

Zaman Serileri Analizi Final

Irem Koyunlu 121517056

25 01 2021

- **ZAMAN SERILERI ANALIZI**
 - *LONDRA HEATHROW YAGISLARI 2010-2019 HER GUN ICIN YAGIS VE ORTALAMA SICAKLIK*
 - *Kaynak :<https://www.kaggle.com/emaforma/ncei-heathrow-2010-2019>*
 - *VERI SETI ACIKLAMASI;*
 - *DEGISKENLER;*
- **1.SORU**
 - A)
 - B)
 - C)
 - D)
- **2.SORU**
 - A)
 - B)
 - C)
 - D)

ZAMAN SERILERI ANALIZI

LONDRA HEATHROW YAGISLARI 2010-2019 HER GUN ICIN YAGIS VE ORTALAMA SICAKLIK

Kaynak :<https://www.kaggle.com/emaforma/ncei-heathrow-2010-2019>



VERİ SETİ AÇIKLAMASI;

Heathrow Havalimanı, İngiltere'nin başkenti Londra'daki altıncı ana havalimanından biridir. Uluslararası yolcu trafiğine göre dünyanın en işlek birinci, toplam yolcu trafiğine göre ise dünyanın en işlek üçüncü havalimanıdır Londra'nın 20 mil batısındadır. Mevcut veri seti, Ocak 2010'den Aralık 2019'ye kadar Londra Heathrow'da her gün için yağış ve ortalama günlük sıcaklıkları içerir. Buradaki fikir, gerçek veriler üzerinde kolay ama etkili görselleştirmelerle deneyler için kullanılacak çok basit bir zaman serisi veri kümesine sahip olmaktır. Veri seti Ulusal Çevresel Bilgi Merkezleri'nden (NCEI) indirilmiştir.

DEĞİSKENLER;

- **DATE:** Veri setimizdeki ilgilenilen tarih.
- **TAVG:** Veri setimizdeki ilgilenilen ortalama sıcaklık.
- **PRCP:** Veri setimizdeki yağış miktarı.

```
library(readr)
library(dplyr)
library(fpp2)
temp <- read_csv("C:/Users/CASPER/Desktop/NCEI Heathrow Meteo Data 2010-2019.csv",
header = T)

temp=temp%>%select(c("DATE", "TAVG"))
head(temp, 10)
```

```
##          DATE  TAVG
## 1  2010-01-01   0.8
## 2  2010-01-02   1.8
## 3  2010-01-03   0.4
## 4  2010-01-04  -2.8
## 5  2010-01-05  -1.3
## 6  2010-01-06  -0.1
## 7  2010-01-07  -2.3
## 8  2010-01-08  -1.4
## 9  2010-01-09  -1.2
## 10 2010-01-10   1.4
```

Kullandığımız verimizin summary kodu ile özetini inceleyerek değişken türlerine bakalım;

```
summary(temp)
```

```
##          DATE          TAVG
## Length:3621      Min.    :-4.10
## Class :character  1st Qu.: 7.40
## Mode  :character  Median :11.60
##                               Mean  :11.64
##                               3rd Qu.:16.00
##                               Max.   :28.60
```

Kullandığımız verimizdeki değişkenlerimizden olan DATE'yi **lubridate** kodu ile Year ve Month olarak ayırıp **group by** kodu ile gruplandıralım. Ayrıca verimize gruplandırdığımız sütunları ekleyelim ve yeni **MonthTemp** verisini oluşturalım;

```
temp$Month<-lubridate::month(temp$DATE)
temp$Year<-lubridate::year(temp$DATE)
temp%>%group_by(Year,Month)%>%summarise(TAVG = mean(TAVG)) ->MonthTemp
head(MonthTemp,10)
```

```
## # A tibble: 10 x 3
## # Groups:   Year [1]
##   Year Month  TAVG
##   <dbl> <dbl> <dbl>
## 1  2010     1  2.18
## 2  2010     2  3.99
## 3  2010     3  7.06
## 4  2010     4 10.4
## 5  2010     5 12.3
## 6  2010     6 17.6
## 7  2010     7 19.4
## 8  2010     8 16.9
## 9  2010     9 14.8
## 10 2010    10 11.4
```

Year ve Month olarak ayirip gruplandirdigimiz yeni MonthTemp verimizin summary kodu ile ozetini inceleyerek degisken turlerine bakalim;

```
summary(MonthTemp)
```

```
##      Year      Month      TAVG
##  Min.   :2010   Min.    : 1.00   Min.    : 1.429
##  1st Qu.:2012   1st Qu.: 3.75   1st Qu.: 7.108
##  Median :2014   Median : 6.50   Median :11.519
##  Mean   :2014   Mean    : 6.50   Mean    :11.594
##  3rd Qu.:2017   3rd Qu.: 9.25   3rd Qu.:16.017
##  Max.   :2019   Max.    :12.00   Max.    :21.803
```

Kullandigimiz verimiz icin MonthTemp olarak olusturdugumuz yeni verimiz aylık bir veri oldugu icin frequency degerimizi 12 alalim ve ts kodu ile aylık verimizi zaman serisine cevirelim;

```
MonthTemp<-ts(MonthTemp[,3],start=c(2010,1),frequency =12)
MonthTemp
```

