Template for Time Series Analysis

James Long, Shyam Attiganal, Manoj Remela, Eric Young

September 30, 2019

Intro

This template shows how to create a basic time series model, then de-seasonalize the data, and then model the new data.

File: 6304 Module 7 Assignment Data.xlsx
Sheet: Quebec Car Sales

Pre-processing

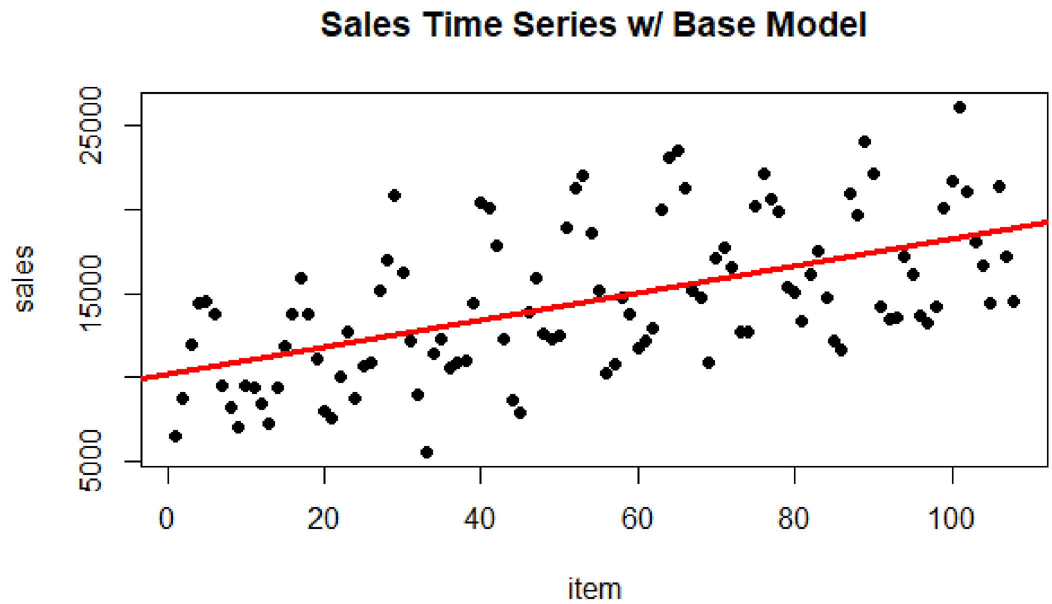
```
setwd(params$wd)
library(readxl)
# Durbin-Watson module
library(car)

## Loading required package: carData

car.sales=read_excel("6304 Module 7 Assignment Data.xlsx", sheet = "Quebec
Car Sales", skip = 3)
colnames(car.sales)=c("yrmo","sales")
names(car.sales)

## [1] "yrmo"  "sales"

car.sales$yr=format(car.sales$yrmo,'%Y')
car.sales$mo=format(car.sales$yrmo,'%m')
car.sales$item=seq(1:nrow(car.sales))
attach(car.sales)
```

