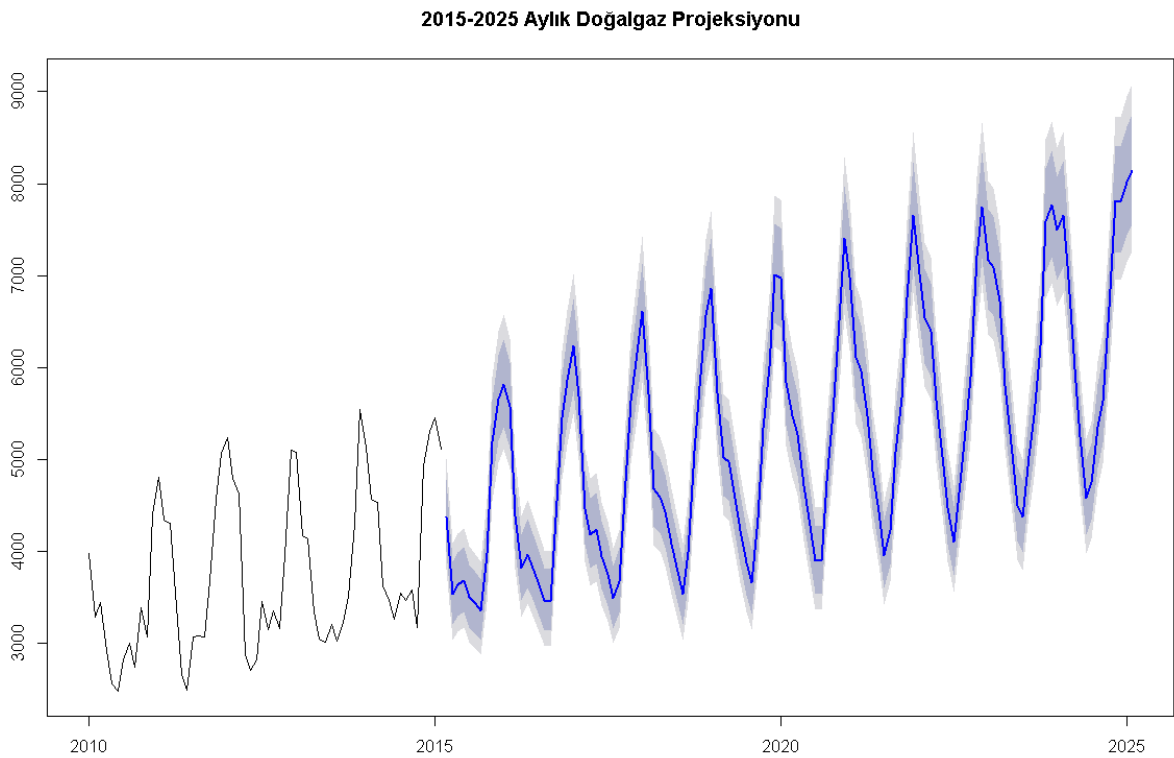
Kısaca

```
forecast(model_verisi, h=ileri_gidilecek_periyod_sayısı*periyoddaki_verisayısı)
```

aslında tüm hesaplamaları yapıyor. Sonuçları sayısal geleceğinden ekrana çizdirmek daha faydalı olacaktır.

```
plot(forecast(model_dg,h=120),main="2015-2025 Aylık Doğalgaz Projeksiyonu")
```

```
> plot(forecast(model_dg,h=120),main="2015-2025 Aylık Doğalgaz Projeksiyonu")  
> |
```



