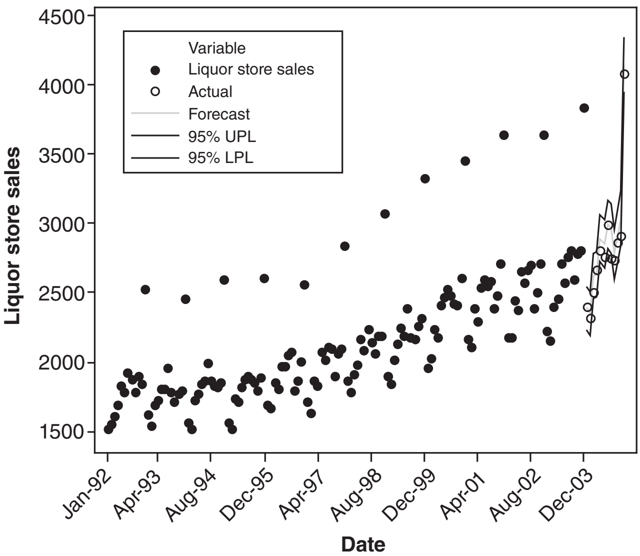


FIGURE 4.32 Forecasts for the liquor store sales for 2004 using the multiplicative model.

where significantly higher than anticipated counts of influenza-like illness might signal a potential bioterrorism attack.

As an example of such syndromic data, Fricker (2013) describes daily counts of respiratory and gastrointestinal complaints for more than 2 $^1/_2$ years at several hospitals in a large metropolitan area. Table 4.12 presents the respiratory count data from one of these hospitals. There are 980 observations. Fifty observations were missing from the original data set. The missing values were replaced with the last value that was observed on the same day of the week. This type of data imputation is a variation of "Hot Deck Imputation" discussed in Section 1.4.3 and in Fricker (2013). It is also sometimes called last observation (or Value) carried forward (LOCF). For additional discussion see the web site: http://missingdata.lshtm.ac.uk/.

Figure 4.33 is a time series plot of the respiratory syndrome count data in Table 4.12. This plot was constructed using the Graph Builder feature in JMP. This software package overlays a smoothed curve on the data. The curve is fitted using **locally weighted regression**, often called **loess**. This is a variation of kernel regression that uses a weighted average of the data in a local neighborhood around a specific location to determine the value to plot at that location. Loess usually uses either first-order linear regression or a quadratic regression model for the weighted least squares fit. For more information on kernel regression and loess see Montgomery, et al. (2012).

TABLE 4.12 Counts of Respiratory Complaints at a Metropolitan Hospital

Day	Count	Day	Count	Day	Count	Day	Count	Day	Count	Day	Count	Day	Count	Day	Count	Day	Count	Day	Count
1	17	101	30	201	31	301	12	401	28	501	35	601	26	701	19	801	41	901	29
2	29	102	21	202	23	302	16	402	26	502	27	602	31	702	12	802	50	902	26
3	31	103	32	203	13	303	24	403	28	503	33	603	23	703	17	803	42	903	36
4	34	104	32	204	18	304	21	404	29	504	30	604	24	704	22	804	56	904	31
5	18	105	43	205	36	305	14	405	33	505	30	605	27	705	20	805	36	905	25
6	43	106	25	206	23	306	15	406	36	506	29	606	24	706	22	306	51	906	31
7	34	107	32	207	22	307	23	407	62	507	30	607	31	707	21	807	40	907	32
8	23	108	31	208	23	308	10	403	31	508	22	608	29	708	24	808	29	908	30
9	23	109	33	209	26	309	16	409	30	509	40	609	36	709	16	809	61	909	31
10	39	110	40	210	22	310	11	410	31	510	40	610	31	710	14	810	42	910	29
11	25	111	37	211	21	311	16	411	27	511	41	611	30	711	14	811	56	911	30
12	15	112	34	212	25	312	16	412	35	512	34	612	27	712	30	812	60	912	35
13	29	113	29	213	20	313	12	413	45	513	30	613	27	713	24	813	38	913	24
14	20	114	50	214	18	314	23	414	37	514	33	614	25	714	25	814	52	914	27
15	21	115	27	215	26	315	10	415	23	515	17	615	34	715	17	815	32	915	22
16	22	116	28	216	32	315	15	416	31	516	32	616	33	716	27	816	43	916	33
17	24	117	23	217	41	317	11	417	33	517	40	617	36	717	25	817	54	917	29
18	19	118	27	218	30	318	17	418	27	518	30	618	26	718	14	818	36	913	37
19	28	119	27	219	34	319	13	419	28	519	27	619	20	719	25	819	51	919	29
20	29	120	41	220	38	320	14	420	46	520	30	620	27	720	25	820	57	920	32
21	26	121	29	221	22	321	20	421	39	521	38	621	25	721	26	821	48	921	27
22	22	122	26	222	35	322	10	422	53	522	22	622	36	722	20	822	70	922	22
23	21	123	28	223	36	323	15	423	33	523	27	623	30	723	21	823	48	923	33

(continued)

TABLE 4.12 (*Continued*)

Day	Count	Day	Count	Day	Count	Day	Count	Day	Count	Day	Count	Day	Count	Day	Count	Day	Count	Day	Count
24	29	124	30	224	37	324	14	424	32	524	19	624	39	724	29	824	54	924	29
25	25	125	49	225	27	325	6	425	45	525	19	625	26	725	16	825	36	925	37
26	20	126	43	226	23	326	17	426	21	526	33	626	20	726	24	826	51	926	29
27	20	127	27	227	31	327	17	427	47	527	45	627	27	727	42	827	52	927	20
28	29	128	32	228	39	328	17	423	23	528	34	628	36	728	44	828	48	928	13
29	29	129	13	229	39	329	23	429	39	529	27	629	43	729	34	829	70	929	27
30	32	130	26	230	31	330	9	430	32	530	31	630	46	730	33	830	48	930	23
31	16	131	34	231	43	331	21	431	27	531	19	631	33	731	26	831	57	931	17
32	25	132	27	232	35	332	13	432	29	532	22	632	26	732	29	832	38	932	26
33	20	133	33	233	41	333	13	433	37	533	23	633	33	733	33	833	44	933	23
34	22	134	42	234	24	334	14	434	32	534	13	634	24	734	34	834	34	934	27
35	27	135	29	235	39	335	25	435	28	535	29	635	23	735	42	835	50	935	28
36	32	136	29	236	44	336	15	436	42	536	13	636	51	736	43	836	39	936	21
37	23	137	29	237	35	337	18	437	33	537	20	637	35	737	33	837	65	937	20
38	31	138	28	238	30	338	21	438	36	538	20	638	26	738	31	838	55	938	25
39	22	139	35	239	29	339	18	439	25	539	23	639	32	739	30	839	46	939	30
40	21	140	33	240	13	340	12	440	19	540	17	640	29	740	35	840	57	940	13
41	27	141	3S	241	23	341	10	441	34	541	31	641	24	741	34	841	43	941	19
42	37	142	23	242	19	342	10	442	34	542	21	642	18	742	43	842	50	942	20
43	28	143	28	243	24	343	17	443	33	543	29	643	36	743	21	843	39	943	27
44	41	144	23	244	19	344	12	444	26	544	20	644	15	744	42	844	55	944	14
45	45	145	31	245	27	345	24	445	43	545	21	645	33	745	30	845	38	945	21
46	40	146	29	246	20	345	22	446	31	546	25	646	21	746	29	846	29	946	32
47	32	147	24	247	19	347	14	447	30	547	35	647	25	747	29	347	32	947	18
48	45	148	22	248	28	348	14	448	41	548	24	648	25	748	41	848	27	948	25

49	48	149	30	249	19	349	9	449	15	549	25	649	19	749	35	849	22	949	13
50	51	150	21	250	29	350	19	450	23	550	23	650	23	750	29	850	23	950	25
51	51	151	24	251	24	351	15	451	25	551	27	651	18	751	37	851	25	951	19
52	21	152	21	252	33	352	9	452	27	552	35	652	26	752	31	852	19	952	27
53	43	153	30	253	20	353	18	453	40	553	36	653	27	753	24	853	29	953	27
54	42	154	25	254	29	354	17	454	40	554	33	654	11	754	47	854	34	954	18
55	56	155	17	255	17	355	15	455	34	555	27	655	20	755	3	855	27	955	25
55	51	155	22	255	19	355	21	455	42	555	33	656	13	755	34	355	30	956	26
57	51	157	18	257	23	357	22	457	12	557	25	657	20	757	35	857	30	957	39
58	60	158	19	258	26	358	17	458	24	558	32	658	23	758	39	858	24	958	59
59	35	159	20	259	25	359	21	459	20	559	23	659	19	759	29	859	33	959	34
60	43	160	22	260	32	360	26	460	26	560	42	660	21	760	41	860	29	960	34
61	42	161	39	261	21	361	23	461	46	561	25	661	21	761	36	861	36	961	24
62	55	162	35	262	15	362	20	462	35	562	33	662	29	762	50	862	29	962	25
63	46	163	29	263	20	363	28	463	46	563	19	663	18	763	33	863	27	963	40
64	49	164	24	264	19	364	34	464	33	564	40	664	25	764	38	864	32	964	19
65	40	165	22	265	13	365	23	465	27	565	35	665	24	765	40	865	30	965	35
66	33	166	26	266	25	366	20	466	35	566	36	666	19	766	41	866	23	966	34
67	45	167	27	267	19	367	37	467	33	567	33	667	15	767	34	867	25	967	33
68	37	168	28	268	9	368	22	468	29	568	25	668	23	768	42	868	23	968	29
69	44	169	36	269	20	369	32	469	45	569	33	669	14	769	40	869	29	969	23
70	50	170	31	270	20	370	41	470	18	570	33	670	16	770	50	870	26	970	29
71	37	171	31	271	21	371	35	471	21	571	27	671	16	771	30	871	29	971	25
72	36	172	34	272	21	372	41	472	35	572	33	672	22	772	34	872	22	972	19
73	43	173	19	273	20	373	43	473	39	573	33	673	13	773	28	873	16	973	34
74	49	174	37	274	13	374	33	474	40	574	39	674	19	774	21	874	25	974	37
75	4C	175	39	275	25	375	32	475	33	575	30	675	23	775	24	875	26	975	34

(continued)

TABLE 4.12 ***(Continued)***

Day	Count	Day	Count	Day	Count	Day	Count	Day	Count	Day	Count	Day	Count	Day	Count	Day	Count	Day	Count
76	65	176	32	276	29	376	28	476	35	576	33	676	14	776	37	876	25	976	29
77	49	177	36	277	16	377	42	477	20	577	26	677	16	777	44	877	24	977	27
78	49	178	42	278	18	378	27	478	36	578	26	678	10	778	39	878	29	978	18
79	34	179	31	279	32	379	25	479	34	579	23	679	14	779	37	879	34	979	26
80	33	180	28	280	32	380	32	480	35	580	24	680	13	780	35	880	35	980	28
81	29	181	35	281	19	381	27	481	36	581	32	681	15	781	32	881	29		
82	32	182	36	282	24	382	35	482	29	582	24	682	15	782	41	882	39		
83	57	183	35	283	18	383	26	483	19	583	32	683	11	783	41	883	31		
84	43	184	32	284	20	384	32	484	36	584	41	634	11	784	51	884	26		
85	40	185	26	285	20	385	42	485	35	585	26	685	18	785	43	885	24		
86	46	186	29	286	20	386	38	486	31	586	28	686	16	786	35	886	31		
87	33	187	25	287	24	387	36	487	23	587	25	687	18	787	33	887	24		
88	30	188	23	288	15	388	26	488	31	588	29	688	15	788	33	888	29		
89	41	189	29	289	22	389	26	489	29	589	40	689	16	789	31	889	26		
90	38	190	29	290	16	390	24	490	44	590	34	690	11	790	43	890	45		
91	29	191	26	291	14	391	30	491	42	591	41	691	11	791	45	891	36		
92	41	192	18	292	17	392	32	492	31	592	37	692	23	792	43	892	29		
93	28	193	19	293	15	393	14	493	31	593	36	693	20	793	42	893	22		
94	47	194	17	294	8	394	27	494	24	594	26	694	18	794	36	894	31		
95	42	195	22	295	23	395	26	495	30	595	42	695	24	795	34	895	38		
96	34	196	25	296	17	396	25	496	26	596	40	696	14	796	30	896	36		
97	40	197	33	297	13	397	23	497	26	597	34	697	22	797	46	897	33		
98	35	198	10	298	15	398	27	498	39	598	41	698	16	798	54	898	34		
99	40	199	25	299	15	399	36	499	35	599	37	699	26	799	52	899	34		
100	24	200	25	300	13	400	40	500	34	600	36	700	17	800	39	900	25		

