

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
NPTEL
NPTEL ONLINE CERTIFICATION COURSE
Business Analytics & Data Mining Modeling
Using R – Part II
Lecture-17
Regression Based Forecasting Methods– Part II
With
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology Roorkee

Business Analytics & Data Mining Modeling Using R - Part II

Lecture-17 Regression Based Forecasting Methods-Part II



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 lecture we started our discussion on Regression Based Forecasting Methods, so therein we talked about the importance of modeling trend and seasonality, we started our discussion with trend, we talked about three specific trend shape linear trend, exponential trend, polynomial trend, we also did an exercise in R studio environment where we were able to model the linear trend and exponential trend.

Regression-Based Forecasting Methods

- Polynomial Trend
 - Specifically, quadratic relationship can be modeled as below:

$$Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \epsilon$$

- Additional predictor: t^2 to capture the quadratic relationship
 - This model fits a multiple linear regression of no. of riders on two predictors (time index and square of it)
- Open RStudio



Now in this particular lecture we'll start with the third type that is polynomial trend, so let's discuss this, and then we'll go back to R studio environment and we'll model this, so if we look at the polynomial trend so specifically we'll focus on the quadratic trend, quadratic relationship if it is there it can be modeled as below, so the equation we can have a look so Y_T and on the RHS side we have $\beta_0 + \beta_1 T + \beta_2 T^2 + \epsilon$, so we can see now we have two you know, two predictors here T and square of T and so these two predictors so our you know if we talk about the specific data set that we are using bicycle ridership, so Y being the number of riders, so now number of riders is going to be the decreased on these two variables T and square of T , where T is the time index.

So the additional predictors, additional predictor that we have T square which is essentially to capture the quadratic relationship, so when we use this quadratic term T square essentially what we are expecting is that series is following a U shaped trend, so if we remember the plot of the original series then that kind of trend that was visible there, it look more like a U shape trend instead of linear or you know exponential trend, so this model particularly fits a multiple linear regression of number of riders on two predictors as we discussed, time index and square of it, so we'll go back to the R studio model, R studio environment and let's try and build this polynomial trend rather quadratic trend, so first we'll compute the new predictor that is T

The screenshot shows RStudio with the following R code in the editor:

```

61 nmetric(dftrain$logriders, mod$fitted.values, c("SSE", "RMSE", "ME"))
62 nmetric(dftest$logriders, modtest1, c("SSE", "RMSE", "ME"))
63
64 # Forecasts in original scale
65 modtestlos=exp(modtest1)
66 rmodtestlos=dftest1$logriders-modtestlos
67 data.frame(modtestlos, rmodtestlos)
68
69 nmetric(dftest1$logriders, modtestlos, c("SSE", "RMSE", "ME"))
70
71 # Polynomial trend
72 tsq=(fulldf$t)^2*(fulldf$t)
73 fulldf=cbind(fulldf, tsq)
74 head(fulldf)
75
76 dftrain2=fulldf[1:147,]
77 dftest2=fulldf[148:159,]
78
79 mod2=lm(Riders~t+tsq, dftrain2[, -c(1,4)])

```

The console output shows the results of the `nmetric` function calls:

```

> nmetric(dftrain$logriders, mod$fitted.values, c("SSE", "RMSE", "ME"))
      SSE      RMSE      ME
3.876251e+06 1.623855e+02 6.164948e-15
> nmetric(dftest$logriders, modtest, c("SSE", "RMSE", "ME"))
      SSE      RMSE      ME
529990.4827 210.1568 169.2548

```

The environment pane shows the following objects:

- `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`: 159
- `rmodtestlos`: Named num [1:12] 211 215 239...

The Residuals plot shows the residuals of the model over time (2004 to 2016), with values ranging from approximately -400 to 200.

square, so we call this TSQ here, so now we are multiplying these two value T and we had already computed time index, so now we are going to multiply it with its own value and we'll get the T square, so let's run this.

Now we are going to append this new variable into the existing data frame that we have, so let's have a look at first 6 observations, now we can see here we have the T square has been added, so in the quadratic modeling that we are going to perform, we'll have T and T square and riders is also there, so we are going to regress this riders on these two predictors T and TSQ.

The screenshot shows RStudio with the following R code in the editor:

```

65 modtestlos=exp(modtest1)
66 rmodtestlos=dftest1$logriders-modtestlos
67 data.frame(modtestlos, rmodtestlos)
68
69 nmetric(dftest1$logriders, modtestlos, c("SSE", "RMSE", "ME"))
70
71 # Polynomial trend
72 tsq=(fulldf$t)^2*(fulldf$t)
73 fulldf=cbind(fulldf, tsq)
74 head(fulldf)
75
76 dftrain2=fulldf[1:147,]
77 dftest2=fulldf[148:159,]
78
79 mod2=lm(Riders~t+tsq, dftrain2[, -c(1,4)])

```

The console output shows the results of the `nmetric` function calls:

```

> nmetric(dftrain$logriders, mod$fitted.values, c("SSE", "RMSE", "ME"))
      SSE      RMSE      ME
3.876251e+06 1.623855e+02 6.164948e-15
> nmetric(dftest$logriders, modtest, c("SSE", "RMSE", "ME"))
      SSE      RMSE      ME
529990.4827 210.1568 169.2548

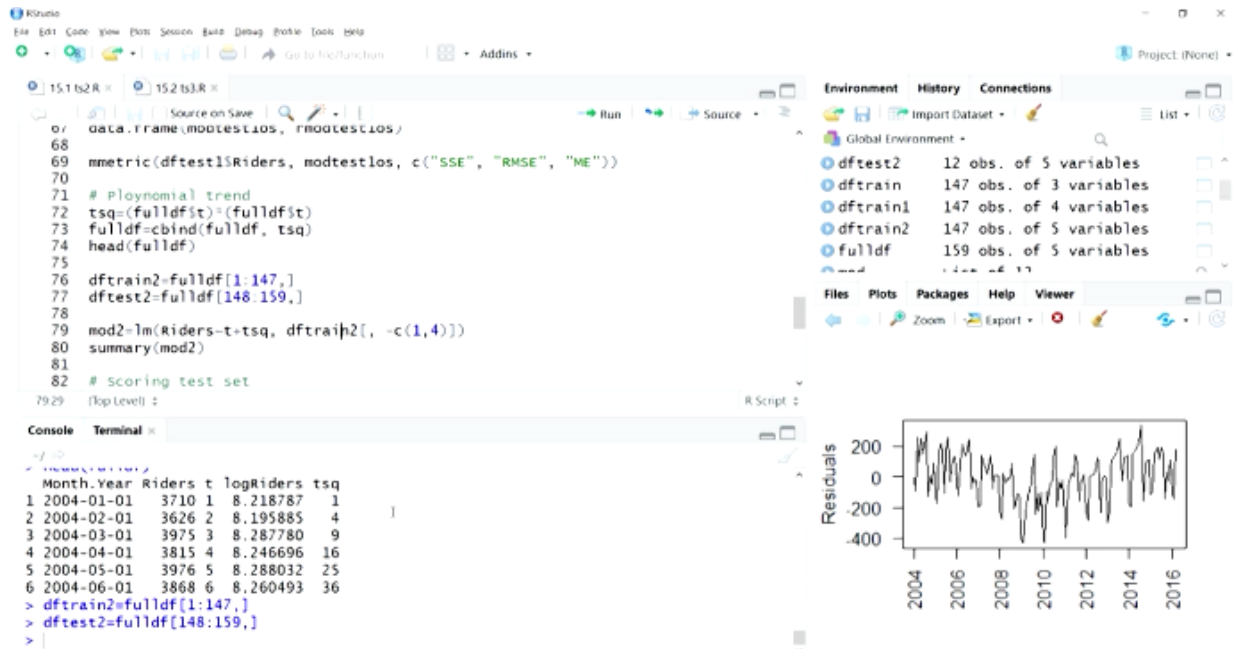
```

The environment pane shows the following objects:

- `fulldf`: 159 obs. of 5 variables
- `mod`: List of 12
- `mod1`: List of 12
- `values`: GCTorture FALSE, logriders num [1:159] 8.22 8.2 8.29 8...