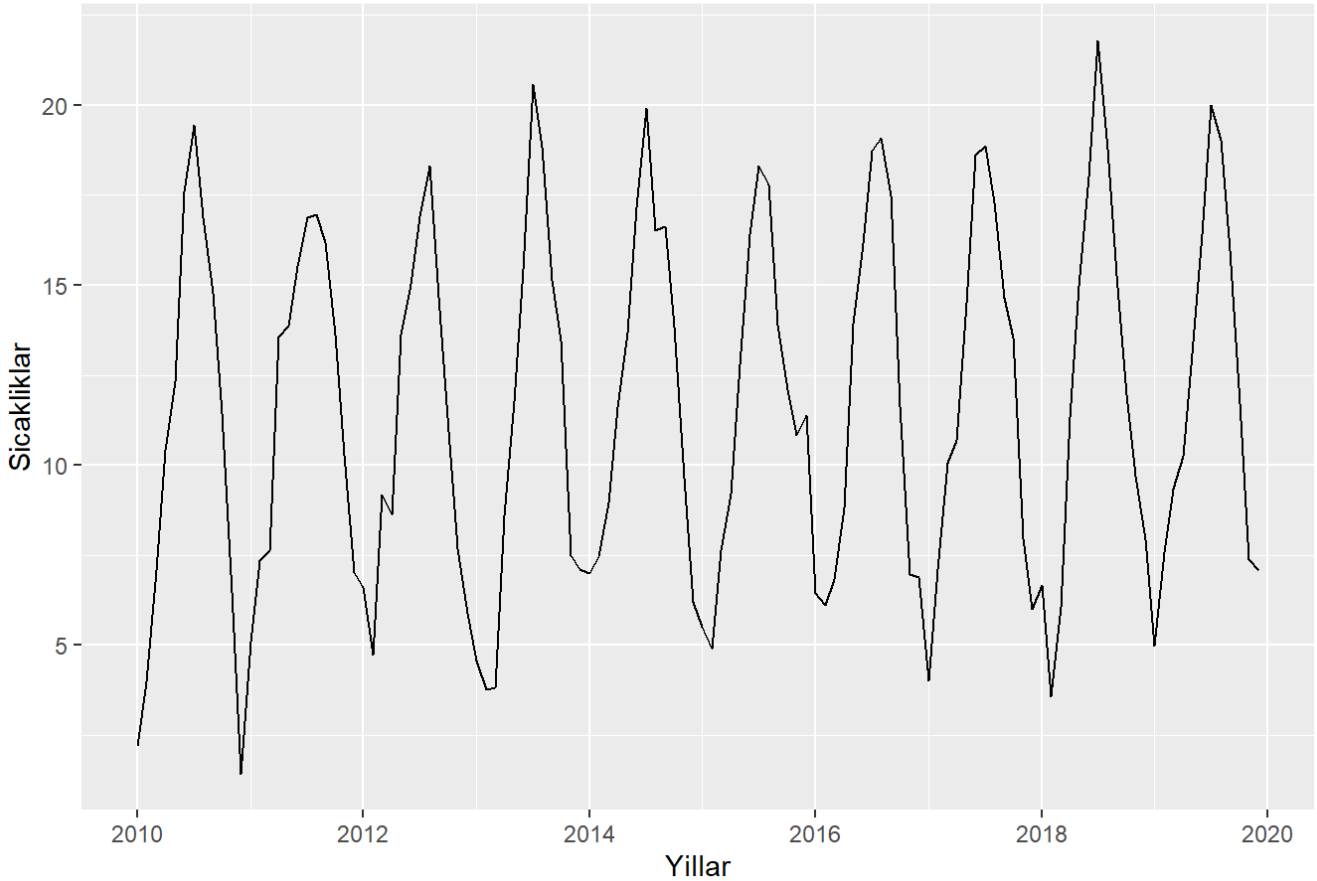
```
##      Jan      Feb      Mar      Apr      May      Jun      Jul
## 2010  2.183871  3.989286  7.064516 10.410000 12.312903 17.586667 19.448387
## 2011  5.083871  7.339286  7.625806 13.576667 13.874194 15.533333 16.864516
## 2012  6.593548  4.734483  9.177419  8.610000 13.619355 15.023333 16.916129
## 2013  4.551613  3.764286  3.819355  8.593333 11.777419 15.436667 20.580645
## 2014  7.006452  7.453571  8.948387 11.600000 13.687097 17.166667 19.912903
## 2015  5.516129  4.896429  7.625806  9.200000 12.793548 16.416667 18.309677
## 2016  6.448387  6.110345  6.819355  8.826667 13.935484 16.150000 18.729032
## 2017  4.003226  7.103571 10.058065 10.696667 14.512903 18.606667 18.870968
## 2018  6.674194  3.557143  6.045161 11.320000 14.964516 18.013333 21.803226
## 2019  4.977419  7.571429  9.345161 10.246667 13.100000 16.396667 19.987097
##      Aug      Sep      Oct      Nov      Dec
## 2010 16.877419 14.770000 11.438710  6.466667  1.429032
## 2011 16.948387 16.186667 13.622581 10.280000  7.019355
```

```
## 2012 18.325806 14.663333 10.929032 7.670000 5.880645
## 2013 18.851613 15.143333 13.445161 7.490000 7.109677
## 2014 16.529032 16.630000 13.829032 9.653333 6.196774
## 2015 17.796774 13.893333 12.112903 10.830000 11.374194
## 2016 19.096774 17.476667 11.619355 6.973333 6.883871
## 2017 17.303226 14.630000 13.538710 7.963333 5.990323
## 2018 18.777419 15.280000 11.974194 9.700000 7.941935
## 2019 19.041935 15.973333 11.722581 7.383333 7.077419
```

Kullandigimiz verimizde olusturdugumuz MonthTemp aylık verimiz için Zaman Serisi Grafigimizi cizdirelim;

```
library(fpp2)
autoplot(MonthTemp) +
ggtitle("Aylık ortalama Hava Sicakliklari") +
xlab("Yillar") +ylab("Sicakliklar")
```

Aylık ortalama Hava Sicakliklari



Kullandigimiz verimizde olusturdugumuz MonthTemp aylık verimiz için Zaman Serisi Grafigimize baktigimizda grafigimizde mevsimsellik olduğu gözükmetedir.Ayrıca grafigimizde az da olsa artan trend olduğu gözükmetedir.

Kullandigimiz verimizde olusturdugumuz MonthTemp aylık verimizi test ve train olarak ikiye ayiralım;

```
MonthTemp.train<-window(MonthTemp,end=c(2017,12))
MonthTemp.test<-window(MonthTemp, start=2018)
```

1.SORU

- **Training setinizi lineer regresyon, lineer spline ve kubik spline regresyon yaklaşımlarını kullanarak modelleyiniz.**

Kullandığımız verimizde oluşturduğumuz MonthTemp aylık verimiz için oluşturduğumuz **MonthTemp.train** setimize Linear Regresyon Modeli , Linear Spline Regresyon Modeli ve Cubic Spline Regresyon Modeli oluşturalım;

- **Linear Regresyon Modeli**

Kullandığımız verimizde **MonthTemp.train** setimizi kullanarak Linear Regresyon Modelini oluşturmak için tek bir lineer değişken kullanarak kurdüğümüz lineer modelimizi tslm kodu ile oluşturarak trend tahmin değerine bakalım. Burada h değerini 10 olarak Linear Regresyon Modelinin 10 adım ilerisi için forecast tahminlerine bakalım;

```
h <- 10
fit.lin <- tslm(MonthTemp.train ~ trend)
fc.lin <- forecast(fit.lin, h = h)
```

Oluşturduğumuz Linear Regresyon Modelimizdeki yani fit.lin'deki trend değerimize bakalım;

```
fit.lin
```

```
##
## Call:
## tslm(formula = MonthTemp.train ~ trend)
##
## Coefficients:
## (Intercept)          trend
##      10.56963          0.01883
```

Oluşturduğumuz Linear Regresyon Modelimizdeki yani fit.lin'deki **trend** değerimize baktığımızda, trend değerimiz 0.01883 çıkmıştır yani pozitif bir trend gözükmektedir.

- **Linear Spline Regresyon Modeli**

Kullandığımız verimizde oluşturduğumuz MonthTemp aylık verimiz için oluşturduğumuz **MonthTemp.train** setimizde üç farklı değişken **t** , **t.break1** ve **t.break2** yi ekleyip verimizi üç parçaya bölerek tslm kodu ile Linear Spline Regresyon Modelimizi oluşturalım.

Eklediğimiz t değişkeni 2014'e kadar olan yıllar , t.break1 ve t.break2 değişkenleri kırılma noktalarıdır. tb1 değişkeni 2014 ve 2017 arasındaki yıllar ve tb2 değişkeni 2017'den sonraki yılları göstermektedir. Burada t.new, tb1.new ve tb2.new değişkenleri t, tb1 ve tb2'ye 10 adımlık tahminleri ekleyerek new data olarak data frame'e dönüştürülmüş halleridir.

```
t <- time(MonthTemp.train)
t.break1 <- 2014
t.break2 <- 2017
tb1 <- ts(pmax(0, t - t.break1), start = 2010)
tb2 <- ts(pmax(0, t - t.break2), start = 2010)

fit.pw <- tslm(MonthTemp.train ~ t + tb1 + tb2)
t.new <- t[length(t)] + seq(h)
tb1.new <- tb1[length(tb1)] + seq(h)
tb2.new <- tb2[length(tb2)] + seq(h)
newdata <- cbind(t=t.new, tb1=tb1.new, tb2=tb2.new) %>%
```

```
as.data.frame()
fc.pw <- forecast(fit.pw, newdata = newdata)
```

Olusturdugumuz Lineer Spline Regresyon Modelimizdeki yani fit.pw'deki trend degerlerimize bakalim;

```
fit.pw
```

```
##
## Call:
## tslm(formula = MonthTemp.train ~ t + tb1 + tb2)
##
## Coefficients:
## (Intercept)          t          tb1          tb2
## -726.9203      0.3668     -0.3933      1.3345
```

Ekledigimiz t degiskeni yani 2014 e kadar olan yillarin **trend** degeri 0.3668 cikmistir, **t** deki bir birimlik artis 0.3668'lik artisa sebep olmaktadır.

tb1 degiskeni yani 2014 ve 2017 arasindaki yillarin **trend** degeri -0.3933 cikmistir, **tb1** deki bir birimlik artis 0.3933'luk azalisa sebep olmaktadır.

tb2 degiskeni yani 2017 den sonraki yillarin **trend** degeri 1.3345 cikmistir, **tb2** deki bir birimlik artis 1.3345'lik artisa sebep olmaktadır.