Create the Base Model

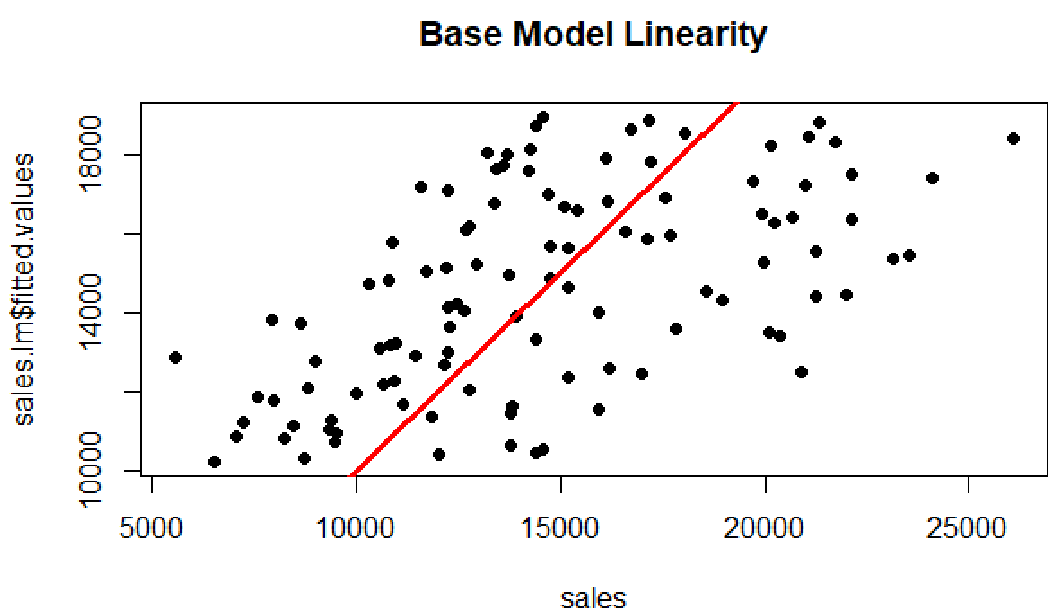
```r
# create the base model
sales.lm=lm(sales~item)
summary(sales.lm)
```

```
##
## Call:
## lm(formula = sales ~ item)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -7281  -3032  -1060   2912   8376
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10169.57     728.76  13.955  < 2e-16 ***
## item           81.20      11.61   6.996 2.47e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3760 on 106 degrees of freedom
## Multiple R-squared:  0.3159, Adjusted R-squared:  0.3094
## F-statistic: 48.94 on 1 and 106 DF,  p-value: 2.466e-10
```

```r
# plot the base model with the original sales data
plot(item,sales,pch=19,main="Sales Time Series w/ Base Model")
abline(sales.lm$coefficients,lwd=3,col="red")
```



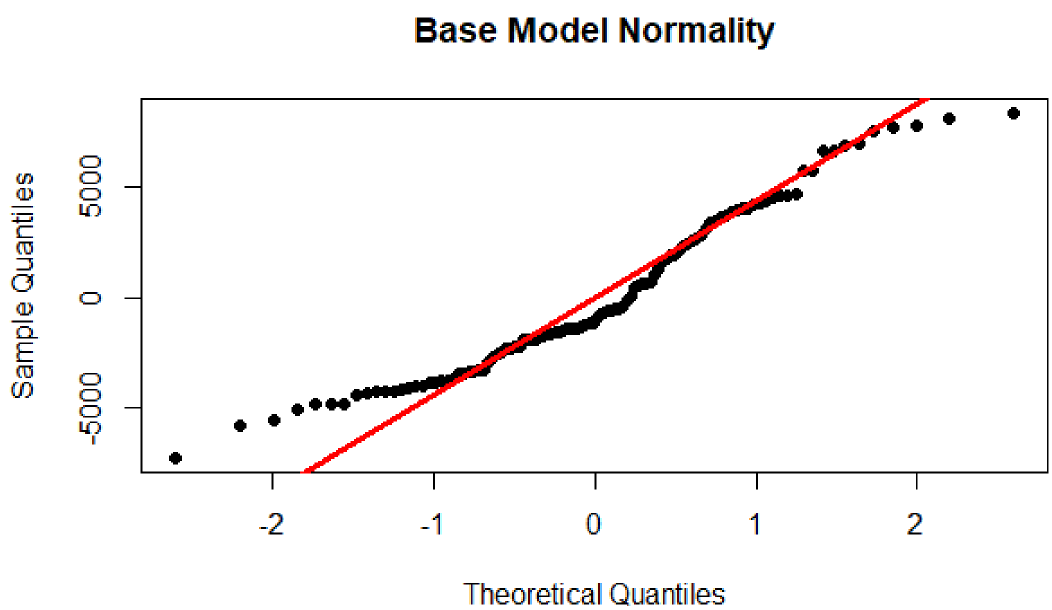

```
# or the plot fitted points instead of the fitted line
# points(item,sales.lm$fitted.values,pch=19,col="blue")

# linearity
plot(sales,sales.lm$fitted.values,pch=19,main="Base Model Linearity")
abline(0,1,lwd=3,col="red")
```