The Residuals plot shows the residuals of the model over time (2004 to 2016), with values ranging from approximately -400 to 200.

So just like you know other modeling that we did, next we are going to trim this dataset, so we'll create these two partition training and testing, you can see we're using the same numbers of observation that we had used for earlier models.

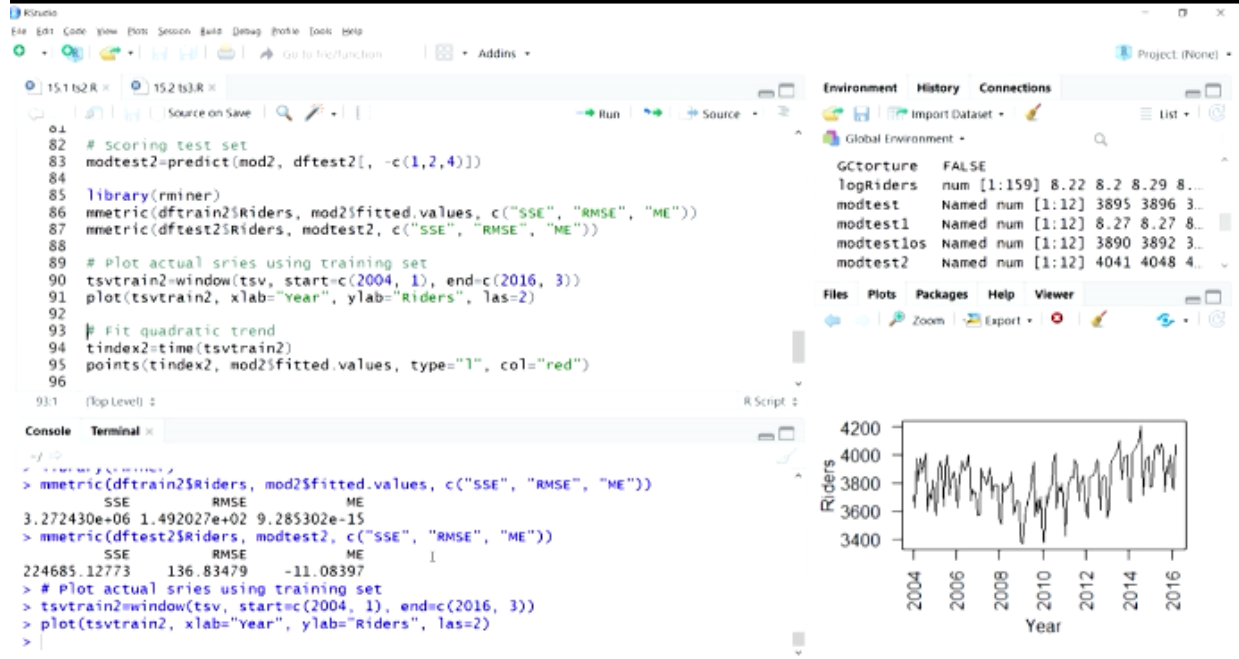


Now if we look at the model equations, so the formula says riders versus T + TSQ, so now in this quadratic model we are regression riders on these two variables, and the dataset, data frame is also appropriately specified, so let's build this model that has been build, let's have a look at the results. So if we look at the result here we can see T and T square and we can see both seem to be significant here, so distance model is also significant so if we are able to remember the linear trend and exponential trend that modeling that we had done in the previous lectures they were also found to be significant, this one which is the quadratic model is also found to be significant. Now we'll compare how the performance, we'll compare the performance of this particular model with previous models as well.

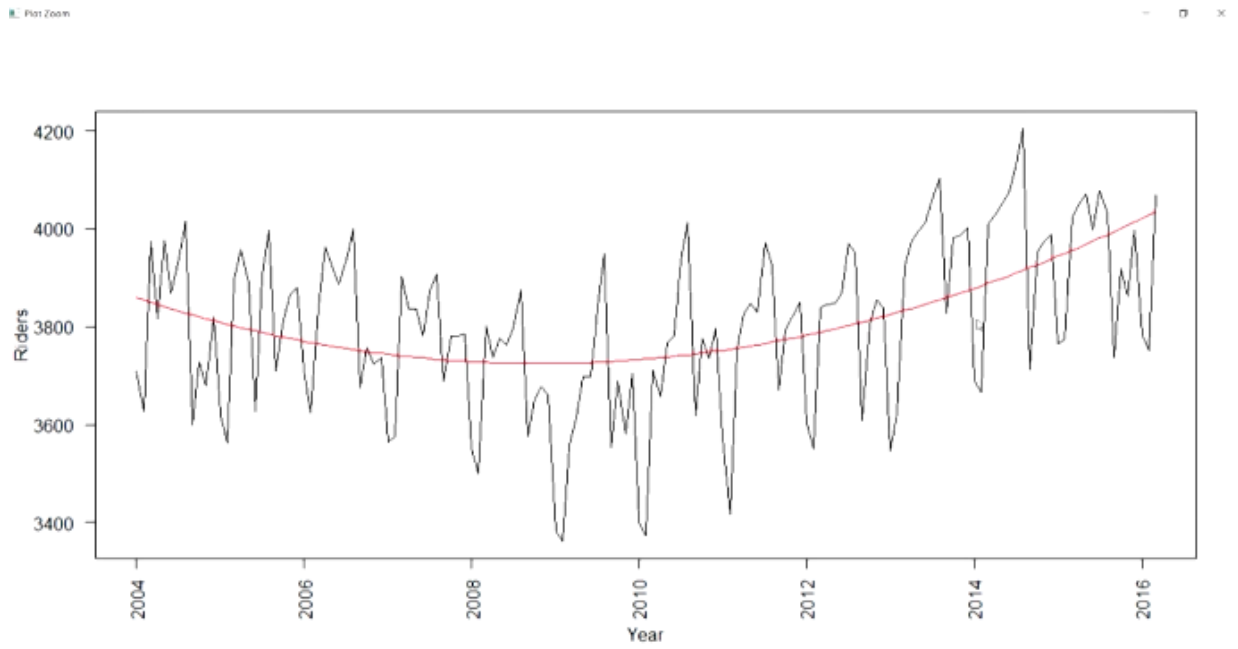


So first let's score the test set, so you can see model object mod 2 and the test partition, so we'll score this off, now let's compute the matrix numbers, so we can see here these are the numbers