- **Cubic Spline Regresyon Modeli**

Kullandigimiz verimizde olusturdugumuz MonthTemp aylik verimiz icin olusturdugumuz **MonthTemp.train** setimize Cubic Spline Regresyon Modelini olusturalim;

```
fit.spline <- tslm(MonthTemp.train ~ t + I(t^2) + I(t^3) +
I(tb1^3) + I(tb2^3))
fc.spl <- forecast(fit.spline, newdata = newdata)
```

A)

- Modelleri AICc kriterine gore kiyaslayip daha uygun oldugunu dusundugunuz modeli belirleyiniz.
- Lineer Regresyon Modelimizin AICc Degeri

Kullandigimiz verimizde MonthTemp.train setimizi kullanarak olusturdugumuz Lineer Regresyon Modelimizin AICc degerine bakalim;

```
CV(fit.lin)
```

```
##          CV          AIC          AICc          BIC          AdjR2
## 2.463378e+01 3.093790e+02 3.096399e+02 3.170721e+02 9.025280e-04
```

Kullandigimiz verimizde MonthTemp.train setimizi kullanarak olusturdugumuz Lineer Regresyon Modelimizin **AICc** degeri 3.096399e+02 cikmistir.

- **Lineer Spline Regresyon Modelimizin AICc Degeri**

Kullandigimiz verimizde MonthTemp.train setimizi kullanarak olusturdugumuz Lineer Spline Regresyon Modelimizin AICc degerine bakalim;

```
CV(fit.pw)
```

##	CV	AIC	AICc	BIC	AdjR2
##	25.93054455	313.19543584	313.86210251	326.01717680	-0.01886674

Kullandigimiz verimizde MonthTemp.train setimizi kullanarak olusturdugumuz Lineer Spline Regresyon Modelimizin **AICc** degeri 313.86210251 cikmistir.

- **Cubic Spline Regresyon Modelimizin AICc Degeri**

Kullandigimiz verimizde MonthTemp.train setimizi kullanarak olusturdugumuz Cubic Spline Regresyon Modelimizin AICc degerine bakalim;

```
CV(fit.spline)
```

##	CV	AIC	AICc	BIC	AdjR2
##	25.75904517	313.67913518	314.62295541	329.06522433	-0.01392125

Kullandigimiz verimizde MonthTemp train setimizi kullanarak olusturdugumuz Cubic Spline Regresyon Modelimizin **AICc** degeri 314.62295541 cikmistir.

Kullandigimiz verimizde **MonthTemp.train** setimizi kullanarak olusturdugumuz Lineer Regresyon Modelimizin ,Lineer Spline Regresyon Modelimizin ve Cubic Spline Regresyon Modelimizin AICc degerlerini karsilastirdigimizda AICc degeri en dusuk cikan modelimiz **Lineer Regresyon Modelimizdir**.

B)

- **Uygun buldugunuz modeli kullanarak test setini tahmin ediniz ve tahminlere iliskin RMSE degerini hesaplayiniz.**

Kullandigimiz verimizde MonthTemp.train setimizi kullanarak olusturdugum Lineer Regresyon Modelimizin test setini tahmin edelim;

```
forecast(fit.lin, h = h)-> fc.lin  
fc.lin
```

##	Point	Forecast	Lo 80	Hi 80	Lo 95	Hi 95
##	Jan 2018	12.39629	5.930807	18.86177	2.449645	22.34293
##	Feb 2018	12.41512	5.945517	18.88472	2.462136	22.36810
##	Mar 2018	12.43395	5.960146	18.90776	2.474501	22.39340
##	Apr 2018	12.45278	5.974693	18.93087	2.486742	22.41882
##	May 2018	12.47161	5.989159	18.95407	2.498858	22.44437
##	Jun 2018	12.49044	6.003544	18.97735	2.510849	22.47004
##	Jul 2018	12.50928	6.017849	19.00070	2.522716	22.49584
##	Aug 2018	12.52811	6.032073	19.02414	2.534459	22.52176
##	Sep 2018	12.54694	6.046216	19.04766	2.546078	22.54780
##	Oct 2018	12.56577	6.060279	19.07126	2.557574	22.57397

Kullandigimiz verimizde MonthTemp.train setimizi kullanarak olusturdugum Lineer Regresyon Modelimizin test setinin tahminlerine iliskin RMSE degerlerini hesaplayalim;

```
accuracy(fc.lin,length(MonthTemp.test))
```

```
##           ME           RMSE          MAE          MPE          MAPE          MASE
## Training set -3.330669e-16  4.855437  4.274361 -31.00309  56.47962  1.740154
## Test set      1.160371e+01 11.603713 11.603713  48.34880  48.34880  4.724040
##           ACF1
## Training set 0.7944443
## Test set      NA
```