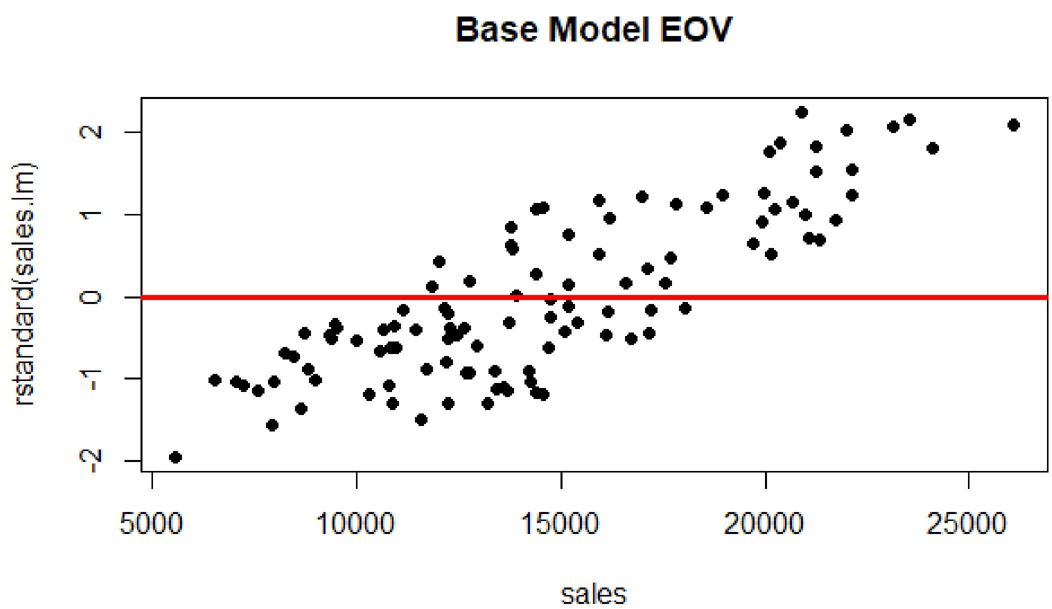
```r
# independence via test for autocorrelation
sales.ac=durbinWatsonTest(sales.lm)
sales.ac
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.5973932     0.7833808       0
##  Alternative hypothesis: rho != 0
```

```r
# normality
qqnorm(sales.lm$residuals,pch=19,main="Base Model Normality")
qqline(sales.lm$residuals,lwd=3,col="red")
```




```r
# EOV
plot(sales,rstandard(sales.lm),pch=19,main="Base Model EOV")
abline(h=0,lwd=3,col="red")
```



