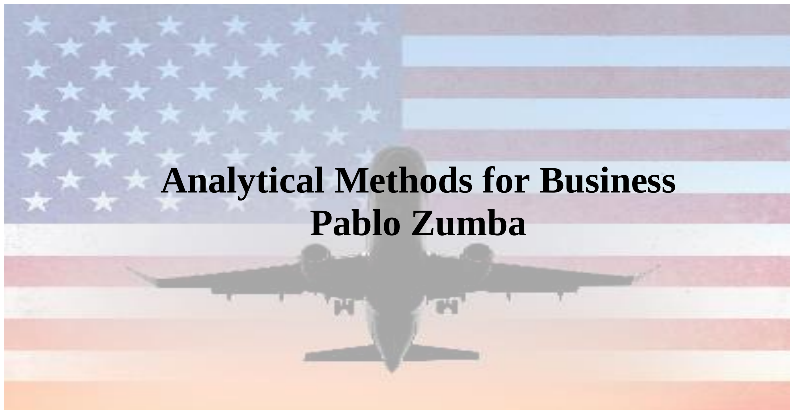


Write a simple R script to execute the following data preprocessing and statistical analysis. Where required show analytical output and interpretations.

Preprocessing

1. Load the file "6304 Module 7 Assignment Data.xlsx" into R. This data shows the monthly count of the number of passengers across all US Domestic flights. The data is recorded in a monthly fashion starting from January 1996 to August 2012. Variables in the data set are:

   Date: the month and year of the specific observation.

   Passengers: The number of passengers carried on all US Domestic flights.

   Month: The numerical month of the observation.

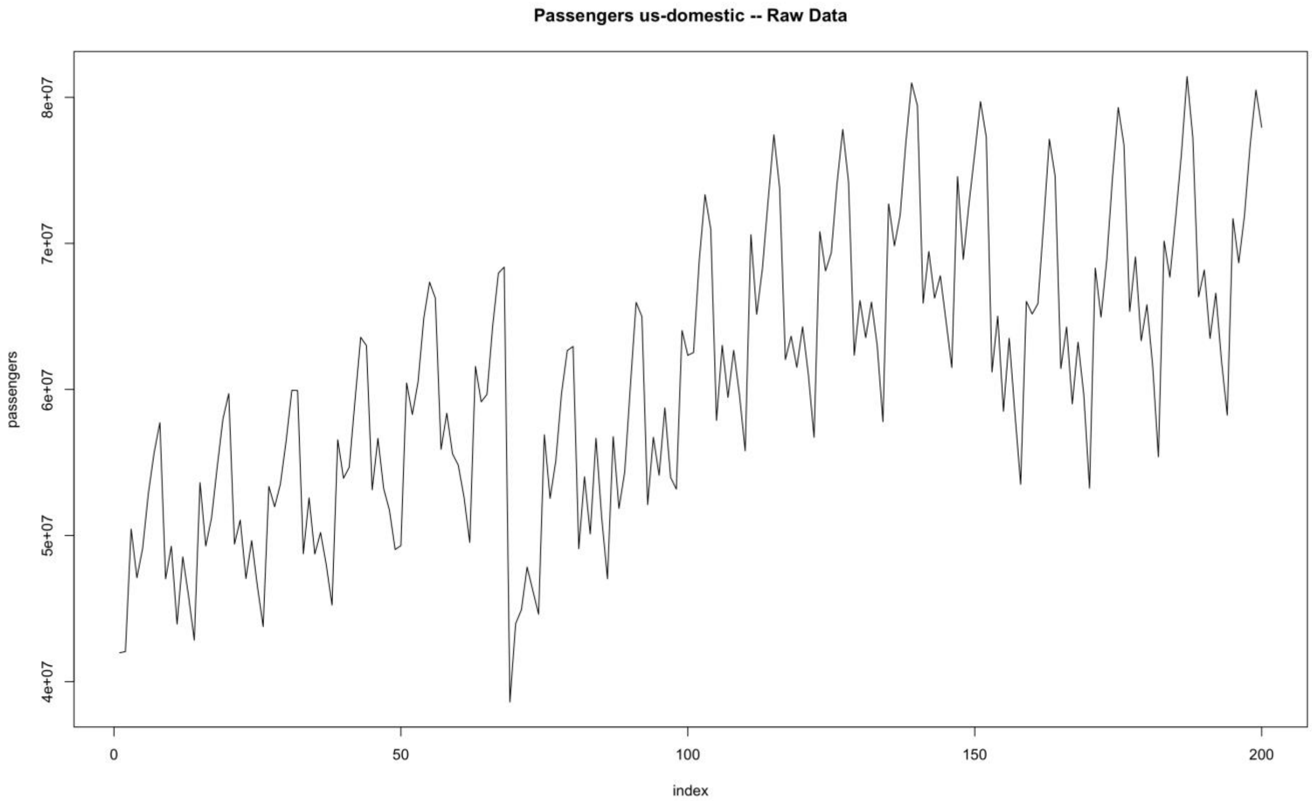
   Year: The numerical year of the observation.

```
#Pablo Zumba
#Processing 1:
rm(list=ls())
set.seed(54252888)
us_Passengers = rio::import("6304 Module 7 Assignment Data.xlsx")
colnames(us_Passengers)=tolower(make.names(colnames(us_Passengers)))
names(us_Passengers)
str(us_Passengers)
> names(us_Passengers)
[1] "date"       "passengers" "month"      "year"
> str(us_Passengers)
'data.frame':   200 obs. of  4 variables:
 $ date      : POSIXct, format: "1996-01-01" "1996-02-01" "1996-03-01" ...
 $ passengers: num  41972194 42054796 50443045 47112397 49118248 ...
 $ month     : num  1 2 3 4 5 6 7 8 9 10 ...
 $ year      : num  1996 1996 1996 1996 1996 ...
```