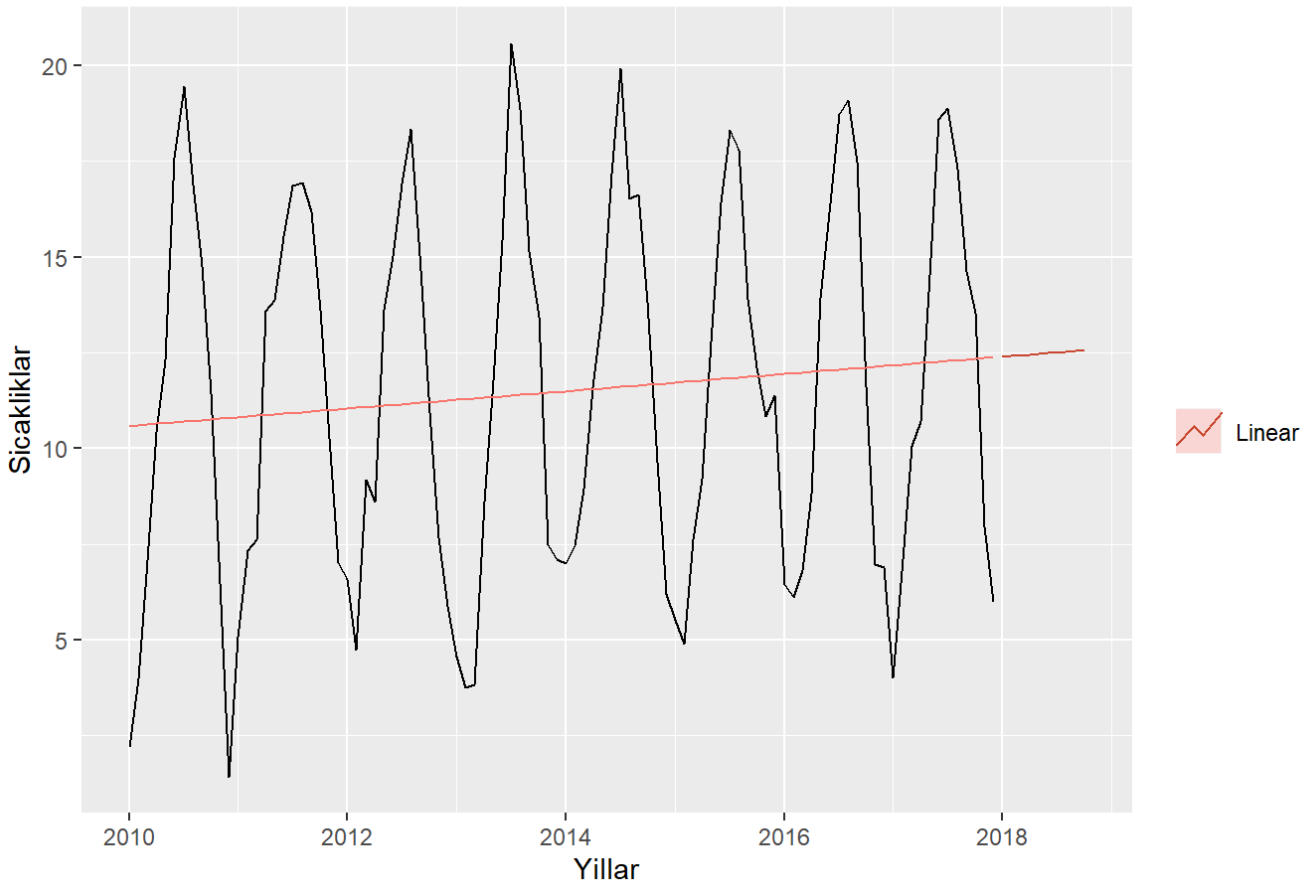
Kullandigimiz verimizde MonthTemp.train setimizi kullanarak olusturdugum Lineer Regresyon Modelimizin test setinin tahminlerine iliskin **RMSE** degeri 11.603713 cikmistir.

C)

- Uygun buldugunuz modelin hem training hem de test seti uzerindeki uyumunu gosteren zaman serisi grafagini cizdiriniz.
- Kullandigimiz verimizde **MonthTemp.train** setimizi kullanarak olusturdugum Lineer Regresyon Modelimizin Zaman Serisi Grafagini cizdirelim;

```
autoplot(MonthTemp.train) +
  autolayer(fitted(fit.lin), series = "Linear") +
  autolayer(fc.lin, series="Linear", PI=FALSE)+
  xlab("Yillar")+ylab("Sicakliklar")+
  ggtitle("Aylik ortalama Hava Sicakliklari")+
  guides(colour = guide_legend(title = " "))
```

Aylik ortalama Hava Sicakliklari



Kullandigimiz verimizde **MonthTemp.train** setimizi kullanarak olusturdugum Lineer Regresyon Modelimizin Zaman Serisi Grafagini cizdirdigimizde lineer trendimizin tahmini lineer modeldeki tek bir

trend degiskeni olan regresyon egrisini vermektedir.Grafikte de goruldugu gibi tahminler cok kaba tahminler yapmaktadir ve grafigimize baktigimizda az da olsa artan trend oldugu gozukmektedir.

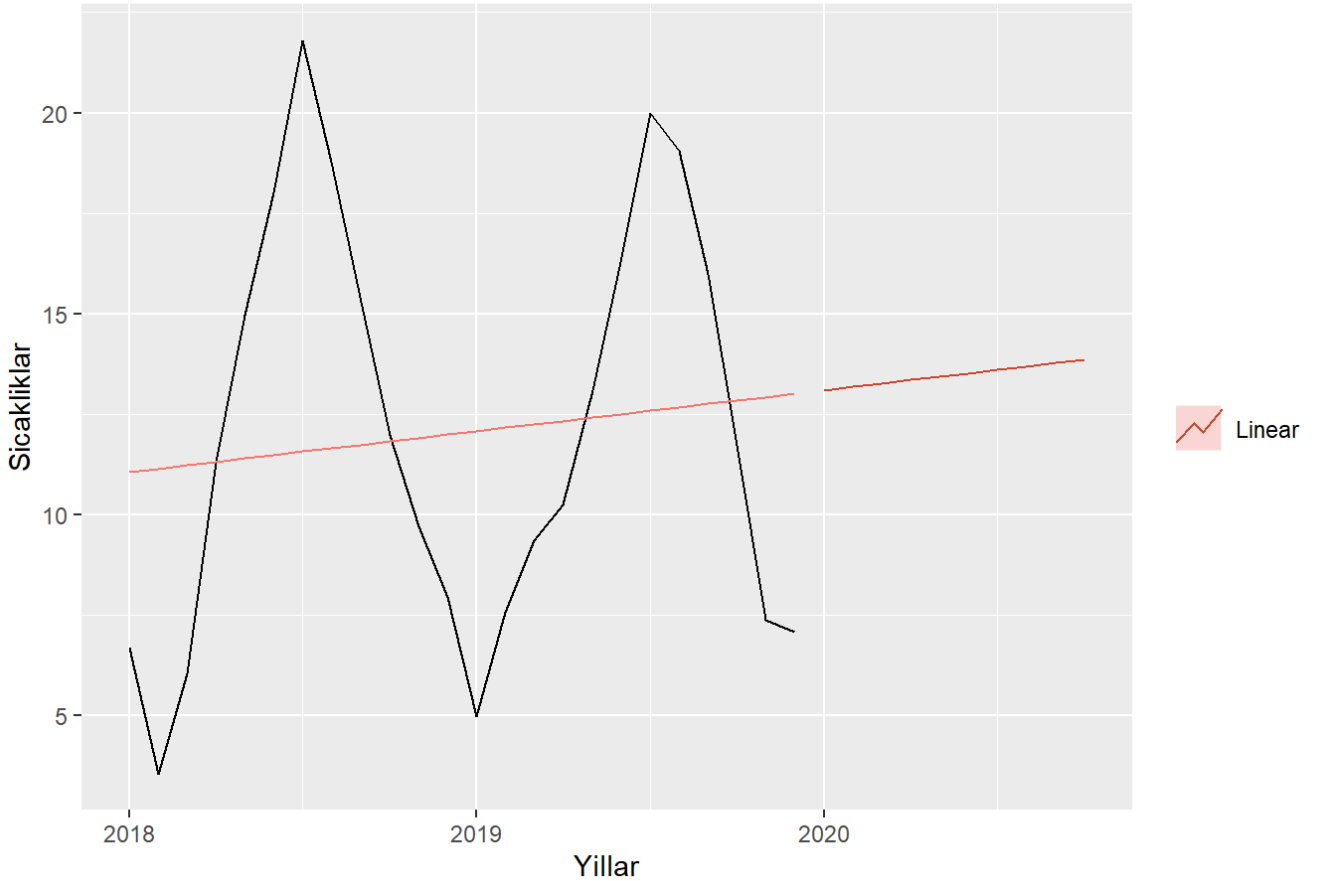
- Oncelikle kullandigimiz verimizde **MonthTemp.test** setimizi kullanarak Lineer Regresyon Modelimizi olusturalim;

```
fit.lintest <- tslm(MonthTemp.test ~ trend)
forecast(fit.lintest, h = h)-> fc.lintest
```