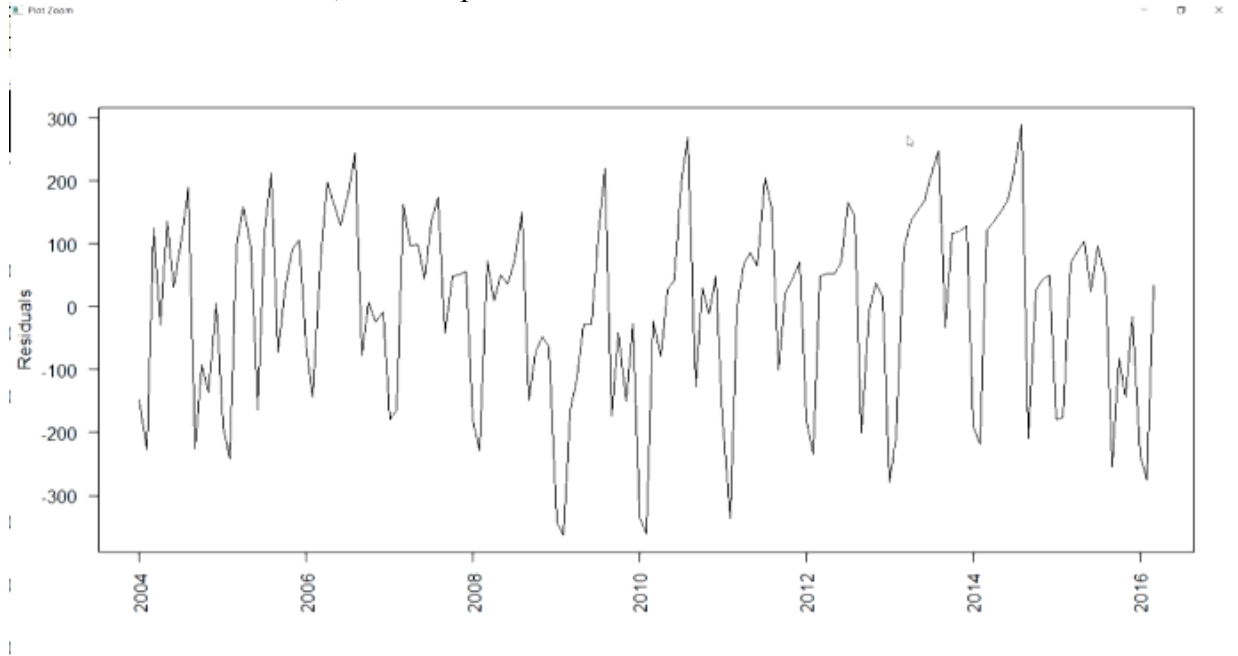
for you know training partition, so RMSE value comes out to be 149, so ME value is quite close to 0, let's compute this matrix for test partition, so you can see the value is now 136, so it seems that the model is performing well on validation partition, rather the model seems to performing a better and validation partition in comparison to what we had, you know its performance on training partition, so if you remember the two other models, candidate models that we had run in previous lecture, the linear trend and the exponential both had RMSE value of more than 200 on you know test set, however if we look at this particular model the RMSE value has come down to 136 which is even a smaller than its you know RMSE value on training set, so it seems that not just this quadratic model is significant, it is also adequately capturing the time series pattern.



So we'll confirm this, so let's plot actual series using training set, so again just like previous model we'll create this you know time series, you know well subset this time series object, and then let's plot this, so this is the plot, this is for you know training, you know training set, now we'll just you know add the fitted model that we have just you know estimated, so first we'll extract the time index format and then we'll use the pointsfunction and the fitted values are going to be plotted, so now we can see here, let's zoom into this plot, so we can see now these



U shaped curve has been fitted this quadratic curve has been fitted on the time series and we can see, it seems to be adequately capturing the shape of you know trend that is there in the time series, and which actually also got reflected in the results on test partition, so let's have a look at the residual series, so we'll plot this one.



Now if you look at the you know residual series, now we can see here you know just remember our modeling for linear trend, the residual series plot actually you know was carrying for, was carrying the U shaped trend along with it, now in this residual plot we don't see any trend here, so that has been adequately captured by the model itself, and it seems that only the seasonality is you know left over in this residual plot because till now we have not modeled seasonality component.

Regression-Based Forecasting Methods

- Polynomial Trend
 - Specifically, quadratic relationship can be modeled as below:

$$Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \epsilon$$

- Additional predictor: t^2 to capture the quadratic relationship
 - This mode fits a multiple linear regression of no. of riders on two predictors (time index and square of it)
- Open RStudio

So let's go back to our discussion, so three types of trends shapes till now we have been able to model, linear trend, exponential trend and polynomial trend, and we saw that how you know, even though the linear trend and the exponential trend were found to be significant but there was you know significant gap in terms of performance on a test partition, test set and when we looked at the polynomial you know trend the quadratic trend, the performance improved on the test set and the shape of trend you know that was present in the time series, original time series that was adequately captured, and the same could be confirmed, the same was confirmed by the residual series plot.

Regression-Based Forecasting Methods

- Fitting other types of trend shapes
 - Can be done if it can be expressed in a mathematical form
 - From modeling perspective
 - Shape should be applicable for the entire series as well as
 - Should be global
 - Future observations that are to be forecasted
 - Selecting an overly complex shape seem to be fitting the training data
 - Can lead to overfitting
 - Poor performance on validation set

So let's move further, so till now we talked about these three types of you know trend shape, so is it possible to fit other types of trend shapes? So yes, that can be done, but certain points that we need to discuss here, so fitting other types of trend shapes so it can be done if this particular trend shape that we want to fit, it can be expressed in a mathematical form just like the linear trend you know we had a mathematical form the equation that could be used to you know model it into using multiple linear regression and then for the exponential trend also and for the polynomial that is quadratic trend that we had used, so if we want to fit any other type of trends shape and if it can be expressed in the mathematical form definitely it can be fitted, however from modeling prospective we need to take care of certain things, for example the shape that we want to fit it should be applicable for the entire series as well as future observations that are to be forecasted, because the main idea is to be able to forecast future you know values, so therefore the shape that we are you know, that we don't plan to fit for the series it should be applicable for the entire series and also the future observation that we want to forecast, only then the model is going to perform well, so you can also see one small point is also mentioned here, it should be global, so like we'll like to refer back to earlier discuss and when we talked about that if we plan the statistical method, the model driven approach where we you know are using regression based kind of methods, so there the kind of trend that is there, the kind of patterns that are available they should be you know global in nature, so that the model is able to perform well on the, not just on the points which are part of the training, but also on the points where we want to you know in forecast future values.

So keeping these things in mind from the modeling prospective the trend shapes should be applicable for the entire series and future observation, and the trend shape if it can be you know presented in a mathematical form then definitely we can fit it using the approach that we have discussed in previous and this lecture as well.

Another point related to this is that if we happen to select an overly complex shape which you know, you know seem to be fitting the training data, so sometime you know we might think about a shape which could be represented in a mathematical form and is also you know look to be fitting to the you know training data points, but this can lead to over fitting because the you know future points might not be following, might not be following this particular shape, so therefore using an overly complex shape typically can lead to over fitting, and also poor performance on validation set, because that shape though, even though it might be fitting the training set observation quite well, but it might not be you know applicable for the validation set points or even for future observations, so therefore we should avoid selecting an overly complex, rather the more important point is we should look for you know any shape that we want to fit, we should look whether it is going to be applicable globally for the time series, for the period of time series under consideration.

Regression-Based Forecasting Methods

- Modeling the seasonality
 - A seasonal pattern in a series means
 - Observations for particular periods (seasons) have consistently higher or lower values in comparison to other periods
 - For example,
 - Day-of-week patterns, monthly patterns, quarterly patterns
 - Seasonality could be of two types
 - Additive seasonality
 - Values are higher or lower by a certain amount for the particular seasons in comparison to other seasons on an average level
 - Y is used as output variable

Let's move forward, so till now the modeling that we have done was mainly about the trend component, so seasonality patterns were not modeled, so now what we are going to do is we'll discuss this particular aspect modeling the seasonality, so let's understand a few points about this, so what is seasonality? So though we have discussed this in previous lectures as well, let's you know let's discuss it again, a seasonal pattern in a series means observations for particular periods or seasons have consistently higher or lower values in comparison to other periods, so as we have talked about in previous lectures we are going to observe you know peak values which are you know consistently higher in comparison to the average level values and you know this appears you know repeatedly, so for certain periods the values are consistently higher, every time that you know that period or season comes again, the values that are being taken there, they are higher or lower you know consistently you know from the average level, so observations for particular periods have consistently higher or lower values in comparison to other periods then we can say that probably some sort of seasonality is present in the series, for example day of week patterns, monthly patterns, quarterly patterns, so what we mean by you know these examples is that you know if the series is weekly then you know for you know particular days the values could be higher you know for example, for Sundays and you know Saturdays the values could be higher in comparison to the you know, in comparison to the working days, so this is what we mean.

