INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
NPTEL
NPTEL ONLINE CERTIFICATION COURSE

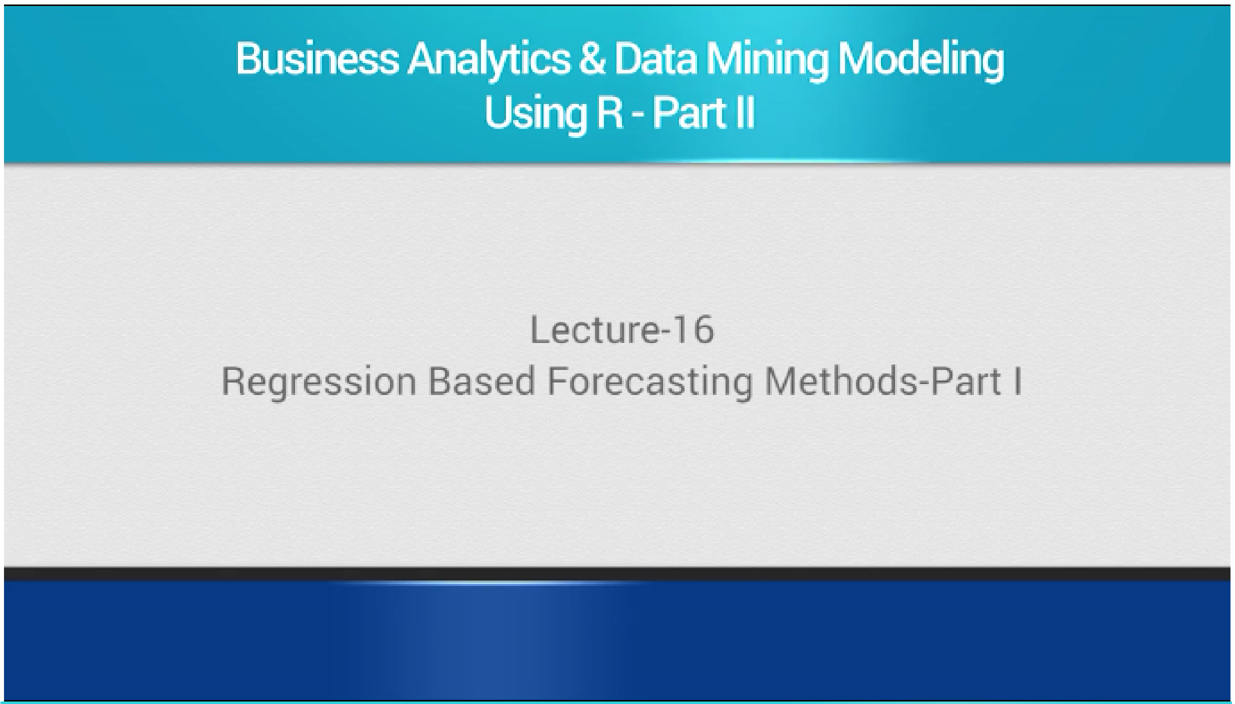
Business Analytics & Data Mining Modeling Using R – Part II

Lecture-16

Regression Based Forecasting Methods– Part I

With

Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology Roorkee

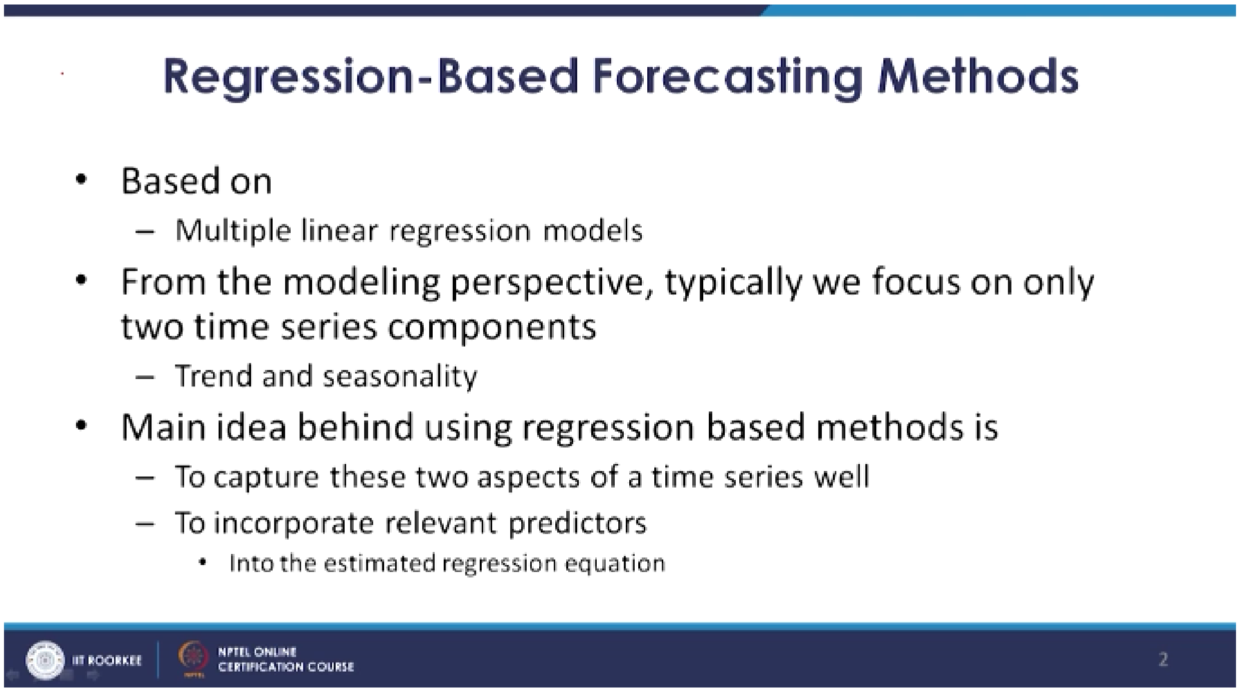


Business Analytics & Data Mining Modeling Using R - Part II

Lecture-16
Regression Based Forecasting Methods-Part I

With
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology Roorkee

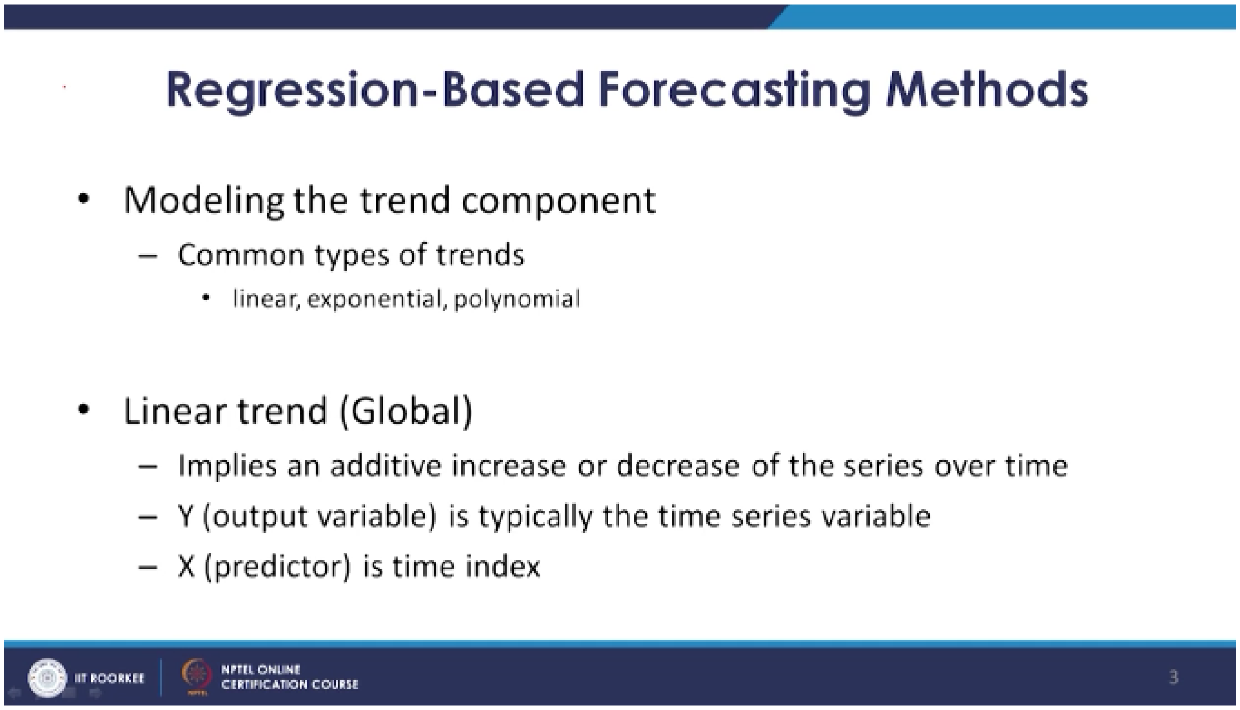
Welcome to the course Business Analytics and Data Mining Modeling Using R – Part 2, so in previous few lectures we were able to discuss our topic understanding time series, so we'll move to our next topic that is Regression Based Forecasting Methods.
So in this particular topic in coming few lectures and this lecture as well, we'll be discussing about how a regression you know based on, how regression models can actually be used to forecast a time series, so let's start our discussion.



Regression-Based Forecasting Methods

- Based on
  - Multiple linear regression models
- From the modeling perspective, typically we focus on only two time series components
  - Trend and seasonality
- Main idea behind using regression based methods is
  - To capture these two aspects of a time series well
  - To incorporate relevant predictors
    - Into the estimated regression equation

So as we can see in the first point itself based on multiple linear regression models if we look at from the modeling prospective, so typically we focus on only two time series components, trend and seasonality, so in our discussion in previous few lectures we talked about four time series components, level, trend, seasonality and noise, wherein we also discussed the importance of trend and seasonality, so in this particular topic, in this particular method regression based forecasting method we'll see that how trend and seasonality are important in terms of building models for forecasting future values, so why regression based forecasting methods are popular for predicting time series, so few points have been discussed here, main idea behind using regression based methods is to capture two aspects, these two aspects of a time series well in the trend and seasonality, so if we are able to you know make certain reasonable assumptions about the structure of data, you know the kind of trend that is there and seasonality that is there, so the seasonal pattern and the you know trend that is there, if you are able to make certain reasonable assumption about these two, then regression based forecasting method can actually be used to model, to capture these two aspect, and also to incorporate relevant predictors, since we are, if we use regression based forecasting methods, so there is always scope that if there is any really win predictor we can also include it in our regression modeling equations.



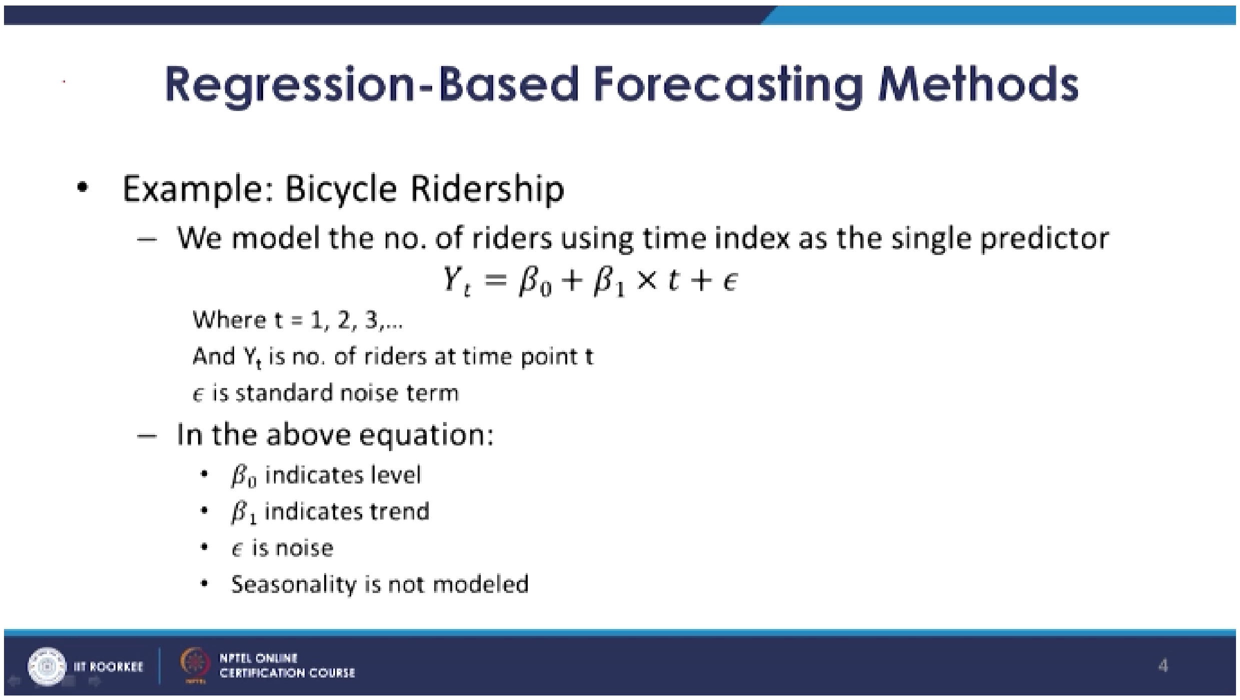
Regression-Based Forecasting Methods

- Modeling the trend component
  - Common types of trends
    - linear, exponential, polynomial
- Linear trend (Global)
  - Implies an additive increase or decrease of the series over time
  - Y (output variable) is typically the time series variable
  - X (predictor) is time index

IIT ROORKEE
NPTEL ONLINE CERTIFICATION COURSE
3

So let’s start with the you know first one, that is trend, so we’ll start our discussion on how we should go about modeling the trend component, so as we have discussed before also there are three common types of trends that we would be focusing on, first one is linear, then exponential and polynomial, so we’ll discuss each of these trends one by one, and we’ll see how you know these trends can actually be model using regression based methods.