Kullandigimiz verimizde **MonthTemp.test** setimizi kullanarak olusturdugum Lineer Regresyon Modelimizin Zaman Serisi Grafigini cizdirelim;

```
autoplot(MonthTemp.test) +
  autolayer(fitted(fit.lintest), series = "Linear") +
  autolayer(fc.lintest, series="Linear", PI=FALSE)+
  xlab("Yillar")+ylab("Sicakliklar")+
  ggtitle("Aylik ortalama Hava Sicakliklari")+
  guides(colour = guide_legend(title = " "))
```

Aylik ortalama Hava Sicakliklari



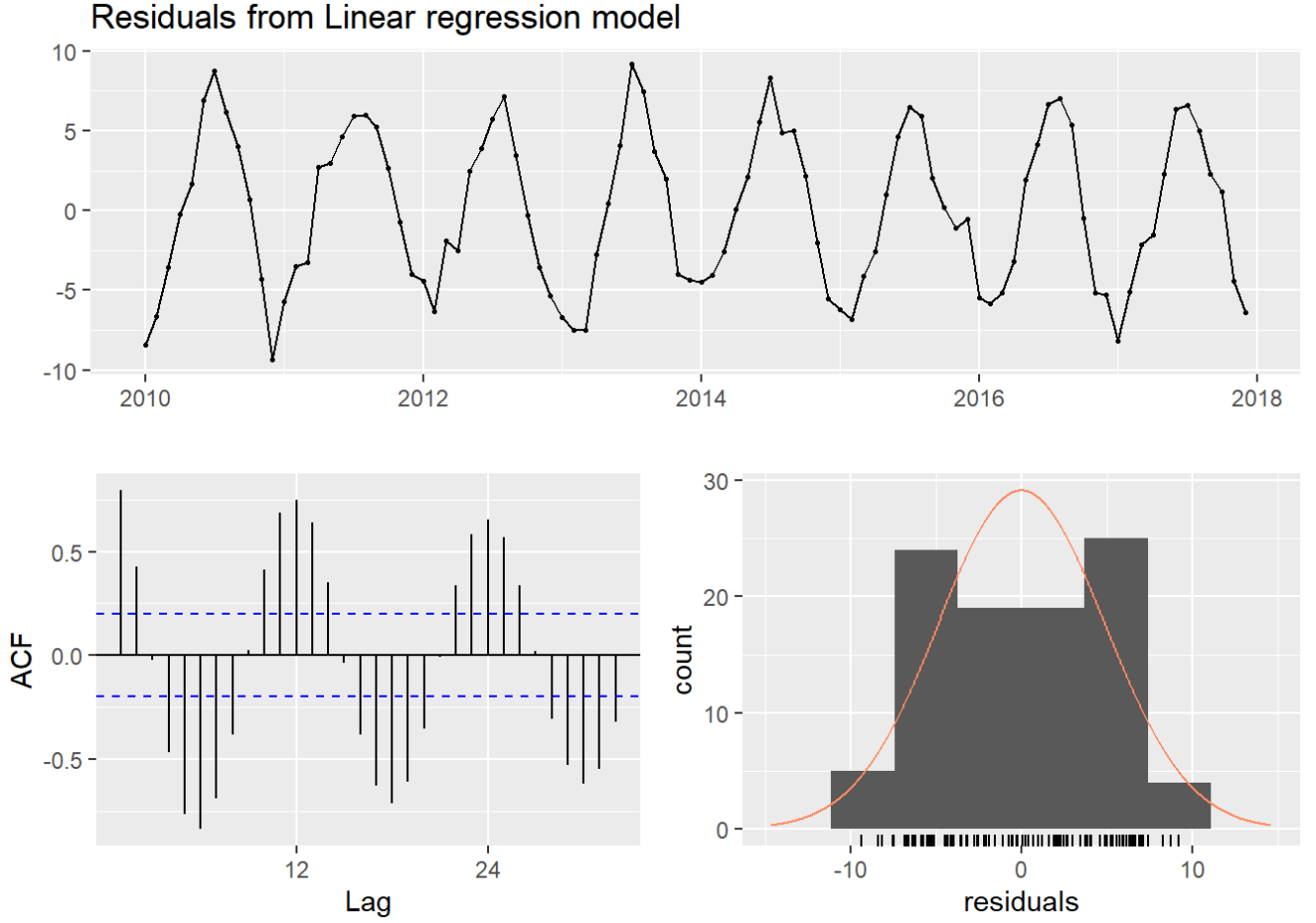
Kullandigimiz verimizde **MonthTemp.test** setimizi kullanarak olusturdugum Lineer Regresyon Modelimizin Zaman Serisi Grafigini cizdirdigimizde lineer trendimizin tahmini lineer modeldeki tek bir trend degiskeni olan regresyon egrisini vermektedir.Grafikte de goruldugu gibi tahminler cok kaba tahminler yapmaktadir ve grafigimize baktigimizda az da olsa artan trend oldugu gozukmektedir.

D)

- Hatalara iliskin varsayimlari kontrol ediniz.

Kullandigimiz verimize en uygun cikan modelimiz **Lineer Regresyon Modelimiz** ciktigindan,Lineer Regresyon Modelimizin artiklari icin checkresiduals komutu ile hatalara iliskin varsayimlari kontrol edelim;

```
checkresiduals(fit.lin)
```



```
##  
## Breusch-Godfrey test for serial correlation of order up to 19  
##  
## data: Residuals from Linear regression model  
## LM test = 84.856, df = 19, p-value = 2.668e-10
```

- Kullandigimiz verimize en uygun cikan modelimiz yani Lineer Regresyon Modelimizin **Breusch-Godfrey** Test sonucumuza baktigimizda p value degerimiz $2.668e-10$ yani yaklasik 0 cikmistir.P value degerimiz 0.05 degerinden kucuk cikti icin **H0** yani **"Artiklar arasinda otokorelasyon problemi yoktur"** hipotezini **red** ederiz. Bu sonuc ile birlikte Lineer Regresyon Modelimizin otokorelasyon problemi oldugu gozukmektedir.
- Sacilim Grafigimize baktigimizda Lineer Regresyon Modelimizin artiklari 0 etrafinda rastgele dagildigi gozukmektedir.
- Otokorelasyon Grafigimize baktigimizda Lineer Regresyon Modelimiz icin laglerden bazilari mavi sinir çizgisini gectigi icin artiklarin otokorelasyon problemi oldugu gozukmektedir.
- Normal Dagilim Grafigimize baktigimizda Lineer Regresyon Modelimiz icin artiklarin normal dagilmadigi gozukmektedir.