Similarly for monthly patterns it could be like that for you know month of May and June which are the summer months, the values could be higher in comparison to other months in the year. Now the next one quarterly patterns, what we mean is for a particular quarter, for example let's say April, May, June, the values for that this particular quarter might be higher in comparison to the other quarters, so this is, in this fashion the seasonality could be present in a particular series.

Now seasonality could be of two types, the way those values are consistently higher or lower it could be you know present in two ways, so let's discuss this two types of seasonality, so first one is additive seasonality, so what we mean by this is that values are higher or lower by a certain amount for the particular seasons in comparison to the other seasons on an average

level, so it is the you know additional amount by which the values are either higher or lower, so in comparison to other seasons, the seasons where we are saying the you know seasonality is present, the values are higher or lower by a certain amount, so when this is the case we say that the additive seasonality is present, so typically when we are modeling this kind of seasonality Y is our output variable, in the other kind of seasonality which is the multiplicative seasonality, the output variable will change to $\log Y$, so $\log Y$ this is going to happen, let's understand what multiplicative seasonality is, so in multiplicative seasonality values are higher or lower by a certain percentage for the particular seasons in comparison to other seasons on an average level, so that you know particular seasons, particular periods where we are observing seasonality the values of those seasons seem to be, seem to be higher or lower by a certain percentage, so because of this the nature of you know, nature of the seasonality is going to be multiplicative, so we can say higher by you know, by a factor of 1.2 or 1.1 or 1.05 or 1.3, so because of this kind of you know this kind of variation due to seasonality we say this, we call this as multiplicative seasonality, so in this case as we have seen, in case of exponential trend that we had taken log of you know Y as the output variable, similarly because of the multiplicative seasonality we'll have to take you know $\log Y$ as the output variable to be able to use the multiple linear regression model for fitting this kind of seasonality.

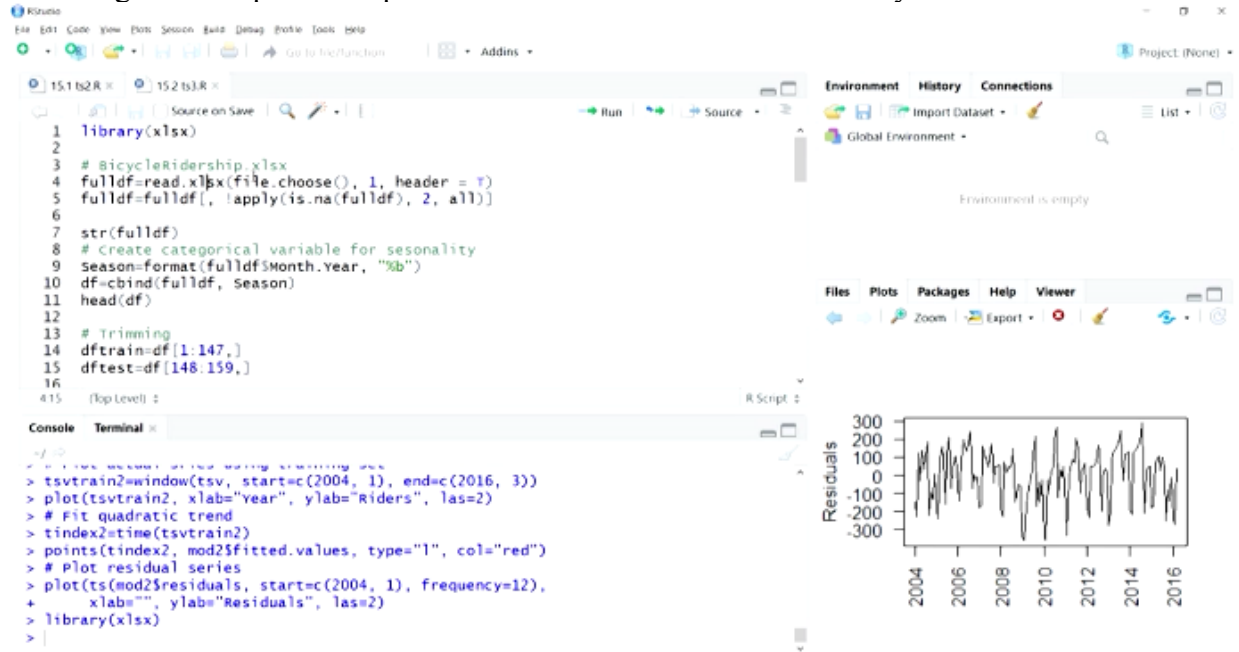
Regression-Based Forecasting Methods

- Modeling the seasonality
 - Seasonality could be of two types
 - Multiplicative seasonality
 - Values are higher or lower by a certain percentage for the particular seasons in comparison to other seasons on an average level
 - $\log Y$ is used as output variable
- Modeling additive seasonality
 - A new categorical variable is created
 - To record the season for each observation

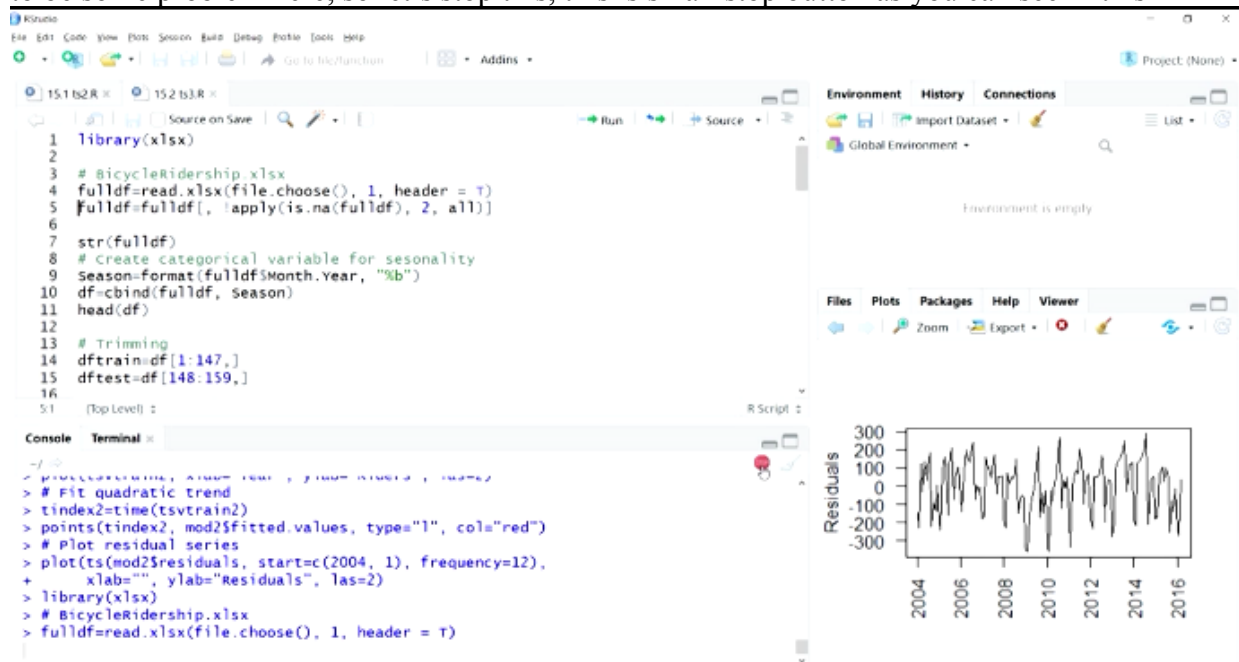
So what we are going to do next is we'll take our Y cycle regressive example and we'll model additive seasonality where values for certain you know period, where we are expecting seasonality, they are going to be either higher or lower by a certain amount.

