Thomas Pattara
STAT 560 Time Series Analysis

1.*Calculate the weights ψ_i, i = 1,2, ⋯ for an ARIMA(1,1,0) model.*

ARIMA (1,1,0) = $(\psi_0 + \psi_1 B + \psi_2 B^2 + \ldots)(1 - \varphi_1 B)(1-B) = 1$

$(\psi_0 - \psi_0 \varphi_1 B + \psi_1 B - \psi_1 \varphi_1 B^2 + \psi_2 B^2 - \psi_2 \varphi_1 B^3 + \ldots)(1-B) = 1$

$(\psi_0 - \psi_0 B - \psi_0 \varphi_1 B + \psi_0 \varphi_1 B^2 + \psi_1 B - \psi_1 B^2 - \psi_1 \varphi_1 B^2 + \psi_1 \varphi_1 B^3 + \psi_2 B^2 - \psi_2 B^3 - \ldots) = 1$

Next we will calculate the weights for B^0, B^1, and B^2.

B^0 : $\psi_0 = 1$

B^1 : $- \psi_0 - \psi_0 \varphi_1 + \psi_1 = 0$

$-1 - \varphi_1 + \psi_1 = 0$

$\psi_1 = \varphi_1 + 1$

B^2: $\psi_0 \varphi_1 - \psi_1 - \psi_1 \varphi_1 + \psi_2 = 0$

$\varphi_1 - (\varphi_1 + 1) - (\varphi_1 + 1)\varphi_1 + \psi_2 = 0$

$-1 - \varphi_1^2 - \varphi_1 + \psi_2 = 0$

$\psi_2 = \varphi_1^2 + \varphi_1 + 1$

Therefore

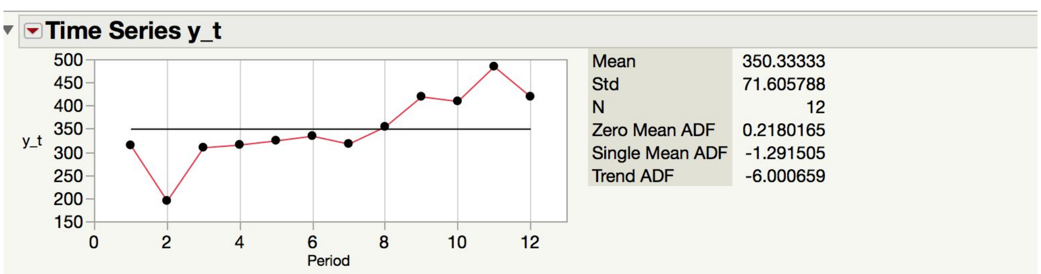
$$\psi_i = \sum_{j=0}^{i} \varphi_1^j$$

2. *Consider the data shown in Table E4.4 below.*

2.1) *Fit an ARIMA model to this time series, excluding the last 12 observations.*

TABLE E4.4 Data for Exercise 4.8

Period	y_t	Period	y_t
1	315	13	460
2	195	14	395
3	310	15	390
4	316	16	450
5	325	17	458
6	335	18	570
7	318	19	520
8	355	20	400
9	420	21	420
10	410	22	580
11	485	23	475
12	420	24	560



Time Series Basic Diagnostics

Lag	AutoCorr	Ljung-Box Q	p-Value	Lag	Partial
0	1.0000	.	.	0	1.0000
1	0.5955	5.4160	0.0200*	1	0.5955
2	0.3351	7.3025	0.0260*	2	-0.0302
3	0.1502	7.7238	0.0521	3	-0.0579
4	-0.0372	7.7528	0.1011	4	-0.1500
5	-0.0462	7.8041	0.1674	5	0.0848
6	-0.1450	8.3927	0.2107	6	-0.1675
7	-0.3215	11.8657	0.1051	7	-0.2693
8	-0.3178	16.1073	0.0409*	8	-0.0035
9	-0.4199	25.9813	0.0021*	9	-0.2504
10	-0.2532	31.3671	0.0005*	10	0.1857
11	-0.0400	31.6360	0.0009*	11	0.0921
12	0.0000	.	.	12	-0.0599
13	0.0000	.	.	13	-0.1637
14	0.0000	.	.	14	-0.0058
15	0.0000	.	.	15	-0.0116
16	0.0000	.	.	16	-0.2362
17	0.0000	.	.	17	0.0292
18	0.0000	.	.	18	-0.0297
19	0.0000	.	.	19	0.0419
20	0.0000	.	.	20	0.0293
21	0.0000	.	.	21	-0.0227
22	0.0000	.	.	22	-0.1175
23	0.0000	.	.	23	-0.0866