So let’s start our discussion with the first one that is the linear trend, you would see in this point I have also mentioned global, so this is to indicate, this is to refer to the discussion that we had in previous lectures where we talked about that you know model driven methods or statistical methods or more you know appropriate when we have the you know presence of trend is global in nature, so we have global trend because you know if we assume certain structure about the data, then that should be applicable in the entire series, so therefore if the nature of the trend is global then probably regression based forecasting methods being statistical methods are going to be better performance, so let’s discuss how this can be model.



Regression-Based Forecasting Methods

- Example: Bicycle Ridership
  - We model the no. of riders using time index as the single predictor

$$Y_t = \beta_0 + \beta_1 \times t + \epsilon$$

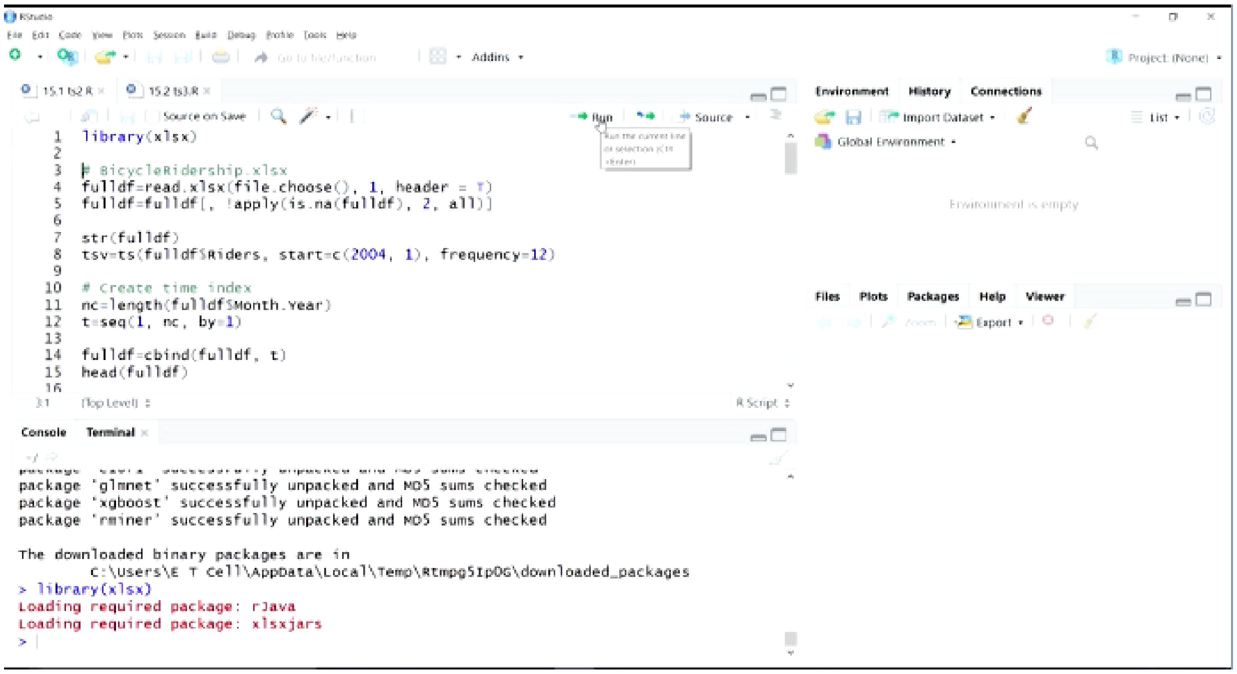
Where t = 1, 2, 3,...

And Y_t is no. of riders at time point t

ϵ is standard noise term

  - In the above equation:
    - β_0 indicates level
    - β_1 indicates trend
    - ϵ is noise
    - Seasonality is not modeled

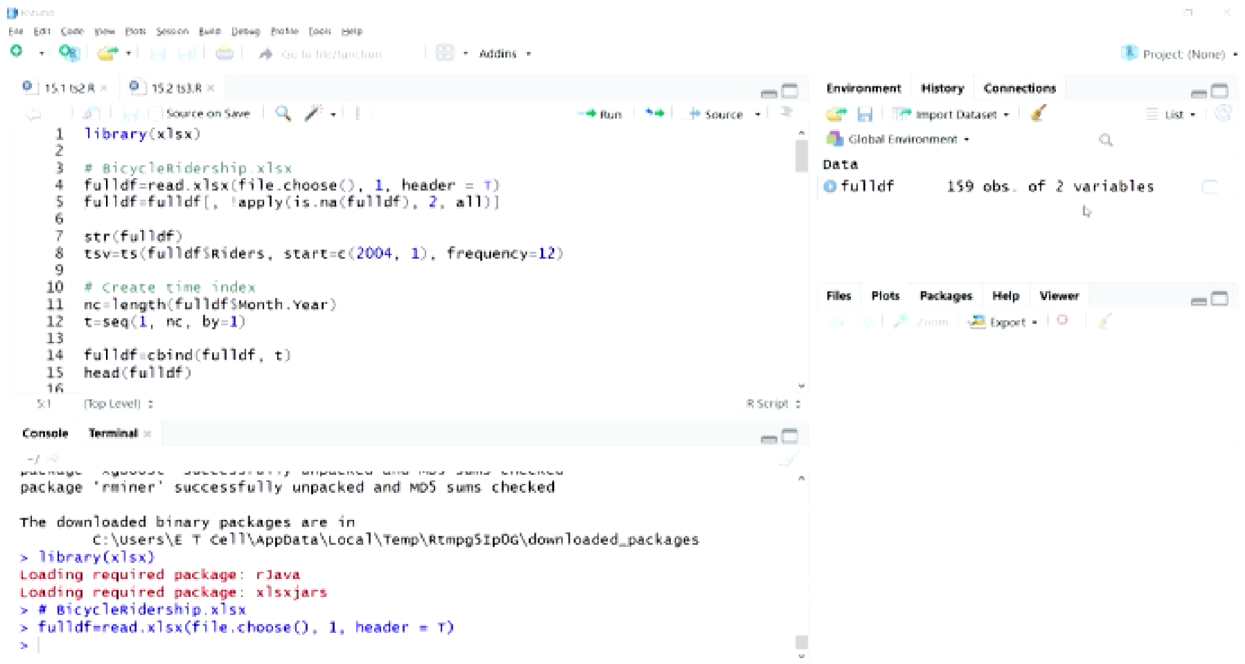
So implies and additive and increase or decrease of the series over time, so this is what we mean by linear trend and additive increase or decrease of the series over time, so depending on the slope, so line depending on the slope of the line so it is going to be either additive increase or decrease, so why output variable is typically the time series variable, for example the dataset that we are using bicycle ridership so it is going to be the number of riders that are there, so this is going to be the output variable because we have monthly data on ridership, so that is the series that we have, so Y is that number of riders that is the time series variable, and X is time index so essentially we would be regressingthis Y, number of riders on this time index variable, so let's see how the model equation can be written, so this is reference to the example that we have been using bicycle ridership and we can model the number of riders using time index as the single predictor so you can see here, regression equation here, YT, where YT is number of riders at time point, at time point T and then we have on the right hand side we have the beta 0 + beta 1T + epsilon, so where T is the numbers like 1, 2, 3 which is actually indicating the time index, so in this if you look at the, if we analyze this regression equation and we try to understand the different components of the time series how they have been modeled into this particular equation, so we can see here beta 0 is actually indicating the level, right so we talked about level and capturing the average values of a particular time series so in this case when we use this particular equation, when we use this particular regression equation then beta 0 is actually capturing the level of the series.



```r
library(xlsx)

# BicycleRidership.xlsx
fulldf=read.xlsx(file.choose(), 1, header = T)
fulldf=fulldf[, !apply(is.na(fulldf), 2, all)]

str(fulldf)
tsv=ts(fulldf$Riders, start=c(2004, 1), frequency=12)

# Create time index
nc=length(fulldf$Month.Year)
t=seq(1, nc, by=1)

fulldf=cbind(fulldf, t)
head(fulldf)
```

```
package 'glmnet' successfully unpacked and MD5 sums checked
package 'xgboost' successfully unpacked and MD5 sums checked
package 'rminer' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\E T Cell\AppData\Local\Temp\RtmpgSIpOG\downloaded_packages
> library(xlsx)
Loading required package: rJava
Loading required package: xlsxjars
> 
```

Then we talked about the beta 1, so beta 1 here is indicating the trend so here we have right now we are modeling the linear trend so we have assumed that a line could actually fit the time series and indicative of the kind of change in the series that is happening, and the change is represented by the slope of the line which is being captured by beta 1. Then we have the epsilon, so that is the invoice so that is the unaccounted part so in this from this equation and we can also say that you know we are assuming that the line has a linear trend and there is you know seasonality is not there, so you can see seasonality is not modeled here, we are just modeling the trend component, so level we can see beta 0, beta 1 is indicating the trend, seasonality is not modeled, and the fourth component is epsilon that is noise, so in this fashion we can see four component and how they are being modeled, so what'll do we'll go back to R studio environment and see how this particular modeling, this modeling of linear trend can actually be applied on bicycle ridership dataset that we have, so let's load this library XLSX, let's import this dataset, so you can see the data has been loaded into the environment section as we can see here 159 observations, 2 variables, so it is the same dataset that we have used in previous few lectures, so let's remove any you know deleted columns in XL.

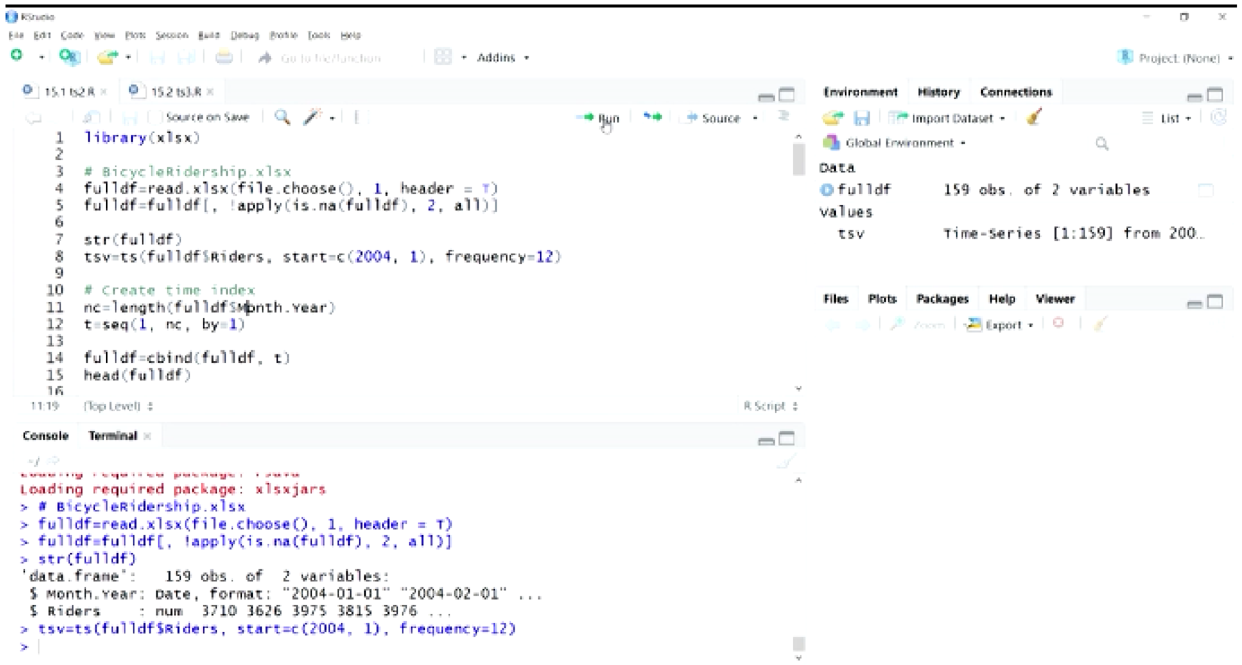


```r
library(xlsx)

# BicycleRidership.xlsx
fulldf=read.xlsx(file.choose(), 1, header = T)
fulldf=fulldf[, !apply(is.na(fulldf), 2, all)]

str(fulldf)
tsv=ts(fulldf$Riders, start=c(2004, 1), frequency=12)

# Create time index
nc=length(fulldf$Month.Year)
t=seq(1, nc, by=1)

fulldf=cbind(fulldf, t)
head(fulldf)
```

```
package 'rminer' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\E T Cell\AppData\Local\Temp\RtmpgSIpOG\downloaded_packages
> library(xlsx)
Loading required package: rJava
Loading required package: xlsxjars
> # BicycleRidership.xlsx
> fulldf=read.xlsx(file.choose(), 1, header = T)
>
```

```
Data
fulldf      159 obs. of 2 variables
```

