به نام خدا

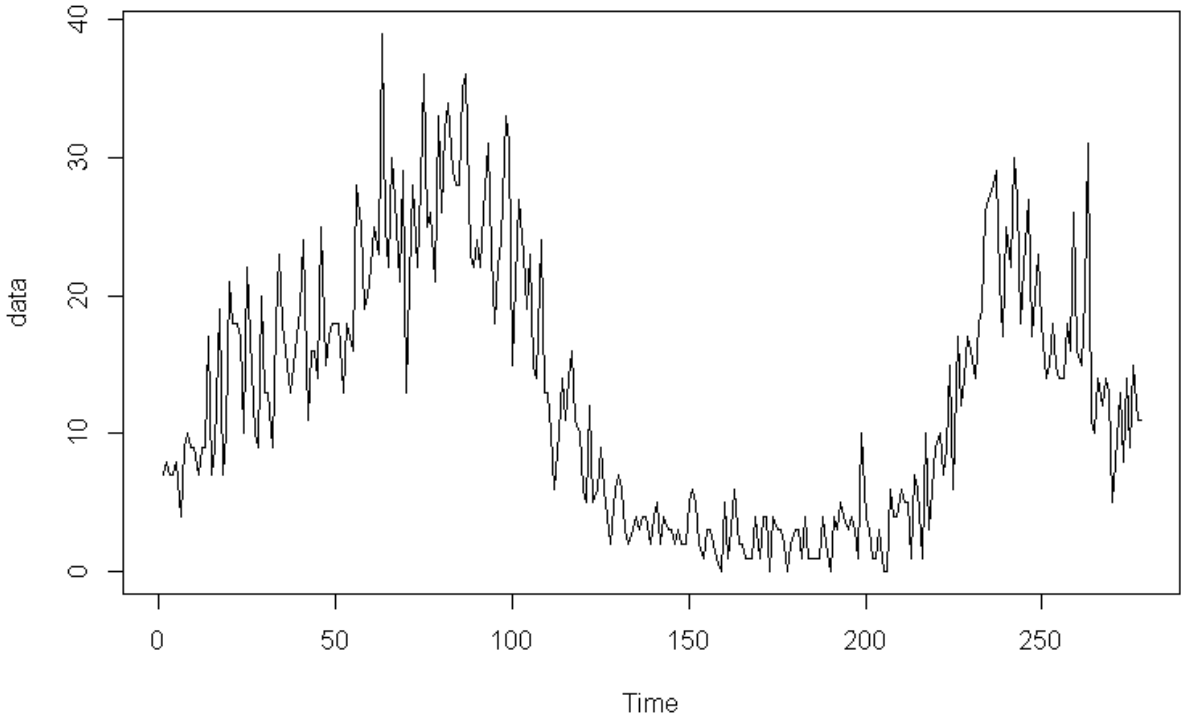
گزارش آمار فوتی در استان اصفهان با نرم افزار R

پرهام حسن

در مرحله اول داده را به نرم افزار پیوست میکنیم و آن را تمییز میکنیم.

```
> ####read data####

> dead = read.csv(file.choose())

> dead[۱۵۳:٤۳۲,٤]

 [۱]  ۷ ۸ ۷ ۷ ۸ ٤ ۹ ۱۰ ۹ ۹ ۷ ۹ ۹ ۱۷ ۷ ۹ ۱۹ ۷ ۱۰ ۲۱ ۱۸ ۱۸ NA ۱۷ ۱۰ ۲۲ ۱۷ ۱۰ ۹ ۲۰

 [۳۱] ۱۳ ۱۳ ۹ ۱۸ ۲۳ ۱۸ ۱٦ ۱۳ ۱۵ ۱۷ ۱۹ ۲٤ ۱۱ ۱٦ ۱٦ ۱٤ ۲۵ ۱۵ ۱۷ ۱۸ ۱۸ ۱۸ ۱۳ ۱۸ ۱۷ ۱٦ ۲۸ ۲۵ ۱۹
۲۰

 [٦۱] ۲۲ ۲۵ ۲۳ ۳۹ ۲۵ ۲۲ ۳۰ ۲۷ NA ۲۱ ۲۹ ۱۳ ۲۱ ۲۸ ۲۲ ۲۸ ۳٦ ۲۵ ۲٦ ۲۱ ۳۳ ۲٦ ۳۲ ۳٤ ۲۹ ۲۸ ۲۸ ۳۵
۳٦ ۲۳

 [۹۱] ۲۲ ۲٤ ۲۲ ۲٦ ۳۱ ۲۳ ۱۸ ۲۲ ۲٤ ۳۳ ۳۱ ۱۵ ۲۱ ۲۷ ۲۳ ۱۹ ۲۳ ۱۵ ۱٤ ۲٤ ۱۳ ۱۳ ۱۰ ٦ ۱۰ ۱٤ ۱۱ ۱٤ ۱٦
۱۱

[۱۲۱] ۱۰ ٦ ۵ ۱۲ ۵ ٦ ۹ ٦ ٤ ۲ ٦ ۷ ٦ ۳ ۲ ۳ ٤ ۳ ٤ ٤ ۲ ٤ ۵ ۲ ٤ ۳ ۳ ۲ ۳ ۲

[۱۵۱] ۲ ۵ ٦ ۵ ۲ ۱ ۳ ۳ ۲ ۱ ۰ ۵ ۱ ۳ ٦ ۲ ۲ ۱ ۱ ۱ ٤ ۱ ٤ ٤ ۰ ٤ ۳ ۳ ۲ ۰

[۱۸۱] ۲ ۳ ۳ ۱ ٤ ۱ ۱ ۱ ۱ ٤ ۲ ۰ ٤ ۳ ۵ ٤ ۳ ٤ ۳ ۱ ۱۰ ٤ ۳ ۱ ۱ ۳ ۰ ۰ ٦ ٤

[۲۱۱] ٤ ٦ ۵ ۵ ۱ ۷ ٦ ۱ ۱۰ ۳ ٦ ۹ ۱۰ ۷ ۹ ۱۵ ٦ ۱۷ ۱۲ ۱٤ ۱۷ ۱٦ ۱٤ ۱۸ ۱۹ ۲٦ ۲۷ ۲۸ ۲۹ ۲۱

[۲٤۱] ۱۷ ۲۵ ۲۲ ۳۰ ۲۷ ۱۸ ۲۲ ۲۷ ۱۷ ۲۰ ۲۳ ۱۸ ۱٤ ۱۵ ۱۸ ۱۵ ۱٤ ۱٤ ۱۸ ۱٦ ۲٦ ۱٦ ۱۵ ۲۰ ۳۱ ۱۱ ۱۰ ۱٤
۱۲ ۱٤

[۲۷۱] ۱۳ ۵ ۸ ۱۳ ۸ ۱٤ ۹ ۱۵ ۱۱ ۱۱

> #dead[۱۵۳:٤۳۲,۲]

> d = data.frame(dead[۱۵۳:٤۳۲,۲],dead[۱۵۳:٤۳۲,٤])

> View(d)

> data = d[,۲]

> data = data[-c(۲۳,٦۹)]
```