Figure 1: Time series plot excluding the last 12 observations

Model Comparison

Model	DF	Variance	AIC	SBC	RSquare	-2LogLH	Weights	MAPE	MA
ARIMA(1, 3, 1)	6	7143.4894	112.45265	113.04432	-4.97	106.45265	0.638863	26.753480	92.17196
ARIMA(1, 2, 1)	7	2774.1204	114.88609	115.79385	-1.38	108.88609	0.189231	17.579141	59.23683
ARIMA(1, 1, 1)	8	1453.6122	117.59395	118.78764	0.525	111.59395	0.048864	12.732458	35.20449
I(3)	8	26264.25	118.06452	118.26175	-6.35	116.06452	0.038619	33.063411	121.5555
IMA(1, 1)	9	1866.947	118.34682	119.14261	0.545	114.34682	0.033535	11.958768	33.47299
ARI(1, 1)	9	2136.4652	118.78429	119.58008	0.435	114.78429	0.026947	14.857633	42.75317
ARIMA(2, 2, 2)	5	3683.2933	120.68355	122.19647	-1.59	110.68355	0.010425	20.181871	67.17500
ARIMA(2, 1, 2)	6	1761.2459	121.19162	123.18109	0.526	111.19162	0.008087	12.588455	34.31636
I(2)	9	11521.389	122.84477	123.14736	-2.19	120.84477	0.003538	21.228695	76.00000
I(1)	10	4267.2727	124.11427	124.51216	0.291	122.11427	0.001876	15.197573	46.23140
AR(1)	10	3743.7611	135.06135	136.03116	0.382	131.06135	0.000008	13.438281	40.37090
MA(1)	10	4292.6146	136.58098	137.55079	0.277	132.58098	0.000004	14.482680	43.87499
ARMA(1, 1)	9	4008.1165	136.74231	138.19703	0.395	130.74231	0.000003	13.291427	40.09422

Figure 2: Model comparison of time series in figure 1.

Model: ARIMA(1, 1, 1)

Model Summary

DF	8	Stable	Yes
Sum of Squared Errors	11628.8974	Invertible	Yes
Variance Estimate	1453.61218		
Standard Deviation	38.1262662		
Akaike's 'A' Information Criterion	117.59395		
Schwarz's Bayesian Criterion	118.787636		
RSquare	0.52476097		
RSquare Adj	0.40595122		
MAPE	12.7324582		
MAE	35.2044946		
-2LogLikelihood	111.59395		

Parameter Estimates

Term	Lag	Estimate	Std Error	t Ratio	Prob>|t|	Constant Estimate	Mu
AR1	1	-0.59147	0.298071	-1.98	0.0825	30.8973783	19.4143487
MA1	1	0.99996	0.309308	3.23	0.0120*		
Intercept	0	19.41435	1.981804	9.80	<.0001*		

Forecast

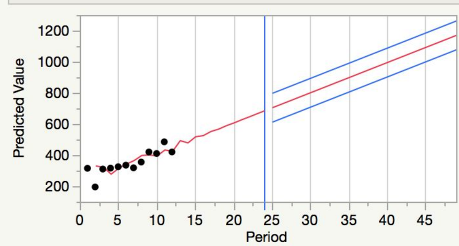


Figure 3: Summary Statistics for ARIMA(1,1,1)

Excluding the last 12 observations, we compare the remaining 12 observations to fit an ARIMA model. Comparing the AIC values of the different models in Figure 2, ARIMA(1,1,1) is the most suitable model. This can be confirmed from the ACF and PACF plot in Figure 1, PACF cut off at the first lag and ACF has an exponential decay, but AR(1) has a bigger AIC than ARIMA(1,1,1), hence we chose ARIMA(1,1,1).