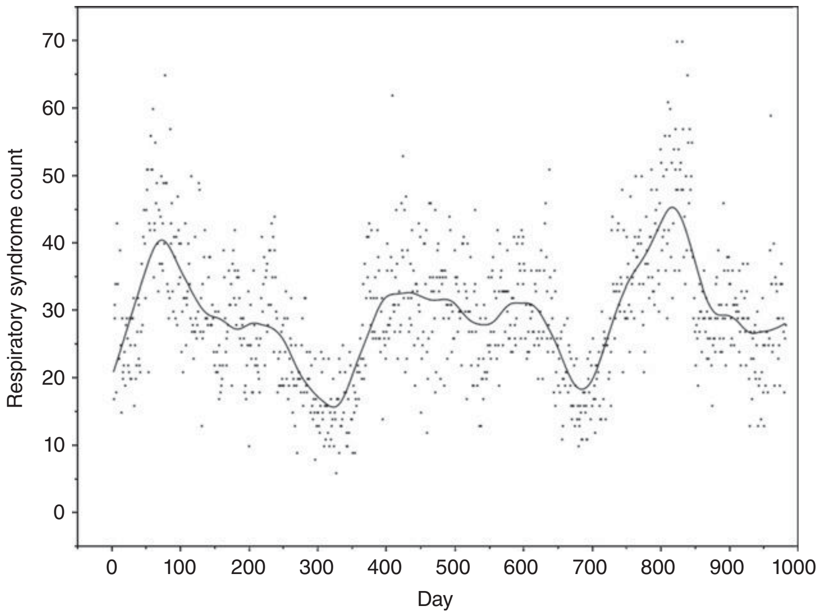


FIGURE 4.33 Time series plot of daily respiratory syndrome count, with kernel-smoothed fitted line. ($\alpha = 0.1$).

Over the 2 $^1/_2$ year period, the daily counts of the respiratory syndrome appear to follow a weak seasonal pattern, with the highest peak in November–December (late fall), a secondary peak in March–April, and then decreasing to the lowest counts in June–August (summer). The amplitude, or range within a year, seems to vary, but counts do not appear to be increasing or decreasing over time.

Not immediately evident from the time series plots is a potential day effect. The box plots of the residuals from the loess smoothed line in Figure 4.33 are plotted in Figure 4.34 versus day of the week. These plots exhibit variation that indicates slightly higher-than-expected counts on Monday and slightly lower-than-expected counts on Thursday, Friday, and Saturday.

The exponential smoothing procedure in JMP was applied to the respiratory syndrome data. The results of first-order or simple exponential smoothing are summarized in Table 4.13 and Figure 4.35, which plots only the last 100 observations along with the smoothed values. JMP reported the value of the smoothing constant that produced the minimum value of the error sum of squares as $\lambda = 0.21$. This value also minimizes the AIC and BIC criteria, and results in the smallest values of the mean absolute

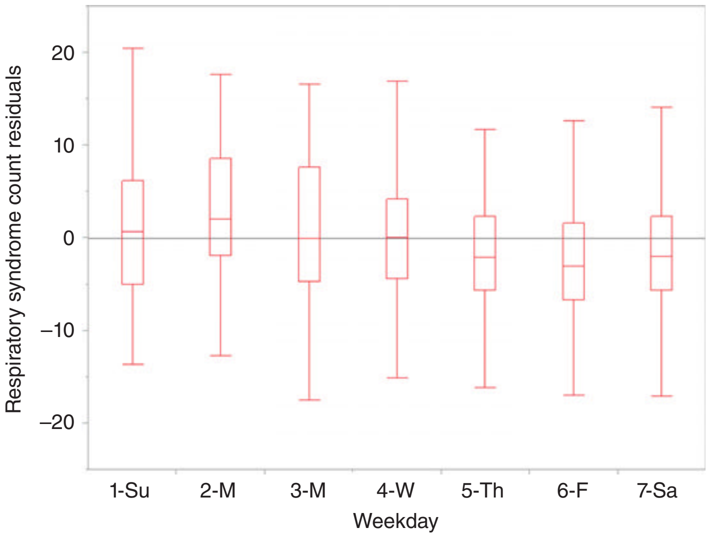


FIGURE 4.34 Box plots of residuals from the kernel-smoothed line fit to daily respiratory syndrome count.

prediction error and the mean absolute, although there is very little difference between the optimal value of $\lambda = 0.21$ and the values $\lambda = 0.1$ and $\lambda = 0.4$.

The results of using second-order exponential smoothing are summarized in Table 4.14 and illustrated graphically for the last 100 observations in Figure 4.36. There is not a lot of difference between the two procedures, although the optimal first-order smoother does perform slightly better and the larger smoothing parameters in the double smoother perform more poorly.

Single and double exponential smoothing do not account for the apparent mild seasonality observed in the original time series plot of the data.

TABLE 4.13 First-Order Simple Exponential Smoothing Applied to the Respiratory Data

Model	Variance	AIC	BIC	MAPE	MAE
First-Order Exponential (min SSE, $\lambda = 0.21$)	52.66	6660.81	6665.70	21.43	5.67
First-Order Exponential ($\lambda = 0.1$)	55.65	6714.67	6714.67	22.23	5.85
First-Order Exponential ($\lambda = 0.4$)	55.21	6705.63	6705.63	21.87	5.82

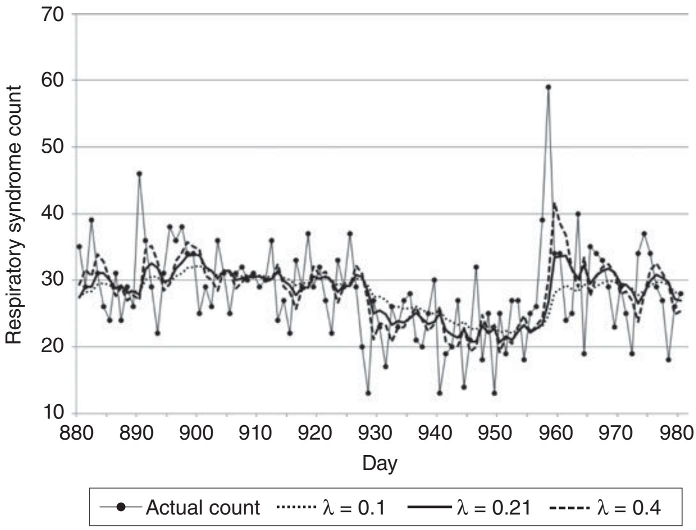


FIGURE 4.35 Respiratory syndrome counts using first-order exponential smoothing with $\lambda = 0.1$, $\lambda = 0.21$ (min SSE), and $\lambda = 0.4$.

We used JMP to fit Winters' additive seasonal model to the respiratory syndrome count data. Because the seasonal patterns are not strong, we investigated seasons of length 3, 7, and 12 periods. The results are summarized in Table 4.15 and illustrated graphically for the last 100 observations in Figure 4.37. The 7-period season works best, probably reflecting the daily seasonal pattern that we observed in Figure 4.34. This is also the best smoother of all the techniques that were investigated. The values of $\lambda = 0$ for the trend and seasonal components in this model are an indication that there is not a significant linear trend in the data and that the seasonal pattern is relatively stable over the period of available data.

TABLE 4.14 Second-Order Simple Exponential Smoothing Applied to the Respiratory Data

Model	Variance	AIC	BIC	MAPE	MAE
Second-Order Exponential (min SSE, $\lambda = 0.10$)	54.37	6690.98	6695.86	21.71	5.78
Second-Order Exponential ($\lambda = 0.2$)	58.22	6754.37	6754.37	22.44	5.98
Second-Order Exponential ($\lambda = 0.4$)	74.46	6992.64	6992.64	25.10	6.74

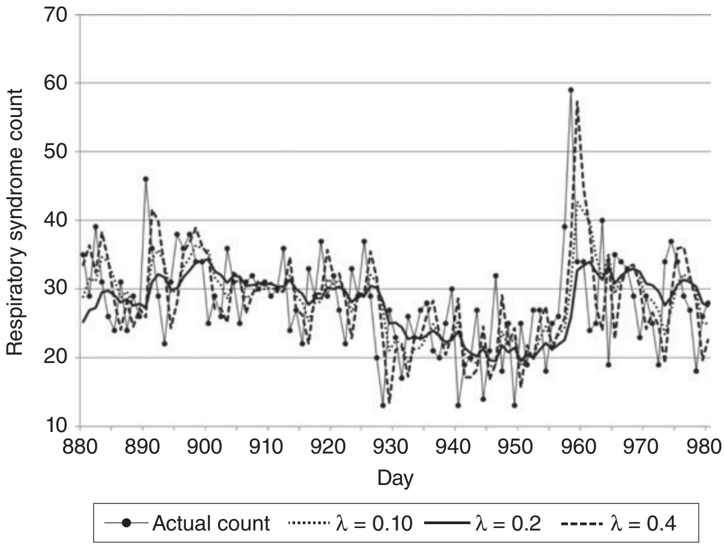


FIGURE 4.36 Respiratory syndrome counts using second-order exponential smoothing with $\lambda = 0.10$ (min SSE), $\lambda = 0.2$, and $\lambda = 0.4$.

TABLE 4.15 Winters' Additive Seasonal Exponential Smoothing Applied to the Respiratory Data

Model	Variance	AIC	BIC	MAPE	MAE
$S = 3$					
Winters Additive (min SSE, $\lambda 1 = 0.21$, $\lambda 2 = 0$, $\lambda 3 = 0$)	52.75	6662.75	6677.40	21.70	5.72
Winters Additive ($\lambda 1 = 0.2$, $\lambda 2 = 0.1$, $\lambda 3 = 0.1$)	57.56	6731.59	6731.59	22.38	5.94
$S = 7$					
Winters Additive (min SSE, $\lambda 1 = 0.22$, $\lambda 2 = 0$, $\lambda 3 = 0$)	49.77	6593.83	6608.47	21.10	5.56
Winters Additive ($\lambda 1 = 0.2$, $\lambda 2 = 0.1$, $\lambda 3 = 0.1$)	54.27	6652.57	6652.57	21.47	5.70
$S = 12$					
Winters Additive (min SSE, $\lambda 1 = 0.21$, $\lambda 2 = 0$, $\lambda 3 = 0$)	52.74	6635.58	6650.21	22.13	5.84
Winters Additive ($\lambda 1 = 0.2$, $\lambda 2 = 0.1$, $\lambda 3 = 0.1$)	58.76	6703.79	6703.79	22.77	6.08

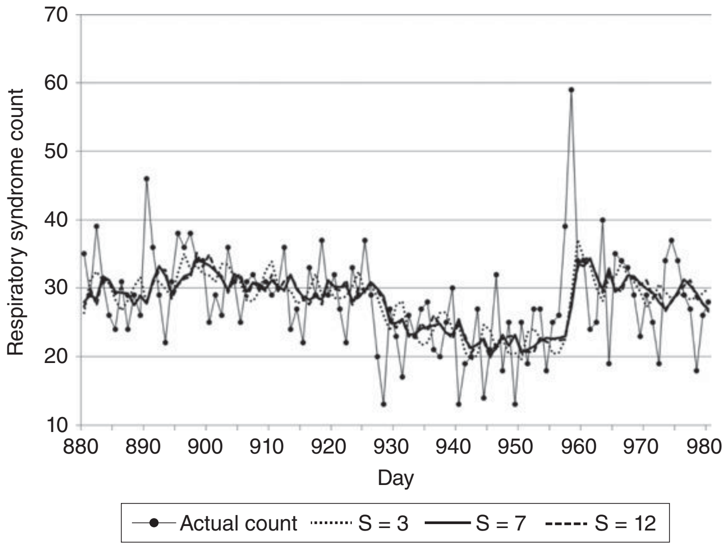


FIGURE 4.37 Respiratory syndrome counts using winters' additive seasonal exponential smoothing with $S = 3$, $S = 7$, and $S = 12$, and smoothing parameters that minimize SSE.