در مرحله بعدی نمودار سری زمانی داده را رسم میکنیم. با استفاده از ts.plot(data)



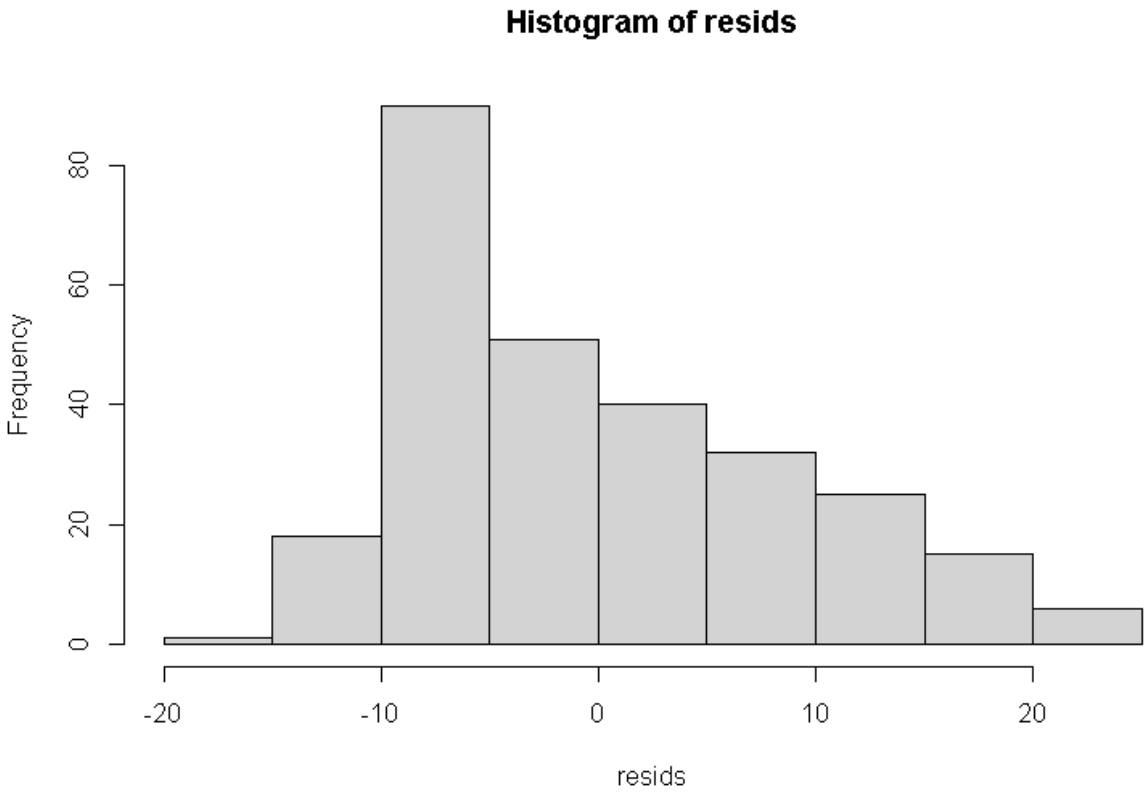
برای برسی اینکه داده ها رگرسیون هستند یا خیر مدل رگرسیونی زیر را به آن برازش می‌دهیم.

```
> ####model####
> t = ١:length(data)
> t٢ = t ^ ٢
> model = lm(data ~ t + t٢)
> summary(model)
Call:
lm(formula = data ~ t + t٢)
Residuals:
   Min     ١Q  Median     ٣Q    Max
-١٦.٥٣٥ -٧.٠٥٤ -١.٥٦٨  ٦.٠١١ ٢٤.٤١٠
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  ٢,١٣٠e+٠١  ١,٦٠٤e+٠٠  ١٣,٢٨٠  < ٢e-١٦ ***
t           -١,٢٩٨e-٠١  ٢,٦٥٥e-٠٢ -٤,٨٨٩ ١,٧٣e-٠٦ ***
t٢           ٣,٦٩٤e-٠٤  ٩,٢١٥e-٠٥  ٤,٠٠٨ ٧,٨٩e-٠٥ ***
---
Signif. codes:  ٠ ‘***’ ٠,٠٠١ ‘**’ ٠,٠١ ‘*’ ٠,٠٥ ‘.’ ٠,١ ‘ ’ ١

Residual standard error: ٨,٨٥ on ٢٧٥ degrees of freedom
Multiple R-squared:  ٠,١٠٥٤,  Adjusted R-squared:  ٠,٠٩٨٨٩
F-statistic:  ١٦,٢ on ٢ and ٢٧٥ DF,  p-value: ٢,٢٣٤e-٠٧

> resids = residuals(model)
> fits = fitted(model)
```