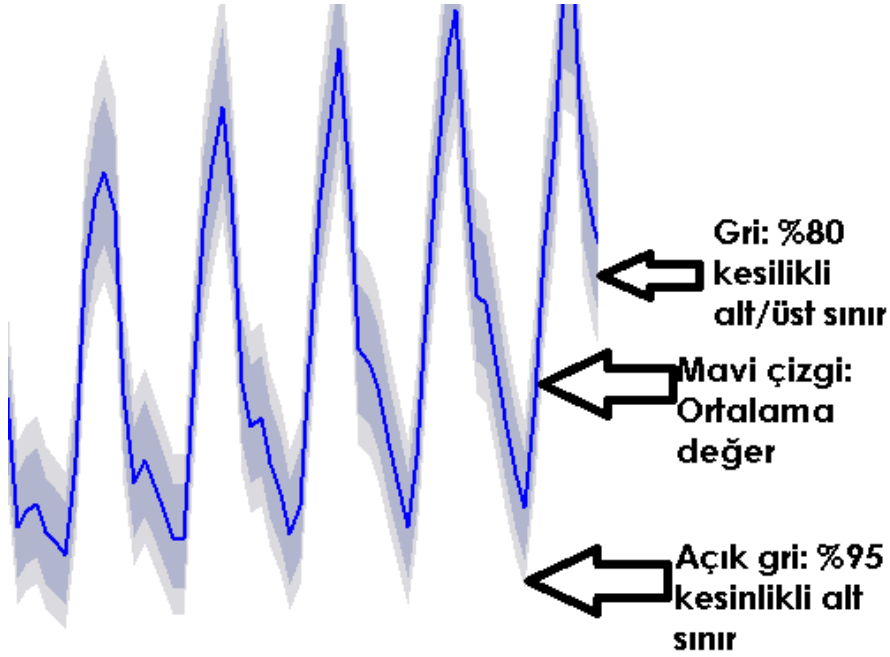
Yukarıda görüldüğü gibi, sadece bir komut ve ileri gidilecek ay sayısını girerek tüm bir projeksiyonu oluşturduk. Burada

- model_dg: tbats komutundan elde ettiğimiz otomatik bir model dosyası (yani sistem en iyi modeli seçti)
- "h=120" kısmı ise 120 ay ileriye gidilmesini belirtti.
- main= ile de başlığı oluşturduk.

Dikkat edilirse geleceğe doğru giden kısım 3 renkli

- mavi renk: ortalama
- gri renk: %80 kesinlikli kısım
- açık gri renk ise %95 kesinlikli sınırı göstermektedir.

Mavi renk ortada olmak üzere alt ve üst kısımlarda gri ve açık gri kısımlar, üst%80'lik veya üst %95'lik kısım ve alt %80'lik veya alt %95'lik kısmı göstermektedir.



9.6 Tüm Projeksiyon Sonuçlarını Görmek

Evet, bir projeksiyon yaptık ve tüm sonuçları ekrana çizdirdik. Peki ya sayısal verileri almak istersek ne yapmamız gerek.

Öncelikle projeksiyonu ekrana yazdırmak yerine bir değişkene almamız gerekiyor.

```
veriler=forecast(model_dg,h=120)
```

komutu bu iş için yeterli olacaktır. Şimdi tüm model kestirim sonuçlarımız bir değişkene atanmış durumdadır. "veriler" değişkeni oldukça veri yoğun bir değişkendir. Bunu yapısından da görebiliriz.

```
str(veriler)
```

```
> veriler=forecast(model_dg,h=120)
> str(veriler)
List of 9
 $ model      :List of 23
  ..$ lambda      : num 0.337
  ..$ alpha       : num 1.4e-08
  ..$ beta        : num 8.65e-05
  ..$ damping.parameter: num 1
  ..$ gamma.one.values : num [1:2] -3.44e-08 2.46e-06
  ..$ gamma.two.values : num [1:2] -9.75e-08 -1.52e-08
  ..$ ar.coefficients : NULL
  ..$ ma.coefficients : NULL
  ..$ likelihood   : num 943
  ..$ optim.return.code: int 0
  ..$ variance     : num 1.22
  ..$ AIC          : num 989
  ..$ parameters   :List of 2
  .. ..$ vect     : num [1:7] 3.37e-01 1.40e-08 8.65e-05 -3.44e-08 2.46e-06 ...
  .. ..$ control:List of 6
  .. .. ..$ use.beta      : logi TRUE
  .. .. ..$ use.box.cox  : logi TRUE
  .. .. ..$ use.damping  : logi FALSE
  .. .. ..$ length.gamma: int 4
  .. .. ..$ p             : num 0
  .. .. ..$ q             : num 0
  ..$ seed.states      : num [1:16, 1] 41.9922 0.0651 -0.708 -0.0433 -0.1486 ...
  ..$ fitted.values    : Time-Series [1:62] from 2010 to 2015: 4263 3906 3608 3222 2449 ...
  .. ..- attr(*, "msts")= num [1:2] 11.7 12
  ..$ errors          : Time-Series [1:62] from 2010 to 2015: -1.11 -2.715 -0.74 -1.415 0.649 ...
```