4.9 EXPONENTIAL SMOOTHERS AND ARIMA MODELS

The first-order exponential smoother presented in Section 4.2 is a very effective model in forecasting. The discount factor, λ, makes this smoother fairly flexible in handling time series data with various characteristics. The first-order exponential smoother is particularly good in forecasting time series data with certain specific characteristics.

Recall that the first-order exponential smoother is given as

$$\tilde{y}_T = \lambda y_T + (1 - \lambda)\tilde{y}_{T-1} \tag{4.58}$$

and the forecast error is defined as

$$e_T = y_T - \hat{y}_{T-1}. \tag{4.59}$$

Similarly, we have

$$e_{T-1} = y_{T-1} - \hat{y}_{T-2}. \tag{4.60}$$

By multiplying Eq. (4.60) by $(1 - \lambda)$ and subtracting it from Eq. (4.59), we obtain

$$\begin{aligned} e_T - (1-\lambda)e_{T-1} &= (y_T - \hat{y}_{T-1}) - (1-\lambda)(y_{T-1} - \hat{y}_{T-2}) \\ &= y_T - y_{T-1} - \hat{y}_{T-1} + \underbrace{\lambda y_{T-1} + (1-\lambda)\hat{y}_{T-2}}_{=\hat{y}_{T-1}} \\ &= y_T - y_{T-1} - \hat{y}_{T-1} + \hat{y}_{T-1} \\ &= y_T - y_{T-1}. \end{aligned} \tag{4.61}$$

We can rewrite Eq. (4.61) as

$$y_T - y_{T-1} = e_T - \theta e_{T-1}, \tag{4.62}$$

where $\theta = 1 - \lambda$. Recall from Chapter 2 the **backshift** *operator, B*, defined as $B(y_t) = y_{t-1}$. Thus Eq. (4.62) becomes

$$(1 - B)y_T = (1 - \theta B)e_T. \tag{4.63}$$

We will see in Chapter 5 that the model in Eq. (4.63) is called the **integrated moving average** model denoted as IMA(1,1), for the backshift operator is used only once on y_T and only once on the error. It can be shown that if the process exhibits the dynamics defined in Eq. (4.63), that is an IMA(1,1) process, the first-order exponential smoother provides minimum mean squared error (MMSE) forecasts (see Muth (1960), Box and Luceno (1997), and Box, Jenkins, and Reinsel (1994)). For more discussion of the equivalence between exponential smoothing techniques and the ARIMA models, see Abraham and Ledolter (1983), Cogger (1974), Goodman (1974), Pandit and Wu (1974), and McKenzie (1984).

4.10 R COMMANDS FOR CHAPTER 4

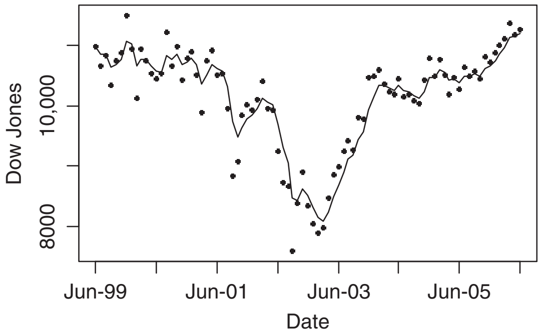
Example 4.1 The Dow Jones index data are in the second column of the array called dji.data in which the first column is the month of the year. We can use the following simple function to obtain the first-order exponential smoothing

```
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
}
```

Note that this function uses the first observation as the starting value by default. One can change this by providing a specific start value when calling the function.

We can then obtain the smoothed version of the data for a specified lambda value and plot the fitted value as the following:

```
dji.smooth1<-firstsmooth(y=dji.data[,2],lambda=0.4)
plot(dji.data[,2],type="p", pch=16,cex=.5,xlab='Date',ylab='Dow
    Jones',xaxt='n')
axis(1, seq(1,85,12), dji.data[seq(1,85,12),1])
lines(dji.smooth1)
```



For the first-order exponential smoothing, measures of accuracy such as MAPE, MAD, and MSD can be obtained from the following function:

```
measacc.fs<- function(y,lambda){
            out<- firstsmooth(y,lambda)
            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)
}

measacc.fs(dji.data[,2],0.4)
         SSE         MAPE          MAD          MSD
1.665968e+07 3.461342e+00 3.356325e+02 1.959962e+05
```

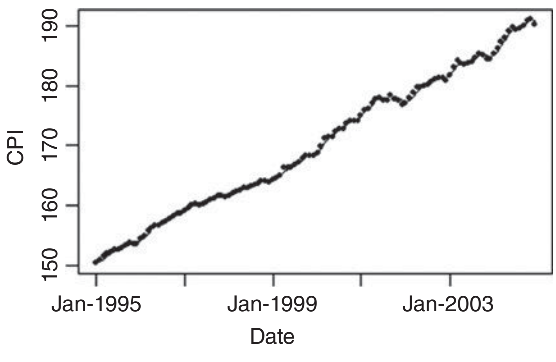
Note that alternatively we could use the Holt–Winters function from the stats package. The function requires three parameters (alpha, beta, and gamma) to be defined. Providing a specific value for alpha and setting beta and gamma to "FALSE" give the first-order exponential as the following

```
dji1.fit<-HoltWinters(dji.data[,2],alpha=.4, beta=FALSE, gamma=FALSE)
```

Beta corresponds to the second-order smoothing (or the trend term) and gamma is for the seasonal effect.

Example 4.2 The US CPI data are in the second column of the array called cpi.data in which the first column is the month of the year. For this case we use the firstsmooth function twice to obtain the double exponential smoothing as

```
cpi.smooth1<-firstsmooth(y=cpi.data[,2],lambda=0.3)
cpi.smooth2<-firstsmooth(y=cpi.smooth1,lambda=0.3)
cpi.hat<-2*cpi.smooth1-cpi.smooth2 #Equation 4.23
plot(cpi.data[,2],type="p", pch=16,cex=.5,xlab='Date',ylab='CPI',
  xaxt='n')
axis(1, seq(1,120,24), cpi.data[seq(1,120,24),1])
lines(cpi.hat)
```

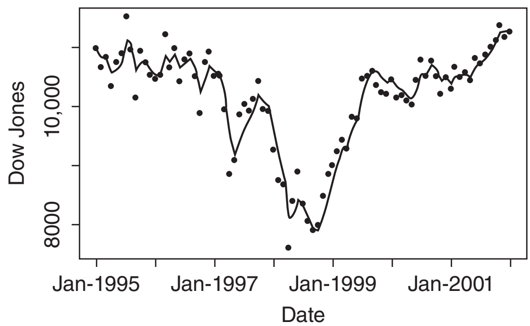


Note that the fitted values are obtained using Eq. (4.23). Also the corresponding command using Holt–Winters function is

```
HoltWinters(cpi.data[,2],alpha=0.3, beta=0.3, gamma=FALSE)
```

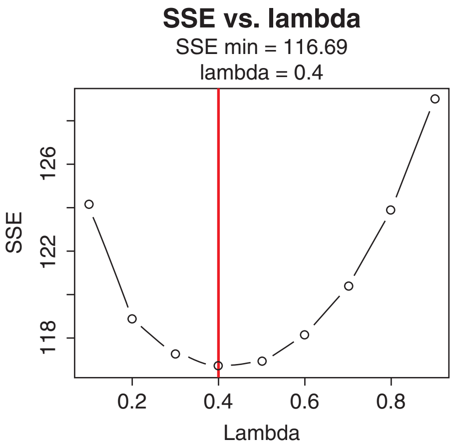
Example 4.3 In this example we use the firstsmooth function twice for the Dow Jones Index data to obtain the double exponential smoothing as in the previous example.

```
dji.smooth1<-firstsmooth(y=dji.data[,2],lambda=0.3)
dji.smooth2<-firstsmooth(y=dji.smooth1,lambda=0.3)
dji.hat<-2*dji.smooth1-dji.smooth2 #Equation 4.23
plot(dji.data[,2],type="p", pch=16,cex=.5,xlab='Date',ylab='Dow
 Jones',xaxt='n')
axis(1, seq(1,85,12), cpi.data[seq(1,85,12),1])
lines(dji.hat)
```



Example 4.4 The average speed data are in the second column of the array called speed.data in which the first column is the index for the week. To find the "best" smoothing constant, we will use the firstsmooth function for various lambda values and obtain the sum of squared one-step-ahead prediction error (SS_E) for each. The lambda value that minimizes the sum of squared prediction errors is deemed the "best" lambda. The obvious option is to apply firstsmooth function in a for loop to obtain SS_E for various lambda values. Even though in this case this may not be an issue, in many cases for loops can slow down the computations in R and are to be avoided if possible. We will do that using sapply function.

```
lambda.vec<-seq(0.1, 0.9, 0.1)
sse.speed<-function(sc){measacc.fs(speed.data[1:78,2],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))
```



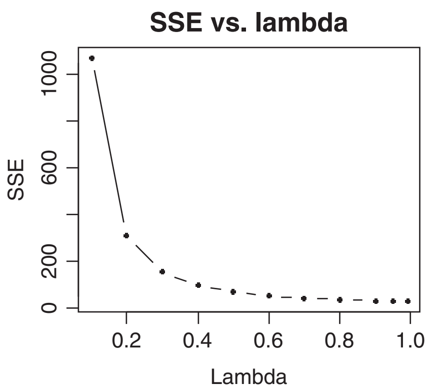
Note that we can also use Holt–Winters function to find the "best" value for the smoothing constant by not specifying the appropriate parameter as the following:

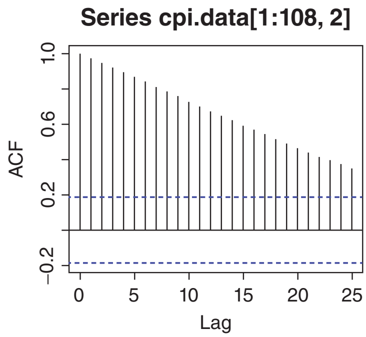
```
HoltWinters(speed.data[,2], beta=FALSE, gamma=FALSE)
```

Example 4.5 We will first try to find the best lambda for the CPI data using first-order exponential smoothing. We will also plot ACF of the data.

Note that we will use the data up to December 2003.

```
lambda.vec<-c(seq(0.1, 0.9, 0.1), .95, .99)
sse.cpi<-function(sc){measacc.fs(cpi.data[1:108,2],sc)[1]}
sse.vec<-sapply(lambda.vec, sse.cpi)
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', pch=16,cex=.5)
acf(cpi.data[1:108,2],lag.max=25)
```

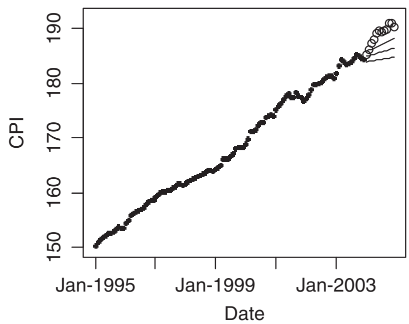




We now use the second-order exponential smoothing with lambda of 0.3. We calculate the forecasts using Eq. (4.31) for the two options suggested in the Example 4.5.