Now let’s look at the structure of this dataset, so we have just two variables, one indicating the time information that is month and year, and then we have the second variable which is riders, number of riders for each of those months, so what we are going to do is just like in previous lecture we will create a time series objects, so for this we are going to create this TSV time series vector, we’ll call this function TS first argument is going to be the time series variable which is the riders, so we’re essentially capturing the number of riders for each month for the given period, so the first argument is the XLSX variable riders and then we are mentioning here, so as you can see in the previous lecture when we created this time series object, we had mentioned the argument start and end and we can see so at that time I had also indicated that end and frequency you know we only need to use, one of those arguments and the other one will be taken care off, given that we specify the values in the appropriate format, for more information you can always refer the help section for this function TS, so in this case as you can see I’m just mentioning the starting point, and the frequency, so here frequency is actually telling me that in one unit of time, so which is here you know assume to be 1 year, how many you know data points are available, so this is monthly series on ridership, so therefore there are 12 for each of those months we have number of riders, information available, so the series is going to be this particular object is going to be created accordingly, so let’s run this.

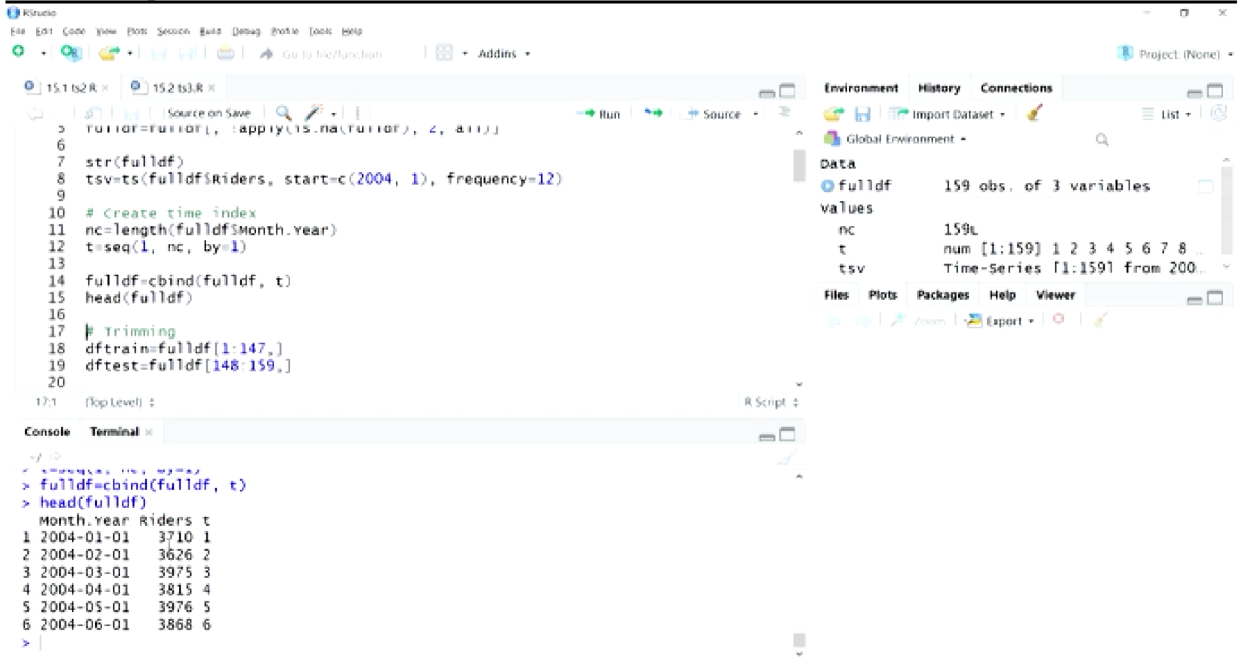


```
library(xlsx)

# BicycleRidership.xlsx
fulldf=read.xlsx(file.choose(), 1, header = T)
fulldf=fulldf[, !apply(is.na(fulldf), 2, all)]

str(fulldf)
tsv=ts(fulldf$Riders, start=c(2004, 1), frequency=12)

# Create time index
nc=length(fulldf$Month.Year)
t=seq(1, nc, by=1)

fulldf=cbind(fulldf, t)
head(fulldf)
```

```
Data
fulldf       159 obs. of 2 variables
Values
  tsv        Time-Series [1:159] from 200…
```

```
Loading required package: xlsxjars
> # BicycleRidership.xlsx
> fulldf=read.xlsx(file.choose(), 1, header = T)
> fulldf=fulldf[, !apply(is.na(fulldf), 2, all)]
> str(fulldf)
'data.frame':   159 obs. of  2 variables:
 $ Month.Year: Date, format: "2004-01-01" "2004-02-01" ...
 $ Riders    : num  3710 3626 3975 3815 3976 ...
> tsv=ts(fulldf$Riders, start=c(2004, 1), frequency=12)
>
```

So you can see time series vector has been created 159 observations, now what we are going to do is we'll create our predictor, the single predictor that we have that is time index, so time index let's first capture the number of observation in this time series vector which is going to be equal to the number of you know index values that we will have, so let's run this, and then we'll create a sequence for this variable T.



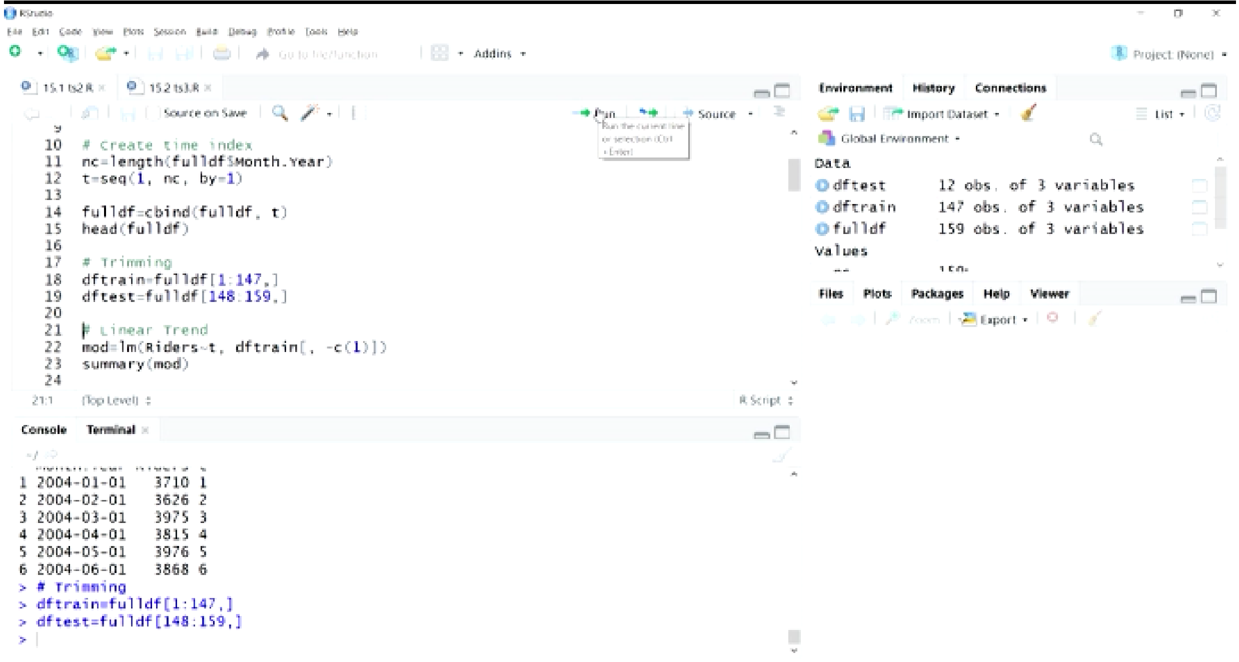
```
fulldf=fulldf[, !apply(is.na(fulldf), 2, all)]

str(fulldf)
tsv=ts(fulldf$Riders, start=c(2004, 1), frequency=12)

# Create time index
nc=length(fulldf$Month.Year)
t=seq(1, nc, by=1)

fulldf=cbind(fulldf, t)
head(fulldf)

# Trimming
dftrain=fulldf[1:147,]
dftest=fulldf[148:159,]
```

```
Data
fulldf       159 obs. of 3 variables
Values
  nc         159L
  t          num [1:159] 1 2 3 4 5 6 7 8 …
  tsv        Time-Series [1:159] from 200…
```

```
> fulldf=cbind(fulldf, t)
> head(fulldf)
  Month.Year Riders t
1 2004-01-01   3710 1
2 2004-02-01   3626 2
3 2004-03-01   3975 3
4 2004-04-01   3815 4
5 2004-05-01   3976 5
6 2004-06-01   3868 6
>
```

Now let's append this variable into this existing data frame, now let's have a look at first 6 observations, so as we can see here, first 6 observation we can see, second column riders, and the third column is having T which is the time index, so as you can see the values of it 1, 2, 3, 4, 5, 6, so in this fashion we are going to use it.

So next step is going to be about trimming the time series, so we'll trim the time, original time series that we have just you know created, and the earlier period is going to be used as training set and the later period is going to be used as a test set, so you can see I'm considering first 147 observation as part of training set and remaining 12 observations as part of test set, so let's go forward with this.

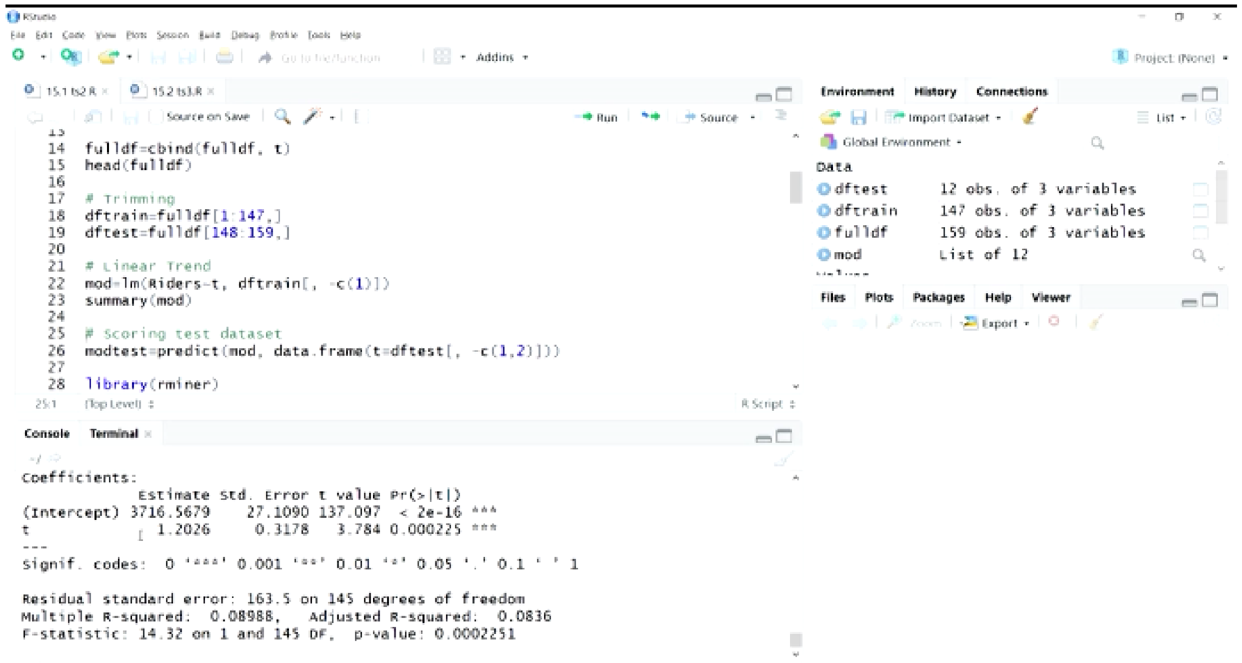


```
# Create time index
nc=length(fulldf$Month.Year)
t=seq(1, nc, by=1)

fulldf=cbind(fulldf, t)
head(fulldf)

# Trimming
dftrain=fulldf[1:147,]
dftest=fulldf[148:159,]

# Linear Trend
mod=lm(Riders~t, dftrain[, -c(1)])
summary(mod)
```

```
1 2004-01-01   3710 1
2 2004-02-01   3626 2
3 2004-03-01   3975 3
4 2004-04-01   3815 4
5 2004-05-01   3976 5
6 2004-06-01   3868 6
> # Trimming
> dftrain=fulldf[1:147,]
> dftest=fulldf[148:159,]
> 
```

```
Data
dftest      12 obs. of 3 variables
dftrain     147 obs. of 3 variables
fulldf      159 obs. of 3 variables
Values
```

So these two sets have been created, now to fit a linear trend because we are going to use regression based forecasting, so you can see I'm calling the same function that we had used when we discussed you know multiple linear regression in our previous course, so in this particular function is being you know used being called here and the first argument is the formula, the model equation that if you're, so we are regressing riders that is number of riders on T, that is the time index and the appropriate data is also has been specified in the second argument, so let's run this.