2.2). *Investigate model adequacy (Diagnostic Checking).*

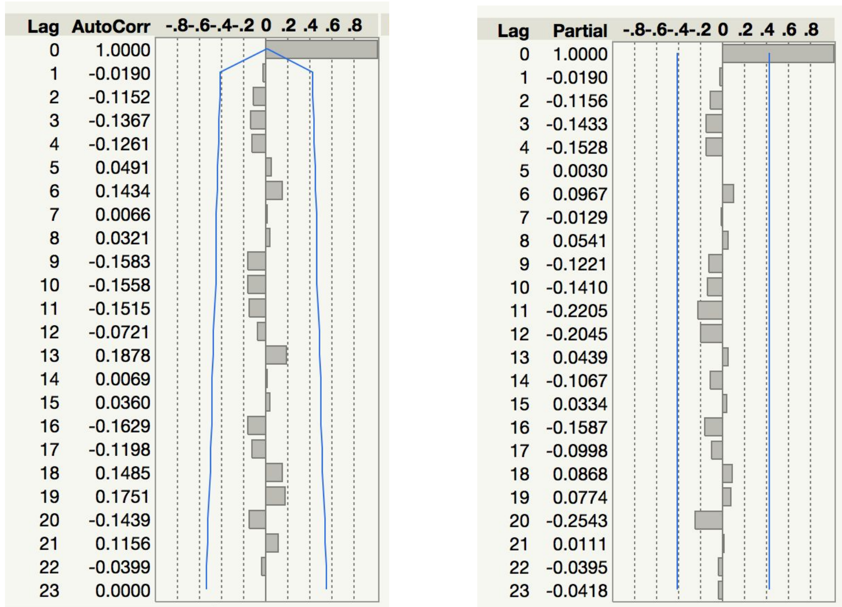


Figure 4 : The sample ACF and PACF of the residuals for the ARIMA(1,1,1) model in Table E4.4.