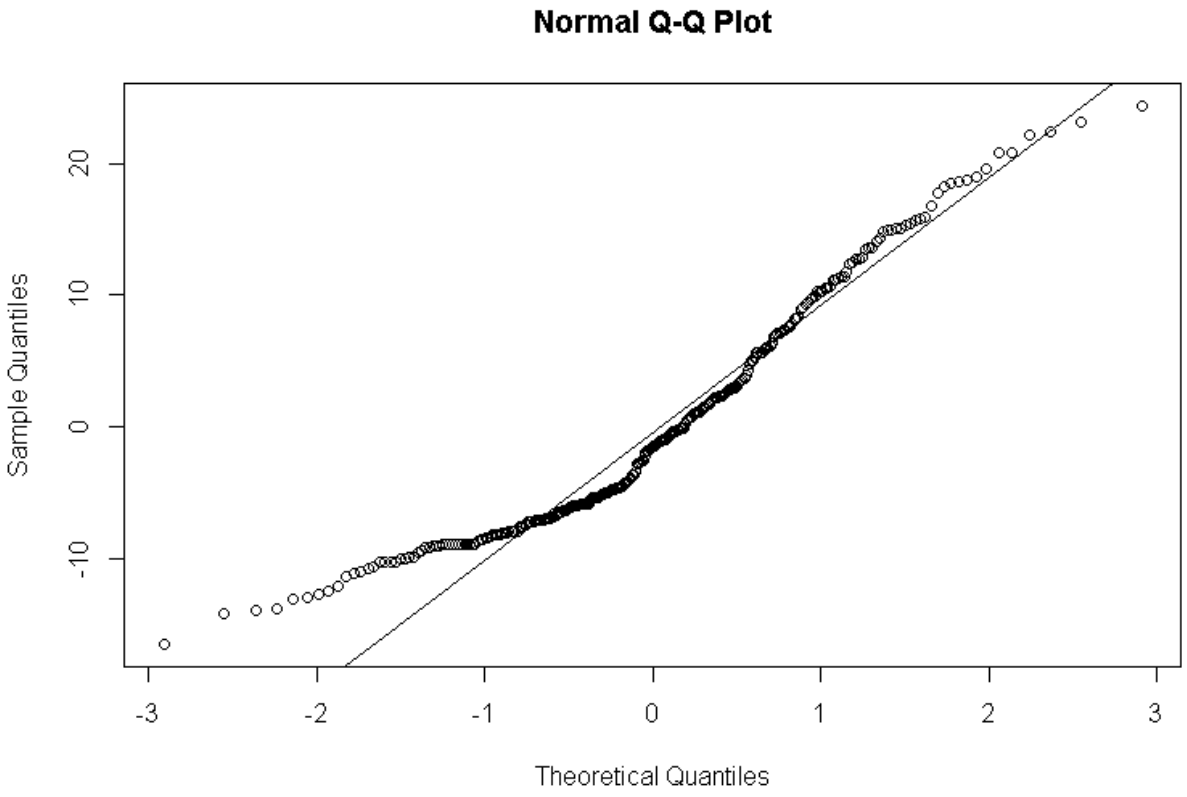
مقادیرp-value نشان می‌دهند که ضرایب رگرسیونی معنی دار هستند و R-squared مدل ۰.۰۹ میباشد و بسیار کم است که به معنای بزرگ بودن SSE یعنی فاصله مقادیر برازش داده شده با داده اصلی می‌باشد.