So what are the steps that needs to be, that need to be taken to implement this, to implement this additive seasonality, to model this additive seasonality, so first thing we need to create a new categorical variable, so main idea about creating this categorical variable is to record the season for each observation, so for each of the observation that is there in the time series, we would like to record, we would like to note down the season for that observations that you know the way we can name it, the way we can identify it, so once that is done so there are going to be a number of seasons for the specified period of time series, so if you know there are N seasons, so this categorical variable R is going to have M levels, so therefore we are going to, we will

have to create M-1 dummy variables and these dummy variables are then, can then be included in the regression equation as predictors and that is how the seasonality can be modeled.



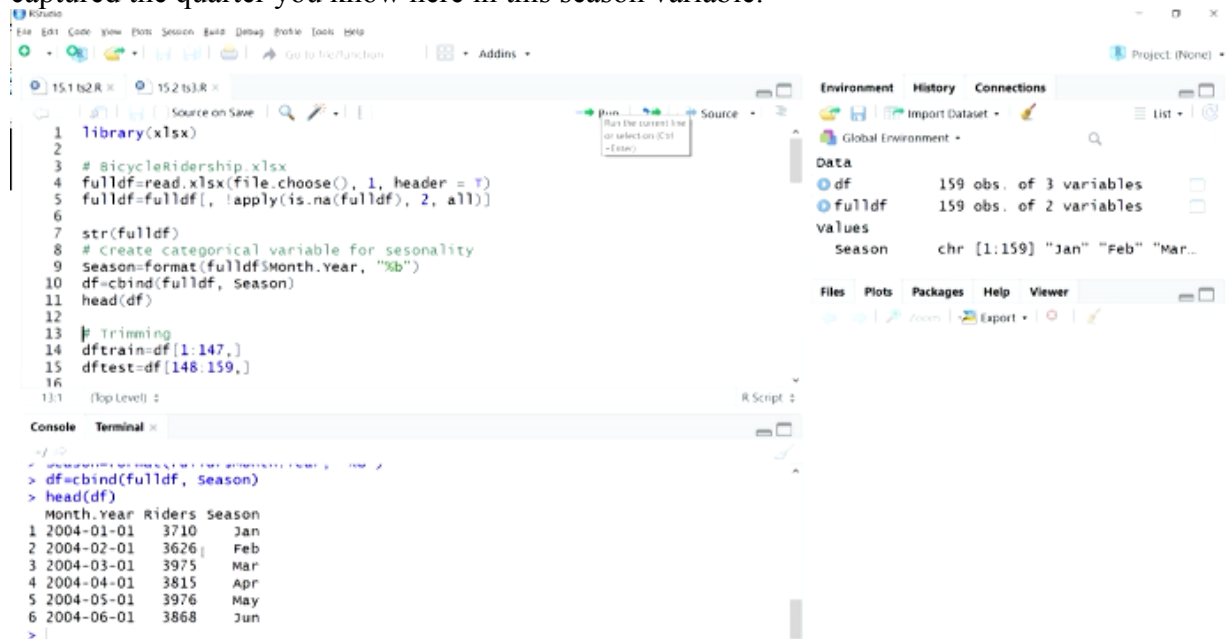
So what we'll do, we'll go back to our R studio and we'll try and model this seasonality into our time series forecasting. So let's load this library, so we are going to import this data set first, bicycle ridership, so once this dataset is imported we'll see that as I talked about, we'll first create a categorical variable to record this seasonality for each of the observation, so there seem to be some problem here, so let's stop this, this is small stop button as you can see in this



particular you know console window, so we'll try and stop this, because, okay so I'll try it again, so let's again, let's load the library, let's import the dataset, so you can see the dataset has been imported we can see in the environment section 149 observation, 2 variables, so let's run

this code as well, let's have a look at the structure, same variables, this is the bicycle ridership data you know set.

Now what we are going to do is we'll create this new categorical variable seasonality, so you can see in the next line of code I have season here, and I'm calling this format function which we have used in previous lectures as well, so month.year this particular variable now this is going to be formatted using this particular function and we'll just have the month, so we'll have, we'll just have the month for each observation. Now if this is you know monthly you know time series and what we are expecting is that the seasonality is month wise, if we remember the you know time plot of the actual time series we saw there were you know few ups and downs for few months, few months the ridership was significantly high in comparison to other months, so the seasonality looked to be monthly and for the same reason you know in the season variable, categorical variable we are going to record the month as season, so using this function we would be able to capture the month for each observation, had this seasonably reason a quarter you know for some certain quarters, the riders, number of riders would have been significantly you know high in comparison to the other quarters, then we would have captured the quarter you know here in this season variable.



```
1 library(xlsx)
2
3 # BicycleRidership.xlsx
4 fulldf=read.xlsx(file.choose(), 1, header = T)
5 fulldf=fulldf[, !apply(is.na(fulldf), 2, all)]
6
7 str(fulldf)
8 # Create categorical variable for seasonality
9 Season=format(fulldf$Month.Year, "%b")
10 df=cbind(fulldf, Season)
11 head(df)
12
13 # Trimming
14 dftrain=df[1:147,]
15 dftest=df[148:159,]
16
17 |> (Top Level) |
```

Environment History Connections

Data

- df 159 obs. of 3 variables
- fulldf 159 obs. of 2 variables

Values

Season chr [1:159] "Jan" "Feb" "Mar..."

```
> df=cbind(fulldf, Season)
> head(df)
  Month.Year Riders Season
1 2004-01-01  3710   Jan
2 2004-02-01  3626   Feb
3 2004-03-01  3975   Mar
4 2004-04-01  3815   Apr
5 2004-05-01  3976   May
6 2004-06-01  3868   Jun
```

So we can have 12 months in a year, so total number of variables that we expect, total number of levels that we expect in this, you know categorical variable are going to be 12, so let's create this variable, now we are going to append this in this our existing data frame, let's have a look at first 6 observation we can see here, the second column is riders that is number of riders for each month and then we have season, so which is capturing the name of the month, so this is because we are expecting the monthly seasonality, so for this reason we have captured the same in this categorical variable.

Now as we have done earlier also, and the next step is going to be trimming of dataset, so let's create the training set which is the earlier period, first 147 observations and then the remaining 12 observation for the test set.

```

6
7 str(fulldf)
8 # Create categorical variable for seasonality
9 Season=Format(fulldf$Month.Year, "%b")
10 df=cbind(fulldf, Season)
11 head(df)
12
13 # Trimming
14 dftrain=df[1:147,]
15 dfctest=df[148:159,]
16
17 mod=lm(Riders~Season, dftrain[, -c(1)])
18 summary(mod)
19
20 # Scoring test set
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400

```

Now if we look at the model, we are the LM function and you see that we are modeling regression riders against season, so you can see we are not including time index here, we are just trying to model only the seasonality component though we have understood from our previous, from this lecture and previous lecture itself that the trend is present and we were able to see the results also that quadratic trend was able to adequately capture the pattern, however in this exercise we're only focusing on the seasonality component and that is why in this equation riders is being regressed as against season, right, and the dataset data frame is a preparatory specified, so let's build this model. Now have a look at the results, so now as we know that for any you know variable you know which is factor will have the, will have the you know this dummy variables created internally in R environment, and you can see in the results

```

0
1
2
3
4
5
6
7
8
9 Season=Format(fulldf$Month.Year, "%b")
10 df=cbind(fulldf, Season)
11 head(df)
12
13 # Trimming
14 dftrain=df[1:147,]
15 dfctest=df[148:159,]
16
17 mod=lm(Riders~Season, dftrain[, -c(1)])
18 summary(mod)
19
20 # Scoring test set
21 modtest=predict(mod, data.frame(Season=dfctest[, -c(1,2)]))
22
23 library(rminer)
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400