2.SORU

A)

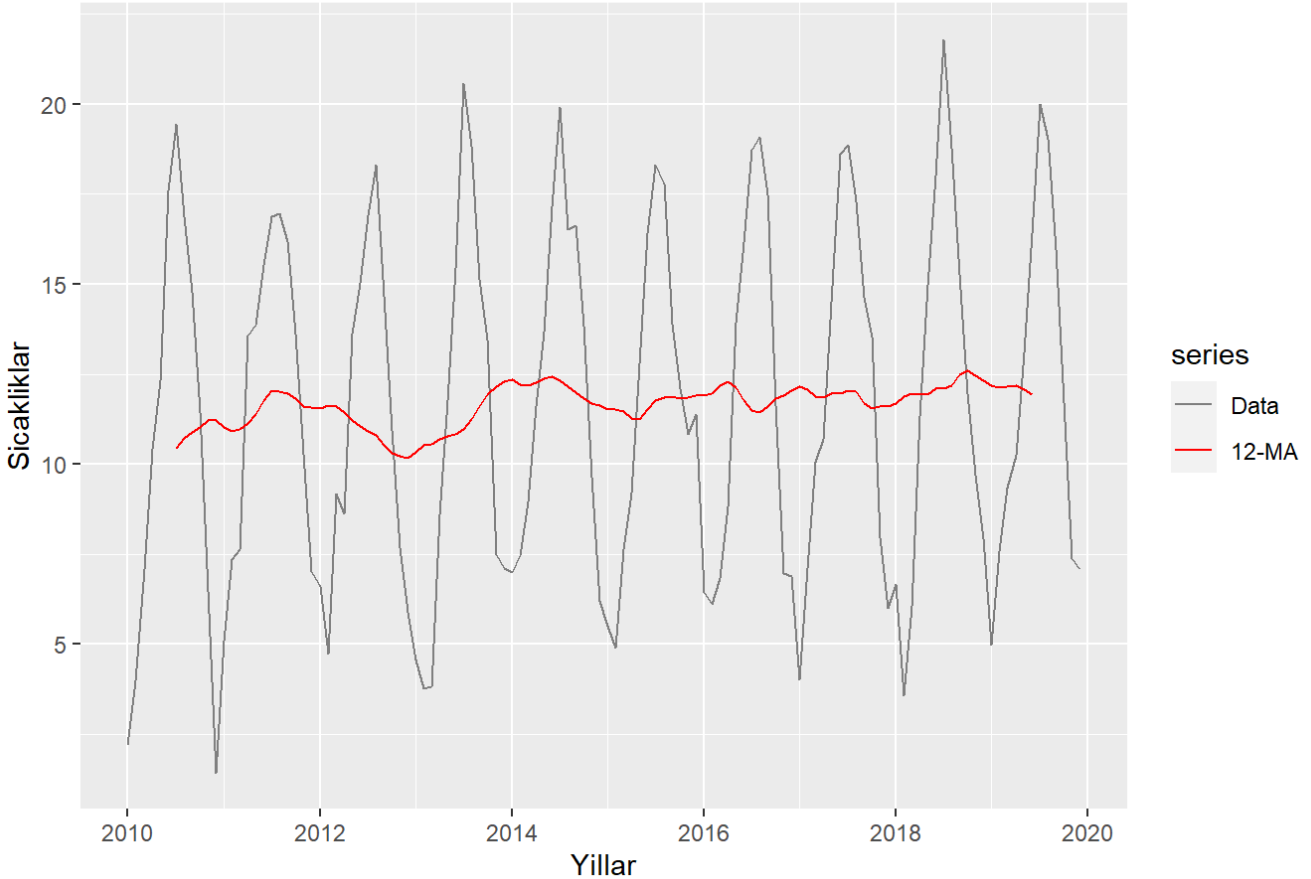
- Uygun hareketli ortalama (moving average) yontemini secip verinizin trend bilesenini belirleyiniz ve orijinal veri uzerinde grafikleyiniz.

Kullandigimiz verimizde olusturdugumuz **MonthTemp** verimiz aylık bir veri oldugundan Moving Average m degerimizi 12 olarak alalım.Burada m degerini 1 aldigimiz zaman bize orijinal seriyi vermektedir. m degerimiz ne kadar yuksekse trend componentimiz o kadar duzlesmektedir ve hareketlilik azalmaktadır.

MonthTemp aylık verimiz için **12-MA** olarak trend dongusunu gormek için orijinal verilerle birlikte Moving Avarage Yonteminin Grafigini cizdirelim;

```
autoplot(MonthTemp, series="Data") +  
  autolayer(ma(MonthTemp,12), series="12-MA") +  
  xlab("Yillar") + ylab("Sicakliklar") +  
  ggtitle("Aylık ortalama Hava Sicakliklari")+  
  scale_colour_manual(values=c("Data"="grey50", "12-MA"="red"),  
    breaks=c("Data", "12-MA"))
```

Aylık ortalama Hava Sicakliklari



Eglim dongusu tahminlerin duzgunlugunu belirler ve ayni zamanda daha buyuk bir m daha yumusak bir egri anlamina gelir.Moving Avarage Yonteminin Grafigini inceledigimizde aylık verilerde eglim dongusunu yani trendi tahmin etmek için 12-MA kullanildigindan trend tahmini genel olarak yapilamamistir.

B)

- Uygun STL ayristirmasi ile verinizi ayristirip grafikleyiniz.
- STL AYRISIMI

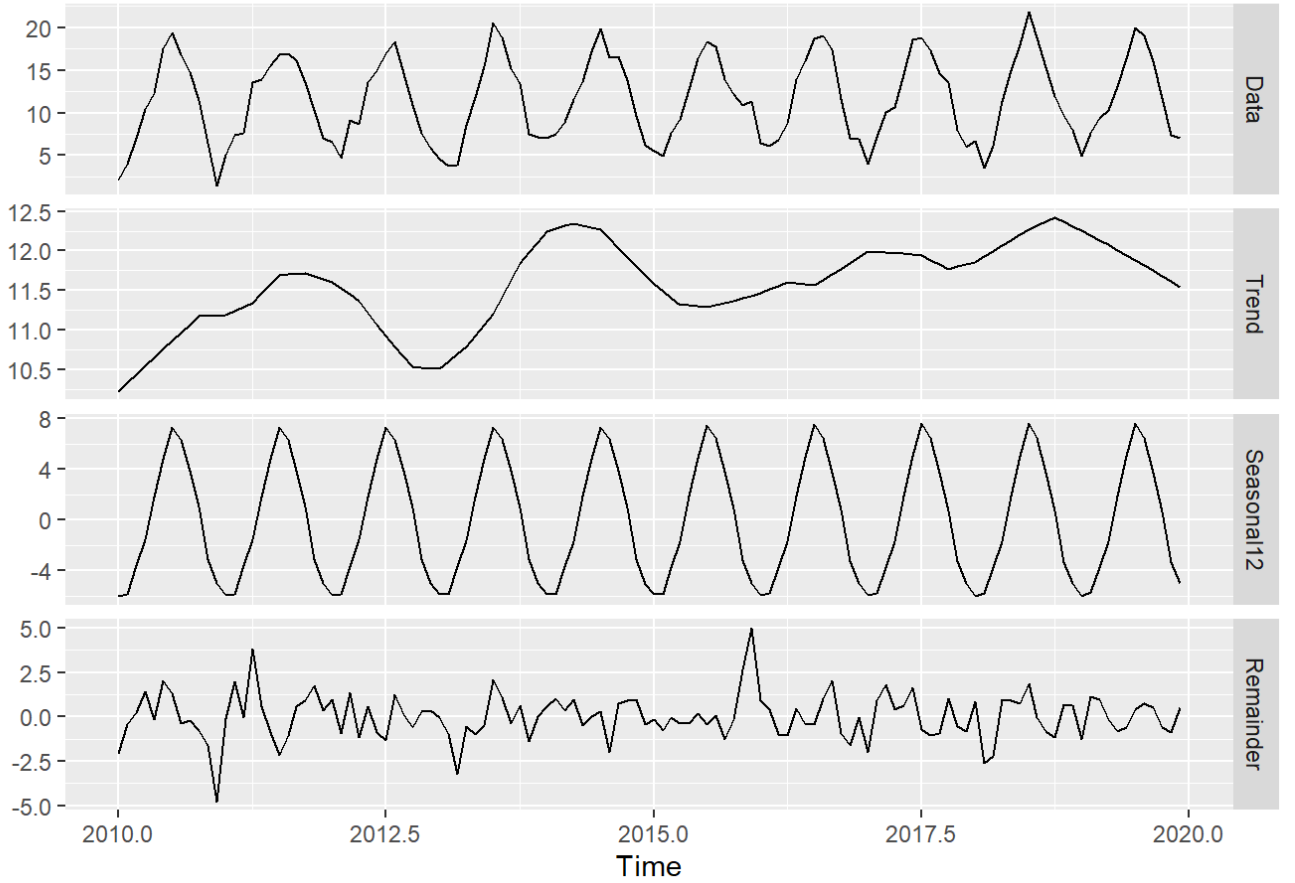
mstl kodu, **s.window =13** kullanarak uygun bir otomatik STL ayristirma saglar ve t.window da otomatik olarak secilir.Bu genellikle mevsime fazla uyum saglamak ve zamanla yavasca degismesine izin vermek arasinda iyi bir denge saglar.