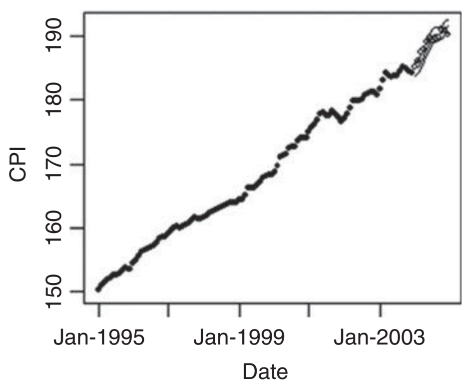
Option 1: On December 2003, make the forecasts for the entire 2004 (1- to 12-step-ahead forecasts).

```
lcpi<-0.3
cpi.smooth1<-firstsmooth(y=cpi.data[1:108,2],lambda=lcpi)
cpi.smooth2<-firstsmooth(y=cpi.smooth1,lambda=lcpi)
cpi.hat<-2*cpi.smooth1-cpi.smooth2
tau<-1:12
T<-length(cpi.smooth1)
cpi.forecast<-(2+tau*(lcpi/(1-lcpi)))*cpi.smooth1[T]-(1+tau*(lcpi/
  (1-lcpi)))*cpi.smooth2[T]
ctau<-sqrt(1+(lcpi/((2-lcpi)^3))*(10-14*lcpi+5*(lcpi^2)+2*tau*lcpi
  *(4-3*lcpi)+2*(tau^2)*(lcpi^2)))
alpha.lev<-.05
sig.est<- sqrt(var(cpi.data[2:108,2]- cpi.hat[1:107]))
cl<-qnorm(1-alpha.lev/2)*(ctau/ctau[1])*sig.est
plot(cpi.data[1:108,2],type="p", pch=16,cex=.5,xlab='Date',
  ylab='CPI',xaxt='n',xlim=c(1,120),ylim=c(150,192))
axis(1, seq(1,120,24), cpi.data[seq(1,120,24),1])
points(109:120,cpi.data[109:120,2])
lines(109:120,cpi.forecast)
lines(109:120,cpi.forecast+cl)
lines(109:120,cpi.forecast-cl)
```



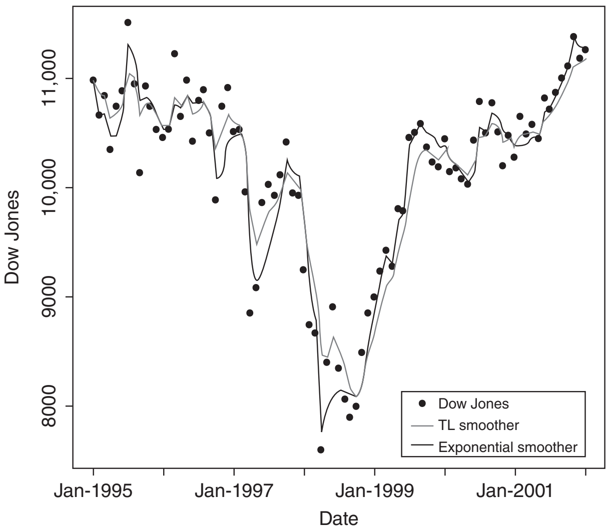
Option 2: On December 2003, make the forecast for January 2004. Then when January 2004 data are available, make the forecast for February 2004 (only one-step-ahead forecasts).

```
lcpi<-0.3
T<-108
tau<-12
alpha.lev<-.05
cpi.forecast<-rep(0,tau)
cl<-rep(0,tau)
cpi.smooth1<-rep(0,T+tau)
cpi.smooth2<-rep(0,T+tau)

for (i in 1:tau) {
      cpi.smooth1[1:(T+i-1)]<-firstsmooth(y=cpi.data[1:(T+i-1),2],
        lambda=lcpi)
      cpi.smooth2[1:(T+i-1)]<-firstsmooth(y=cpi.smooth1[1:(T+i-1)],
        lambda=lcpi)
      cpi.forecast[i]<-(2+(lcpi/(1-lcpi)))*cpi.smooth1[T+i-1]-
        (1+(lcpi/(1-lcpi)))*cpi.smooth2[T+i-1]
      cpi.hat<-2*cpi.smooth1[1:(T+i-1)]-cpi.smooth2[1:(T+i-1)]
      sig.est<- sqrt(var(cpi.data[2:(T+i-1),2]- cpi.hat[1:(T+i-2)]))
      cl[i]<-qnorm(1-alpha.lev/2)*sig.est
      }

plot(cpi.data[1:T,2],type="p", pch=16,cex=.5,xlab='Date',ylab='CPI',
  xaxt='n',xlim=c(1,T+tau),ylim=c(150,192))
axis(1, seq(1,T+tau,24), cpi.data[seq(1,T+tau,24),1])
points((T+1):(T+tau),cpi.data[(T+1):(T+tau),2],cex=.5)
lines((T+1):(T+tau),cpi.forecast)
lines((T+1):(T+tau),cpi.forecast+cl)
lines((T+1):(T+tau),cpi.forecast-cl)
```



Example 4.6 The function for the Trigg–Leach smoother is given as:

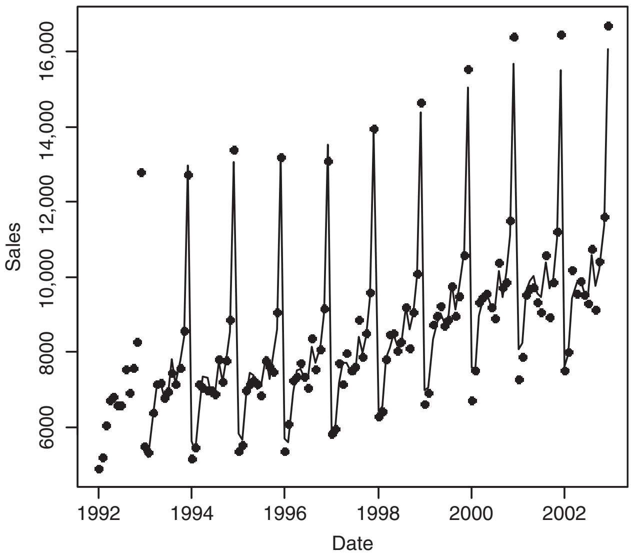
```
tlsmooth<-function(y,gamma,y.tilde.start=y[1],lambda.start=1){
      T<-length(y)

#Initialize the vectors
      Qt<-vector()
      Dt<-vector()
      y.tilde<-vector()
      lambda<-vector()
      err<-vector()
#Set the starting values for the vectors
      lambda[1]=lambda.start
      y.tilde[1]=y.tilde.start
      Qt[1]<-0
      Dt[1]<-0
      err[1]<-0

      for (i in 2:T){
            err[i]<-y[i]-y.tilde[i-1]
            Qt[i]<-gamma*err[i]+(1-gamma)*Qt[i-1]
            Dt[i]<-gamma*abs(err[i])+(1-gamma)*Dt[i-1]
            lambda[i]<-abs(Qt[i]/Dt[i])
            y.tilde[i]=lambda[i]*y[i] + (1-lambda[i])*y.tilde[i-1]
      }
return(cbind(y.tilde,lambda,err,Qt,Dt))
}

#Obtain the TL smoother for Dow Jones Index
out.tl.dji<-tlsmooth(dji.data[,2],0.3)
```



```
#Obtain the exponential smoother for Dow Jones Index
dji.smooth1<-firstsmooth(y=dji.data[,2],lambda=0.4)

#Plot the data together with TL and exponential smoother for
  comparison
plot(dji.data[,2],type="p", pch=16,cex=.5,xlab='Date',ylab='Dow
  Jones',xaxt='n')
axis(1, seq(1,85,12), cpi.data[seq(1,85,12),1])
lines(out.tl.dji[,1])
lines(dji.smooth1,col="grey40")
legend(60,8000,c("Dow Jones","TL Smoother","Exponential Smoother"),
  pch=c(16, NA, NA),lwd=c(NA,.5,.5),cex=.55,col=c("black",
  "black","grey40"))
```

Example 4.7 The clothing sales data are in the second column of the array called closales.data in which the first column is the month of the year. We will use the data up to December 2002 to fit the model and make forecasts for the coming year (2003). We will use Holt–Winters function given in stats package. The model is additive seasonal model with all parameters equal to 0.2.

```
dat.ts = ts(closales.data[,2], start = c(1992,1), freq = 12)
y1<-closales.data[1:132,]
# convert data to ts object
y1.ts<-ts(y1[,2], start = c(1992,1), freq = 12)
clo.hw1<-HoltWinters(y1.ts,alpha=0.2,beta=0.2,gamma=0.2,seasonal
  ="additive")
plot(y1.ts,type="p", pch=16,cex=.5,xlab='Date',ylab='Sales')
lines(clo.hw1$fitted[,1])
```



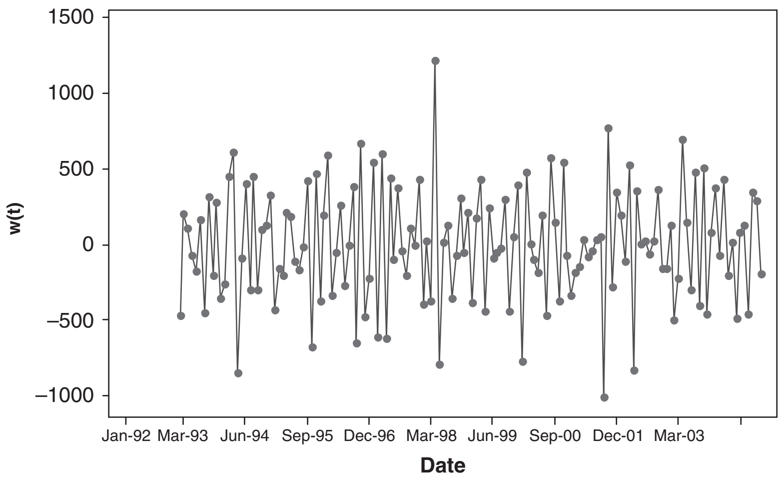


FIGURE 5.26 Time series plot of $w_t = (1 - B)(1 - B^{12})y_t$ for the US clothing sales data.

decaying values at the first 8 lags suggest that a nonseasonal MA(1) model should be used.

The interpretation of the remaining seasonality is a bit more difficult. For that we should focus on the sample ACF and PACF values at lags 12, 24, 36, and so on. The sample ACF at lag 12 seems to be significant and the sample PACF at lags 12, 24, 36 (albeit not significant) seems to be alternating in sign. That suggests that a seasonal MA(1) model can be used as well. Hence an ARIMA$(0, 1, 1) \times (0, 1, 1)_{12}$ model is used to model the data, y_t. The output from Minitab is given in Table 5.9. Both MA(1) and seasonal MA(1) coefficient estimates are significant. As we can see from the sample ACF and PACF plots in Figure 5.28, while there are still some

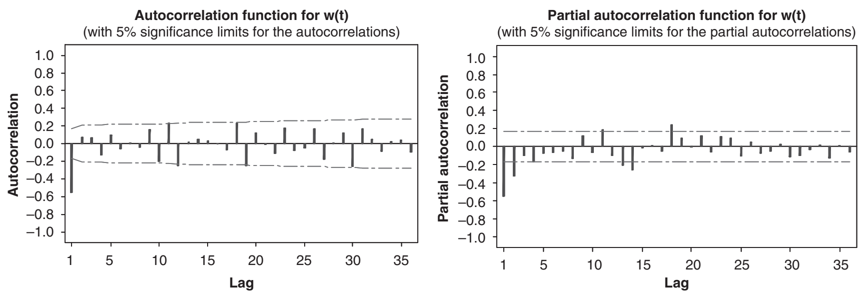


FIGURE 5.27 Sample ACF and PACF plots of $w_t = (1 - B)(1 - B^{12})y_t$.

TABLE 5.9 Minitab Output for the ARIMA(0, 1, 1) × (0, 1, 1)$_{12}$ Model for the US Clothing Sales Data

```
Final Estimates of Parameters

Type        Coef   SE Coef       T      P
MA   1    0.7626    0.0542  14.06  0.000
SMA  12   0.5080    0.0771   6.59  0.000


Differencing: 1 regular, 1 seasonal of order 12
Number of observations:  Original series 155, after
differencing 142
Residuals:    SS =  10033560 (backforecasts excluded)
              MS =  71668   DF = 140


Modified Box-Pierce (Ljung-Box) Chi-Square statistic

Lag              12      24      36      48
Chi-Square     15.8    37.7    68.9    92.6
DF               10      22      34      46
P-Value       0.107   0.020   0.000   0.000
```

small significant values, as indicated by the modified Box pierce statistic most of the autocorrelation is now modeled out.

The residual plots in Figure 5.29 provided by Minitab seem to be acceptable as well.

Finally, the time series plot of the actual and fitted values in Figure 5.30 suggests that the ARIMA(0, 1, 1) × (0, 1, 1)$_{12}$ model provides a reasonable fit to this highly seasonal and nonstationary time series data.