De-seasonalize the data

```
# create seasonalize indexes the hard way
des.sales=data.frame(index=1:12,sales=0,count=0,average=0,seas.index=0)
car.sales$des.sales=0
sales.avg=mean(car.sales$sales)

for(i in item){
    if(i%%12==1){
    des.sales$sales[1]=des.sales$sales[1]+car.sales$sales[i]
    des.sales$count[1]=des.sales$count[1]+1
    des.sales$average[1]=des.sales$sales[1]/des.sales$count[1]
    des.sales$seas.index[1]=des.sales$average[1]/sales.avg
    }
    if(i%%12==2){
        des.sales$sales[2]=des.sales$sales[2]+car.sales$sales[i]
        des.sales$count[2]=des.sales$count[2]+1
        des.sales$average[2]=des.sales$sales[2]/des.sales$count[2]
        des.sales$seas.index[2]=des.sales$average[2]/sales.avg
    }
    if(i%%12==3){
        des.sales$sales[3]=des.sales$sales[3]+car.sales$sales[i]
        des.sales$count[3]=des.sales$count[3]+1
        des.sales$average[3]=des.sales$sales[3]/des.sales$count[3]
```

```
        des.sales$seas.index[3]=des.sales$average[3]/sales.avg
    }
    if(i%%12==4){
        des.sales$sales[4]=des.sales$sales[4]+car.sales$sales[i]
        des.sales$count[4]=des.sales$count[4]+1
        des.sales$average[4]=des.sales$sales[4]/des.sales$count[4]
        des.sales$seas.index[4]=des.sales$average[4]/sales.avg
    }
    if(i%%12==5){
        des.sales$sales[5]=des.sales$sales[5]+car.sales$sales[i]
        des.sales$count[5]=des.sales$count[5]+1
        des.sales$average[5]=des.sales$sales[5]/des.sales$count[5]
        des.sales$seas.index[5]=des.sales$average[5]/sales.avg
    }
    if(i%%12==6){
        des.sales$sales[6]=des.sales$sales[6]+car.sales$sales[i]
        des.sales$count[6]=des.sales$count[6]+1
        des.sales$average[6]=des.sales$sales[6]/des.sales$count[6]
        des.sales$seas.index[6]=des.sales$average[6]/sales.avg
    }
    if(i%%12==7){
        des.sales$sales[7]=des.sales$sales[7]+car.sales$sales[i]
        des.sales$count[7]=des.sales$count[7]+1
        des.sales$average[7]=des.sales$sales[7]/des.sales$count[7]
        des.sales$seas.index[7]=des.sales$average[7]/sales.avg
    }
    if(i%%12==8){
        des.sales$sales[8]=des.sales$sales[8]+car.sales$sales[i]
        des.sales$count[8]=des.sales$count[8]+1
        des.sales$average[8]=des.sales$sales[8]/des.sales$count[8]
        des.sales$seas.index[8]=des.sales$average[8]/sales.avg
    }
    if(i%%12==9){
        des.sales$sales[9]=des.sales$sales[9]+car.sales$sales[i]
        des.sales$count[9]=des.sales$count[9]+1
        des.sales$average[9]=des.sales$sales[9]/des.sales$count[9]
        des.sales$seas.index[9]=des.sales$average[9]/sales.avg
    }
    if(i%%12==10){
        des.sales$sales[10]=des.sales$sales[10]+car.sales$sales[i]
        des.sales$count[10]=des.sales$count[10]+1
        des.sales$average[10]=des.sales$sales[10]/des.sales$count[10]
        des.sales$seas.index[10]=des.sales$average[10]/sales.avg
    }
    if(i%%12==11){
        des.sales$sales[11]=des.sales$sales[11]+car.sales$sales[i]
        des.sales$count[11]=des.sales$count[11]+1
        des.sales$average[11]=des.sales$sales[11]/des.sales$count[11]
        des.sales$seas.index[11]=des.sales$average[11]/sales.avg
    }
```

```
    if(i%%12==0){
        des.sales$sales[12]=des.sales$sales[12]+car.sales$sales[i]
        des.sales$count[12]=des.sales$count[12]+1
        des.sales$average[12]=des.sales$sales[12]/des.sales$count[12]
        des.sales$seas.index[12]=des.sales$average[12]/sales.avg
    }
}

# # create seasonalize indexes the easy way
# # code by Dr. Ron Satterfield
# indices=data.frame(month=1:12,average=0,index=0)
# for(i in 1:12) {
#     count=0
#     for(j in 1:nrow(car.sales)) {
#         if(i==car.sales$month[j]) {
#             indices$average[i]=indices$average[i]+car.sales$sales[j]
#             count=count+1
#         }
#     }
#     indices$average[i]=indices$average[i]/count
#     indices$index[i]=indices$average[i]/mean(car.sales$sales)}

# de-seasonalize the sales data the hard way
for(i in item){
    if(i%%12==1){
        car.sales$des.sales[i]=car.sales$sales[i]/des.sales$seas.index[1]
    }
    if(i%%12==2){
        car.sales$des.sales[i]=car.sales$sales[i]/des.sales$seas.index[2]
    }
    if(i%%12==3){
        car.sales$des.sales[i]=car.sales$sales[i]/des.sales$seas.index[3]
    }
    if(i%%12==4){
        car.sales$des.sales[i]=car.sales$sales[i]/des.sales$seas.index[4]
    }
    if(i%%12==5){
        car.sales$des.sales[i]=car.sales$sales[i]/des.sales$seas.index[5]
    }
    if(i%%12==6){
        car.sales$des.sales[i]=car.sales$sales[i]/des.sales$seas.index[6]
    }
    if(i%%12==7){
        car.sales$des.sales[i]=car.sales$sales[i]/des.sales$seas.index[7]
    }
    if(i%%12==8){
        car.sales$des.sales[i]=car.sales$sales[i]/des.sales$seas.index[8]
    }
    if(i%%12==9){
        car.sales$des.sales[i]=car.sales$sales[i]/des.sales$seas.index[9]
```