Analysis

1. Show a line plot of the data using the index as 'x' and passengers as the "y" variable.

```
#Analysis 1:
us_Passengers$index=seq(1:nrow(us_Passengers))
names(us_Passengers)
attach(us_Passengers)
plot(index,passengers,type="l",pch=19,
     main="Passengers us-domestic -- Raw Data")
```



Interpretation: Plotting the dependent variable "passengers" in the y axis as type= "l", we can observe an incremental cyclical pattern over time. A remarkable drop between the 50-100 index can be interpreted as a potential outlier and it will increase the residual value at that point when performing the regression, but after analyzing the data, we should be able to explain it based on the business problem.

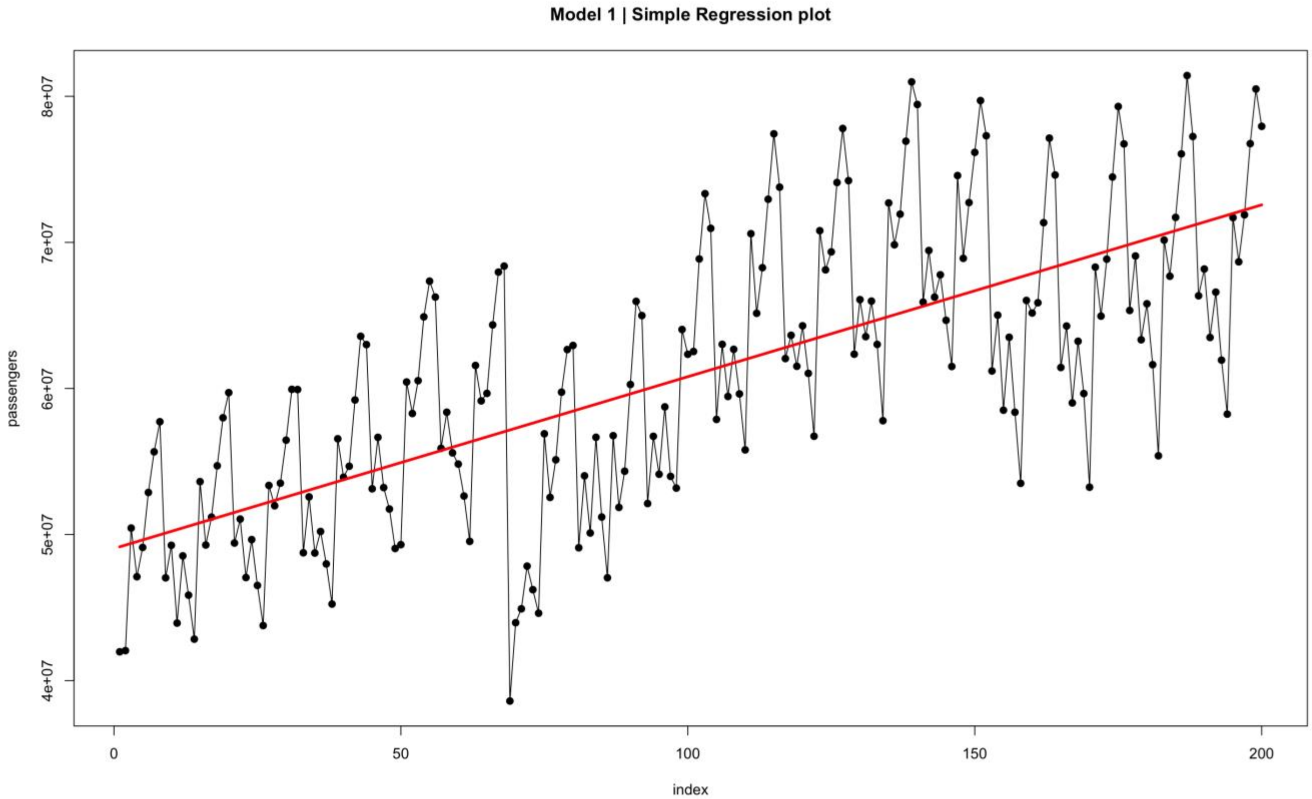
2. Using all the rows parameterize a base time series simple regression model using "index" as the independent variable and passengers as the dependent variable. Show the summary of your regression output.

```
model_1.out=lm(passengers~index,data=us_Passengers)
summary(model_1.out)
> summary(model_1.out)

Call:
lm(formula = passengers ~ index, data = us_Passengers)

Residuals:
      Min        1Q    Median        3Q       Max
-18556757  -4696514   -487347   5128043  15595104

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 49041652     961448   51.01   <2e-16 ***
index         117637       8295   14.18   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6773000 on 198 degrees of freedom
Multiple R-squared:  0.5039,    Adjusted R-squared:  0.5014
F-statistic: 201.1 on 1 and 198 DF,  p-value: < 2.2e-16
```