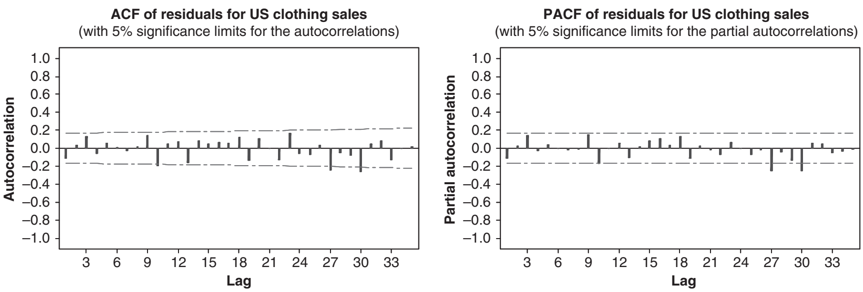


FIGURE 5.28 Sample ACF and PACF plots of residuals from the ARIMA(0, 1, 1) × (0, 1, 1)$_{12}$ model.

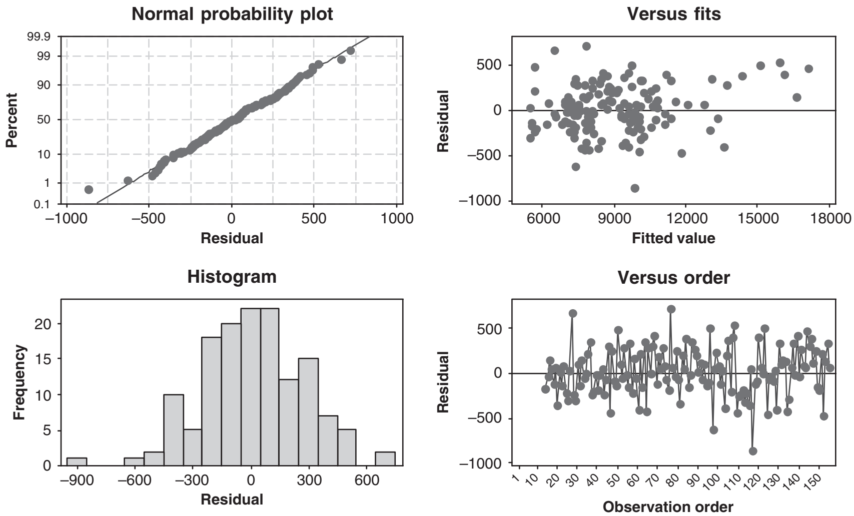


FIGURE 5.29 Residual plots from the ARIMA(0, 1, 1) × $(0, 1, 1)_{12}$ model for the US clothing sales data.

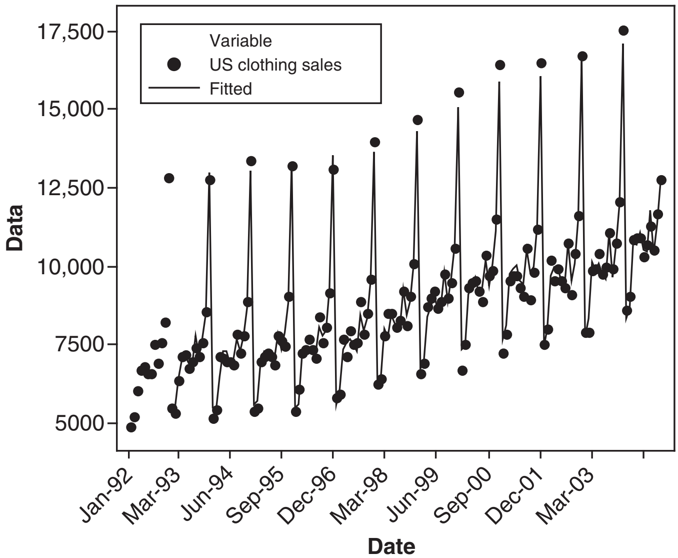


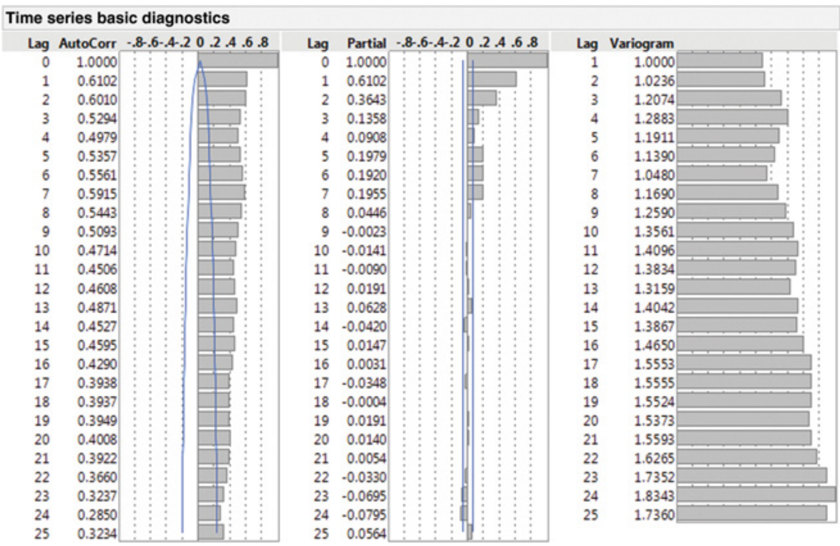
FIGURE 5.30 Time series plot of the actual data and fitted values from the ARIMA(0, 1, 1) × $(0, 1, 1)_{12}$ model for the US clothing sales data.

5.10 ARIMA MODELING OF BIOSURVEILLANCE DATA

In Section 4.8 we introduced the daily counts of respiratory and gastrointestinal complaints for more than 2-$^1/_2$ years at several hospitals in a large metropolitan area from Fricker (2013). Table 4.12 presents the 980 observations from one of these hospitals. Section 4.8 described modeling the respiratory count data with exponential smoothing. We now present an ARIMA modeling approach. Figure 5.31 presents the sample ACF, PACF, and the variogram from JMP for these data. Examination of the original time series plot in Figure 4.35 and the ACF and variogram indicate that the daily respiratory syndrome counts may be nonstationary and that the data should be differenced to obtain a stationary time series for ARIMA modeling.

The ACF for the differenced series ($d = 1$) shown in Figure 5.32 cuts off after lag 1 while the PACF appears to be a mixture of exponential decays. This suggests either an ARIMA(1, 1, 1) or ARIMA(2, 1, 1) model.

The Time Series Modeling platform in JMP allows a group of ARIMA models to be fit by specifying ranges for the AR, difference, and MA terms. Table 5.10 summarizes the fits obtained for a constant difference (d = 1), and both AR (p) and MA (q) parameters ranging from 0 to 2.



Time series basic diagnostics

Lag	AutoCorr	Lag	Partial	Lag	Variogram
0	1.0000	0	1.0000	1	1.0000
1	0.6102	1	0.6102	2	1.0236
2	0.6010	2	0.3643	3	1.2074
3	0.5294	3	0.1358	4	1.2883
4	0.4979	4	0.0908	5	1.1911
5	0.5357	5	0.1979	6	1.1390
6	0.5561	6	0.1920	7	1.0480
7	0.5915	7	0.1955	8	1.1690
8	0.5443	8	0.0446	9	1.2590
9	0.5093	9	-0.0023	10	1.3561
10	0.4714	10	-0.0141	11	1.4096
11	0.4506	11	-0.0090	12	1.3834
12	0.4608	12	0.0191	13	1.3159
13	0.4871	13	0.0628	14	1.4042
14	0.4527	14	-0.0420	15	1.3867
15	0.4595	15	0.0147	16	1.4650
16	0.4290	16	0.0031	17	1.5553
17	0.3938	17	-0.0348	18	1.5555
18	0.3937	18	-0.0004	19	1.5524
19	0.3949	19	0.0191	20	1.5373
20	0.4008	20	0.0140	21	1.5593
21	0.3922	21	0.0054	22	1.6265
22	0.3660	22	-0.0330	23	1.7352
23	0.3237	23	-0.0695	24	1.8343
24	0.2850	24	-0.0795	25	1.7360
25	0.3234	25	0.0564		

FIGURE 5.31 ACF, PACF, and variogram for daily respiratory syndrome counts.

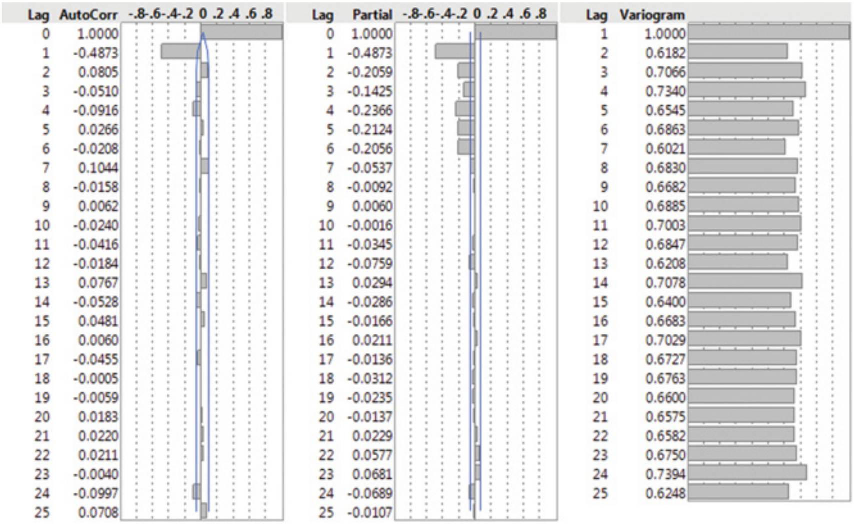


FIGURE 5.32 ACF, PACF, and variogram for the first difference of the daily respiratory syndrome counts.

TABLE 5.10 Summary of Models fit to the Respiratory Syndrome Count Data

Model	Variance	AIC	BIC	RSquare	MAPE	MAE
AR(1)	65.7	6885.5	6895.3	0.4	24.8	6.3
AR(2)	57.1	6748.2	6762.9	0.5	22.9	5.9
MA(1)	81.6	7096.9	7106.6	0.2	28.5	6.9
MA(2)	69.3	6937.6	6952.3	0.3	26.2	6.4
ARMA(1, 1)	52.2	6661.2	6675.9	0.5	21.6	5.6
ARMA(1, 2)	52.1	6661.2	6680.7	0.5	21.6	5.6
ARMA(2, 1)	52.1	6660.7	6680.3	0.5	21.6	5.6
ARMA(2, 2)	52.3	6664.3	6688.7	0.5	21.6	5.6
ARIMA(0, 0, 0)	104.7	7340.4	7345.3	0.0	33.2	8.0
ARIMA(0, 1, 0)*	81.6	7088.2	7093.1	0.2	26.2	7.0
ARIMA(0, 1, 1)*	52.7	6662.8	6672.6	0.5	21.4	5.7
ARIMA(0, 1, 2)	52.6	6662.1	6676.7	0.5	21.4	5.7
ARIMA(1, 1, 0)*	62.2	6824.4	6834.2	0.4	23.2	6.2
ARIMA(1, 1, 1)	52.6	6661.4	6676.1	0.5	21.4	5.7
ARIMA(1, 1, 2)	52.6	6661.9	6681.5	0.5	21.4	5.7
ARIMA(2, 1, 0)	59.6	6783.5	6798.1	0.4	22.7	6.1
ARIMA(2, 1, 1)	52.3	6657.1	6676.6	0.5	21.4	5.6
ARIMA(2, 1, 2)	52.3	6657.8	6682.2	0.5	21.3	5.6

*Indicates that objective function failed during parameter estimation.

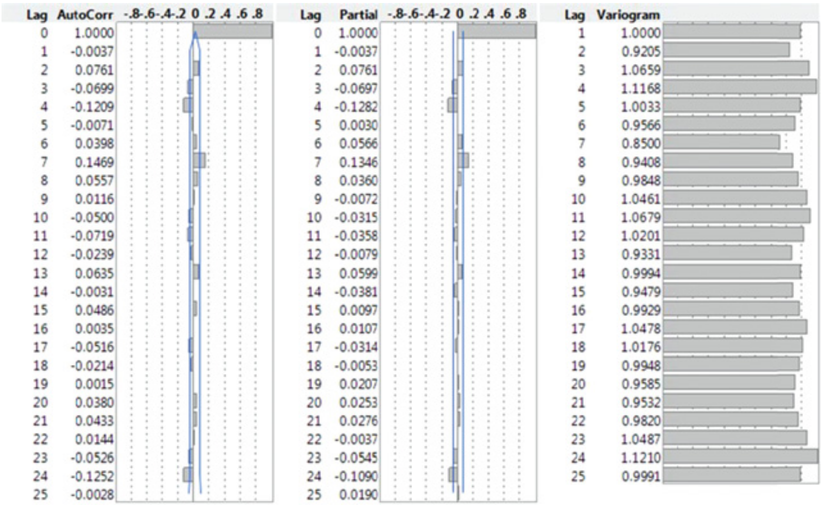


FIGURE 5.33 ACF, PACF, and variogram for the residuals of ARIMA(1, 1, 1) fit to daily respiratory syndrome counts.