در مرحله بعدی فرض های اصلی رگرسیون را برسی میکنیم.

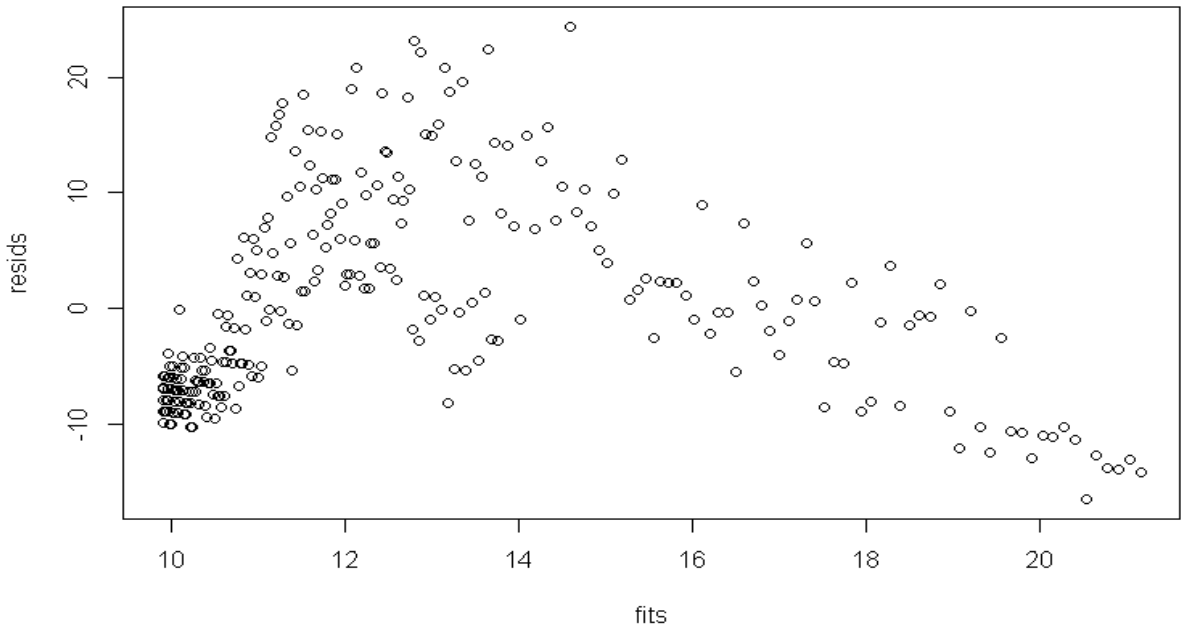
نمودارهای هیستوگرام ، qq و مانده ها در برابر مقادیر برازش داد شده و مانده ها در برابر زمان را رسم میکنیم.



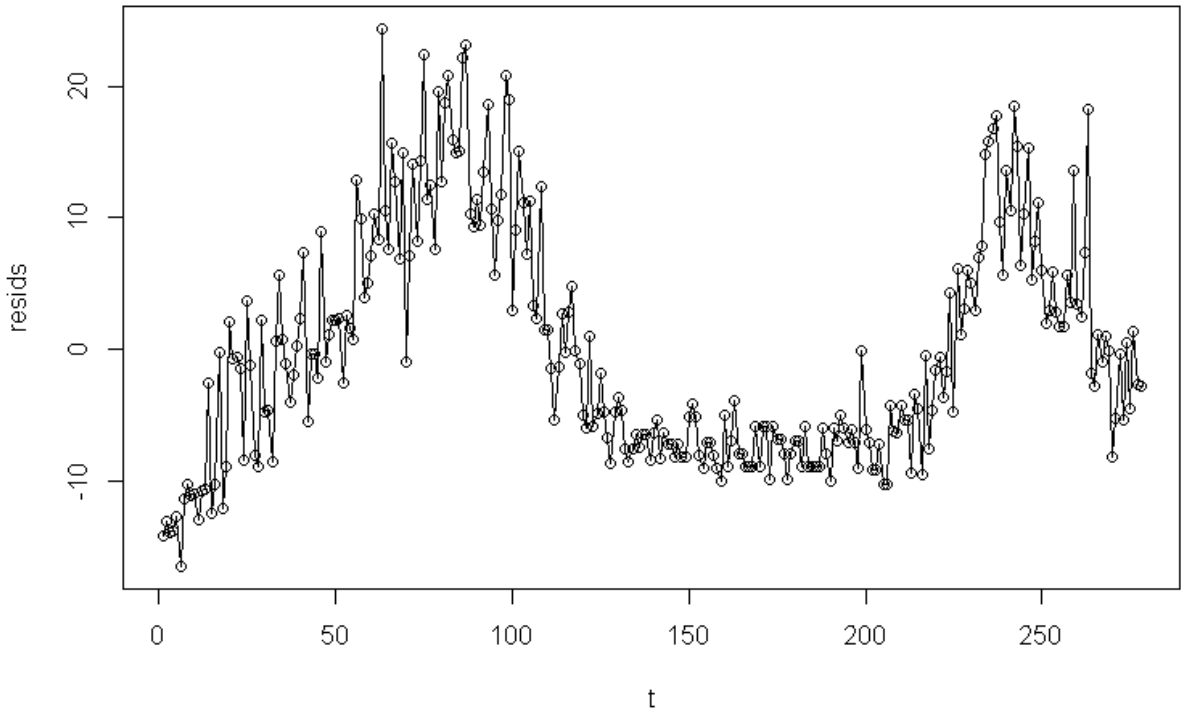
نمودار هیستوگرام شبیه به توزیع نرمال نمی‌باشد.