```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3858.50	33.97	113.590	< 2e-16 ***
SeasonAug	139.42	48.04	2.902	0.00433 **
SeasonDec	-20.00	48.04	-0.416	0.67783
SeasonFeb	-288.96	47.11	-6.134	8.88e-09 ***
SeasonJan	-251.42	47.11	-5.337	3.88e-07 ***
SeasonJul	94.42	48.04	1.965	0.05142 .
SeasonJun	-9.50	48.04	-0.198	0.84353
SeasonMar	11.42	47.11	0.242	0.80876
SeasonMay	31.42	48.04	0.654	0.51473

all this you know, all this dummy variables are there, so we don't have to explicitly do it in R environment, so you can see season August, season December, season Feb, so total you know, these seasons, 11 seasons are going to be there, one is going to be one might be there for the reference.

```

8 # Create categorical variable for seasonality
9 season=format(fulldf$Month.year, "%b")
10 df=cbind(fulldf, season)
11 head(df)
12
13 # Trimming
14 dftrain=df[1:147,]
15 dfctest=df[148:159,]
16
17 mod=lm(Riders~Season, dftrain[, -c(1)])
18 summary(mod)
19
20 # Scoring test set
21 modtest=predict(mod, data.frame(Season=dfctest[, -c(1,2)]))
22
23 library(rminer)
201
  
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3858.50	33.97	113.590	< 2e-16 ***
SeasonAug	139.42	48.04	2.902	0.00433 **
SeasonDec	-20.00	48.04	-0.416	0.67783
SeasonFeb	-288.96	47.11	-6.134	8.88e-09 ***
SeasonJan	-251.42	47.11	-5.337	3.88e-07 ***
SeasonJul	94.42	48.04	1.965	0.05142 .
SeasonJun	-9.50	48.04	-0.198	0.84353
SeasonMar	11.42	47.11	0.242	0.80876
SeasonMay	31.42	48.04	0.654	0.51423
SeasonNov	-63.25	48.04	-1.317	0.19019

So now if you look at the results here, so August this season seems to be significant then Feb is significant, then Jan is significant and here we can see that July is also seem to be significant, any other season let's scroll down, so we can see that September is also significant, so it's seems that July, August, and September those 3 months they seem to be significant and Jan and Feb are also seem to be significant, so these are the months which are having you know, you know significant the higher or lower value, depending on the estimated coefficient, so if you look at the estimated coefficient here, so for August it is 139 so it seems that for August which is significant the values are on an average level higher for month of August and by 139 if we look at the Feb and Jan, so the values are on an average level lower by you know for Feb it is lower by 288 and for Jan it is lower by 251, so for some months values are higher, some months values are lower, so for the month August it was higher.

Now if we look at the July it is also on the higher side, if we look at the September it is on the lower side, so it is July and August where the August is having more higher level of significance, the values are in particular on the higher side and also in July, for month of Feb and Jan values are on the lower side, and also for the you know month of September values are on the lower side, so we can see for certain months values are on the higher side for certain month values on the lower side.


```

15 dfTest=df[148:159,]
16
17 mod=lm(Riders~Season, dftrain[, -c(1)])
18 summary(mod)
19
20 # Scoring test set
21 modtest=predict(mod, data.frame(Season=dfTest[, -c(1,2)]))
22
23 library(rminer)
24 mmetric(dftrain$Riders, mod$fitted.values, c("SSE", "RMSE", "ME"))
25 mmetric(dfTest$Riders, modtest, c("SSE", "RMSE", "ME"))
26
27 # Plot actual series using training set
28 dftrain[147,]
29 tsvtrain=ts(fulldf$Riders, start=c(2004, 1), end=c(2016, 3), frequency = 12)
30 plot(tsvtrain, xlab="year", ylab="Riders", las=2)
27.1 (top Level)