Veriler’de modelin tüm parametreleri ve sonuçlar, hatalar vs bir çok veri bulunmaktadır. Önce modelimizin hangi aralıklarda hesap yaptığını bakalım. Bunun için:

```
veriler$level
```

```
> veriler$level
[1] 80 95
> |
```

Komutu bize %80 ve %95 alt ve üst sınırlarda hesap yaptığını göstermektedir.

Yani sadece projeksiyon yapmayıp, aynı zamanda alt ve üst kesinlik aralıkları ile kestirim sonuçları bulunmaktadır.

Bu %80 ve %95 sınırlar “veriler” parametresi içinde:

```
$ upper      : num [1:120, 1:2] 4819 4144 3970 3755 3825 ...
$ lower      : num [1:120, 1:2] 4076 3474 3319 3128 3191 ...
+ fitted     : Time-Series [1:62] from 2010 to 2015: 4263 3906 3608 3222 2449 ...
```

upper ve lower dizilerinde bulunmaktadır. Her iki dizi de iki sütundan oluşmaktadır. Bir de ortalama olduğunu düşünürsek, toplamda elimizde 5 senaryo olacaktır. Tüm bunları dosyaya yazdırabiliriz.

9.7 Tüm senaryoları bir araya getirmek

%80 ve %95 hesapları maalesef zaman serisinin orijinal elemanlarına sahip değildir. Bu yüzden işimiz biraz Lego vari bir işe dönmektedir. Verilerimizin bulunduğu dizi değişkenleri aşağıda verilmiştir.

	%95 üst sınır	%80 üst sınır	ortalama	%80 alt sınır	%95 alt sınır
Veri noktası	veriler\$upper[,2]	veriler\$upper[,1]	veriler\$mean	veriler\$lower[,1]	veriler\$lower[,2]

Bu verilerdeki değişkenleri şu şekilde dizmeye çalışacağız. En başta söylediğimiz gibi dg[,13] bizim tüketim tarihsel verileriydi. 2010’dan 2015 2.aya uzanıyordu.

sütunlar	%95 üst sınır	%80 üst sınır	ortalama	%80 alt sınır	%95 alt sınır
----------	---------------	---------------	----------	---------------	---------------

2010 2015	dg[,13]	dg[,13]	dg[,13]	dg[,13]	dg[,13]
2015- 2024	veriler\$upper[,2]	veriler\$upper[,1]	veriler\$mean	veriler\$lower[,1]	veriler\$lower[,2]

Bu yüzden

- rbind: yani satırları alt alta yapıştırma

- cbind : sütunları yan yana yapıştırma

Komutunu kullanacağız,

Önce üst-alt sınır ve ortalamayı yan yana sütun sütun bir araya getireceğiz.

```
a1=cbind(veriler$upper[,2],veriler$upper[,1],veriler$mean,veriler$lower[,1],veriler$lower[,2])
```

bir "a1" verimiz oldu. Şimdi 5 tane de tarihsel tüketimi yan yana yapıştıralım.

```
a2=cbind(dg[,13],dg[,13],dg[,13],dg[,13],dg[,13])
```