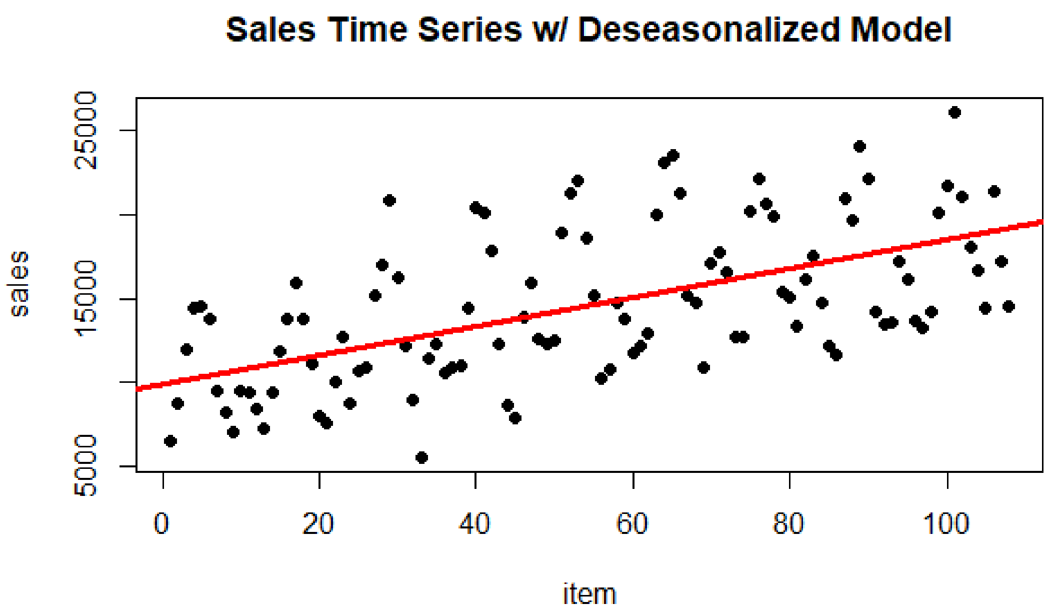
```r
    }
    if(i%%12==10){
        car.sales$des.sales[i]=car.sales$sales[i]/des.sales$seas.index[10]
    }
    if(i%%12==11){
        car.sales$des.sales[i]=car.sales$sales[i]/des.sales$seas.index[11]
    }
    if(i%%12==0){
        car.sales$des.sales[i]=car.sales$sales[i]/des.sales$seas.index[12]
    }
}

# # de-seasonalize the sales data the easy way
# # code by Dr. Ron Satterfield
# for(i in 1:12){
#     for(j in 1:nrow(car.sales)){
#         if(i==car.sales$month[j]){
#             car.sales$deseason.sales[j]=car.sales$sales[j]/indices$index[i]
#         }
#     }
# }
```

Create the New Model

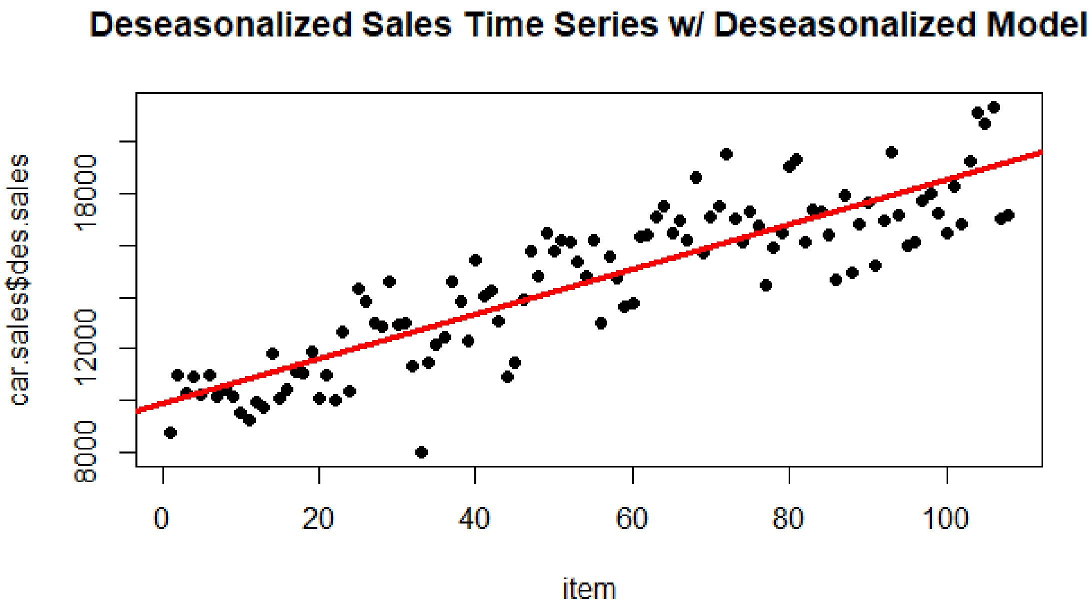
```r
# create the de-seasonalized model
des.sales.lm=lm(car.sales$des.sales~item)
summary(des.sales.lm)

##
## Call:
## lm(formula = car.sales$des.sales ~ item)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4732.4 -1029.5    26.3   943.1  3409.7
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 9909.741    289.297   34.26   <2e-16 ***
## item          85.970      4.608   18.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1493 on 106 degrees of freedom
```

```
## Multiple R-squared:  0.7666, Adjusted R-squared:  0.7644
## F-statistic: 348.1 on 1 and 106 DF,  p-value: < 2.2e-16
```