برای مقادیر منفی فاصله نقاط از خط زیاد می‌شود.



از نمودار مانده ها در برابر مقادیر برازش یافته همگن نبودن واریانس و روند داشتن مانده ها قابل مشاهده است.



تست نرمال بودن مانده های رگرسیونی را با استفاده از شاپیرو و اندرسون دارلینگ انجام میدهیم.

نرمال بودن مانده های رگرسیونی را با استفاده از شاپیرو و اندرسون دارلینگ انجام میدهیم.

```
> ####norm test####
> shapiro.test(resids)

     Shapiro-Wilk normality test

data:  resids
W = 0.94087, p-value = 4.024e-09

> library(nortest)
> ad.test(resids)

     Anderson-Darling normality test

data:  resids
A = 5.6747, p-value = 5.19e-14
```

مقادیر p-value در هر دو آزمون کمتر از ۰.۰۵ هستند که به معنای رد شدن فرض صفر این آزمون می‌باشد و فرض نرمال بودن مانده‌ها رد می‌شود.

از آزمون ناپارامتری runs برای برسی تصادفی بودن یا روند داشتن مانده ها استفاده میکنیم.

```
> ####runs test####
> library(randtests)
> runs.test(resids)

        Runs Test

data:  resids
statistic = -١١,٨٩٧, runs = ٤١, n١ = ١٣٩, n٢ = ١٣٩, n = ٢٧٨, p-value < ٢,٢e-١٦
alternative hypothesis: nonrandomness
```

چون p-value این آزمون کمتر از ۰.۰۵ است پس داده ها روند دارند و تصادفی نیستند.

و در نهایت آزمون همگنی واریانس را با استفاده از دستور آزمون leveneTest انجام می‌دهیم و برای اینکار داده ها را به ۲۳ گروه ۱۲ تایی دسته بندی میکنیم.

```
> ####variance test####
> fac = factor(rep(١:٢٣, each = ١٢))
> fac = c(fac , c(٢٣,٢٣))
> #fac = c(fac , c(٣٦,٣٦,٣٦))
> library(car)
Loading required package: carData
> leveneTest(resids, group = fac)
Levene's Test for Homogeneity of Variance (center = median)
      Df F value    Pr(>F)
group  ٢٢  ٣,٣٩٠٢ ١,٤٠١e-٠٦ ***
    ٢٥٥
---
Signif. codes:  ٠ ‘***’ ٠,٠٠١ ‘**’ ٠,٠١ ‘*’ ٠,٠٥ ‘.’ ٠,١ ‘ ’ ١
Warning message:
In leveneTest.default(resids, group = fac) : fac coerced to factor.
```

چون p-value این آزمون کمتر از ۰.۰۵ است پس فرض همگن بودن واریانس مانده ها رد می‌شود.

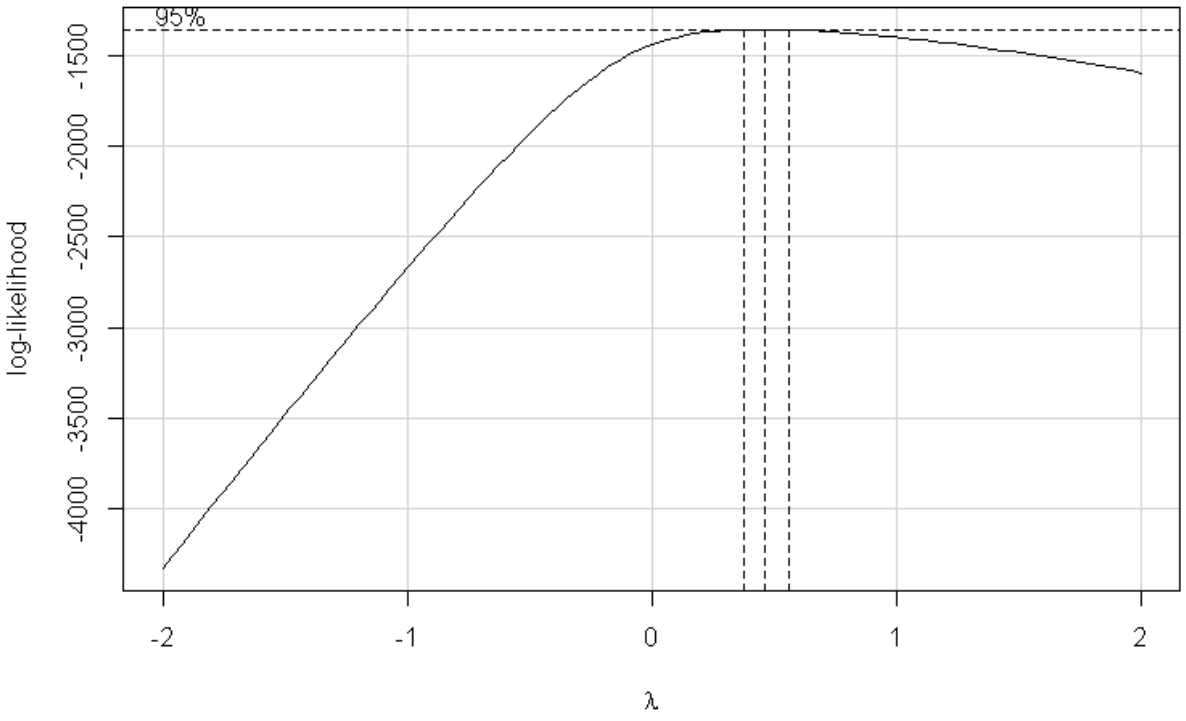
و درنتیجه با توجه به رد شدن هر سه فرض اصلی رگرسیون ، پس شرایط استفاده از رگرسیون را نداریم و ادامه کار را با سری های زمانی انجام می‌دهیم.