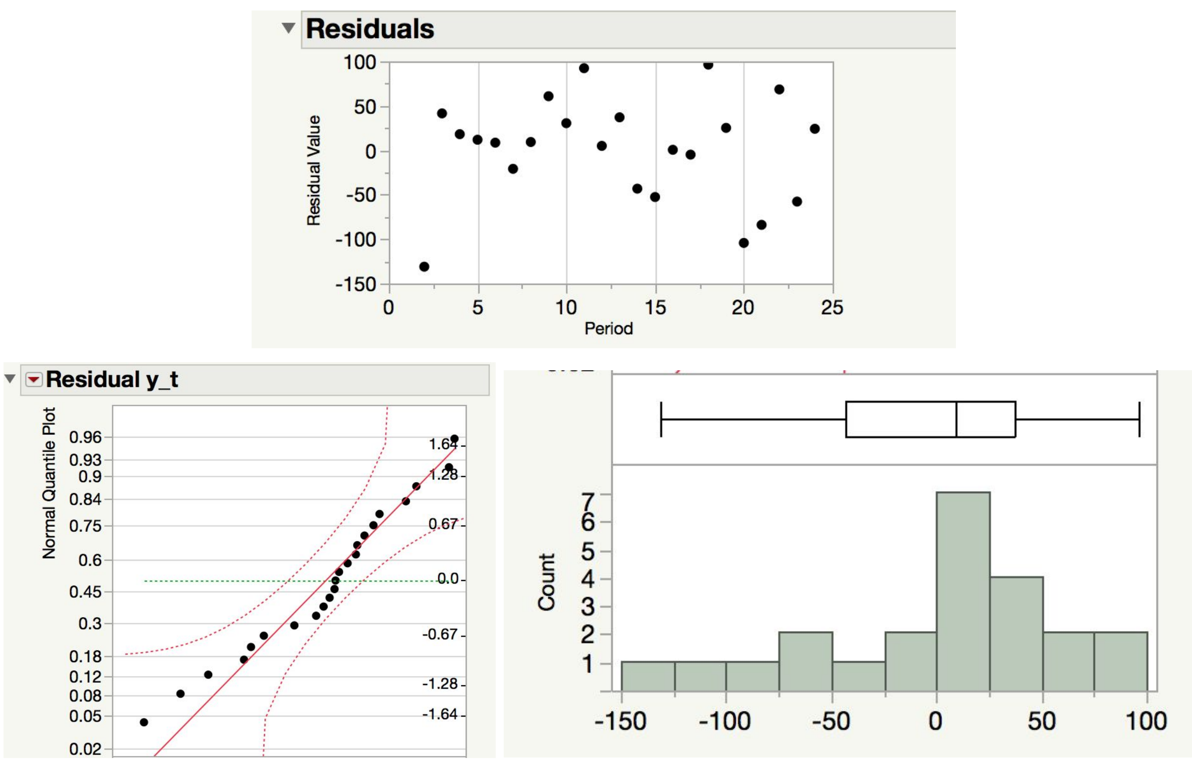


Figure 5 : Residual plots for the ARIMA(1,1,1) model in Table E4.4.

2.3). *Forecast the last 12 observations.*

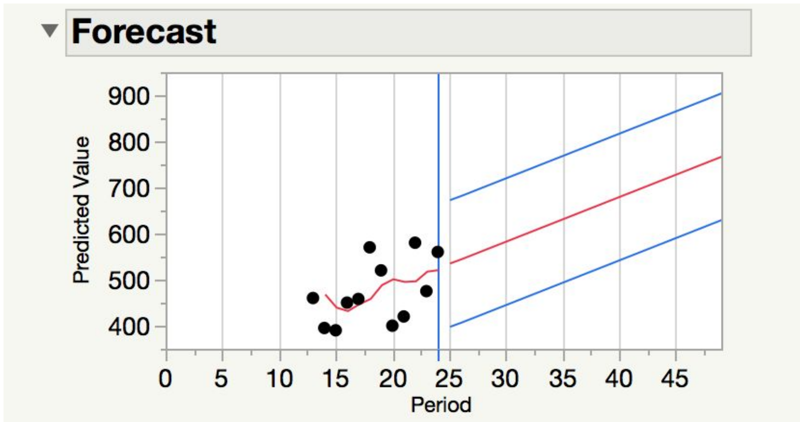
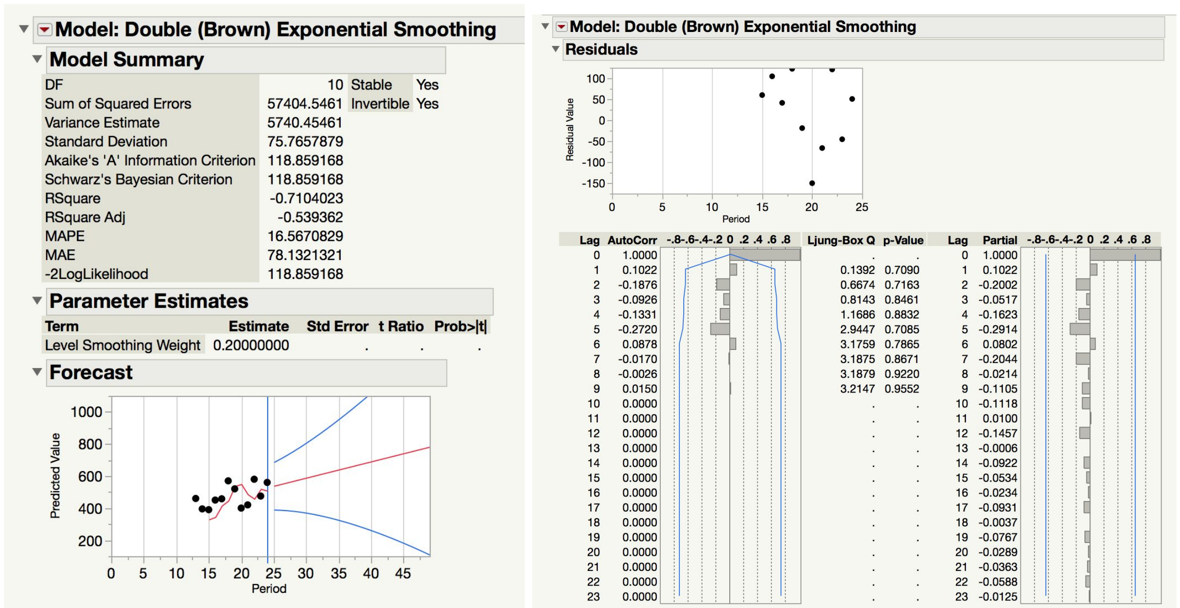