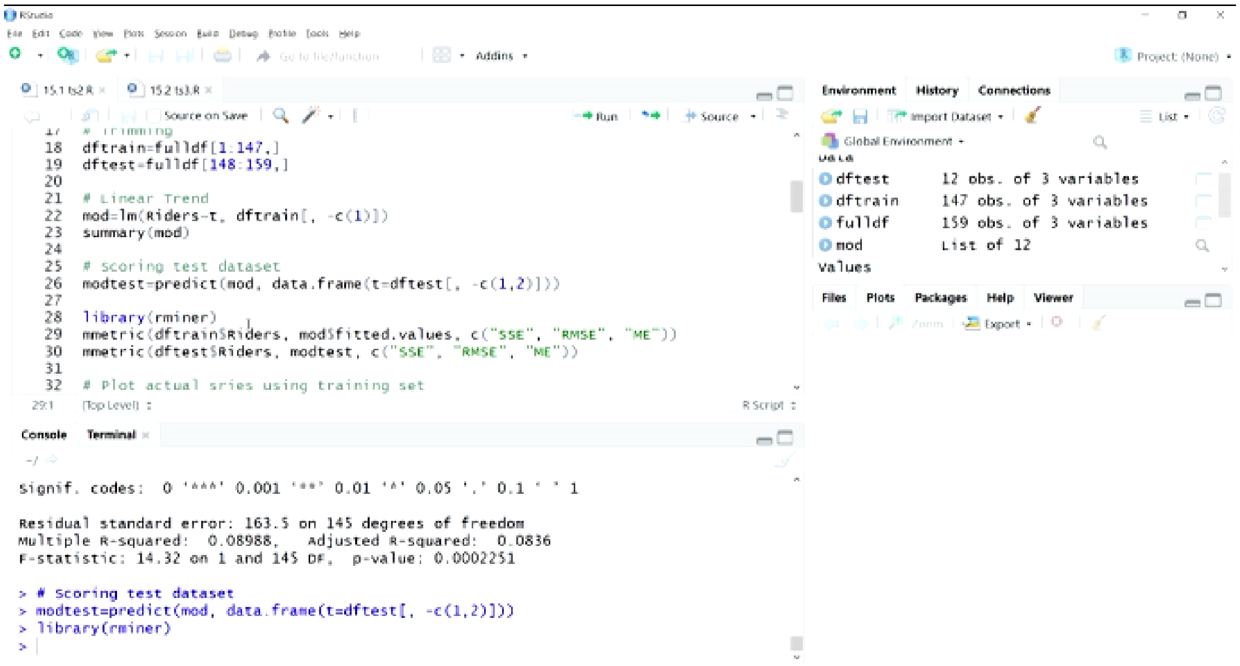


```r
fulldf=cbind(fulldf, t)
head(fulldf)

# Trimming
dftrain=fulldf[1:147,]
dftest=fulldf[148:159,]

# Linear Trend
mod=lm(Riders~t, dftrain[, -c(1)])
summary(mod)

# Scoring test dataset
modtest=predict(mod, data.frame(t=dftest[, -c(1,2)]))

library(rminer)
```

```
Data
dftest      12 obs. of 3 variables
dftrain     147 obs. of 3 variables
fulldf      159 obs. of 3 variables
mod         List of 12
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 3716.5679    27.1090 137.097  < 2e-16 ***
t              1.2026     0.3178   3.784 0.000225 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 163.5 on 145 degrees of freedom
Multiple R-squared:  0.08988,	Adjusted R-squared:  0.0836 
F-statistic: 14.32 on 1 and 145 DF,  p-value: 0.0002251
```

Now let's analyze the results, so you can see both intercept and this single predictor T both are significant, the model is also significant as indicated by other statistics given in this result, so there seem to be the model significant, the value of the estimated coefficient is 1.2 for this single predictor T, so significance of this solo linear trend we can say that this linear trend is significant and it is fitting to the, it can be fit to the series, but we will have to check for the performance, right, so what we'll do is now we will score the test dataset that we have created to be able to compare the performance of this linear trend model on training on test, so let's call this function predict, first argument is the model object that we have just created and we'll pass on the test dataset, so let's conclude this, so it has been computed, now we are going to compute certain matrix like we use to do in previous course, so as we have talked about in previous few lectures that the matrix that we are going to use in time series forecasting are seeing like we used in supervise learning techniques module, so this is the package that we typically use library or minor, so this package has this function M metric which can be used to compute values for different matrix like SSE, RMSE, ME so we are going to focus on these three matrix in particular, so let's load this particular package.

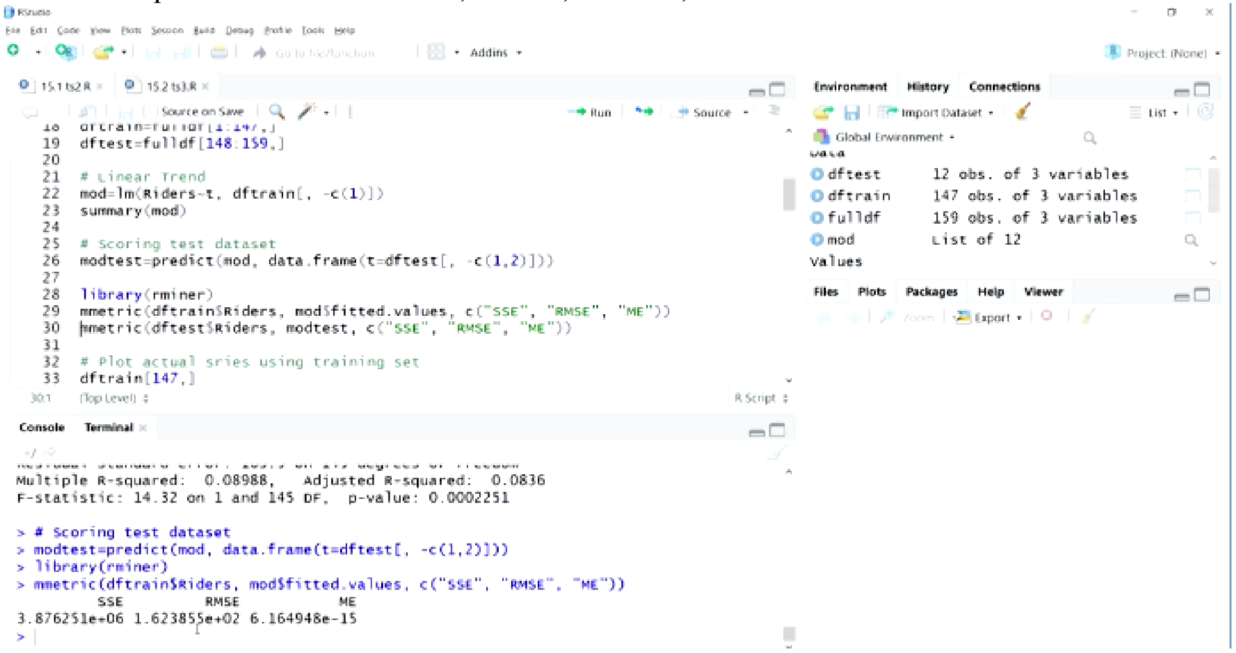


```r
# Trimming
dftrain=fulldf[1:147,]
dftest=fulldf[148:159,]

# Linear Trend
mod=lm(Riders~t, dftrain[, -c(1)])
summary(mod)

# Scoring test dataset
modtest=predict(mod, data.frame(t=dftest[, -c(1,2)]))

library(rminer)
mmetric(dftrain$Riders, mod$fitted.values, c("SSE", "RMSE", "ME"))
mmetric(dftest$Riders, modtest, c("SSE", "RMSE", "ME"))

# Plot actual sries using training set
```

```
dftest      12 obs. of 3 variables
dftrain     147 obs. of 3 variables
fulldf      159 obs. of 3 variables
mod         List of 12
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 163.5 on 145 degrees of freedom
Multiple R-squared:  0.08988,	Adjusted R-squared:  0.0836 
F-statistic: 14.32 on 1 and 145 DF,  p-value: 0.0002251

> # Scoring test dataset
> modtest=predict(mod, data.frame(t=dftest[, -c(1,2)]))
> library(rminer)
> 
```

So now let's call this function M metric, so in the first argument as you can see we are passing on the, this is being applied on the training set and we are using the first you know first argument is about the riders, number of riders that are there, then the fitted values, so the values which have been forecasted by the model, the linear trend model that we have, and we'll be able to compute these 3 metric SSE, RMSE, and ME, so let's run this.

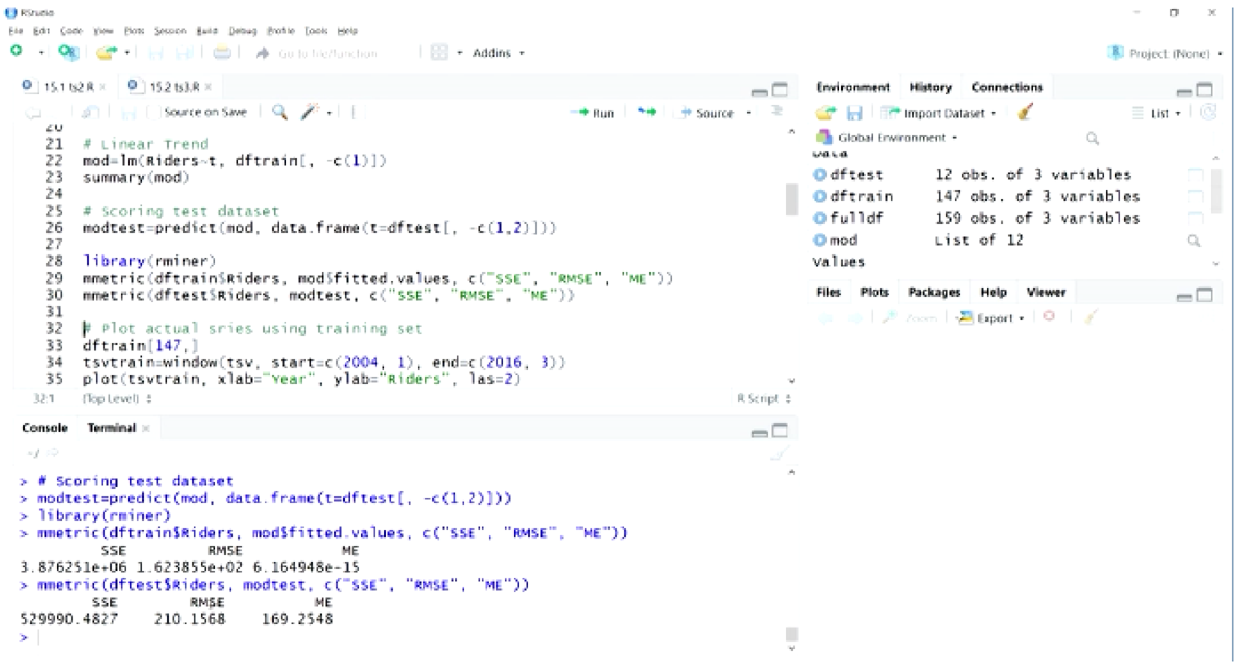


```r
dftrain=fulldf[1:147,]
dftest=fulldf[148:159,]

# Linear Trend
mod=lm(Riders~t, dftrain[, -c(1)])
summary(mod)

# Scoring test dataset
modtest=predict(mod, data.frame(t=dftest[, -c(1,2)]))

library(rminer)
mmetric(dftrain$Riders, mod$fitted.values, c("SSE", "RMSE", "ME"))
mmetric(dftest$Riders, modtest, c("SSE", "RMSE", "ME"))

# Plot actual sries using training set
dftrain[147,]
```

```
dftest      12 obs. of 3 variables
dftrain     147 obs. of 3 variables
fulldf      159 obs. of 3 variables
mod         List of 12
```

```
Multiple R-squared:  0.08988,	Adjusted R-squared:  0.0836 
F-statistic: 14.32 on 1 and 145 DF,  p-value: 0.0002251

> # Scoring test dataset
> modtest=predict(mod, data.frame(t=dftest[, -c(1,2)]))
> library(rminer)
> mmetric(dftrain$Riders, mod$fitted.values, c("SSE", "RMSE", "ME"))
         SSE         RMSE           ME 
3.876251e+06 1.623855e+02 6.164948e-15 
> 
```