در ابتدا برسی میکنیم که آیا داده های اصلی در واریانس مانا هستند یا خیرمشابه بالا این کار را انجام می‌دهیم با استفاده از ۲۳ گروه ۱۲ تایی.

```
> ####variance test####
> fac = factor(rep(1:23, each = 12))
> fac = c(fac , c(23,23))
> #fac = c(fac , c(36,36,36))
> library(car)
> leveneTest(data, group = fac)
Levene's Test for Homogeneity of Variance (center = median)
       Df F value    Pr(>F)
group  22  3.3993 1.323e-06 ***
      255
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’
1
Warning message:
In leveneTest.default(data, group = fac) : fac coerced to
factor.
```

چون p-value این آزمون کمتر از ۰.۰۵ است پس فرض مانایی در واریانس را نداریم و از تبدیل باکس کاکس برای مانایی در واریانس استفاده می‌کنیم.

```
> ####box-cox####
> t = 1:length(data)
> library(MASS)
> library(car)
> data = data + 0.01
> boxCox(data ~ t, lambda = seq(from = -2, to = 2, by = .1))
> data.new = (data^0.5-1)/0.5
> leveneTest(data.new, group = fac)
Levene's Test for Homogeneity of Variance (center = median)
       Df F value Pr(>F)
group  22  1.1434 0.3011
      255
Warning message:
In leveneTest.default(data.new, group = fac) : fac coerced to
factor.
```