Figure 6 : ARIMA(1,1,1) Forecast of last 10 observation in Table E4.4

When looking at the ACF plot in Figure 4, we can see that the autocorrelation does not differ much from zero at any lags other than 1. Therefore we can conclude that our model is adequate.

2.4). *In exercise 4.8, you were asked to use 2nd (double) exponential smoothing methods to smooth the data, and to forecast the last 12 observations. (Hint: The optimum smoothing constant is chosen by finding the value of \lambda that minimizes the error sum of squares*

*Reminder: use the error sum of squares after **2nd** smoothing, **not the 1st** smoothing. To simply the question, you can assume the smoothing lambda that used in the 1st and 2nd smoothing are same.)*



Model: Double (Brown) Exponential Smoothing

Model Summary

DF	10	Stable	Yes
Sum of Squared Errors	57404.5461	Invertible	Yes
Variance Estimate	5740.45461		
Standard Deviation	75.7657879		
Akaike's 'A' Information Criterion	118.859168		
Schwarz's Bayesian Criterion	118.859168		
RSquare	-0.7104023		
RSquare Adj	-0.539362		
MAPE	16.5670829		
MAE	78.1321321		
-2LogLikelihood	118.859168		

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Level Smoothing Weight	0.20000000	.	.	.

Forecast

Model: Double (Brown) Exponential Smoothing

Residuals

Lag	AutoCorr	Ljung-Box Q	p-Value	Lag	Partial
0	1.0000	.	.	0	1.0000
1	0.1022	0.1392	0.7090	1	0.1022
2	-0.1876	0.6674	0.7163	2	-0.2002
3	-0.0926	0.8143	0.8461	3	-0.0517
4	-0.1331	1.1686	0.8832	4	-0.1623
5	-0.2720	2.9447	0.7085	5	-0.2914
6	0.0878	3.1759	0.7865	6	0.0802
7	-0.0170	3.1875	0.8671	7	-0.2044
8	-0.0026	3.1879	0.9220	8	-0.0214
9	0.0150	3.2147	0.9552	9	-0.1105
10	0.0000	.	.	10	-0.1118
11	0.0000	.	.	11	0.0100
12	0.0000	.	.	12	-0.1457
13	0.0000	.	.	13	-0.0006
14	0.0000	.	.	14	-0.0922
15	0.0000	.	.	15	-0.0534
16	0.0000	.	.	16	-0.0234
17	0.0000	.	.	17	-0.0931
18	0.0000	.	.	18	-0.0037
19	0.0000	.	.	19	-0.0767
20	0.0000	.	.	20	-0.0289
21	0.0000	.	.	21	-0.0363
22	0.0000	.	.	22	-0.0588
23	0.0000	.	.	23	-0.0125

Figure 7 : Double exponential smoothing of data in Table E4.4

2.5). *Compare the ARIMA and exponential smoothing forecasts. Which forecasting method do you prefer?*