Interpretation: We can see that the p-value at the index variable is very low (<5%), indicating we can reject the null hypothesis and accept that there is a relationship between the index and passenger data. Having a multiple R-squared value of 0.5 tells us that we can only explain roughly half (50%) of the variability in the passenger's data. As a result of applying the square root to the Multiple R-squared value, we obtain an “r” value of 0.71; not a very strong relationship but indicates a positive linear relationship. The Beta value can be interpreted as: For every additional month in the future, we expect the number of passengers to increase by 117637.

3. Drawing on Analysis Part 1 above, show a properly titled plot of the time series data with the simple regression line layered on the graph in a contrasting color.

```
plot(index,passengers,type="o",pch=19,
     main="Simple Regression plot")
points(model_1.out$fitted.values, type="l", lwd=3, col="red")
```



4. Execute and interpret a Durbin-Watson test on your model results.

```
durbin_test = car::dwt(model_1.out)
durbin_test
lag Autocorrelation D-W Statistic p-value
   1       0.5605341     0.8700628       0
 Alternative hypothesis: rho != 0
```

Interpretation: Based on the Durbin Watson Test previously performed, we get an autocorrelation value of 0.56 (approaching 0), which indicates a positive serial autocorrelation in our dependent variable 'passenger'. It suggests that the total number of passengers carried on all US Domestic flights last month shows a positive correlation with the total number of passengers carried on all US Domestic flights the current month. So, if the total number of passengers increased say July, it would most likely increase by August. Likewise, if the total number of passengers fell last month, it is "likely" to fall this month as well.