So we can see the numbers here RMSE value and ME value, so the error seems to be about 162 there, now let's look at the performance numbers of test partition, so here also first argument is as you can see here we are applying this on test partition, number of riders is the first argument, the second argument is the scored values using the model that we have built using linear trend and the same matrix, so let's run this, you can see the numbers, now if we compare these numbers then we can see the RMSE value for the training partition were the 162 and these

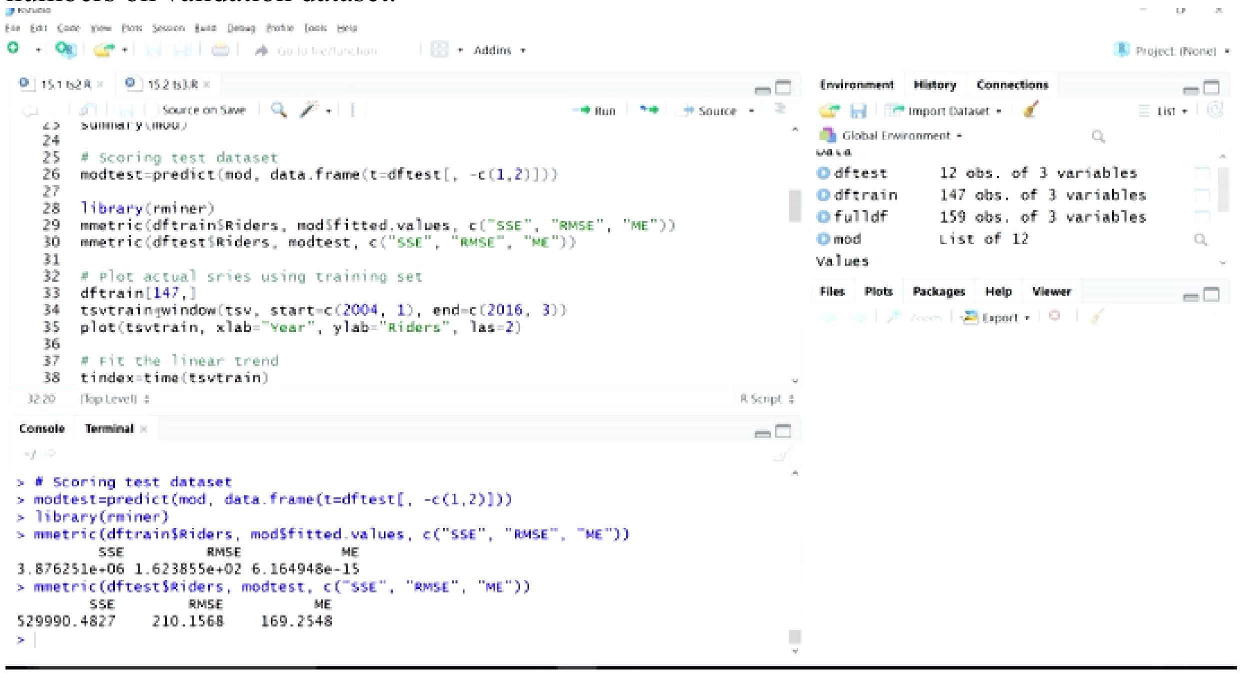


```
# Linear Trend
mod=lm(Riders~t, dftrain[, -c(1)])
summary(mod)

# Scoring test dataset
modtest=predict(mod, data.frame(t=dftest[, -c(1,2)]))

library(rminer)
mmetric(dftrain$Riders, mod$fitted.values, c("SSE", "RMSE", "ME"))
mmetric(dftest$Riders, modtest, c("SSE", "RMSE", "ME"))

# Plot actual sries using training set
dftrain[147,]
tsvtrain=window(tsv, start=c(2004, 1), end=c(2016, 3))
plot(tsvtrain, xlab="Year", ylab="Riders", las=2)
```

```
Data
dftest      12 obs. of 3 variables
dftrain     147 obs. of 3 variables
fulldf      159 obs. of 3 variables
mod         List of 12
Values
```

```
> # Scoring test dataset
> modtest=predict(mod, data.frame(t=dftest[, -c(1,2)]))
> library(rminer)
> mmetric(dftrain$Riders, mod$fitted.values, c("SSE", "RMSE", "ME"))
         SSE         RMSE           ME
3.876251e+06 1.623855e+02 6.164948e-15
> mmetric(dftest$Riders, modtest, c("SSE", "RMSE", "ME"))
        SSE        RMSE          ME
529990.4827    210.1568    169.2548
>
```

RMSE value for the test partition is 210, so there is significant you know difference in these two value, so the model doesn't seem to be performing well on the validation partition, so even though the model was found to be significant, the single predictor that we had used T, that is time index was that relationship between the riders and T was found to be significant, however looking at the numbers the performance number, these value RMSE values we can see that, the model is not doing linear trend, model is not doing a good job of predicting the ridership numbers on validation dataset.

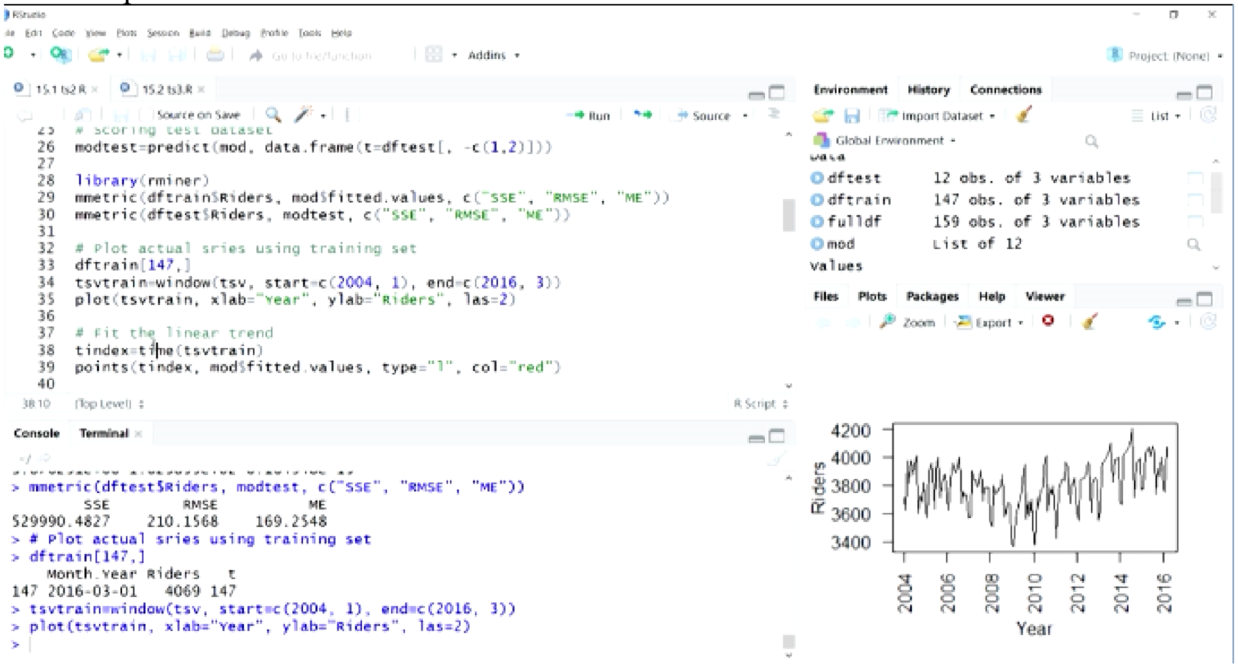


```
summary(mod)

# Scoring test dataset
modtest=predict(mod, data.frame(t=dftest[, -c(1,2)]))

library(rminer)
mmetric(dftrain$Riders, mod$fitted.values, c("SSE", "RMSE", "ME"))
mmetric(dftest$Riders, modtest, c("SSE", "RMSE", "ME"))

# Plot actual sries using training set
dftrain[147,]
tsvtrain=window(tsv, start=c(2004, 1), end=c(2016, 3))
plot(tsvtrain, xlab="Year", ylab="Riders", las=2)

# Fit the linear trend
tindex=time(tsvtrain)
```

```
Data
dftest      12 obs. of 3 variables
dftrain     147 obs. of 3 variables
fulldf      159 obs. of 3 variables
mod         List of 12
Values
```

```
> # Scoring test dataset
> modtest=predict(mod, data.frame(t=dftest[, -c(1,2)]))
> library(rminer)
> mmetric(dftrain$Riders, mod$fitted.values, c("SSE", "RMSE", "ME"))
         SSE         RMSE           ME
3.876251e+06 1.623855e+02 6.164948e-15
> mmetric(dftest$Riders, modtest, c("SSE", "RMSE", "ME"))
        SSE        RMSE          ME
529990.4827    210.1568    169.2548
>
```

Similarly if we look at the third matrix that is ME, there also if we look at the training partition the value is near about 0, we can see here, this value is closer to 0, if we look at the ME value for the test partition it is about 169, so the model is you know over predicting by a significant high margin, so this doesn't seem to be an educate fit, so let's now analyze the same thing using the visual inspection like we talked about in previous lecture that visual inspection of time plots and other techniques that we talked about that is going to be very important in terms of analyzing whether a model is educate enough or not, so we are going to plot create few graphics for the training set, so last observation for the training set was this one so this value was March 2016 so now you can see we are creating a new time series object rather subsiding any new time series object from TSV which started from 2004, just like the original time series vector and ends at this point which is also the end point for the training dataset that we have, so let's run this, we'll get this series, training dataset series let's plot this series, so we can see here this has been plotted.

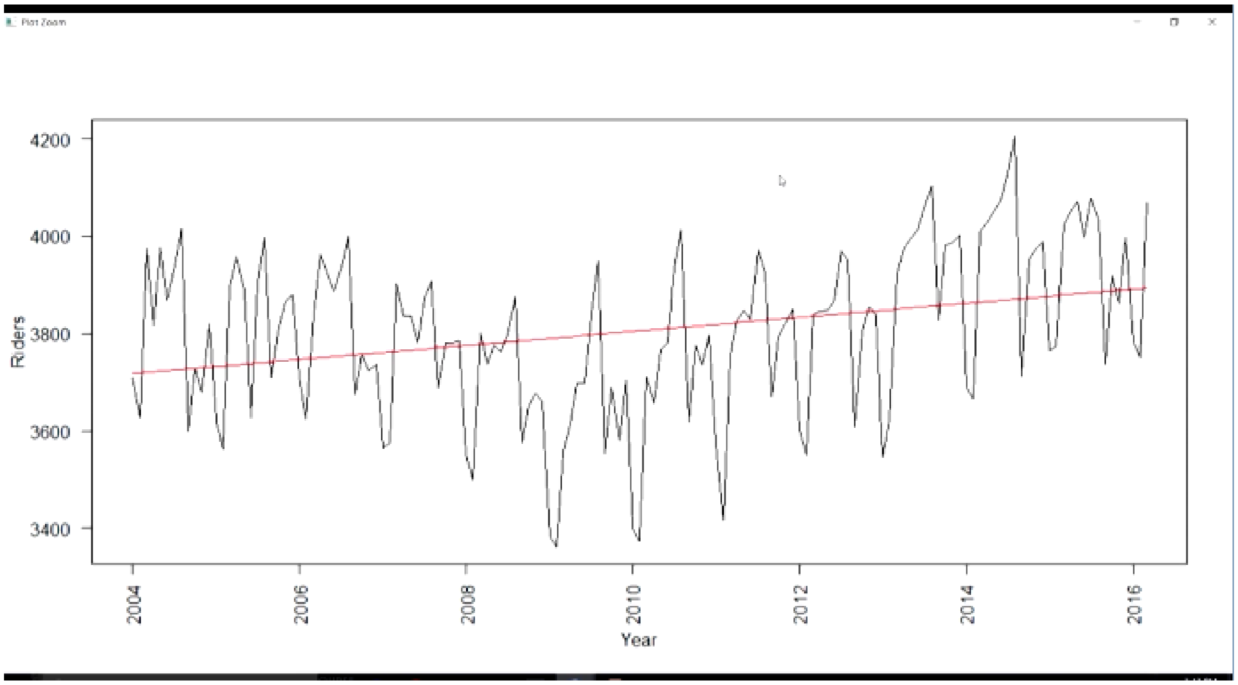


```
# scoring test dataset
modtest=predict(mod, data.frame(t=dftest[, -c(1,2)]))

library(rminer)
mmetric(dftrain$Riders, mod$fitted.values, c("SSE", "RMSE", "ME"))
mmetric(dftest$Riders, modtest, c("SSE", "RMSE", "ME"))

# Plot actual sries using training set
dftrain[147,]
tsvtrain=window(tsv, start=c(2004, 1), end=c(2016, 3))
plot(tsvtrain, xlab="Year", ylab="Riders", las=2)

# Fit the linear trend
tindex=time(tsvtrain)
points(tindex, mod$fitted.values, type="l", col="red")
```