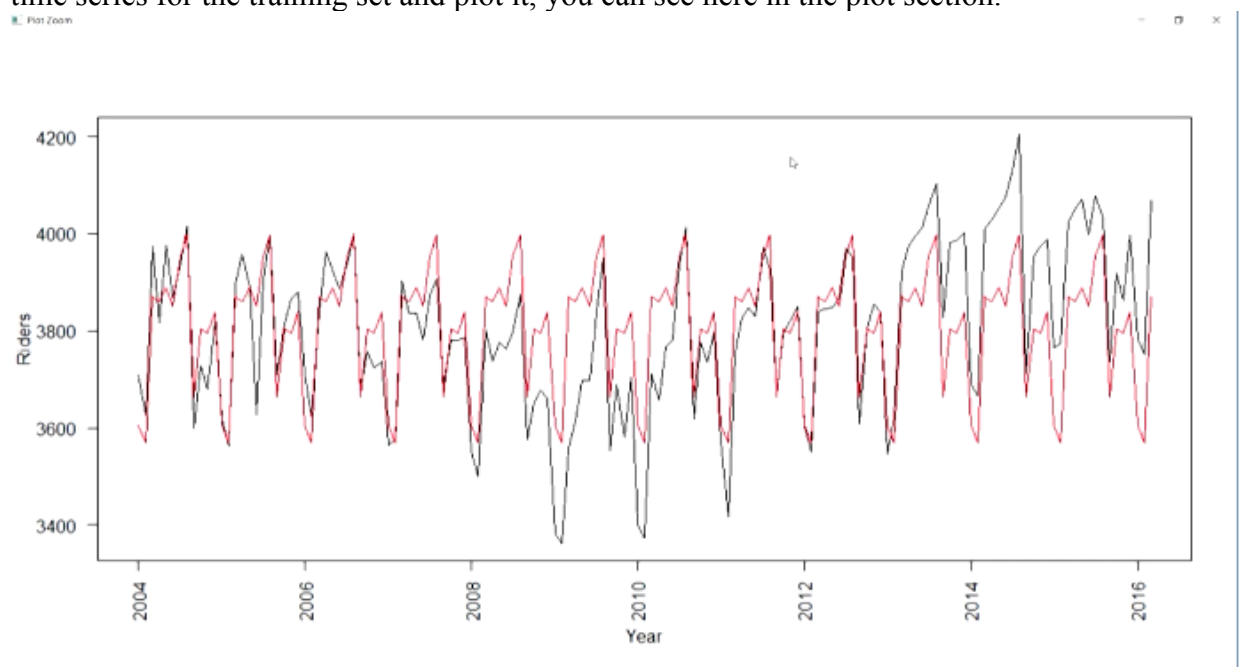
```

```

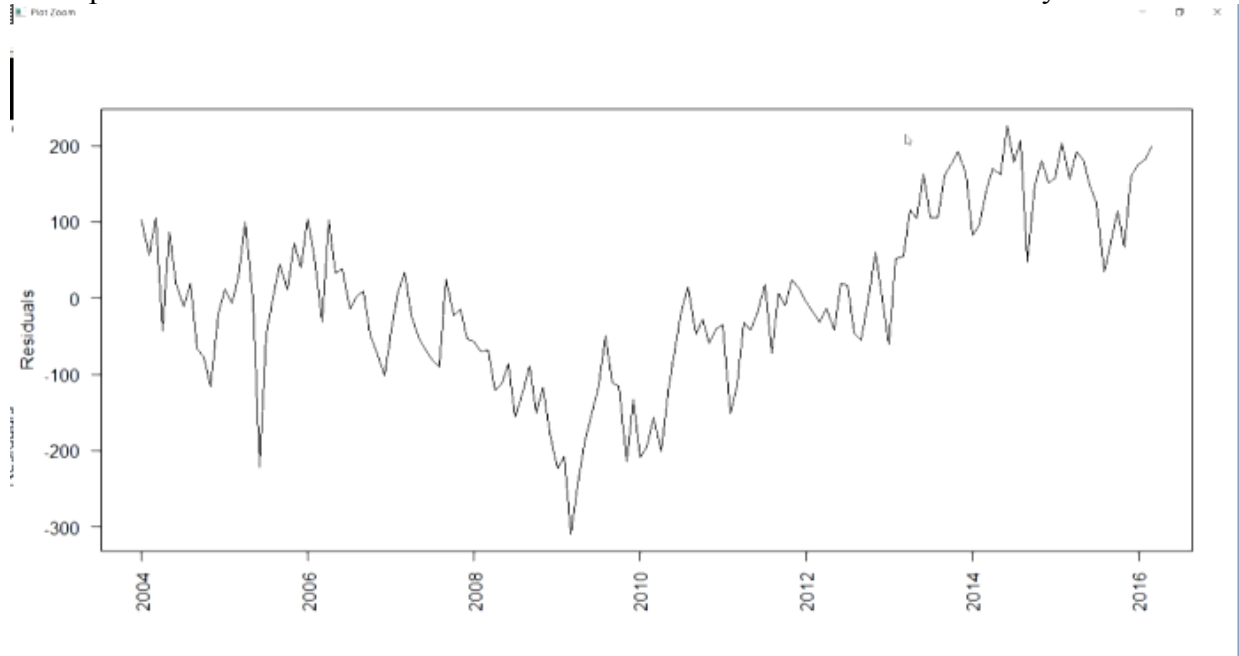
> # Scoring test set
> modtest=predict(mod, data.frame(Season=dfTest[, -c(1,2)]))
> library(rminer)
> mmetric(dftrain$Riders, mod$fitted.values, c("SSE", "RMSE", "ME"))
      SSE      RMSE      ME
1.869258e+06 1.127654e+02 -6.170234e-15
> mmetric(dfTest$Riders, modtest, c("SSE", "RMSE", "ME"))
      SSE      RMSE      ME
841988.1810 264.8881 262.2885
>

```

So now let's score the test dataset, now what we'll do is we'll compute the matrix for these two you know sets, so let's have a look at the numbers, let's compute for test set also, so you can see the RMSE value is 264 which is on test dataset which is higher in comparison to previous other models that we have built using linear trend it was about 210 you know exponential it was also about 210 for the polynomial the quadratic you know trend the value came down to 130, 140, and this value has gone up, this is about 264 so it seems you know that, even though the seasonality is present we saw that few dummy variables were significant, you know modeling just the seasonality component actually is not good enough to capture the patterns in the series, so this is what we understand so we can confirm the same using other plots, so let's create the time series for the training set and plot it, you can see here in the plot section.



Now we'll plot the fitted seasonality, so first we'll compute the time index and we'll add the fitted points, so we can see here let's zoom into the plot, so we can see that seasonality has been fitted, red lines have been used, and we can see that those swings are being you know adequately captured here, however the trend is not being captured, and that is what is resulting in higher error for this model, seasonality only model, same thing we can confirm if we you know plot this residual series so we will see that if we look at the residual series you can see

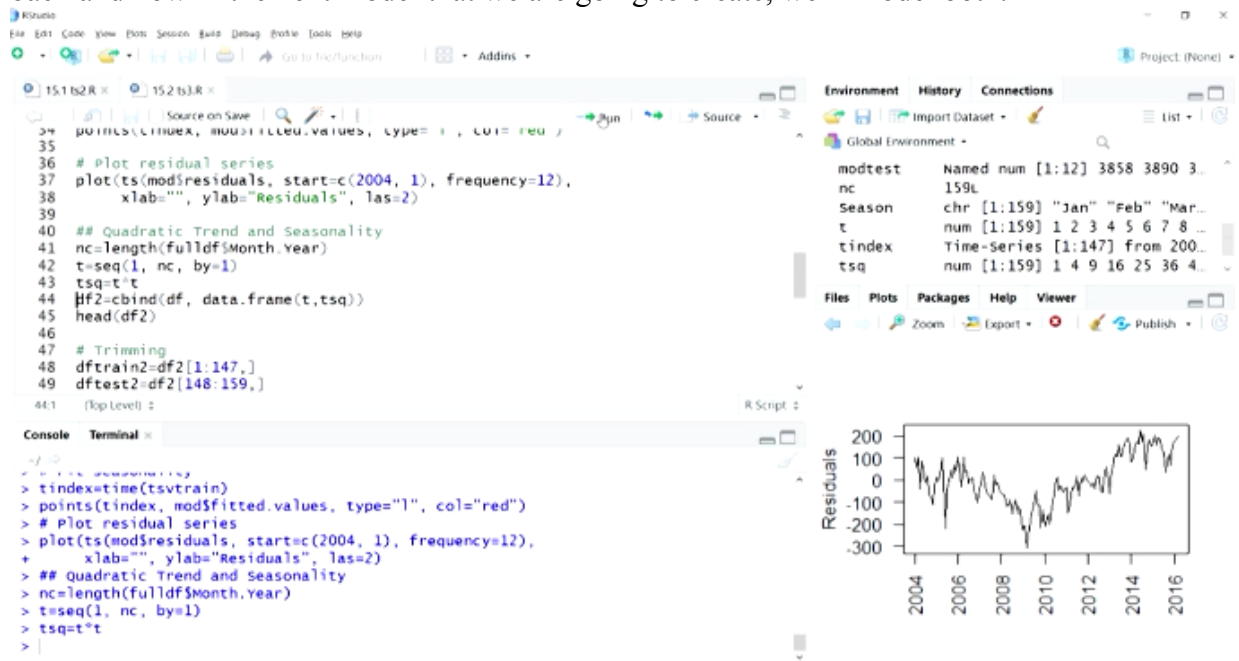


that U shaped trend that we were able to adequately capture using the quadratic model, now this is part of the residual series that means it was not captured using just the seasonality component and it has become part of residual series, so what we need to do here is that probably we need to model both trend and seasonality, so quadratic model was able to capture the U shaped trend

Regression-Based Forecasting Methods

- Modeling additive seasonality
 - If this categorical variable has m seasons
 - $m-1$ dummy variables are created to be included as predictors in the regression equation
- Modeling trend and seasonality
- Open RStudio

that was clearly present in the series, and the seasonality we have seen that monthly seasonality is present for some month it was in the dummy variables were clearly significant, so therefore we need to model both trend and seasonality the quadratic trend, and seasonality, so let's go back and now in the next model that we are going to create, we'll model both.

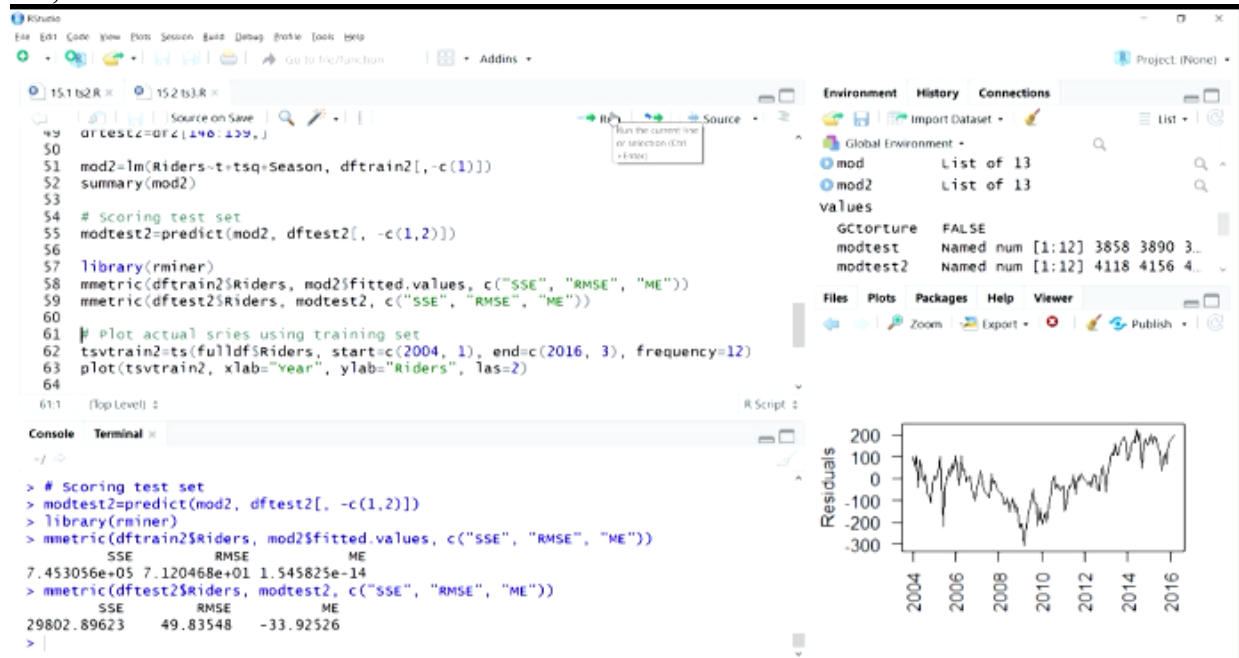