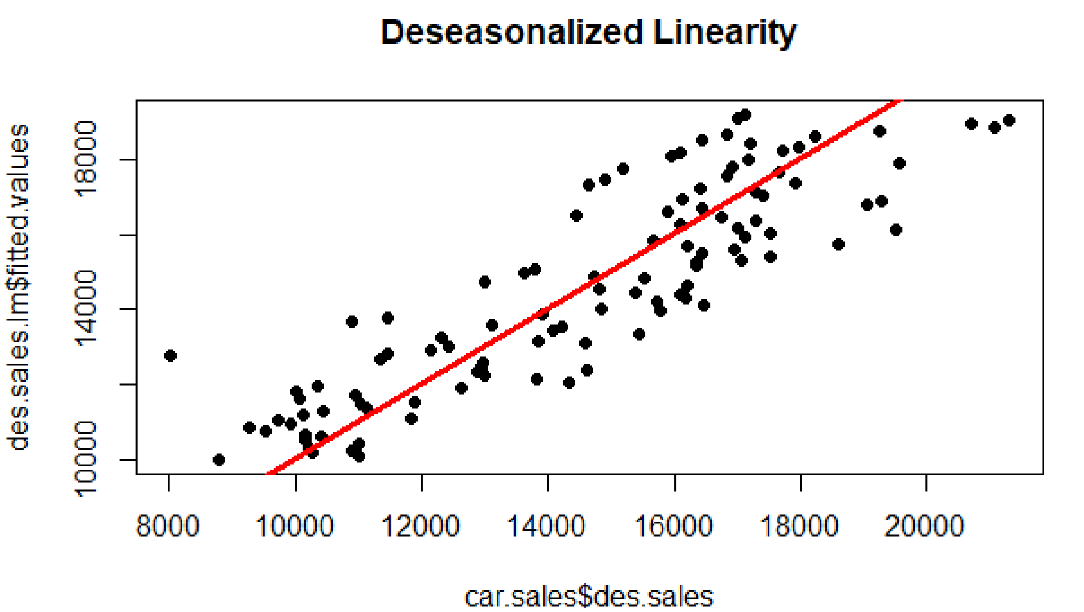
```
# plot the deseasonalized model with the original sales data
plot(item,sales,pch=19,main="Sales Time Series w/ Deseasonalized Model")
abline(des.sales.lm$coefficients,lwd=3,col="red")
```




```
# or the plot fitted points instead of the fitted line
# points(item,des.sales.lm$fitted.values,pch=19,col="blue")

# plot the deseasonalized model with the de-seasonalized sales data
plot(item,car.sales$des.sales,pch=19,main="Deseasonalized Sales Time Series
w/ Deseasonalized Model")
abline(des.sales.lm$coefficients,lwd=3,col="red")
```