şimdi iki veriyi alt alta yapıştıralım. Önce a2 sonra altına a1 gelecek şekilde veri tablomuzu oluşturalım.

```
a=rbind(a2,a1)
```

Verimize bakarsak "head(a)" komutu ile aşağıdaki tabloyu buluruz.

```
> head(a)
      veriler$upper[, 2] veriler$upper[, 1] veriler$mean veriler$lower[, 1] veriler$lower[, 2]
[1,]                3985                3985                3985                3985                3985
[2,]                3286                3286                3286                3286                3286
[3,]                3441                3441                3441                3441                3441
[4,]                2930                2930                2930                2930                2930
[5,]                2566                2566                2566                2566                2566
[6,]                2479                2479                2479                2479                2479
```

Ardından da yeni oluşan veri tablomuzu (2010 Ocak'tan 2024'e kadar olan verimizi) zaman serileri ile zaman damgalarını oluşturalım. Sonra da "sonuc-dg.csv" dosyasına yazalım. Burada unutabilirsiniz diye tekrar library(zoo) komutunu ekledim. Çünkü zaman serisi ile yazmak için en zahmetsiz komut zoo kütüphanesindeki "write.zoo" komutudur.

Burada dikkat edilmesi gereken "a" adıyla tanımladığımız veri tablosunun üzerinde

```
a<-ts(a,start=c(2010,1),frequency=12)
```

ile 2010 Ocak ayından başlayan ve 12'şer aylık frekanslı veriyi tanımlıyoruz. Elde ettiğimiz zaman serisi haline getirilmiş bir a tablosudur.

```
a<-ts(a,start=c(2010,1),frequency=12)
library(zoo)
write.zoo(a,"sonuc-dg.csv",sep=";")
```



```
> a1=cbind(veriler$upper[,2],veriler$upper[,1],veriler$mean,veriler$lower[,1],veriler$lower[,2])
> a2=cbind(dg[,13],dg[,13],dg[,13],dg[,13])
> a=rbind(a2,a1)
> a<-ts(a,start=c(2010,1),frequency=12)
> library(zoo)
> write.zoo(a,"sonuc-dg.csv",sep=";")
> |
```

“sonuc-dg.csv” dosyasını açarsak da aşağıdaki şekilde sonuçları görürüz.

	A	B	C	D	E	F
1	Index	veriler\$upper[, 2]	veriler\$upper[, 1]	veriler\$mean	veriler\$lower[, 1]	veriler\$lower[, 2]
2	2010	3985	3985	3985	3985	3985
3	2010.083333	3286	3286	3286	3286	3286
4	2010.166667	3441	3441	3441	3441	3441
5	2010.25	2930	2930	2930	2930	2930
6	2010.333333	2566	2566	2566	2566	2566
7	2010.416667	2479	2479	2479	2479	2479
8	2010.5	2826	2826	2826	2826	2826
9	2010.583333	3002	3002	3002	3002	3002
10	2010.666667	2743	2743	2743	2743	2743
11	2010.75	3386	3386	3386	3386	3386
12	2010.833333	3071	3071	3071	3071	3071
13	2010.916667	4404	4404	4404	4404	4404
14	2011	4802	4802	4802	4802	4802
15	2011.083333	4342	4342	4342	4342	4342
16	2011.166667	4305	4305	4305	4305	4305