5. Note the original data appears to have a pronounced cyclical pattern. Assuming the complete cycles are 12 months long, construct a set of seasonal indices which describe the typical annual fluctuations in passengers. Use these indices to de-seasonalize the passenger's data.

```
seasonal_indices=data.frame(month=1:12,average=0,index2=0)
for(i in 1:12) {
  count=0
  for(j in 1:nrow(us_Passengers)) {
    if(i==us_Passengers$month[j]) {

seasonal_indices$average[i]=seasonal_indices$average[i]+us_Passengers$passen
gers[j]
      count=count+1}
  }
  seasonal_indices$average[i]=seasonal_indices$average[i]/count

seasonal_indices$index2[i]=seasonal_indices$average[i]/mean(us_Passengers$pa
ssengers)
}

for(i in 1:12){
  for(j in 1:nrow(us_Passengers)){
    if(i==us_Passengers$month[j]){

us_Passengers$deseason.index[j]=us_Passengers$passengers[j]/seasonal_indices
$index2[i]}
  }
}
```

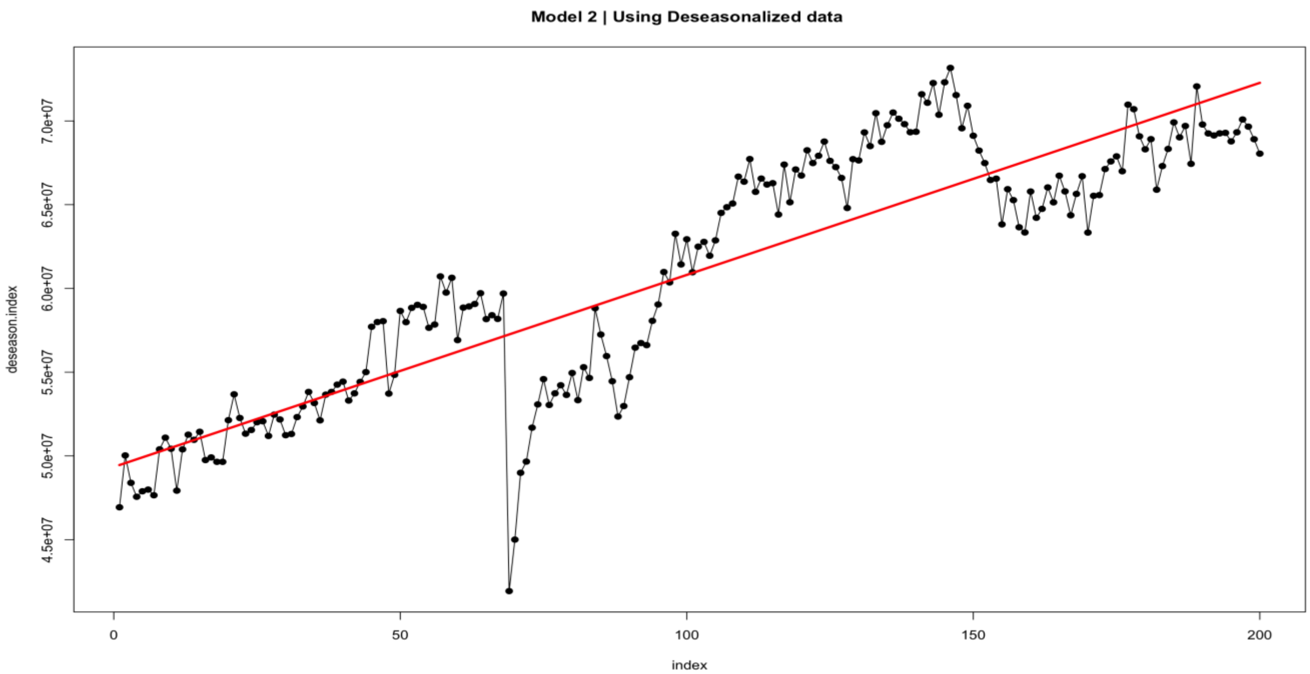
	month	average	index2
1	1	54432319	0.8943241
2	2	51163705	0.8406207
3	3	63444743	1.0423984
4	4	60290829	0.9905795
5	5	62428112	1.0256951
6	6	67069801	1.1019581
7	7	71102164	1.1682100
8	8	69717354	1.1454575
9	9	56035273	0.9206606
10	10	59460577	0.9769384
11	11	55800412	0.9168019
12	12	58629147	0.9632780