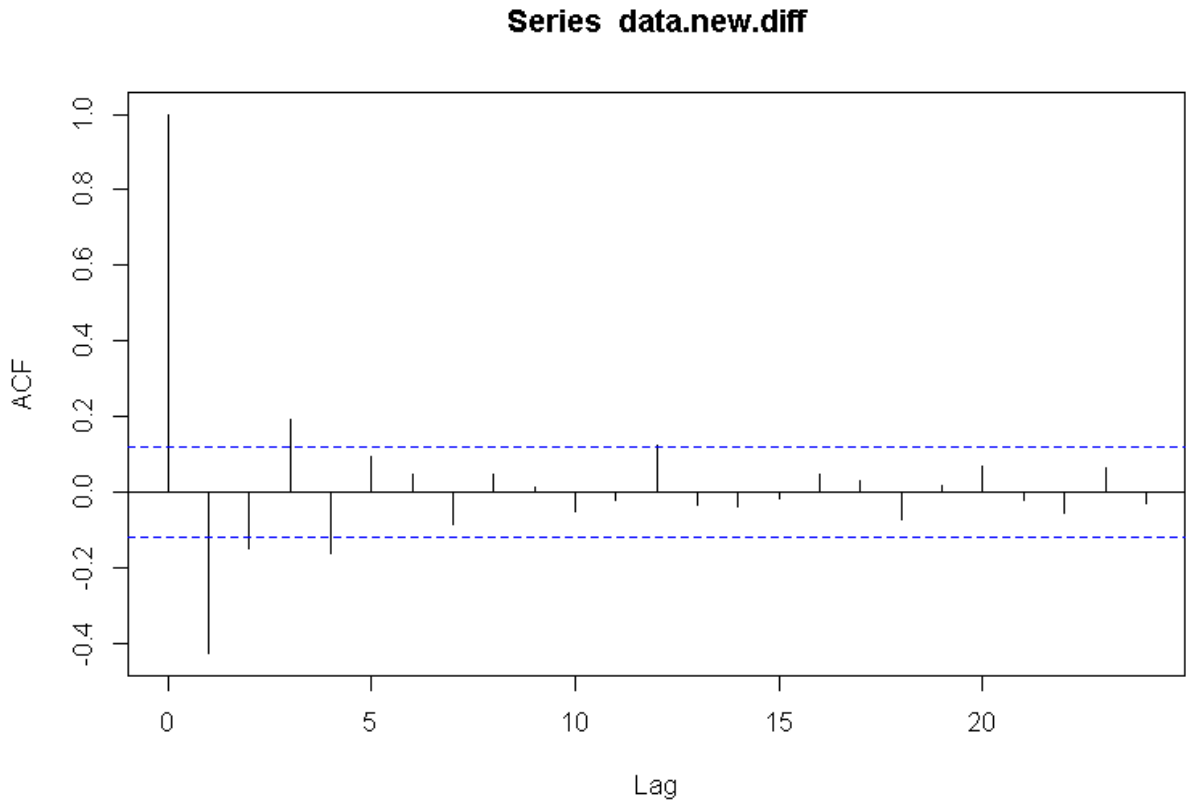


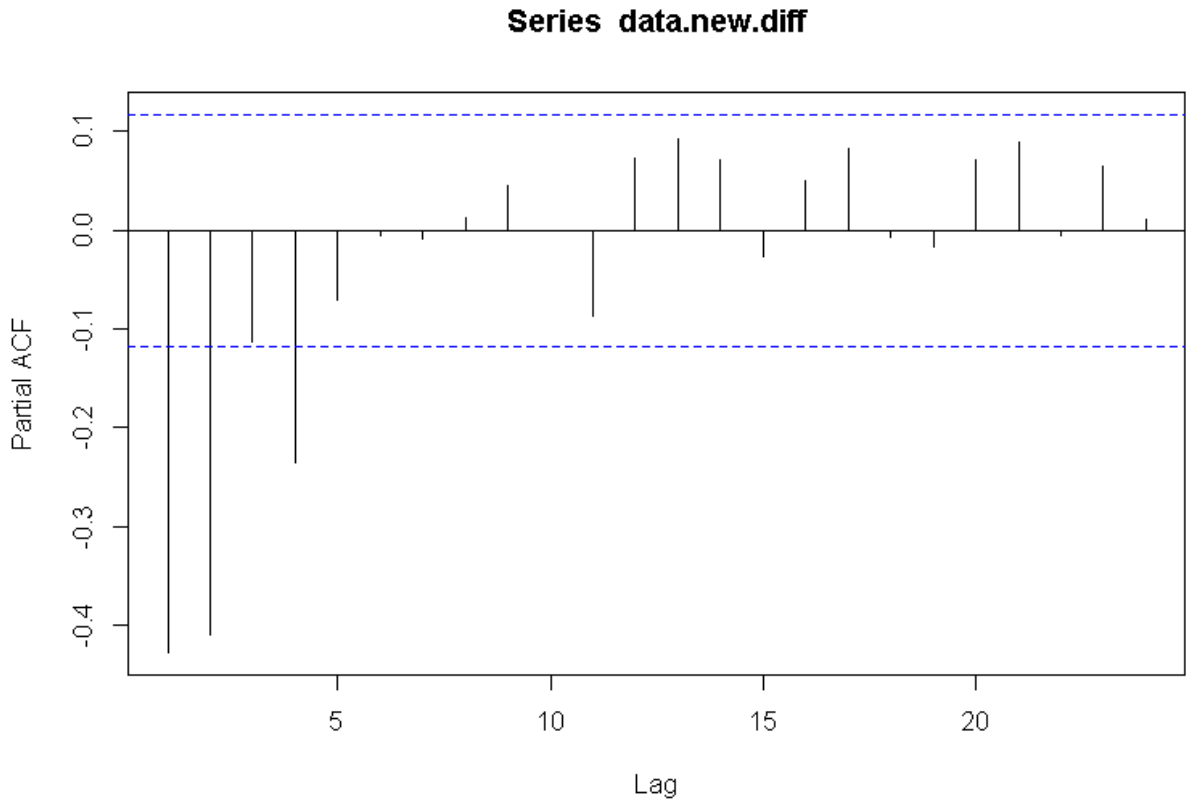
چون در داده ها روز هایی بدون فوتی هم داشتیم یعنی تعداد فوتی برابر صفر بوده و از آنجایی که در تبدیل باکس کاکس باید داده ها مثبت باشند پس به همه عدد ۰.۰۱ را اضافه می‌کنیم. چون p-value آزمون نزدیک ۰.۳۰۱ است و بزرگتر از ۰.۰۵ است پس فرض مانایی در واریانس تایید می‌شود.