In terms of the model summary statistics variance of the errors, AIC and mean absolute prediction error (MAPE) several models look potentially reasonable. For the ARIMA(1, 1, 1) we obtained the following results from JMP:

Parameter estimates

Term	Lag	Estimate	Std error	t Ratio	Prob>|t|	Constant estimate
AR1	1	0.07307009	0.0394408	1.85	0.0642	0.00069557
MA1	1	0.81584055	0.0223680	36.47	<.0001*	
Intercept	0	0.00075040	0.0036018	0.21	0.8350	

Figure 5.33 presents the ACF, PACF, and variogram of the residuals from this model. Other residual plots are in Figure 5.34.

For comparison purposes we also fit the ARIMA(2, 1, 1) model. The parameter estimates obtained from JMP are:

Parameter Estimates

Term	Lag	Estimate	Std Error	t Ratio	Prob>|t|	Constant Estimate
AR1	1	0.09953471	0.0402040	2.48	0.0135*	0.00097496
AR2	2	0.09408008	0.0375486	2.51	0.0124*	
MA1	1	0.84755625	0.0231814	36.56	<.0001*	
Intercept	0	0.00120905	0.0088678	0.14	0.8916	

P-value < α (0.05)
Diff. is sig.

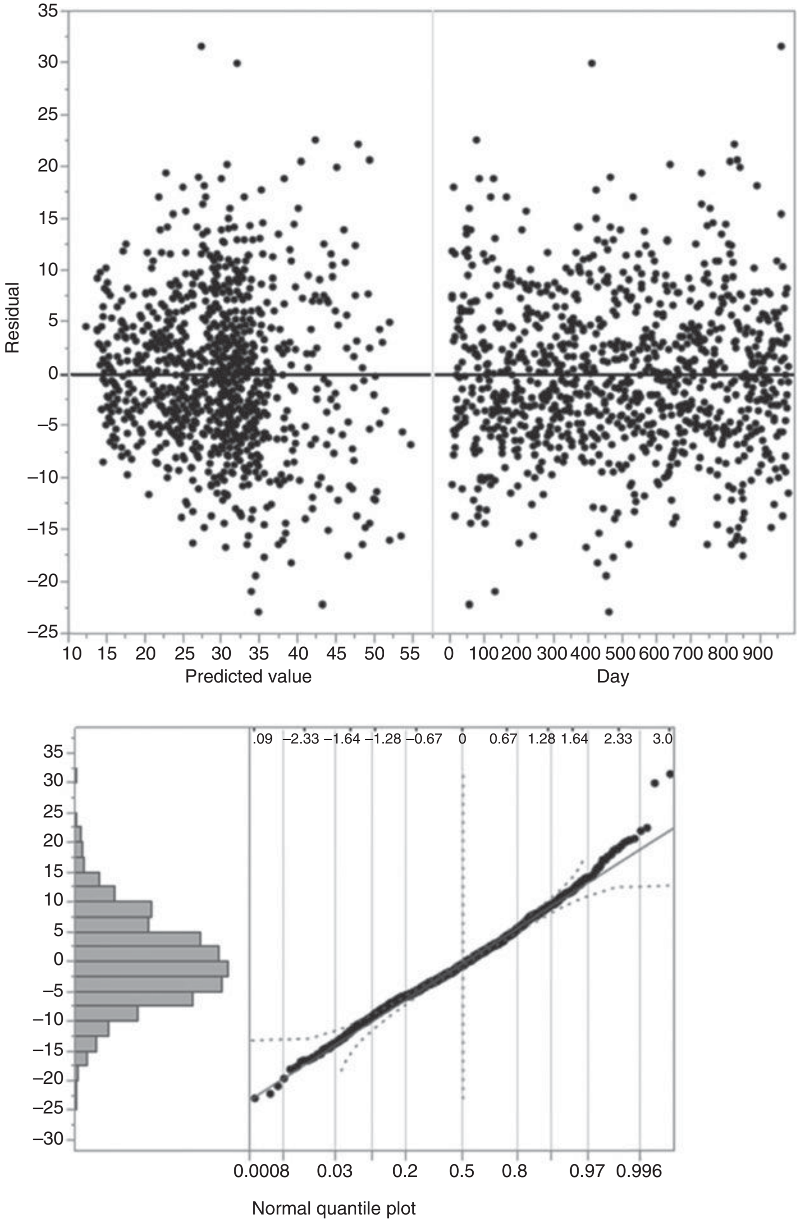


FIGURE 5.34 Plots of residuals from ARIMA(1, 1, 1) fit to daily respiratory syndrome counts.

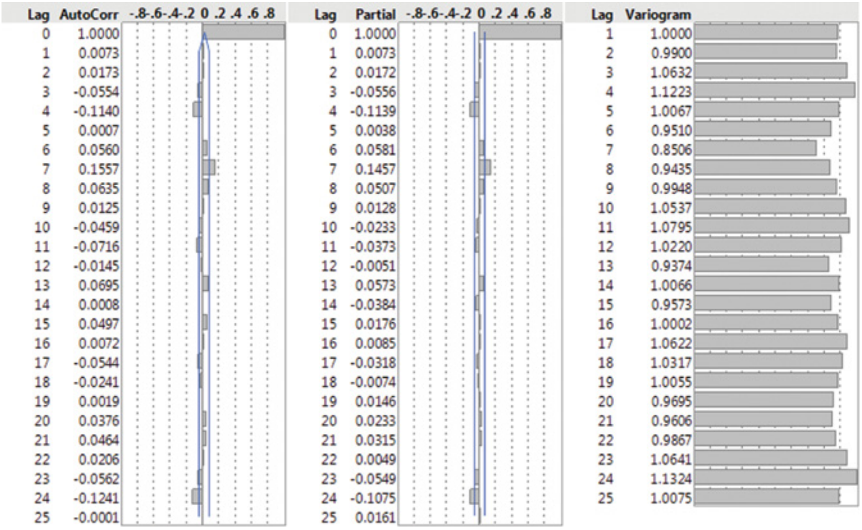


FIGURE 5.35 ACF, PACF, and variogram for residuals of ARIMA(2, 1, 1) fit to daily respiratory syndrome counts.

The lag 2 AR parameter is highly significant. Figure 5.35 presents the plots of the ACF, PACF, and variogram of the residuals from ARIMA(2, 1, 1). Other residual plots are shown in Figure 5.36. Based on the significant lag 2 AR parameter, this model is preferable to the ARIMA(1, 1, 1) model fit previously.

Considering the variation in counts by day of week that was observed previously, a seasonal ARIMA model with a seasonal period of 7 days may be appropriate. The resulting model has an error variance of 50.9, smaller than for the ARIMA(1, 1, 1) and ARIMA(2, 1, 1) models. The AIC is also smaller. Notice that all of the model parameters are highly significant. The residual ACF, PACF, and variogram shown in Figure 5.37 do not suggest any remaining structure. Other residual plots are in Figure 5.38.

Model	Variance	AIC	BIC	RSquare	MAPE	MAE
ARIMA(2, 1, 1)(0, 0, 1)$_7$	50.9	6631.9	6656.4	0.5	21.1	5.6

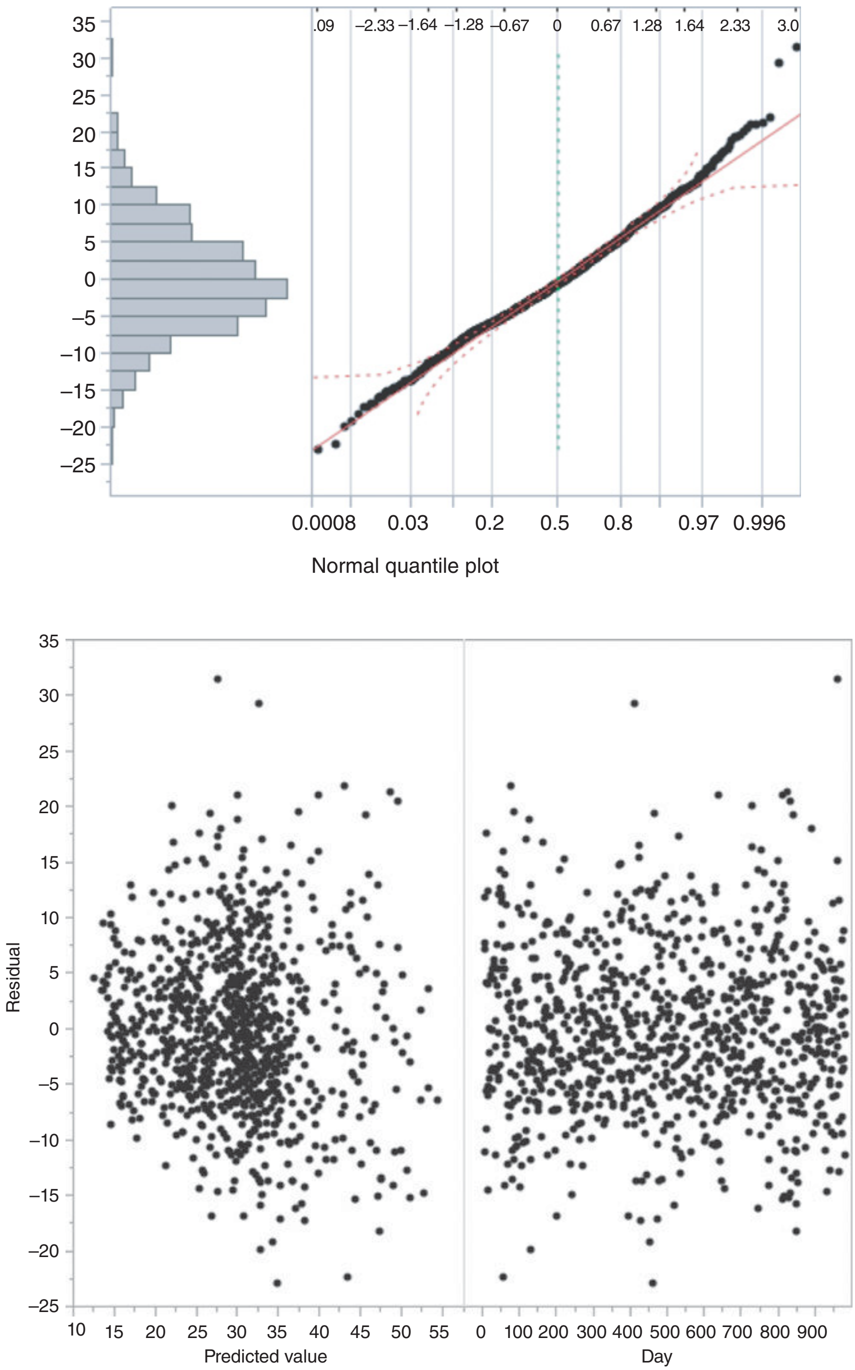


FIGURE 5.36 Plots of residuals from ARIMA(2, 1, 1) fit to daily respiratory syndrome counts.