	date	passengers	month	year	index	deseason.index
1	1996-01-01	41972194	1	1996	1	46931751
2	1996-02-01	42054796	2	1996	2	50028267
3	1996-03-01	50443045	3	1996	3	48391331
4	1996-04-01	47112397	4	1996	4	47560440
5	1996-05-01	49118248	5	1996	5	47887767
6	1996-06-01	52880510	6	1996	6	47987767
7	1996-07-01	55664750	7	1996	7	47649611
8	1996-08-01	57723208	8	1996	8	50393147
9	1996-09-01	47035464	9	1996	9	51088819
10	1996-10-01	49263120	10	1996	10	50426022
11	1996-11-01	43937074	11	1996	11	47924285
12	1996-12-01	48539606	12	1996	12	50390027
13	1997-01-01	45850623	1	1997	13	51268466
14	1997-02-01	42838949	2	1997	14	50961093
15	1997-03-01	53620994	3	1997	15	51440021
16	1997-04-01	49282817	4	1997	16	49751501
17	1997-05-01	51191842	5	1997	17	49909415
18	1997-06-01	54707221	6	1997	18	49645462
19	1997-07-01	57995025	7	1997	19	49644351
20	1997-08-01	59715433	8	1997	20	52132387

Interpretation: If the index2 variable is exactly 1.00 means the total number of Passengers is exactly at the 12-month average. If it's greater than 1, means above the 12-month average and vice-versa.

6. Parameterize a simple regression model using the deseasonalized passenger data. Show the summary of this model's output and a plot showing a) index as the x variable, b) the deseasonalized values as the y variable, and c) your deseasonalized regression line with this data.

```
attach(us_Passengers)
model_2_des=lm(deseason.index~index,data=us_Passengers)
summary(model_2_des)
plot(index,deseason.index,type="o",pch=19,
     main="Model 2 | Deseasonalized Regression ")
points(model_2_des$fitted.values, type="l", lwd=3, col="red")
> summary(model_2_des)

Call:
lm(formula = deseason.index ~ index, data = us_Passengers)

Residuals:
      Min        1Q    Median        3Q       Max
-15323571  -2214856   -111228   3019320   7083693

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 49339612     498274   99.02   <2e-16 ***
index         114672       4299   26.67   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3510000 on 198 degrees of freedom
Multiple R-squared:  0.7823,     Adjusted R-squared:  0.7812
F-statistic: 711.5 on 1 and 198 DF,  p-value: < 2.2e-16
```