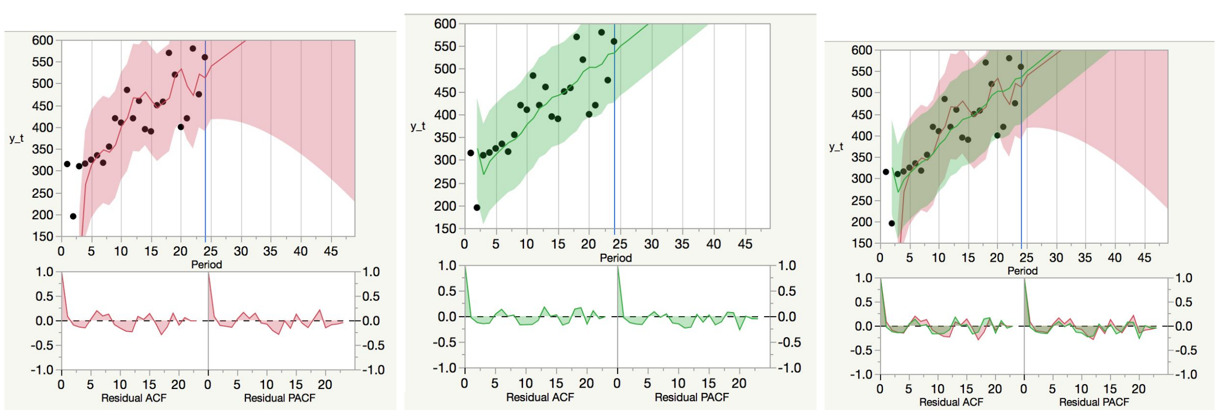


Figure 8 : Exponential smoothing, ARIMA(1,1,1), and both together respectively.

In Figure 8 above, we are comparing exponential smoothing forecast with the lambada of 0.2 to the ARIMA(1,1,1) forecasting model. We prefer the ARIMA(1,1,1) forecasting model because the forecast seems to better fit the observations. Also when comparing the residual plots for both forecasting models the ARIMA(1,1,1) model seems to differ from 0 less than the exponential smoothing forecast. Therefore we can conclude that the ARIMA(1,1,1) model is more adequate.

2.6). *How would prediction intervals be obtained for the ARIMA forecasts?*
Your answer should include three parts:

2.6.1 Specify the ψ_i, i = 1,2, ⋯. You can use the results that you obtained from question 1 if you are using the same model as question 1. Write down the general formula of $var[e_T(\tau)]$ and 100(1 − α) percent prediction interval for

$y_{T+\tau}$. You don't have to simplify your formula or plug ψ_i, i = 1,2, ⋯ into the formulas. You can do the calculation by R.

$$
\begin{aligned}
var[e_T(\tau)] &= \sigma^2 \sum_{i=0}^{\tau-1} \psi_i^{\,2} \\
&= \sigma^2 \sum_{i=0}^{\tau-1} \varphi_1^{\,2i} \\
&= \sigma^2 \left(\frac{1-\varphi_1^{2i}}{1-\varphi_1^{2}}\right)
\end{aligned}
$$

Prediction interval for $y_{T+\tau} = \hat{y}_{T+\tau}(T) \pm Z_{\alpha/2} \sqrt{var[e_T(\tau)]}$

$$= \hat{y}_{T+\tau}(T) \pm Z_{\alpha/2} \sqrt{\sigma^2 \left(\frac{1-\varphi_1^{2i}}{1-\varphi_1^{2}}\right)}$$