Parameter estimates

Term	Factor	Lag	Estimate	Std error	t Ratio	Prob>|t|	Constant estimate
AR1,1	1	1	0.1090685	0.0395388	2.76	0.0059*	0.00083453
AR1,2	1	2	0.1186083	0.0376471	3.15	0.0017*	
MA1,1	1	1	0.8730535	0.0225127	38.78	<.0001*	
MA2,7	2	7	-0.1744415	0.0328363	-5.31	<.0001*	
Intercept	1	0	0.0010805	0.0051887	0.21	0.8351	

< 0.05

Lag	AutoCorr
0	1.0000
1	0.0046
2	0.0137
3	-0.0184
4	-0.0875
5	0.0138
6	0.0593
7	-0.0031
8	0.0664
9	0.0151
10	-0.0300
11	-0.0467
12	-0.0106
13	0.0601
14	0.0001
15	0.0386
16	0.0173
17	-0.0271
18	-0.0160
19	0.0076
20	0.0263
21	0.0368
22	0.0179
23	-0.0496
24	-0.1188
25	0.0065

Lag	Partial
0	1.0000
1	0.0046
2	0.0136
3	-0.0185
4	-0.0876
5	0.0152
6	0.0619
7	-0.0073
8	0.0579
9	0.0198
10	-0.0225
11	-0.0483
12	-0.0015
13	0.0638
14	-0.0145
15	0.0282
16	0.0201
17	-0.0158
18	-0.0165
19	0.0148
20	0.0329
21	0.0179
22	0.0098
23	-0.0466
24	-0.1142
25	0.0148

Lag	Variogram
1	1.0000
2	0.9909
3	1.0231
4	1.0925
5	0.9907
6	0.9450
7	1.0077
8	0.9378
9	0.9894
10	1.0347
11	1.0515
12	1.0152
13	0.9442
14	1.0045
15	0.9658
16	0.9872
17	1.0318
18	1.0206
19	0.9969
20	0.9782
21	0.9676
22	0.9867
23	1.0544
24	1.1239
25	0.9981

FIGURE 5.37 ACF, PACF, and variogram of residuals from ARIMA(2, 1, 1) × (0, 0, 1)$_7$ fit to daily respiratory syndrome counts.

5.11 FINAL COMMENTS

ARIMA models (a.k.a. Box–Jenkins models) present a very powerful and flexible class of models for time series analysis and forecasting. Over the years, they have been very successfully applied to many problems in research and practice. However, there might be certain situations where they may fall short on providing the "right" answers. For example, in ARIMA models, forecasting future observations primarily relies on the past data and implicitly assumes that the conditions at which the data is collected will remain the same in the future as well. In many situations this assumption may (and most likely will) not be appropriate. For those cases, the transfer function–noise models, where a set of input variables that may have an effect on the time series are added to the model, provide suitable options. We shall discuss these models in the next chapter. For an excellent discussion of this matter and of time series analysis and forecasting in general, see Jenkins (1979).

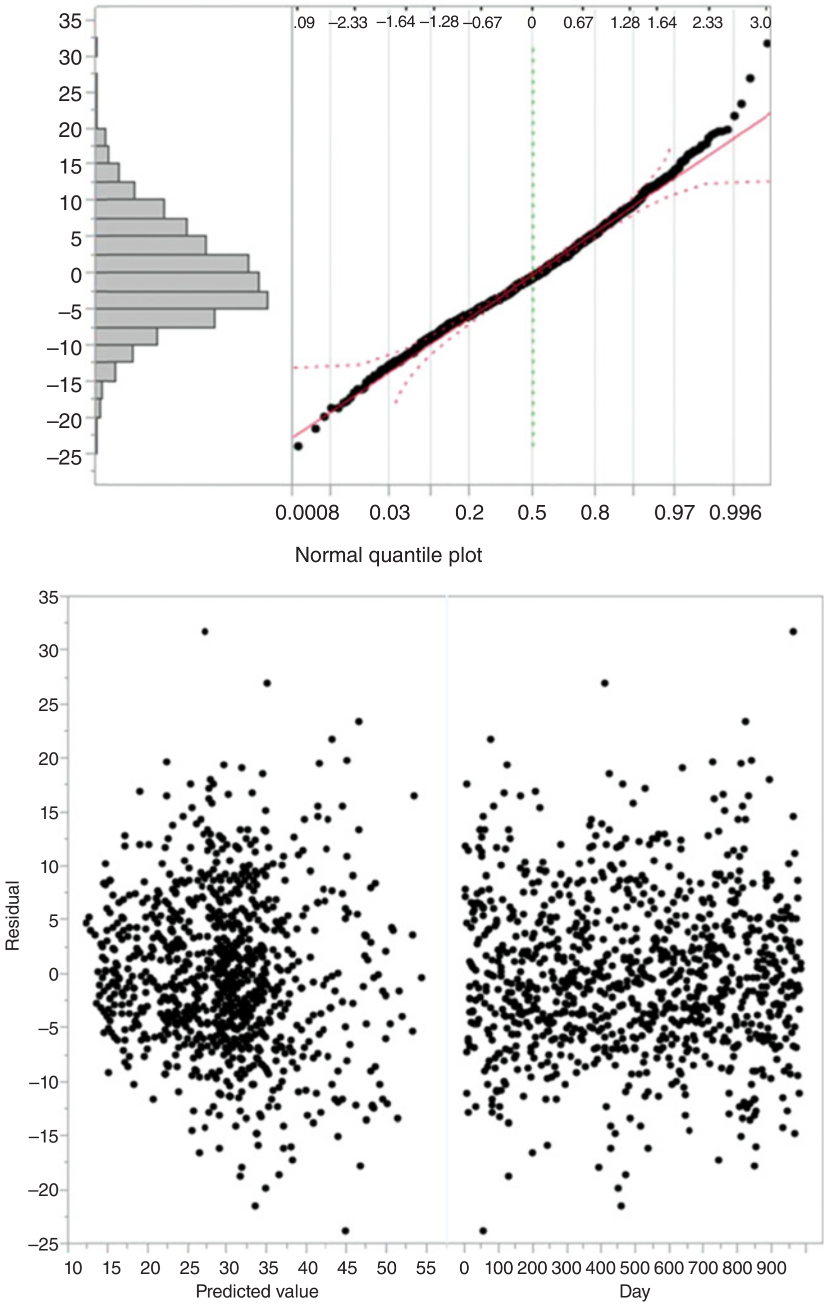
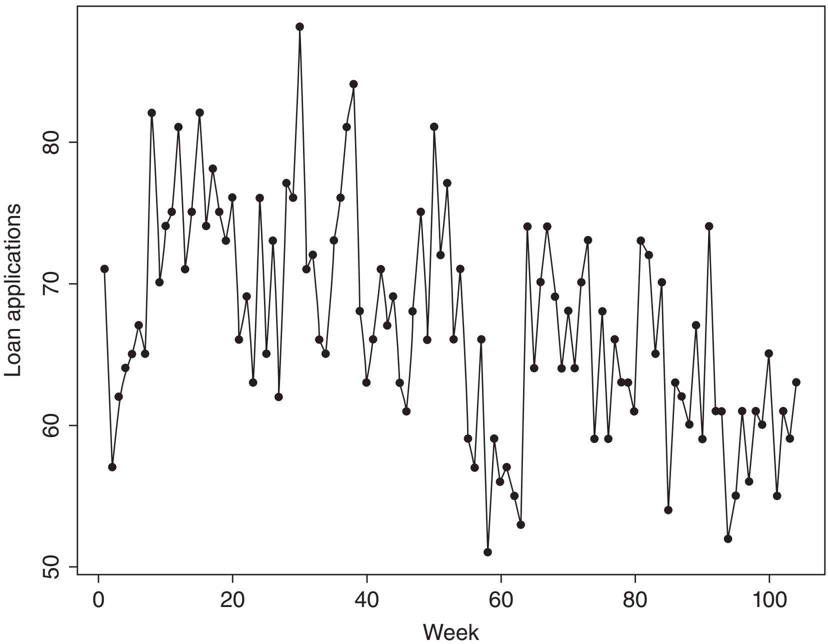


FIGURE 5.38 Plots of residuals from ARIMA(2, 1, 1) × $(0, 0, 1)_7$ fit to daily respiratory syndrome counts.

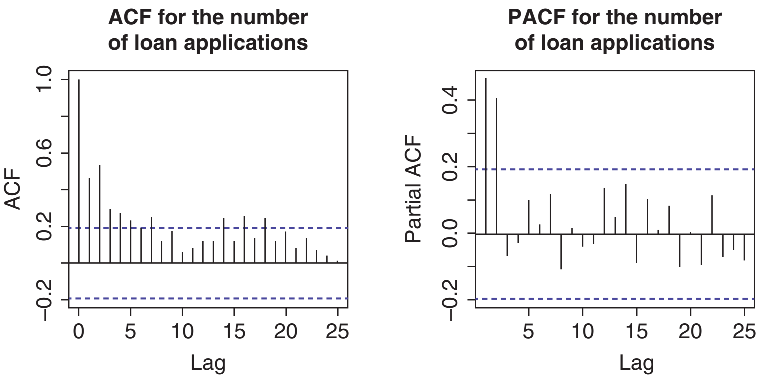
5.12 R COMMANDS FOR CHAPTER 5

Example 5.1 The loan applications data are in the second column of the array called loan.data in which the first column is the number of weeks. We first plot the data as well as the ACF and PACF.

```
plot(loan.data[,2],type="o",pch=16,cex=.5,xlab='Week',ylab='Loan
Applications')
```



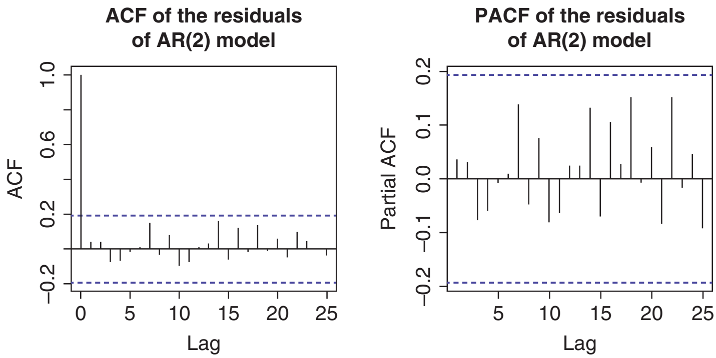

```
par(mfrow=c(1,2),oma=c(0,0,0,0))

acf(loan.data[,2],lag.max=25,type="correlation",main="ACF for the
Number \nof Loan Applications")

acf(loan.data[,2], lag.max=25,type="partial",main="PACF for the
Number \nof Loan Applications")
```



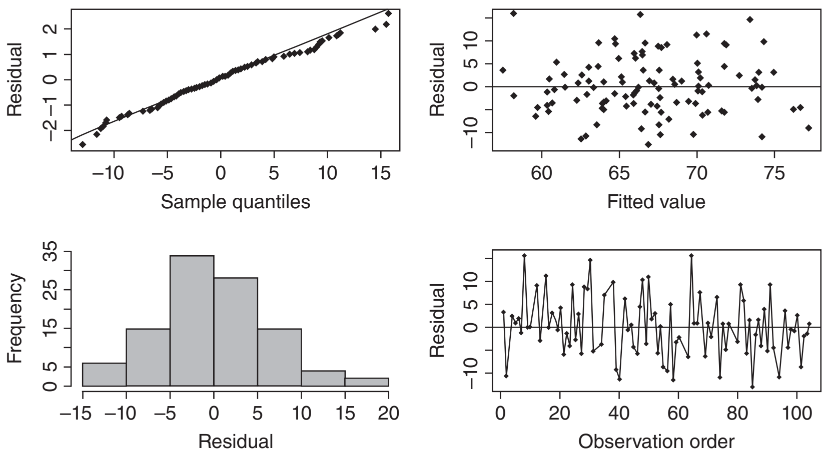
Fit an ARIMA(2,0,0) model to the data using arima function in the stats package.

```
loan.fit.ar2<-arima(loan.data[,2],order=c(2, 0, 0))
loan.fit.ar2

       Call:
       arima(x = loan.data[, 2], order = c(2, 0, 0))

       Coefficients:
                ar1     ar2   intercept
             0.2659  0.4130     66.8538
       s.e.  0.0890  0.0901      1.8334

       sigma^2 estimated as 38.32:  log likelihood = -337.46,
       aic = 682.92


res.loan.ar2<-as.vector(residuals(loan.fit.ar2))
#to obtain the fitted values we use the function fitted() from
#the forecast package
library(forecast)
fit.loan.ar2<-as.vector(fitted(loan.fit.ar2))

Box.test(res.loan.ar2,lag=48,fitdf=3,type="Ljung")

                Box-Ljung test

       data:  res.loan.ar2
       X-squared = 31.8924, df = 45, p-value = 0.9295


#ACF and PACF of the Residuals
par(mfrow=c(1,2),oma=c(0,0,0,0))
```

```
acf(res.loan.ar2,lag.max=25,type="correlation",main="ACF of the
Residuals \nof AR(2) Model")

acf(res.loan.ar2, lag.max=25,type="partial",main="PACF of the
Residuals \nof AR(2) Model")
```