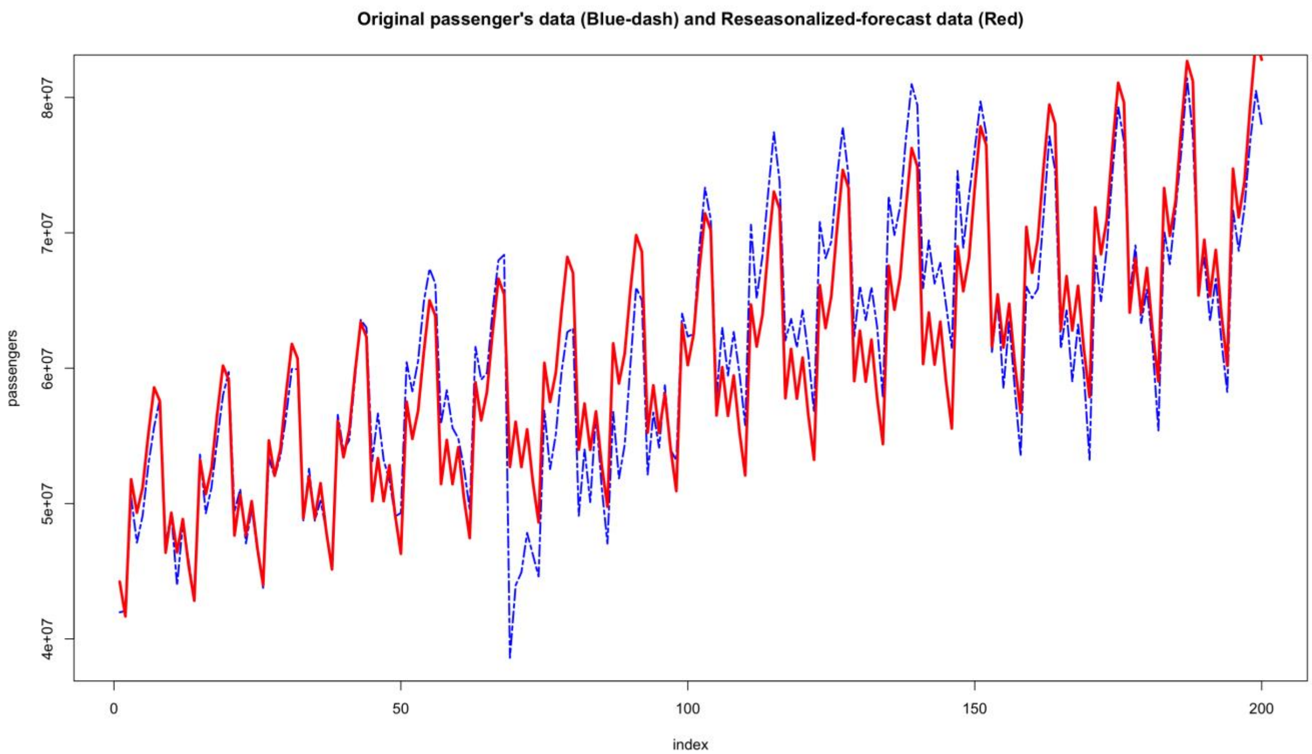


Interpretation: We can see that the p-value at the index variable is still very low (<5%), indicating that there is a relationship between the index and deseasonalized data. The new multiple R-squared value is greater than the previous one (0.78) indicating more explainability. An "r" value of 0.88; is also greater than the previous value, indicating a stronger positive linear relationship. The Beta value can be interpreted as: For every additional month in the future, we expect the number of deseasonalized value to increase by 114672.

7. Reseasonalize the fitted values for each of the two models. Construct a plot showing the original data and the re-seasonalized fitted values for each of the two regression models. Also, print the correlation between the passengers and re-seasoned attributes.
8. In a single plot and using the index value as the x, show a) the original passengers' data, and b) the re-seasonalized forecasts. Use contrasting colors and title the graph appropriately.

```
us_Passengers$deseason.forecast=model_2_des$fitted.values
for(i in 1:12){
  for(j in 1:nrow(us_Passengers)){
    if(i==us_Passengers$month[j]){

us_Passengers$reseason.forecast[j]=us_Passengers$deseason.forecast[j]*season
al_indices$index2[i]}
  }
}
attach(us_Passengers)
plot(index,passengers,type="l",pch=19, lwd=2, col="blue", lty="twodash",
main="Original passenger's data (Blue-dash) and Reseasonalized-forecast data
(Red)")
points(index,reseason.forecast, type="l",pch=19,col="red", lwd=3)
```

```
points(model_1.out$fitted.values, type="l", lwd=1, col="blue",
lty="twodash")
points(model_2_des$fitted.values, type="l", lwd=2, col="red")
cor(us_Passengers$passengers, model_1.out$fitted.values)
cor(deseason.index, model_2_des$fitted.values)
> cor(us_Passengers$passengers, model_1.out$fitted.values)
[1] 0.7098539
> cor(deseason.index, model_2_des$fitted.values)
[1] 0.8844753
us_Passengers[which.min(rstandard(model_1.out)),c(1,3)]
> us_Passengers[which.min(rstandard(model_1.out)),c(1,3)]
         date month
69 2001-09-01     9
```



Interpretation: In comparing the first model (using raw data) with the second model (using the seasonal index), we can see that the second model performs a more accurate forecast. There is still a pattern in the passenger's data, and the remarkable decline has been corrected. In the previous steps, we have noticed that "r" and "Multiple R-square" metrics have improved and now, by comparing them using correlation, we can see that our second model has a much better correlation than the first (0.71 in the first model and 0.88 in the second model). Using seasonal indexes in time series regression is an effective way to improve some of the most important metrics in cyclical pattern data. Finally, using which.min(), we see that the significant drop in the total number of passengers occurred in September of 2021 (World Trade Center 9-11), which makes sense.