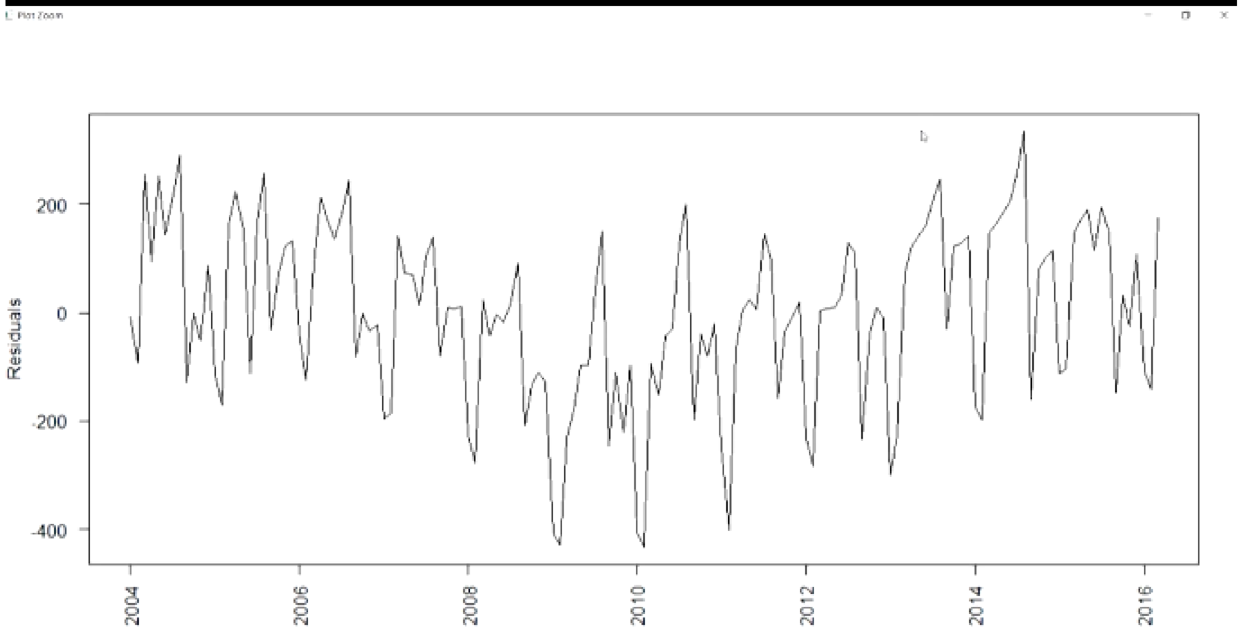
```
dftest      12 obs. of 3 variables
dftrain     147 obs. of 3 variables
fulldf      159 obs. of 3 variables
mod         List of 12
values
```

```
> mmetric(dftest$Riders, modtest, c("SSE", "RMSE", "ME"))
       SSE        RMSE          ME
529990.4827    210.1568    169.2548
> # Plot actual sries using training set
> dftrain[147,]
    Month.Year Riders   t
147 2016-03-01   4069 147
> tsvtrain=window(tsv, start=c(2004, 1), end=c(2016, 3))
> plot(tsvtrain, xlab="Year", ylab="Riders", las=2)
>
```

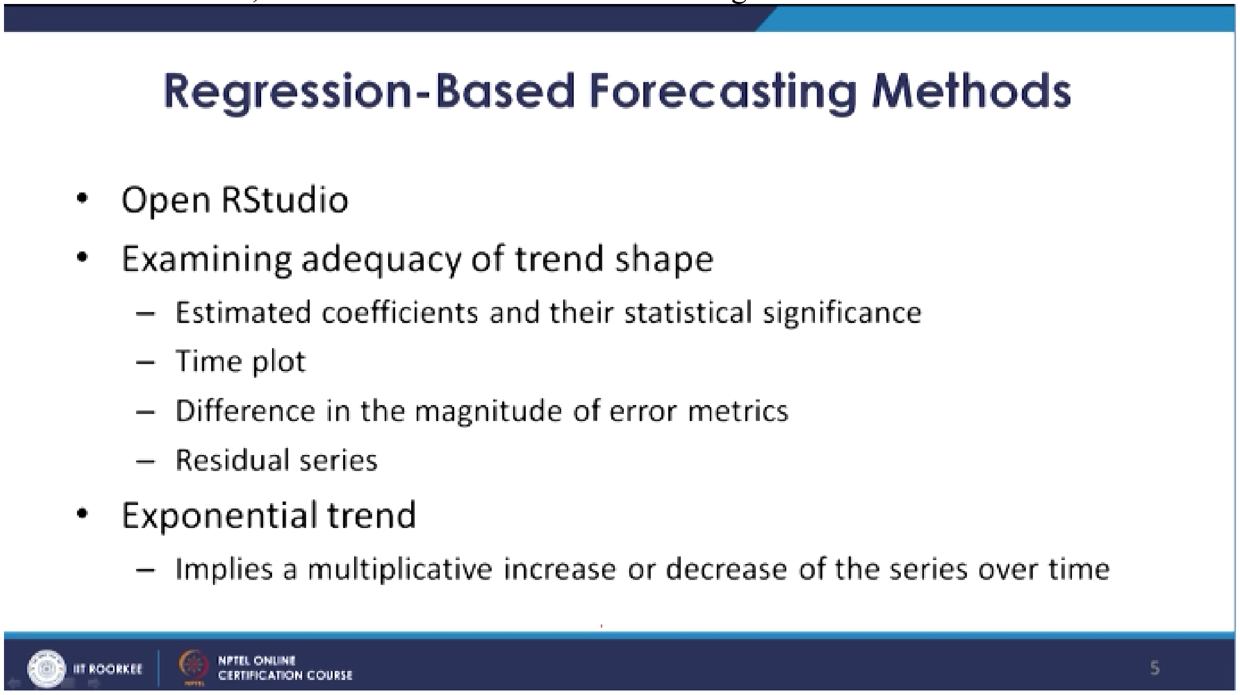
Now the linear trend model that we have just fitted, we are going to plot this particular fitted model, so first we'll extract the time index from this you know time series object because that is going to be the you know later on we can see in the second line, we are calling this points function which is actually going to create a line for us because you can see here third argument we have specified type as L which is actually the line segments, so using this particular function, point function and the coordinates T index indicating X coordinate and the second mod fitted values indicating the Y coordinate we are going to plot the fitted models, so let's run this two lines, now let's call this you can see because the colour we had specified as red, so if we zoom in to the model that zoom in to the plot that we have just created, so this was the



original in the black, we can see the original you know training set series and the linear trend model that we have just fitted we can see in the red colour it is just going right through it and we can see here from the visual inspection of the time series itself we can see that time series looks to be following more of a U trend, U shaped trend however what we have done we have fitted a linear line, so even though it was found to be you know significant which is also quite obvious from this plot also, so but it is not adequately capturing the series trend, so the same can also be confirmed by plotting residual series so that is the next thing that we are going to do, so let's create another plot, now this plot is the residual series plot, so the model that we have just build linear trend model if we create this plot we can see we are creating a time series object out of residuals and the same is being plotted, so let's run this, so this is the plot, now if



we zoom into this plot we can see that despite you know the model and the prediction that has happened, the model was linear trend model was found to be significant, if we look at the residual series clearly the pattern that was visible in the original series the same pattern is still visible in the residual series, so therefore this pattern has not been adequately captured by the linear trend models, so therefore we need to do something else.

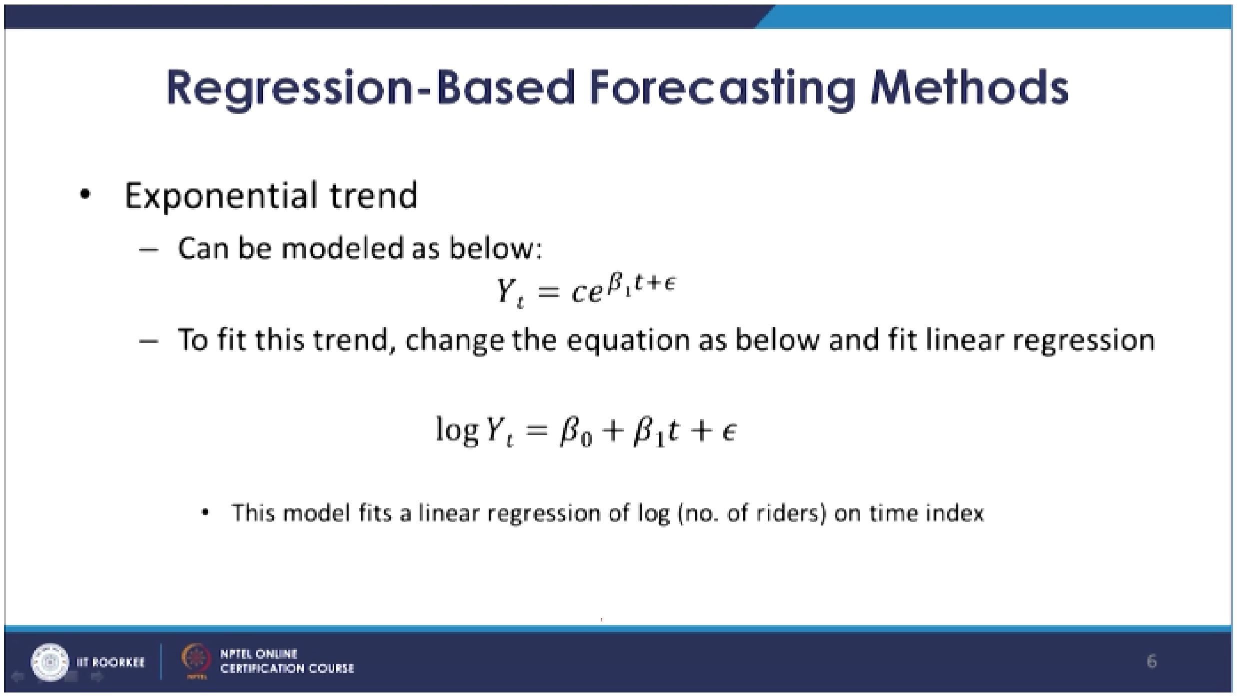


Regression-Based Forecasting Methods

- Open RStudio
- Examining adequacy of trend shape
  - Estimated coefficients and their statistical significance
  - Time plot
  - Difference in the magnitude of error metrics
  - Residual series
- Exponential trend
  - Implies a multiplicative increase or decrease of the series over time

IIT ROORKEE
NPTEL ONLINE CERTIFICATION COURSE
5

So let’s go back to our discussion, so as we talked about the adequacy of trend shapes so we saw that linear trend model was found to be significant, but whether it was adequate trend shape or not, so these are few things that can be done to find this out, estimated coefficient and their statistical significance so that indicated that it should be adequate then from the time plot we could see that the fitted you know when we plotted the fitted model you know there seems to be you know it was not seem to be you know adequate fit, then we looked at the difference in the magnitude of error matrix, so RMSE and average ME those numbers clearly indicated that the model was not doing good on a new set that is validation set, and in the residual series also we saw the shape that was there in the original series was a still part of the residual series, so therefore model was not able to adequately capture it.