Kullandigimiz verimizde olusturdugumuz **MonthTemp** verimizde STL Ayrisimi yapmak icin mstl kodumuzu kullanarak fit.stl modelimizi olusturalim;

```
MonthTemp %>% mstl(s.window=13, robust=TRUE) -> fit.stl
```

Kullandigimiz verimizde olusturdugumuz **MonthTemp** verimizde STL Ayrisimi yapmak icin **mstl** kodumuzu kullanarak olusturdugumu fit.stl modelimizin grafigini cizdirelim;

```
autoplot(fit.stl)
```



- Kullandigimiz verimiz icin olusturdugumuz MonthTemp verimizde **STL Ayrisimi** yaptigimizda bu ayrisim mevsimsellik ve trendi net bir sekilde gormemizde ve verimizin yapisini incelememizde bize yardimci olur.
- Kullandigimiz verimiz icin olusturdugumuz MonthTemp verimizde cizdirdigimiz **Data Ciktimiz** orijinal serinin Zaman Serisi Grafigini yani autoplotunu vermektedir.Bu grafige baktigimizda mevsimselligin varligi gozukmektedir.
- Kullandigimiz verimiz icin olusturdugumuz MonthTemp verimizde cizdirdigimiz Moving Average ile olusturulan **Trend Ciktimize** baktigimizda az da olsa artan bir trend oldugu gozukmektedir.
- Kullandigimiz verimiz icin olusturdugumuz MonthTemp verimizde cizdirdigimiz **Seasonal12 Ciktimize** baktigimizda mevsimsellik oldugu gozukmektedir.Grafige baktigimizda mevsimselligin artis veya azalis gostermedigini yani mevsimselligin her yil sabit kaldigi gozukmektedir.

- Kullandigimiz verimiz icin olusturdugumuz MonthTemp verimizde cizdirdigimiz **Remainder Ciktimiza** baktigimizda aylık verimizden trend ve mevsimselligin cikarildigi zamanki verimizin grafigi gozukmektedir. Ayrica Remainder grafigi bize verimizin trend ve mevsimsellik tarafından aciklanamayan kismini gostermektedir.

C)

- **Verinizdeki trend ve mevsimsellik bileşenlerinin gücünü belirleyiniz.**

Kullandigimiz verimizde degisken varyanslilik heteroscedasticity problemi var ise yani zaman ile verinin varyasyonunda bir degisiklik oluyor ise carpimsallik soz konusudur,degisken varyanslilik heteroscedasticity problemi yok ise veride toplamsal bir ayrisim kullanilir.Soruda bizden degisken varyansliliği test etmemiz istenmedigi icin trend ve mevsimselligin gücünü iki sekilde de inceleyelim.

Kullandigimiz verimiz icin olusturdugumuz MonthTemp verimizdeki trend ve mevsimsellik bileşenlerinin gücünü belirlememiz icin;

Mevsimsel bileşenini belirlemek icin **seasonal** kodunu kullanalim;

```
seasonal (fit.stl) -> s
```

Trend bileşenini belirlemek icin **trendcycle** kodunu kullanalim;

```
trendcycle (fit.stl) -> t
```

Ramainder bileşenini belirlemek icin **remainder** kodunu kullanalim;

```
remainder (fit.stl) -> r
```

Kullandigimiz verimiz icin olusturdugumuz MonthTemp verimizdeki trend ve mevsimsellik bileşenlerinin degerlerine bakalim;

- **Toplamsal Trend Ve Toplamsal Mevsimsellik**

```
ToplamsalTrend<-max(0, (1-(var(r)/var(t+r))))  
ToplamsalTrend
```

```
## [1] 0.1878766
```

```
ToplamsalMevsimsellik<-max(0, (1-(var(r)/var(s+r))))  
ToplamsalMevsimsellik
```

```
## [1] 0.9305752
```

- Olusturdugumuz MonthTemp verimizde guclu bir toplamsal trend var ise toplamsal trend degerimizin sonucu 1'e daha yakin cikmalidir. Verimizdeki toplamsal trend degerimiz 0.1878766 olarak cikmistir. Bu deger 1'e yakin olmadigindan verimizde dusuk bir toplamsal trend vardir.
- Olusturdugumuz MonthTemp verimizde guclu bir toplamsal mevsimsellik var ise toplamsal mevsimsellik degerimizin sonucu 1'e daha yakin cikmalidir. Verimizdeki toplamsal mevsimsellik degerimiz 0.9305752 olarak cikmistir. Bu deger 1'e yakin oldugundan verimizde guclu bir toplamsal mevsimsellik vardir.

- **Carpimsal Trend Ve Carpimsal Mevsimsellik**

```
CarpimsalTrend<-max(0, (1-(var(r)/var(t*r))))  
CarpimsalTrend
```

```
## [1] 0.9924062
```

```
CarpimsalMevsimsellik<-max(0, (1-(var(r)/var(s*r))))  
CarpimsalMevsimsellik
```

```
## [1] 0.9562956
```

- Olusturdugumuz MonthTemp verimizde guclu bir carpimsal trend var ise carpimsal trend degerimizin sonucu 1'e daha yakin cikmalidir. Verimizdeki carpimsal trend degerimiz 0.9924062 olarak cikmistir. Bu deger 1'e yakin oldugundan verimizde guclu bir carpimsal trend vardır.
- Olusturdugumuz MonthTemp verimizde guclu bir carpimsal mevsimsellik var ise carpimsal mevsimsellik degerimizin sonucu 1'e daha yakin cikmalidir. Verimizdeki carpimsal mevsimsellik degerimiz 0.9562956 olarak cikmistir. Bu deger 1'e yakin oldugundan verimizde guclu bir carpimsal mevsimsellik vardır.