2.6.2 *List the 95% PI for the next 10 periods (25-34).*

Table 1

	Actual y_t	Period	Predicted y_t	Std Err Pred y_t	Residual y_t	Upper CL (0.95) y_t	Lower CL (0.95) y_t
1	315	1	•	•	•	•	•
2	195	2	326.09102373	56.232074449	-131.0910237	436.30386442	215.87818303
3	310	3	268.52047589	56.232074449	41.479524106	378.73331659	158.3076352
4	316	4	297.9326138	56.232074449	18.067386195	408.1454545	187.71977311
5	325	5	313.18847422	56.232074449	11.811525784	423.40131491	202.97563352
6	335	6	326.49615407	56.232074449	8.503845935	436.70899476	216.28331337
7	318	7	338.92942285	56.232074449	-20.92942285	449.14226354	228.71658215
8	355	8	345.73711007	56.232074449	9.2628899316	455.94995076	235.52426937
9	420	9	359.20344798	56.232074449	60.79655202	469.41628868	248.99060728
10	410	10	379.53839315	56.232074449	30.461606852	489.75123384	269.32555245
11	485	11	392.64279382	56.232074449	92.357206179	502.85563452	282.42995313
12	420	12	415.09267507	56.232074449	4.907324928	525.30551577	304.87983438
13	460	13	422.97187121	56.232074449	37.028128787	533.18471191	312.75903052
14	395	14	438.26385148	56.232074449	-43.26385148	548.47669217	328.05101078
15	390	15	442.66908149	56.232074449	-52.66908149	552.88192218	332.45624079
16	450	16	449.50707253	56.232074449	0.4929274688	559.71991323	339.29423184
17	458	17	462.95386535	56.232074449	-4.953865348	573.16670604	352.74102465
18	570	18	473.60823788	56.232074449	96.391762121	583.82107857	363.39539718
19	520	19	494.82207787	56.232074449	25.177922129	605.03491857	384.60923718
20	400	20	504.32703808	56.232074449	-104.3270381	614.53987877	394.11419738
21	420	21	503.99549693	56.232074449	-83.99549693	614.20833762	393.78265623
22	580	22	511.5291942	56.232074449	68.470805797	621.7420349	401.31635351
23	475	23	532.79766123	56.232074449	-57.79766123	643.01050193	422.58482054
24	560	24	535.867589	56.232074449	24.132411001	646.08042969	425.6547483
25	•	25	551.4736271	56.232074449	•	661.6864678	441.26078641
26	•	26	561.63203201	56.295682952	•	671.96954308	451.29452094
27	•	27	572.67871867	56.295831408	•	683.01652071	462.34091663
28	•	28	583.7676346	56.295832011	•	694.10543782	473.42983137
29	•	29	594.85855812	56.295832066	•	705.19636145	484.52075479
30	•	30	605.94957708	56.29583211	•	716.2873805	495.61177367
31	•	31	617.04060058	56.295832153	•	727.37840408	506.70279708
32	•	32	628.1316243	56.295832196	•	738.46942788	517.79382072
33	•	33	639.22264803	56.295832239	•	749.56045169	528.88484436
34	•	34	650.31367175	56.295832282	•	760.65147551	539.975868

The above table displays the 95% CI for the next 10 observations.