Regression-Based Forecasting Methods

- Exponential trend
  - Can be modeled as below:

$$Y_t = ce^{\beta_1 t+\epsilon}$$

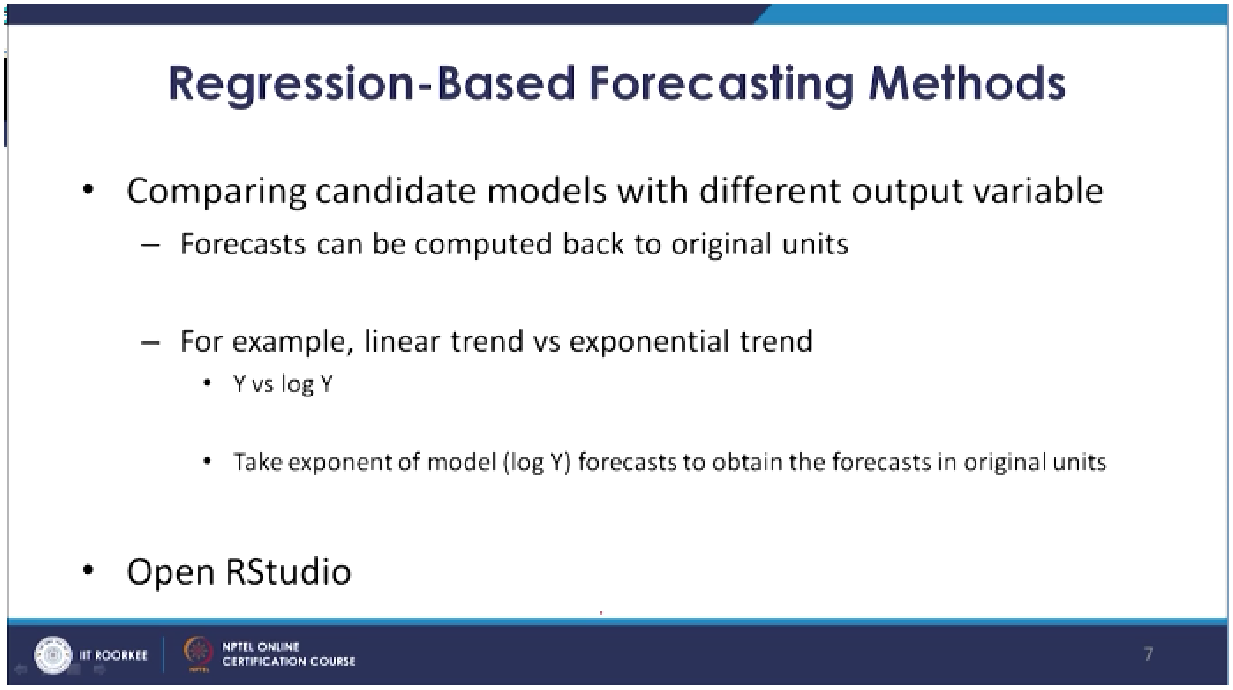
  - To fit this trend, change the equation as below and fit linear regression

$$\log Y_t = \beta_0 + \beta_1 t + \epsilon$$

    - This model fits a linear regression of log (no. of riders) on time index

Now the next trend shape that we are going to model is the exponential trend, so what we mean by exponential trend, so this typically implies a multiplicative increase or decrease of the series over time, so how we can express this in the mathematical form, so we can see here YT and we can see C multiplied by E to the power beta 1T + epsilon, so this is how we can express the exponential trend, if we want to fit this trend using the linear regression model we can rewrite, we can take log of this equation and rewrite it in this fashion, log YT beta 0 + beta 1T + epsilon, so now we can see that this model fits a linear regression of log of number of riders on time index, so therefore the output variable will change instead of Y, now it is going to be log of Y and the R just part is going to be remain same where we have this single predictor that is time index.

So since the output variable has changed, so because of this if we want to compare the results of linear trend and exponential trend so we need to do certain you know, we need to do certain computations before that, so few points are mentioned here so comparing candidate models with different output variables, so we are comparing linear trend model with the exponential trend model, linear having the output variable Y and the exponential model having the you know output variable as log of Y, so forecast can be you know forecast for example exponential trend model those forecast can be completed back to original units, so once that is done then we



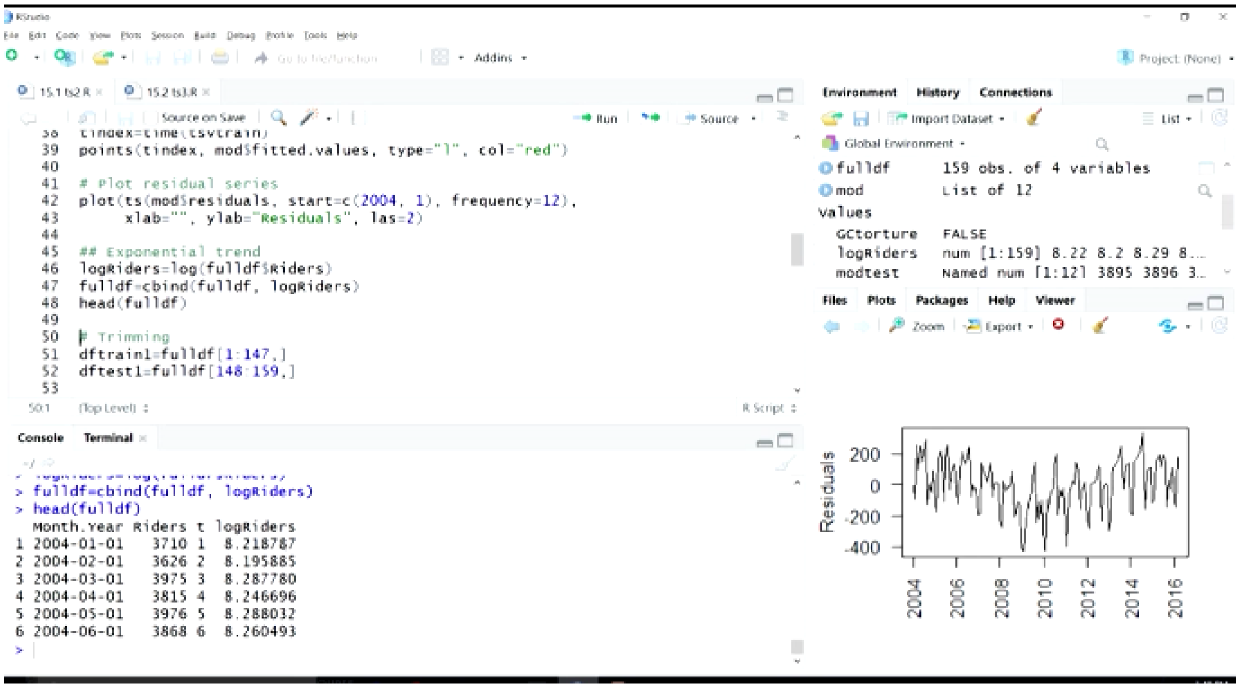
Regression-Based Forecasting Methods

- Comparing candidate models with different output variable
  - Forecasts can be computed back to original units
  - For example, linear trend vs exponential trend
    - Y vs log Y
    - Take exponent of model (log Y) forecasts to obtain the forecasts in original units
- Open RStudio

IIT ROORKEE | NPTEL ONLINE CERTIFICATION COURSE 7

can of course compare the results of the model, so as you can see here linear trend versus exponential trend if these are the two you know candidate models so essentially the output variables is going to be Y versus log Y, so we can take exponent of you know model forecast for this one log Y model and we can obtain the forecast in original units, and once that is done then the numbers that we compute after that they would, they can be compared and we can compare the performance of these two models.

So let's go back to R studio and let see how this can be done, so first we need to compute this you know output variable that is log Y, that is log riders so you can see here we are just computing first line, we are computing we are taking log of this value and we'll compute this variable then we'll append it to the existing data frame, so let's have a look at the first 6 observations, now we can see here another column has been added, log riders and the values are there.

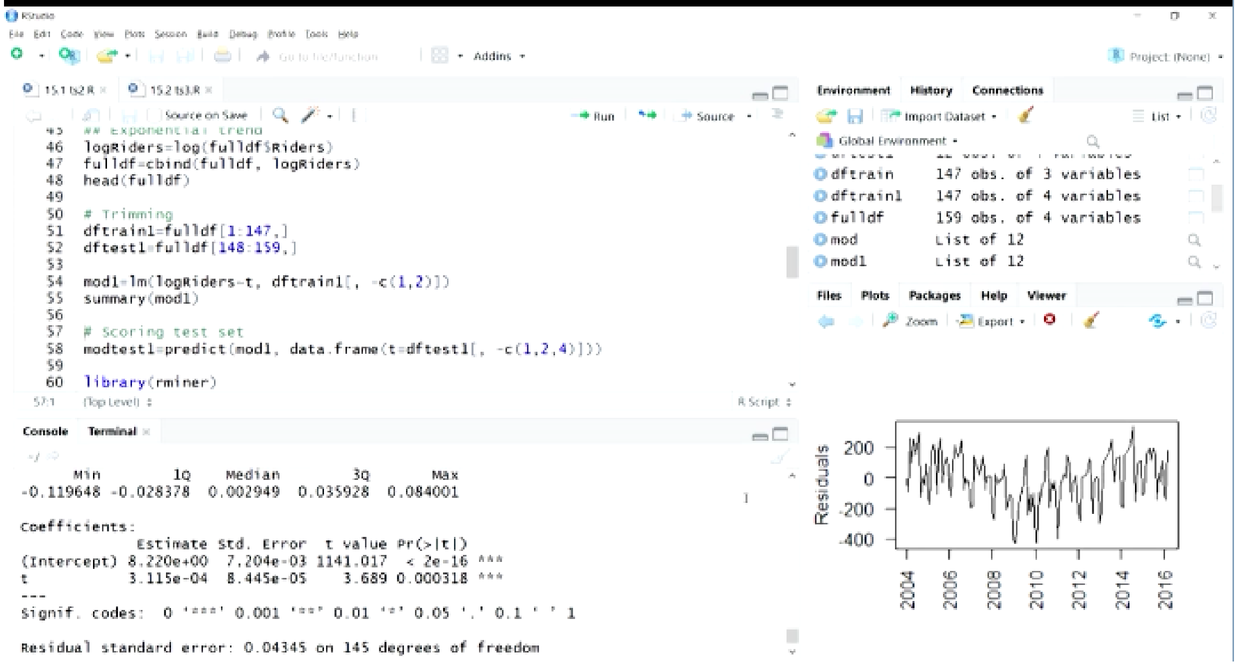


```
tindex=time(tsvtrain)
points(tindex, mod$fitted.values, type="l", col="red")

# Plot residual series
plot(ts(mod$residuals, start=c(2004, 1), frequency=12),
     xlab="", ylab="Residuals", las=2)

## Exponential trend
logRiders=log(fulldf$Riders)
fulldf=cbind(fulldf, logRiders)
head(fulldf)

# Trimming
dftrain1=fulldf[1:147,]
dftest1=fulldf[148:159,]
```

```
> fulldf=cbind(fulldf, logRiders)
> head(fulldf)
  Month.Year Riders t logRiders
1 2004-01-01   3710 1  8.218787
2 2004-02-01   3626 2  8.195885
3 2004-03-01   3975 3  8.287780
4 2004-04-01   3815 4  8.246696
5 2004-05-01   3976 5  8.288032
6 2004-06-01   3868 6  8.260493
>
```

```
fulldf      159 obs. of 4 variables
mod         List of 12
Values
  GCtorture   FALSE
  logRiders   num [1:159] 8.22 8.2 8.29 8...
  modtest     Named num [1:12] 3895 3896 3...
```

Now we are going to repeat the same process, first we'll trim the dataset, so let's run this, training set and the test set, now if we look at the you know model equation we are again calling alum function and now instead of you know riders that we had used in modeling linear trend, now we are using log riders, and then regression it on T, so let's run this code. Let's have