برای برسی فرض مانایی در میانگین از آزمون ناپارامتری کروسکال والیس استفاده میکنیم که فرض توزیعی ندارد.

```
> ####mean test####
> kruskal.test(data.new, g = fac)


     Kruskal-Wallis rank sum test


data:  data.new and fac
Kruskal-Wallis chi-squared = 238.79, df = 22, p-value <
2.2e-16


> data.new.diff = diff(data.new)
> fac.diff = fac[-1]
> kruskal.test(data.new.diff, g = fac.diff)


     Kruskal-Wallis rank sum test


data:  data.new.diff and fac.diff
Kruskal-Wallis chi-squared = 4.435, df = 22, p-value = 1


> leveneTest(data.new.diff, group = fac.diff)
Levene's Test for Homogeneity of Variance (center = median)
       Df F value Pr(>F)
group  22  1.2729 0.1895
      254
Warning message:
In leveneTest.default(data.new.diff, group = fac.diff) :
  fac.diff coerced to factor.
```

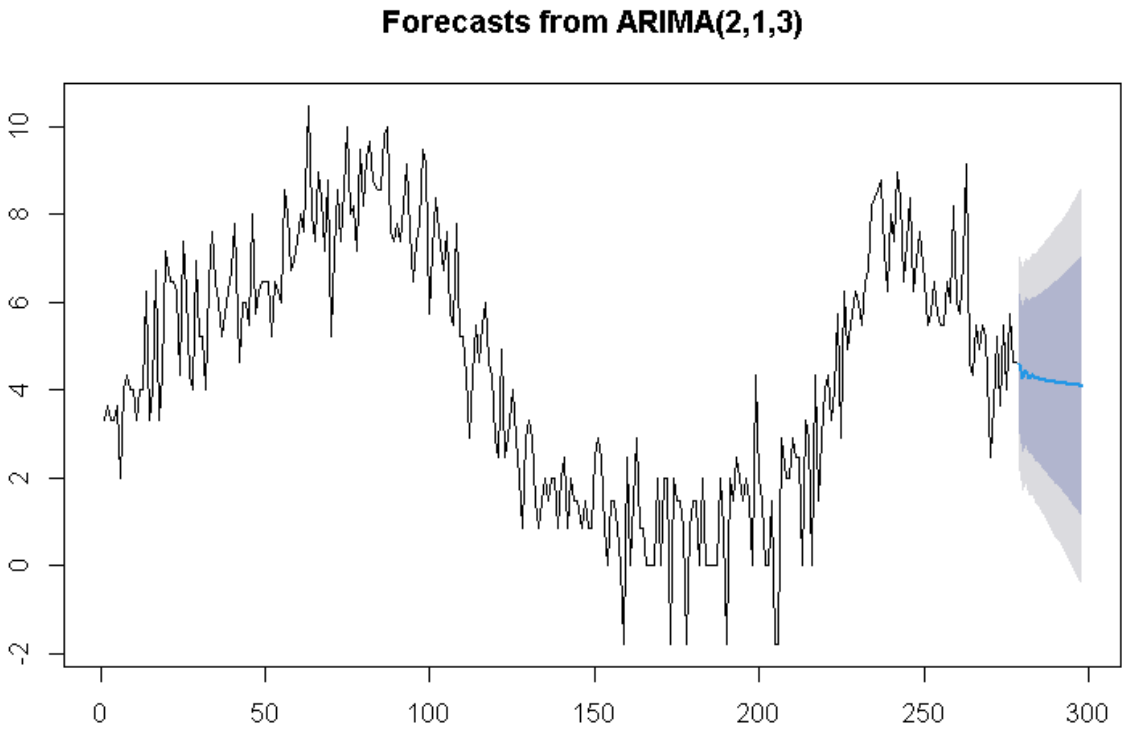
در ابتدا p-value ۰.۰۵ است و فرض مانایی رد می‌شود همچنین پس از یک مرتبه تفاضلی کردن روی داده ها p-value نزدیک یک می‌شود و داده ها در میانگین مانا می‌شوند. پس از مانایی در میانگین و واریانس نمودار های acf , pacf را برای داده های جدید رسم میکنیم.


Series data.new.diff
ACF
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
0
5
10
15
20
Lag


Series data.new.diff
Partial ACF
0.1
0.0
-0.1
-0.2
-0.3
-0.4
5
10
15
20
Lag

با توجه به نمودار های بالا برای اتورگرسیو ۲ مرتبه و برای میانگین متحرک ۳ مرتبه را در نظر می‌گیریم.

```
> ####model####
> y = ts(data = data.new)
> order = c(2, 1, 3)
> library(forecast)
> fit <- Arima(y = y, order = order)
> summary(fit)
Series: y
ARIMA(2,1,3)

Coefficients:
         ar1     ar2      ma1      ma2     ma3
      0.2874  0.6051  -0.9725  -0.5688  0.6011
s.e.  0.1416  0.1340   0.1282   0.1920  0.0876

sigma^2 estimated as 1.543:  log likelihood=-451.17
AIC=914.34   AICc=914.65   BIC=936.08

Training set error measures:
                     ME    RMSE       MAE        MPE
MAPE      MASE         ACF1
Training set 0.002518319 1.22874 0.9715128 -705.2714
۷۳۶.۷۲۳۲ ۰.۸۱۹۸۹۰۱ -۰.۰۳۰۰۱۲۳۵
> ####forecast####
> plot(forecast(fit, h = 20))
```


Forecasts from ARIMA(2,1,3)
10
8
6
4
2
0
-2
0
50
100
150
200
250
300