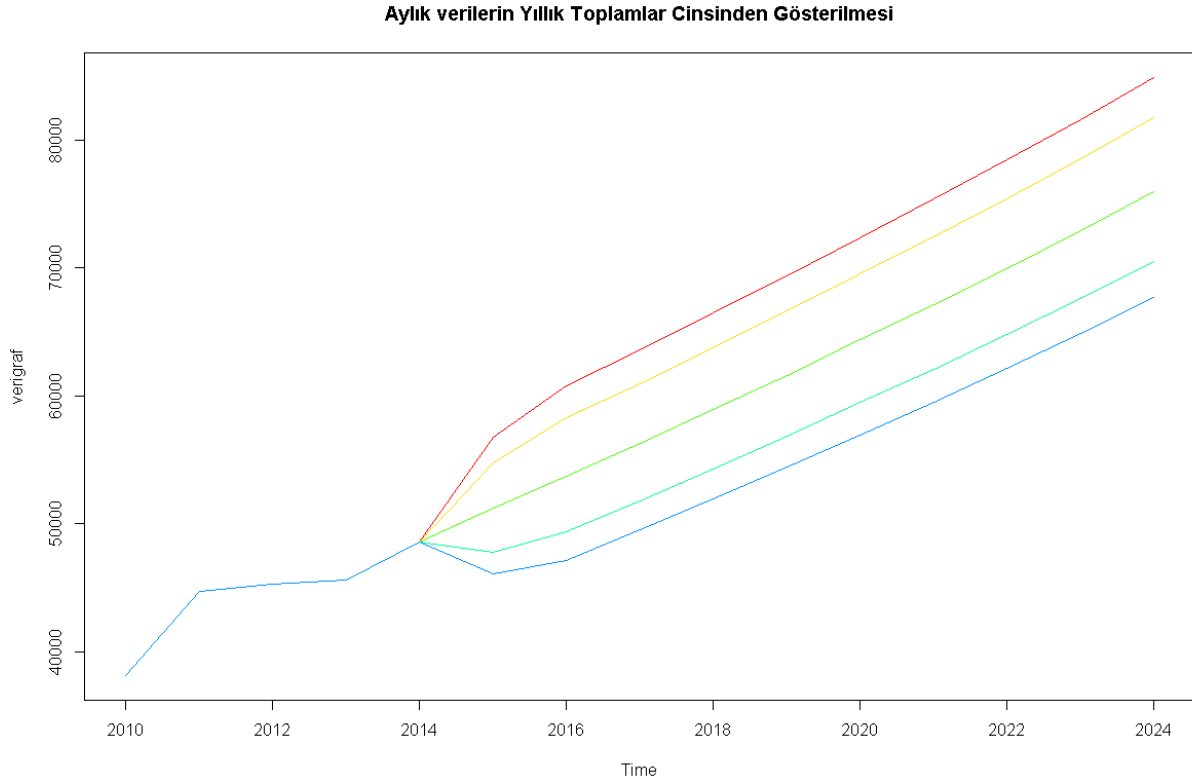
9.7 Aylık Öngörülerini Yıllığa Çevirmek

Aslında R’da herşey kolay gibi gözükmesine rağmen zaman serilerinde işler gerçekten çok karışabiliyor. Bu yüzden, bu kısmı internetteki kaynaklardan tekrar gözden geçirin. Çünkü veri serilerinin zamansal bağlantıları doğru okunamaz ise sonuçlar çok karışabilir.

Bu sebeple işlerimizi kolaylaştıracak “zoo” kütüphanesini kullanıyoruz.. library(zoo) ile kütüphaneyi seçiyoruz.

Önce verilerimizi yıllık toplamalar haline getirmeye çalışıyoruz. Bunun için “aggregate” komutunu kullanıyoruz. Bu komutta iki parametre önemli birincisi “nfrequency=1”, yani verimizde 12 ay da bir dönem atıyordu ya, dönemsel hale getir, ikincisi olan “FUN=sum” ise, her dönem sonunda “sum” yani toplam komutu ile dönemdeki tüm verilerin toplamını elde edelim.

```
> verigraf=aggregate(a,nfrequency = 1,FUN=sum)
> palette(rainbow(7))
> plot(verigraf,plot.type="s",col=as.integer(time(verigraf)),main="Aylık verilerin Yıllık Toplamlar Cinsinden Gösterilmesi")
```



Senaryolarımızın yıllık toplamları da görülmektedir. Bu en üst ve en alttaki çizgiler, doğal gaz tüketiminin %95 ihtimal ile bu iki aradan birinde olacağını söylemektedir.