D)

- **Training set üzerinde STL ayristirmasini kullanarak, test setinizi tahmin ediniz. Orjinal veri üzerine tahminlerinizi grafikleyip tahminlere iliskin RMSE degerini hesaplayiniz.**

Kullandigimiz verimizde **MonthTemp.train** setimiz üzerinde STL Ayrisimi yapmak için mstl kodumuzu kullanarak train.fit.stl modelimizi olusturalim;

```
MonthTemp.train %>% mstl(s.window=13, robust=TRUE) -> train.fit.stl
```

Kullandigimiz verimizde MonthTemp.train setimizi kullanarak olusturdugumuz STL Ayrisimi Modelimizin test setini tahmin edelim;

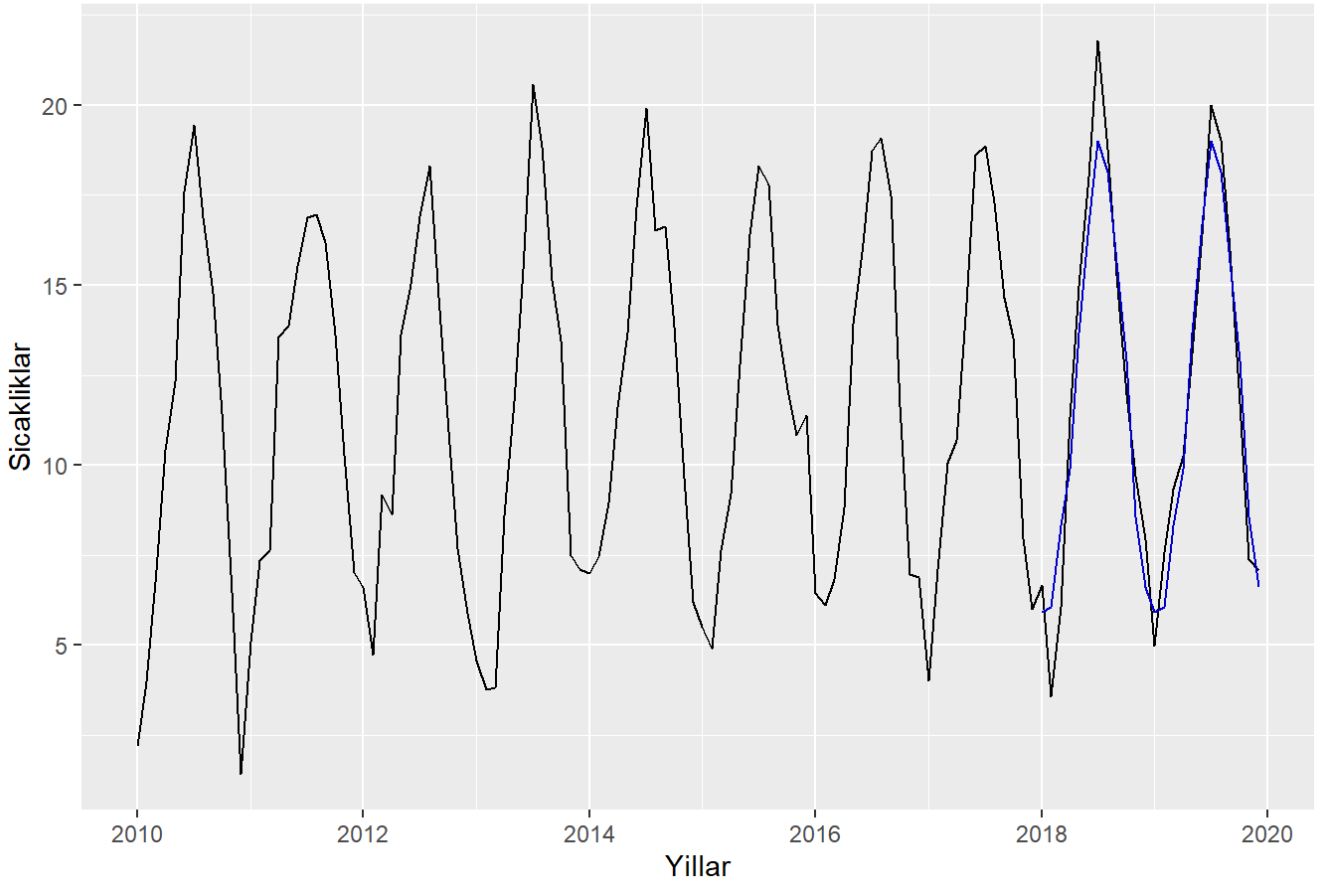
```
fc.train.fit.stl<-forecast(train.fit.stl)  
fc.train.fit.stl
```

##	Point	Forecast	Lo 80	Hi 80	Lo 95	Hi 95
##	Jan 2018	5.905677	3.993992	7.817363	2.982006	8.829349
##	Feb 2018	6.047658	4.133356	7.961961	3.119985	8.975331
##	Mar 2018	8.331163	6.414247	10.248078	5.399493	11.262832
##	Apr 2018	9.884954	7.965429	11.804479	6.949294	12.820615
##	May 2018	13.665107	11.742976	15.587238	10.725461	16.604752
##	Jun 2018	16.690435	14.765702	18.615169	13.746810	19.634061
##	Jul 2018	19.001936	17.074604	20.929269	16.054336	21.949537
##	Aug 2018	18.158705	16.228777	20.088632	15.207135	21.110274
##	Sep 2018	15.636505	13.703985	17.569024	12.680971	18.592038
##	Oct 2018	12.908816	10.973708	14.843924	9.949323	15.868308
##	Nov 2018	8.556192	6.618499	10.493885	5.592746	11.519638
##	Dec 2018	6.613544	4.673269	8.553818	3.646150	9.580938
##	Jan 2019	5.905677	3.962825	7.848530	2.934341	8.877014
##	Feb 2019	6.047658	4.102231	7.993086	3.072384	9.022933

```
## Mar 2019      8.331163  6.383164 10.279161  5.351956 11.310369
## Apr 2019      9.884954  7.934388 11.835521  6.901820 12.868088
## May 2019     13.665107 11.711976 15.618238 10.678051 16.652163
## Jun 2019     16.690435 14.734743 18.646127 13.699462 19.681408
## Jul 2019     19.001936 17.043686 20.960186 16.007051 21.996821
## Aug 2019     18.158705 16.197900 20.119509 15.159913 21.157496
## Sep 2019     15.636505 13.673149 17.599860 12.633811 18.639198
## Oct 2019     12.908816 10.942912 14.874719  9.902226 15.915405
## Nov 2019      8.556192  6.587744 10.524640  5.545710 11.566673
## Dec 2019      6.613544  4.642555  8.584533  3.599176  9.627912
```

Kullandigimiz aylık MonthTemp verimiz yani orijinal verimiz üzerinde tahminlerinizi grafikleyelim;

```
autoplot(MonthTemp) +
  autolayer(fc.train.fit.stl,PI=FALSE)+xlab("Yillar")+ylab("Sicakliklar")+guides(colour=guide_legend(title="Forecast"))
```



Kullandigimiz aylık MonthTemp verimiz yani orijinal verimiz üzerinde tahminlerinizin grafigini inceledigimizde; tahminlerimiz yani grafikte gozuken mavi çizgiler orijinal verimizin yani siyah çizgilerin tam ustunde oldugu için orijinal verimizi iyi tahmin etmistir.

Kullandigimiz verimizde MonthTemp.train setimizi kullanarak olusturdugumuz STL Ayrisimi Modelimizin tahminlerine ilişkin RMSE degerlerini hesaplayalım;

```
accuracy(fc.train.fit.stl,length(MonthTemp.test))
```

```
##           ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.151314 1.476076 1.137732 -4.587959 15.98990 0.4631871
```

```
## Test set      18.094323 18.094323 18.094323 75.393011 75.39301 7.3664619
##              ACF1
## Training set 0.3098812
## Test set      NA
```

Kullandigimiz verimizde MonthTemp.train setimizi kullanarak olusturdugumuz STL Ayrisimi Modelimizin tahminlerine iliskin **RMSE** degerimiz 18.094323 cikmistir.