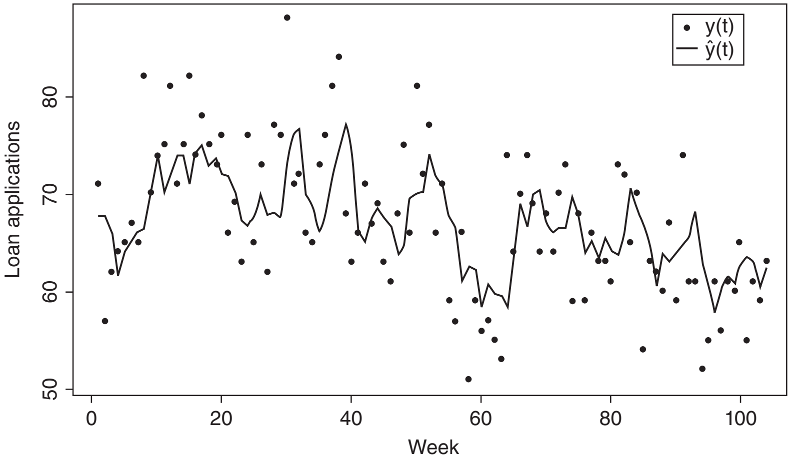



```
#4-in-1 plot of the residuals
par(mfrow=c(2,2),oma=c(0,0,0,0))
qqnorm(res.loan.ar2,datax=TRUE,pch=16,xlab='Residual',main='')
qqline(res.loan.ar2,datax=TRUE)
plot(fit.loan.ar2,res.loan.ar2,pch=16, xlab='Fitted Value',
ylab='Residual')
abline(h=0)
hist(res.loan.ar2,col="gray",xlab='Residual',main='')
plot(res.loan.ar2,type="l",xlab='Observation Order',
ylab='Residual')
points(res.loan.ar2,pch=16,cex=.5)
abline(h=0)
```



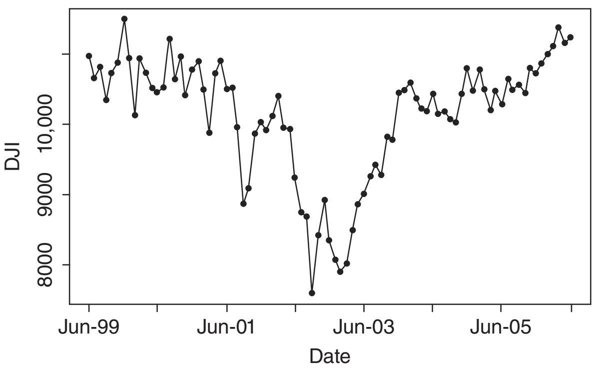
Plot fitted values

```
plot(loan.data[,2],type="p",pch=16,cex=.5,xlab='Week',ylab='Loan
Applications')
lines(fit.loan.ar2)
legend(95,88,c("y(t)","yhat(t)"), pch=c(16, NA),lwd=c(NA,.5),
cex=.55)
```

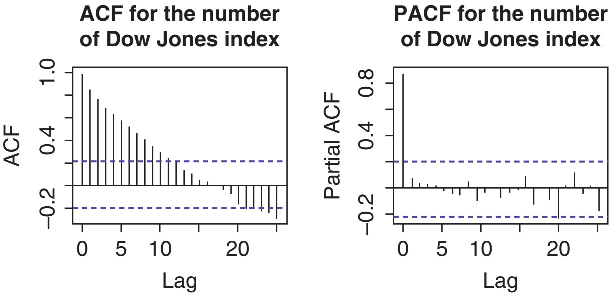


Example 5.2 The Dow Jones index data are in the second column of the array called dji.data in which the first column is the month of the year. We first plot the data as well as the ACF and PACF.

```
plot(dji.data[,2],type="o",pch=16,cex=.5,xlab='Date',ylab='DJI',
xaxt='n')
axis(1, seq(1,85,12), dji.data[seq(1,85,12),1])
```



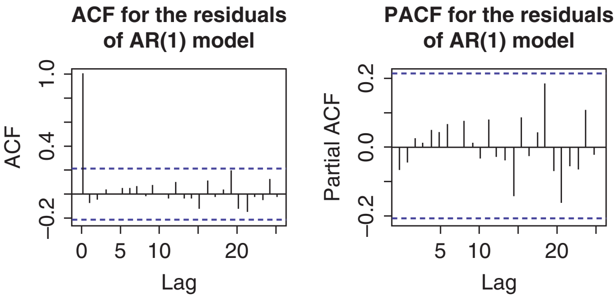
```
par(mfrow=c(1,2),oma=c(0,0,0,0))
acf(dji.data[,2],lag.max=25,type="correlation",main="ACF for the
Number \nof Dow Jones Index")

acf(dji.data[,2], lag.max=25,type="partial",main="PACF for the
Number \nof Dow Jones Index ")
```



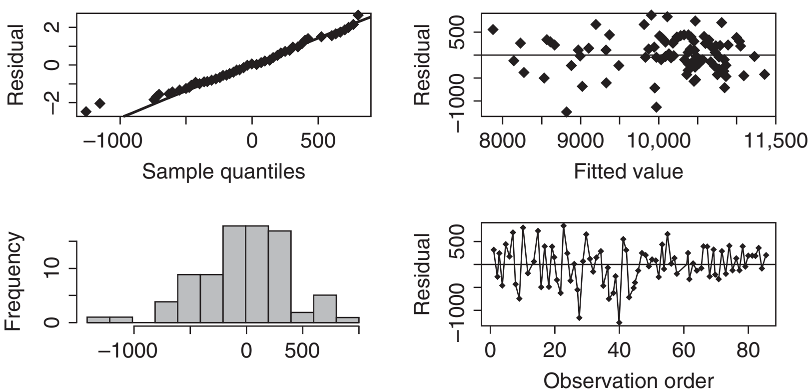
We first fit an ARIMA(1,0,0) model to the data using arima function in the stats package.

```
dji.fit.ar1<-arima(dji.data[,2],order=c(1, 0, 0))
dji.fit.ar1
       Call:
       arima(x = dji.data[, 2], order = c(1, 0, 0))

       Coefficients:
                ar1   intercept
             0.8934  10291.2984
       s.e.  0.0473    373.8723

       sigma^2 estimated as 156691:  log likelihood = -629.8,
       aic = 1265.59


res.dji.ar1<-as.vector(residuals(dji.fit.ar1))
#to obtain the fitted values we use the function fitted() from
#the forecast package
library(forecast)
fit.dji.ar1<-as.vector(fitted(dji.fit.ar1))

Box.test(res.dji.ar1,lag=48,fitdf=3,type="Ljung")

        Box-Ljung test

        data:  res.dji.ar1
        X-squared = 29.9747, df = 45, p-value = 0.9584
```

```
#ACF and PACF of the Residuals
par(mfrow=c(1,2),oma=c(0,0,0,0))
acf(res.dji.ar1,lag.max=25,type="correlation",main="ACF of the
Residuals \nof AR(1) Model")

acf(res.dji.ar1, lag.max=25,type="partial",main="PACF of the
Residuals \nof AR(1) Model")
```

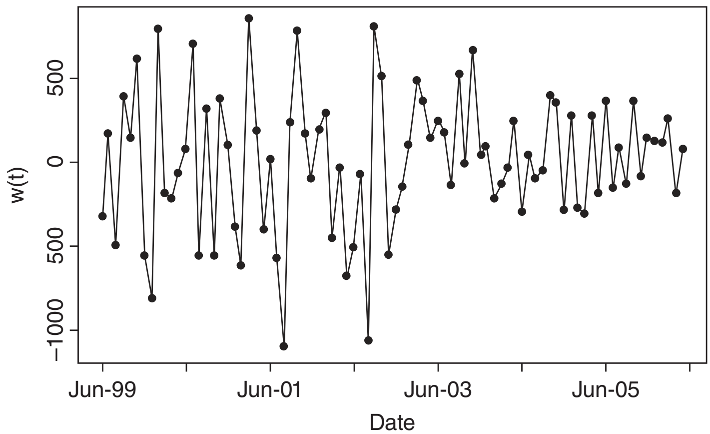



```
#4-in-1 plot of the residuals
par(mfrow=c(2,2),oma=c(0,0,0,0))
qqnorm(res.dji.ar1,datax=TRUE,pch=16,xlab='Residual',main='')
qqline(res.dji.ar1,datax=TRUE)
plot(fit.dji.ar1,res.dji.ar1,pch=16, xlab='Fitted Value',
ylab='Residual')
abline(h=0)
hist(res.dji.ar1,col="gray",xlab='Residual',main='')
plot(res.dji.ar1,type="l",xlab='Observation Order',
ylab='Residual')
points(res.dji.ar1,pch=16,cex=.5)
abline(h=0)
```

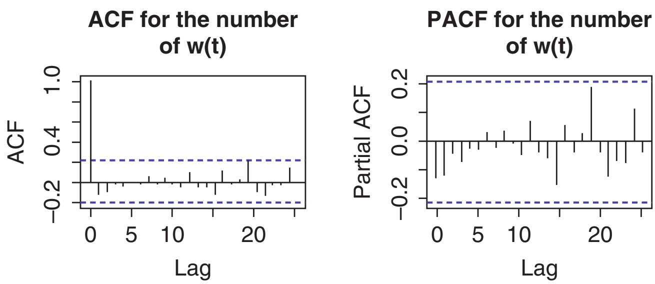


We now consider the first difference of the Dow Jones index.

```
wt.dji<-diff(dji.data[,2])
plot(wt.dji,type="o",pch=16,cex=.5,xlab='Date',ylab='w(t)',
xaxt='n')
axis(1, seq(1,85,12), dji.data[seq(1,85,12),1])
```




```
par(mfrow=c(1,2),oma=c(0,0,0,0))
acf(wt.dji,lag.max=25,type="correlation",main="ACF for the
Number \nof w(t)")

acf(wt.dji, lag.max=25,type="partial",main="PACF for the
Number \nof w(t)")
```



Example 5.6 The loan applications data are in the second column of the array called loan.data in which the first column is the number of weeks. We use the AR(2) model to make the forecasts.

```
loan.fit.ar2<-arima(loan.data[,2],order=c(2, 0, 0))
#to obtain the 1- to 12-step ahead forecasts, we use the
#function forecast() from the forecast package
library(forecast)
loan.ar2.forecast<-as.array(forecast(loan.fit.ar2,h=12))
loan.ar2.forecast
    Point  Forecast     Lo 80     Hi 80     Lo 95     Hi 95
105        62.58571  54.65250  70.51892  50.45291  74.71851
106        64.12744  55.91858  72.33629  51.57307  76.68180
107        64.36628  55.30492  73.42764  50.50812  78.22444
108        65.06647  55.80983  74.32312  50.90965  79.22330
109        65.35129  55.86218  74.84039  50.83895  79.86362
110        65.71617  56.13346  75.29889  51.06068  80.37167
111        65.93081  56.27109  75.59054  51.15754  80.70409
112        66.13857  56.43926  75.83789  51.30475  80.97240
113        66.28246  56.55529  76.00962  51.40605  81.15887
114        66.40651  56.66341  76.14961  51.50572  81.30730
115        66.49892  56.74534  76.25249  51.58211  81.41572
116        66.57472  56.81486  76.33458  51.64830  81.50114
```

Note that forecast function provides a list with forecasts as well as 80% and 95% prediction limits. To see the elements of the list, we can do

```
ls(loan.ar2.forecast)
       [1] "fitted"     "level"  "lower"  "mean"  "method"  "model"
       [7] "residuals"  "upper"  "x"      "xname"
```

In this list, "mean" stands for the forecasts while "lower" and "upper" provide the 80 and 95% lower and upper prediction limits, respectively. To plot the forecasts and the prediction limits, we have

```
plot(loan.data[,2],type="p",pch=16,cex=.5,xlab='Date',ylab='Loan
Applications',xaxt='n',xlim=c(1,120))
axis(1, seq(1,120,24), dji.data[seq(1,120,24),1])
lines(105:116,loan.ar2.forecast$mean,col="grey40")
lines(105:116,loan.ar2.forecast$lower[,2])
lines(105:116,loan.ar2.forecast$upper[,2])
legend(72,88,c("y","Forecast","95% LPL","95% UPL"), pch=c(16, NA,
NA,NA),lwd=c(NA,.5,.5,.5),cex=.55,col=c("black","grey40","black",
"black"))
```