2.6.3 *Plot the time series plot and forecasts.*

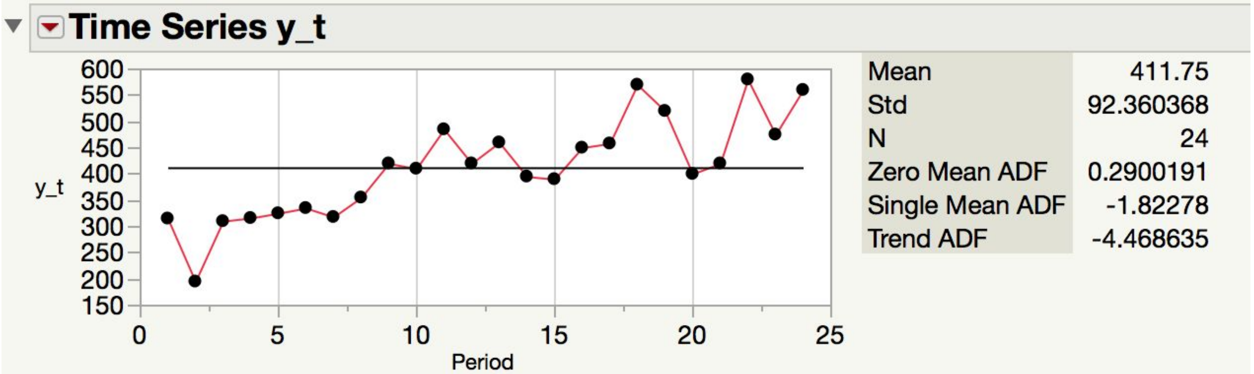


Figure 9 : Time series plot of data in Table E4.4

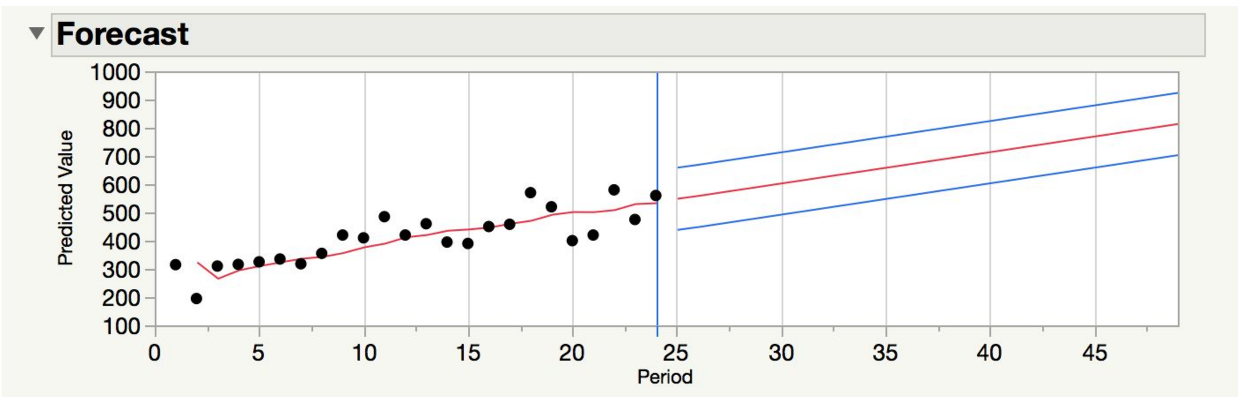


Figure 10 : Forecast of data in Table E4.4

Appendix

```
E4.4 <- read_excel("/Users/Thomas/Documents/E4_4.xlsx")

firstsmooth<-function(y,lambda,start=y[1]){
  ytilde<-y
  ytilde[1]<-lambda*y[1]+(1-lambda)*start
  for (i in 2:length(y)){
    ytilde[i]<-lambda*y[i]+(1-lambda)*ytilde[i-1]
  }
  ytilde
```

```
}

measacc.fs<- function(y,lambda){
  out1<- firstsmooth(y,lambda) #first smoothing
  out2<- firstsmooth(out1,lambda) # second smoothing
  out<- 2*out1-out2 # forecast
  T<-length(y)
  #Smoothed version of the original is the one step ahead prediction
  #Hence the predictions (forecasts) are given as
  pred<-c(y[1],out[1:(T-1)])
  prederr<- y-pred
  SSE<-sum(prederr^2)
  MAPE<-100*sum(abs(prederr/y))/T
  MAD<-sum(abs(prederr))/T
  MSD<-sum(prederr^2)/T
  ret1<-c(SSE,MAPE,MAD,MSD)
  names(ret1)<-c("SSE","MAPE","MAD","MSD")
  return(ret1)
}

lambda.vec<-seq(0.1, 1, 0.05)
sse.speed<-function(sc){measacc.fs(E4.4$y,sc)[1]}
sse.vec<-sapply(lambda.vec, sse.speed)
opt.lambda<-lambda.vec[sse.vec == min(sse.vec)]
plot(lambda.vec, sse.vec, type="b", main = "SSE vs. lambda\n",
   xlab='lambda\n',ylab='SSE')
abline(v=opt.lambda, col = 'red')

mtext(text = paste("SSE min = ", round(min(sse.vec),2), "\n lambda
           = ", opt.lambda))
```