Her grafiği birbirinden ayrı ekranlarda göstermek için ise

x11()

komutunu kullanabilirsiniz. Böylelikle çizdiğiniz tüm grafikler ayrı ekranlarda yer alır.

Sonuç

Bu bölümde sırası ile:

- JodiDB.org sitesinden Türkiye gaz tüketimi ile ilgili verileri nasıl indireceğimizi,
- Bu verileri nasıl zaman serisi haline getireceğimizi,
- Doğal gaz verilerinde bulunan 3 dönemselliği (6 aylık, 11.64 aylık, 12 aylık) nasıl sisteme ekleyerek, verimizi alt bileşenlere ayıracağımızı,
- Forecast komutu ile elde ettiğimiz otomatik modeli nasıl belirli bir dönem ileri götürebileceğimizi,
- Forecast sonucundan Şupper ve Şlower alt dizileri ile %80 ve %95 kesinlikli alt sınırları alabileceğimizi,
- rbind ve cbind ile veri serilerini birbirine ekleyerek büyük veri tabloları oluşturabildiğimizi,
- Veri setimizi düzene sokarak csv yani Excel'in okuyabileceği hale getirebileceğimizi,
- Veri setimizdeki dönemsellikleri değiştirerek (aylık-> yıllık) ile aylık verilerden yıllık toplamlara nasıl gidebileceğimizi

gördük.

Doğal gaz talep tahminlerinde ise :

- On günlük Doğalgaz ve Elektrik Modeli, <http://www.barissanli.com/calismalar/2013/ODET-v01.pdf>

Gibi yayınlarda daha farklı değişkenler ve daha analitik çalışmalar yapmak mümkündür. Bu çalışma da ise otomatik model seçiminin de aslında fena sonuçlar doğurmadığını gördük.

Veri sonuçlarımız ise bu adrestedir:

<http://www.barissanli.com/calismalar/dersler/r/sonuc-dg.csv>

Komutlar

```
#önce doğalgaz verilerinin olduğu dizine gidelim
#bu dizin siz de masaüstü de olabilir
setwd("C:/dogalgaz-projeksiyon/")
# dir komutu dizindeki dosyaları ekrana yazar
dir()

#doğru dizindeyse csv dosyasını yükleyelim
dg=read.csv("dogalgaz-jodi.csv",sep = ";")

#verinin yapısına bakalım, neden? Eğer verileri doğru okudu ise verileri numeric
#gösterecektir. Mutlaka plot ile ekrana çizin, verileri sayısal olarak okudunuz mu diye
str(dg)

# 13. Sütundaki verinin tüm satırlarının ( dg[satır, sütun]) çizdirip başlık atalım
plot(dg[,13],type="l",main="Tüketim Gelişimi")

library("forecast")
library(zoo)

plot(model_dg,main="Sezonsal yapıtaşlarına ayrılmış TBATS Modeli")
t_dg=msts(dg[,13],start=2010,seasonal.periods=c(12*354.37/365,12))
model_dg<-tbats(t_dg)
#x11()
plot(model_dg)
x11()
plot(forecast(model_dg,h=120),main="2015-2025 Aylık Doğalgaz Projeksiyonu")
veriler=forecast(model_dg,h=120)
a1=cbind(veriler$upper[,2],veriler$upper[,1],veriler$mean,veriler$lower[,1],veriler$lower[,2])
a2=cbind(dg[,13],dg[,13],dg[,13],dg[,13],dg[,13])
a=rbind(a2,a1)
a<-ts(a,start=c(2010,1),frequency=12)
write.zoo(a,"sonuc-dg.csv",sep=";")
verigraf=aggregate(a,nfrequency = 1,FUN=sum)
x11()
plot(verigraf,plot.type="s")
```

```
x11()  
palette(rainbow(7))  
plot(verigraf,plot.type="s",col=as.integer(time(verigraf)),main="Aylık verilerin Yıllık  
Toplamlar Cinsinden Gösterilmesi")
```