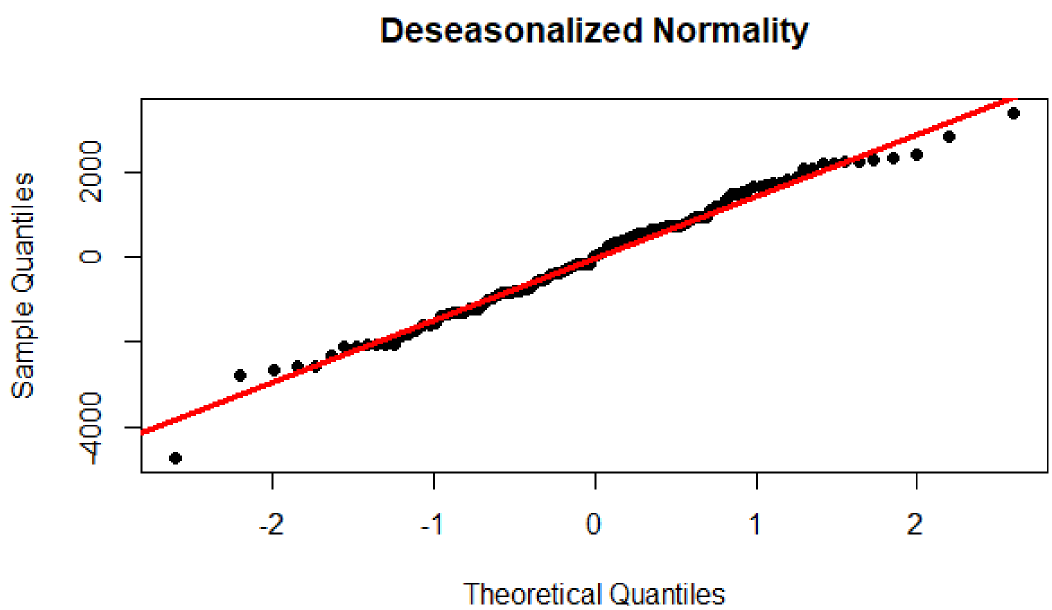

```
# or the plot fitted points instead of the fitted line
# points(item,des.sales.lm$fitted.values,pch=19,col="blue")

# linearity
plot(car.sales$des.sales,des.sales.lm$fitted.values,pch=19,main="Deseasonaliz
ed Linearity")
abline(0,1,lwd=3,col="red")
```



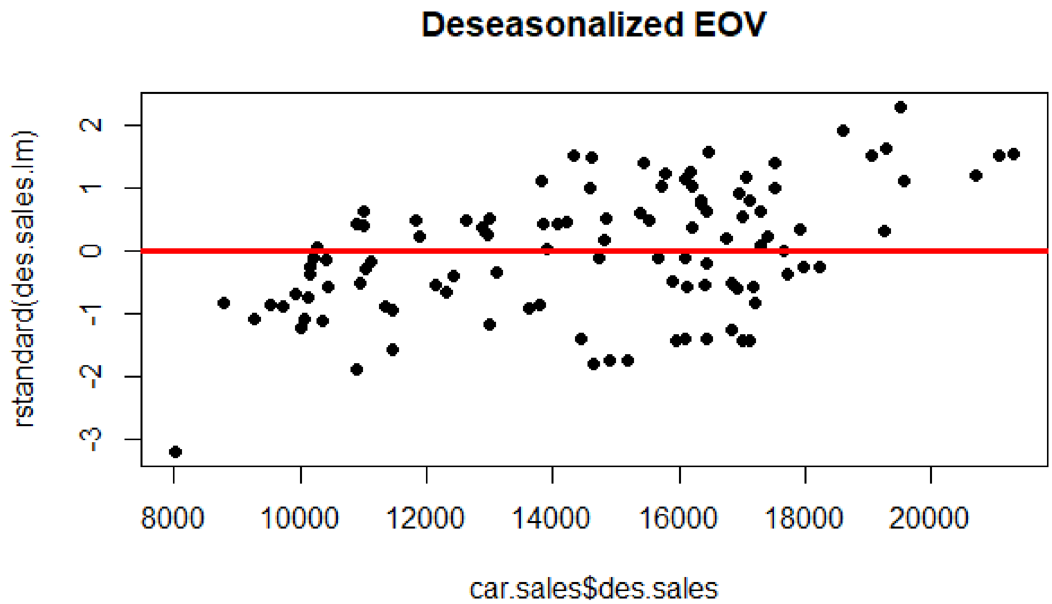
```r
# independence via test for autocorrelation
des.sales.ac=durbinWatsonTest(des.sales.lm)
des.sales.ac
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.3586422      1.258348       0
##  Alternative hypothesis: rho != 0
```

```r
# normality
qqnorm(des.sales.lm$residuals,pch=19,main="Deseasonalized Normality")
qqline(des.sales.lm$residuals,lwd=3,col="red")
```




```r
# EOV
plot(car.sales$des.sales,rstandard(des.sales.lm),pch=19,main="Deseasonalized
EOV")
abline(h=0,lwd=3,col="red")
```



Conclusions

- The base and new models graphically fit the original data about the same.
- The new model graphically fits the de-seasonalized data better.
- The new model has better linearity.
- Both models have autocorrelation, but the new model has less.
- The new model has better normality.
- Neither model has EOV, but the new model has less inequality.