So let's first compute this two predictors T and T square, so we'll compute T and T square like we did in previous lectures as well, we'll add these two variables in our existing data frame, let's have a look at the first 6 observations, now in this data frame we have riders, season, T and T square.

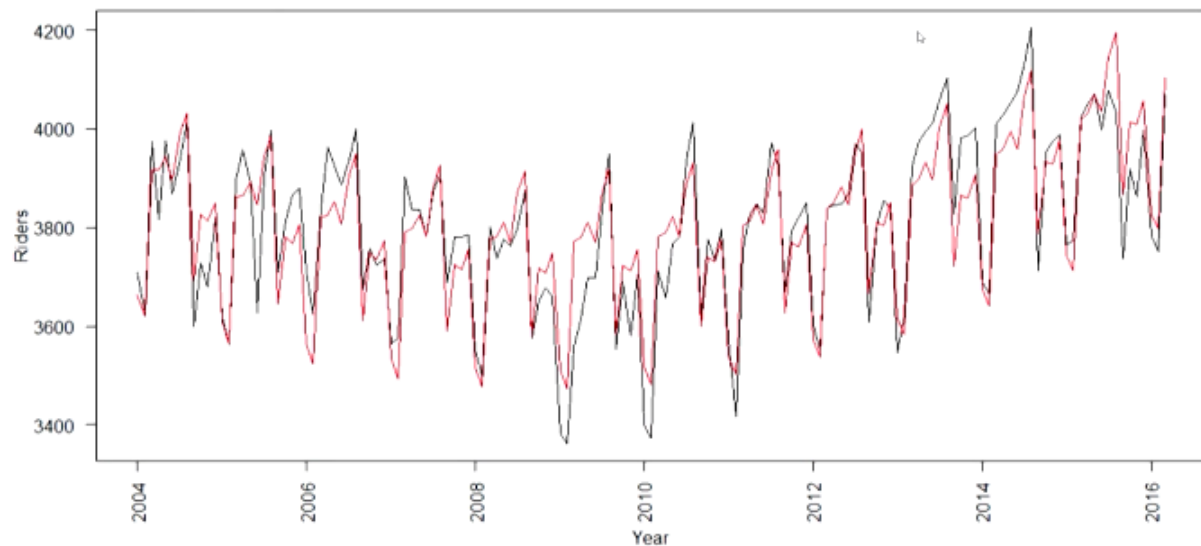


Now next step is typically trimming, training set and test set, now if we look at the model equation here, we are regression riders against T and T square + season which is the categorical

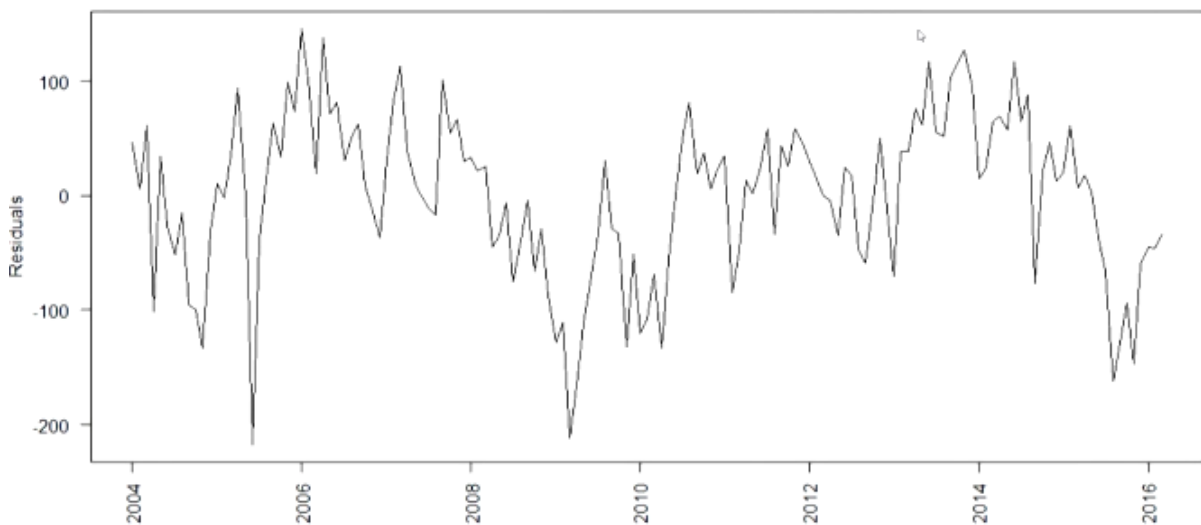
variable, now we'll build this model, let's look at the results, so we can see here September is now significant, October, November is also significant, now this is happening because of the presence of two predictors, T and T square, we can see now that July, Jan, Feb these two are earlier you know significant as well, but for July the significance level has gone up, August is significant again, so seasonality is significant, seasonality is present, now T and T square are also significant, so it seems that this model is going to perform quite well, so let's check that out, so we'll score the test set.



Now let's compute the matrix, so first training partition and then test partition, now let's have a look, you know RMSE value for training partition has also come down and RMSE value for you know test partition has been come down, so if we look at the, focus on the test partition the RMSE value has come down to 49, remember the best RMSE value till now that we had was when we you know model the quadratic trend it came down to 136 something and now when we model both quadratic trend and seasonality, now this RMSE value has come down to 49, so it looks like that the model has improved significantly and the same thing can be you know confirmed using some of these plots, so we'll create a you know time series plot for training set, so this is the plot, now let's look at this, let's fit the fitted model which has quadratic trend as well as seasonality, so we'll fit this.



Now let's zoom into this plot, now we see that the fitted model in the red line and the actual you know series in the black, so we can see the swings are being captured and the fitted model is also you know taking that shape of the actual series, so it is you know of course it is not perfectly fitting the actual series, but now it is closely resembling the actual series, the red line is closely resembling the, closely following the actual series, same we can confirm by plotting the residual series, so let's run this code, so this is the residual you know series plot.



Now if we look at the residual series plot we can see, you know no you know extra pattern or something clearly visible so you know random variation seem to be there, so it seems that we have been able to adequately capture trend and seasonality, and therefore the model, so with

this we'll stop here, and we'll continue our discussion on regression based forecasting methods in the next lecture. 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. Nibedita Bisoyi

Neetesh Kumar

Jitender Kumar

Vivek Kumar

Dharamveer Singh

Gaurav Kumar
An educational Technology cell
IIT Roorkee Production
© Copyright All Rights Reserved
WANT TO SEE MORE LIKE THIS
SUBSCRIBE