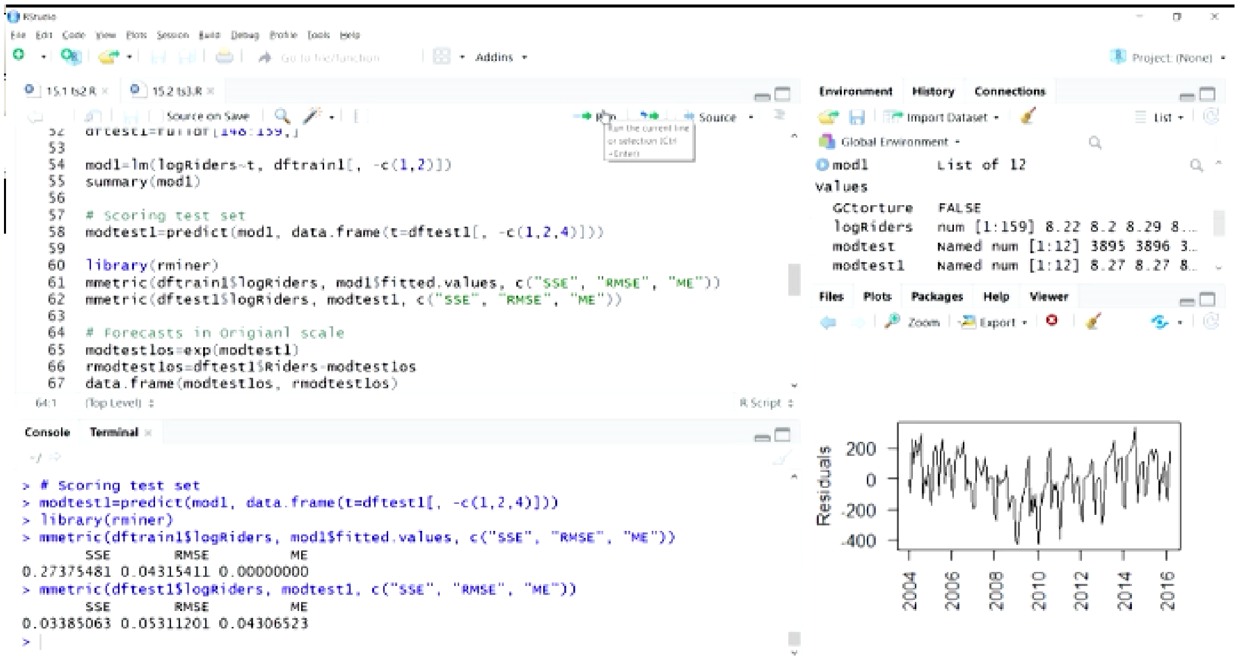


```
## Exponential trend
logRiders=log(fulldf$Riders)
fulldf=cbind(fulldf, logRiders)
head(fulldf)

# Trimming
dftrain1=fulldf[1:147,]
dftest1=fulldf[148:159,]

mod1=lm(logRiders~t, dftrain1[, -c(1,2)])
summary(mod1)

# Scoring test set
modtest1=predict(mod1, data.frame(t=dftest1[, -c(1,2,4)]))

library(rminer)
```

```
      Min        1Q    Median        3Q       Max
-0.119648 -0.028378  0.002949  0.035928  0.084001

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept) 8.220e+00  7.204e-03 1141.017  < 2e-16 ***
t           3.115e-04  8.445e-05    3.689 0.000318 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04345 on 145 degrees of freedom
```

```
dftrain     147 obs. of 3 variables
dftrain1    147 obs. of 4 variables
fulldf      159 obs. of 4 variables
mod         List of 12
mod1        List of 12
```

a look at the results, now you can see still this single predictor T it is still significant, so from this model result also we can say that exponential level and exponential trend is fitting to the data, you know, as per the results linear regression results it is comes out to be significant.



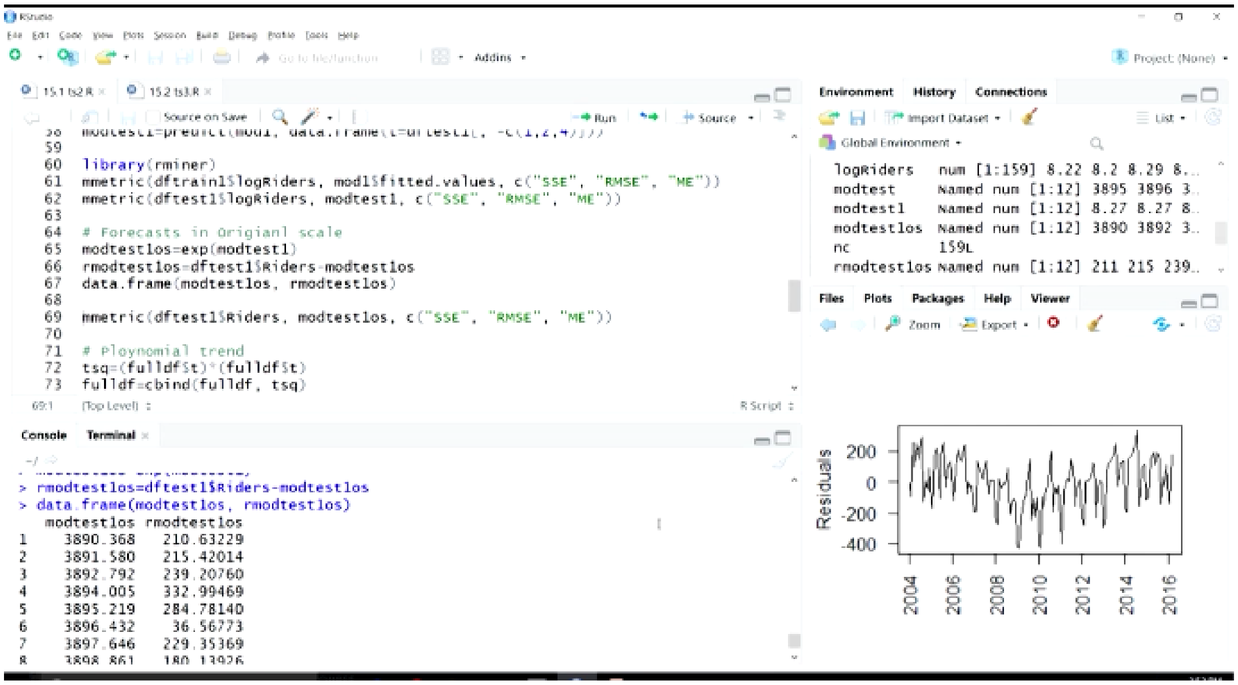
```
mod1=lm(logRiders~t, dftrain1[, -c(1,2)])
summary(mod1)

# Scoring test set
modtest1=predict(mod1, data.frame(t=dftest1[, -c(1,2,4)]))

library(rminer)
mmetric(dftrain1$logRiders, mod1$fitted.values, c("SSE", "RMSE", "ME"))
mmetric(dftest1$logRiders, modtest1, c("SSE", "RMSE", "ME"))

# Forecasts in Origianl scale
modtestlos=exp(modtest1)
rmodtestlos=dftest1$Riders-modtestlos
data.frame(modtestlos, rmodtestlos)
```

```
> # Scoring test set
> modtest1=predict(mod1, data.frame(t=dftest1[, -c(1,2,4)]))
> library(rminer)
> mmetric(dftrain1$logRiders, mod1$fitted.values, c("SSE", "RMSE", "ME"))
       SSE       RMSE         ME
0.27375481 0.04315411 0.00000000
> mmetric(dftest1$logRiders, modtest1, c("SSE", "RMSE", "ME"))
       SSE       RMSE         ME
0.03385063 0.05311201 0.04306523
> |
```

```
mod1          List of 12
Values
  GCtorture   FALSE
  logRiders   num [1:159] 8.22 8.2 8.29 8...
  modtest     Named num [1:12] 3895 3896 3...
  modtest1    Named num [1:12] 8.27 8.27 8...
```

So now let's score the test data set values, now let's compute the matrix, this is for training partition, this is for test partition, so we can see here that in terms of performance of model on test partition, now the gap you know in the performances you know good enough on the test partition, if we compare this back to the performance of linear model on validation set, so there was significant gap here, but here you know the model seems to be improving in terms of performing on new set that is validation set, if we look the RMSE value and also ME value so that differences that gap is closing down.

Now we you know look at the, because this output variable is different so if we want to compare the results of linear trend model with exponential trend model first we need to re-compute these forecasted values to original scale, so we are just doing to do this, we'll call this function EXP to take the exponent of forecasted results, let's compute this, now we'll have to compute the new residual values in the original scale, so let's do this.



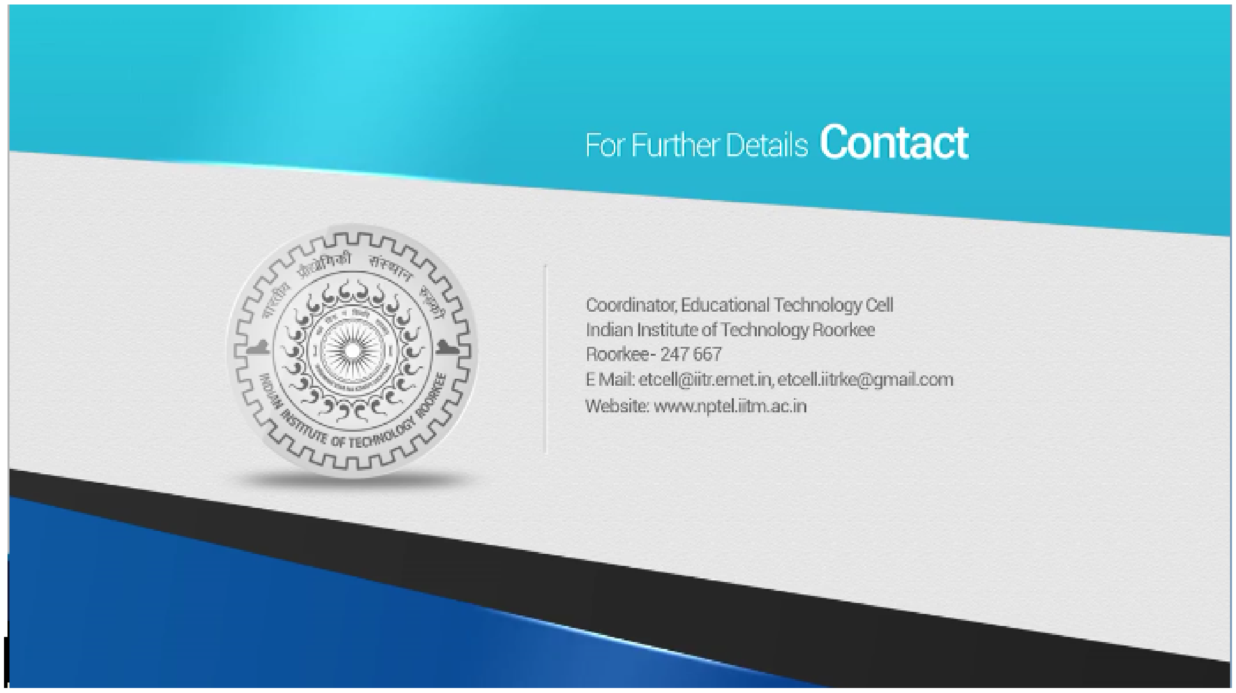
```r
library(rminer)
mmetric(dftrain1$logRiders, mod1$fitted.values, c("SSE", "RMSE", "ME"))
mmetric(dftest1$logRiders, modtest1, c("SSE", "RMSE", "ME"))

# Forecasts in Origianl scale
modtestlos=exp(modtest1)
rmodtestlos=dftest1$Riders-modtestlos
data.frame(modtestlos, rmodtestlos)

mmetric(dftest1$Riders, modtestlos, c("SSE", "RMSE", "ME"))

# Ploynomial trend
tsq=(fulldf$t)*(fulldf$t)
fulldf=cbind(fulldf, tsq)
```

```
> rmodtestlos=dftest1$Riders-modtestlos
> data.frame(modtestlos, rmodtestlos)
  modtestlos rmodtestlos
1   3890.368   210.63229
2   3891.580   215.42014
3   3892.792   239.20760
4   3894.005   332.99469
5   3895.219   284.78140
6   3896.432    36.56773
7   3897.646   229.35369
8   3898.861   180.13926
```

```
logRiders    num [1:159] 8.22 8.2 8.29 8...
modtest      Named num [1:12] 3895 3896 3..
modtest1     Named num [1:12] 8.27 8.27 8..
modtestlos   Named num [1:12] 3890 3892 3..
nc           159L
rmodtestlos  Named num [1:12] 211 215 239..
```

Now these are going to be the values for the test data set, we can see in the original scales for exponential model, so you can see predicted values and residuals are here, this is for test partition, now we can compute the matrix also using the original scale values, so we can see here, the value is comes out to be RMSE 213, now if we go back so this is the result of exponential model on test data set, so RMSE value is 213 and ME value is 173, if we go back to the results that we had earlier for linear trend model, so we will have to go back, so you can see here 210 and 169, so in terms of numbers if we see that you know linear trend and exponential trend it seems that linear trend is doing better than exponential trend on test data in terms of numbers, right, RMSE value so when we go back to the original scales it seems that linear trend is doing better than the exponential trend.

So till now you know in this lecture we covered the linear trend modeling and exponential trend modeling, so we'll continue this discussion in the next lecture and we'll cover the polynomial trend, so we'll stop here. Thank you.



For Further Details Contact

Coordinator, Educational Technology Cell

Indian Institute of Technology Roorkee

Roorkee- 247 667

E Mail: etcell@iitr.ernet.in, etcell.iitrke@gmail.com

Website: www.nptel.iitm.ac.in

For Further Details Contact
Coordinator Educational Technology Cell
Indian Institute of Technology Roorkee
Roorkee – 247 667
E Mail:-etcell@iitr.ernet.in, iitrke@gmail.com
Website: www.nptel.iitm.ac.in
Acknowledgement
Prof. Ajit Kumar Chaturvedi
Director, IIT Roorkee
NPTEL Coordinator
IIT Roorkee
Prof. B. K Gandhi
Subject Expert
Dr. Gaurav Dixit
Department of Management Studies
IIT Roorkee
Produced by
Mohan Raj.S
Graphics
Binoy V.P
Web Team
Dr. NibeditaBisoyi
Neetesh Kumar
Jitender Kumar
Vivek Kumar
Dharamveer Singh
Gaurav Kumar
An educational Technology cell