A Biomed Data Analyst Training Program

Time series

Professor Ron S. Kenett

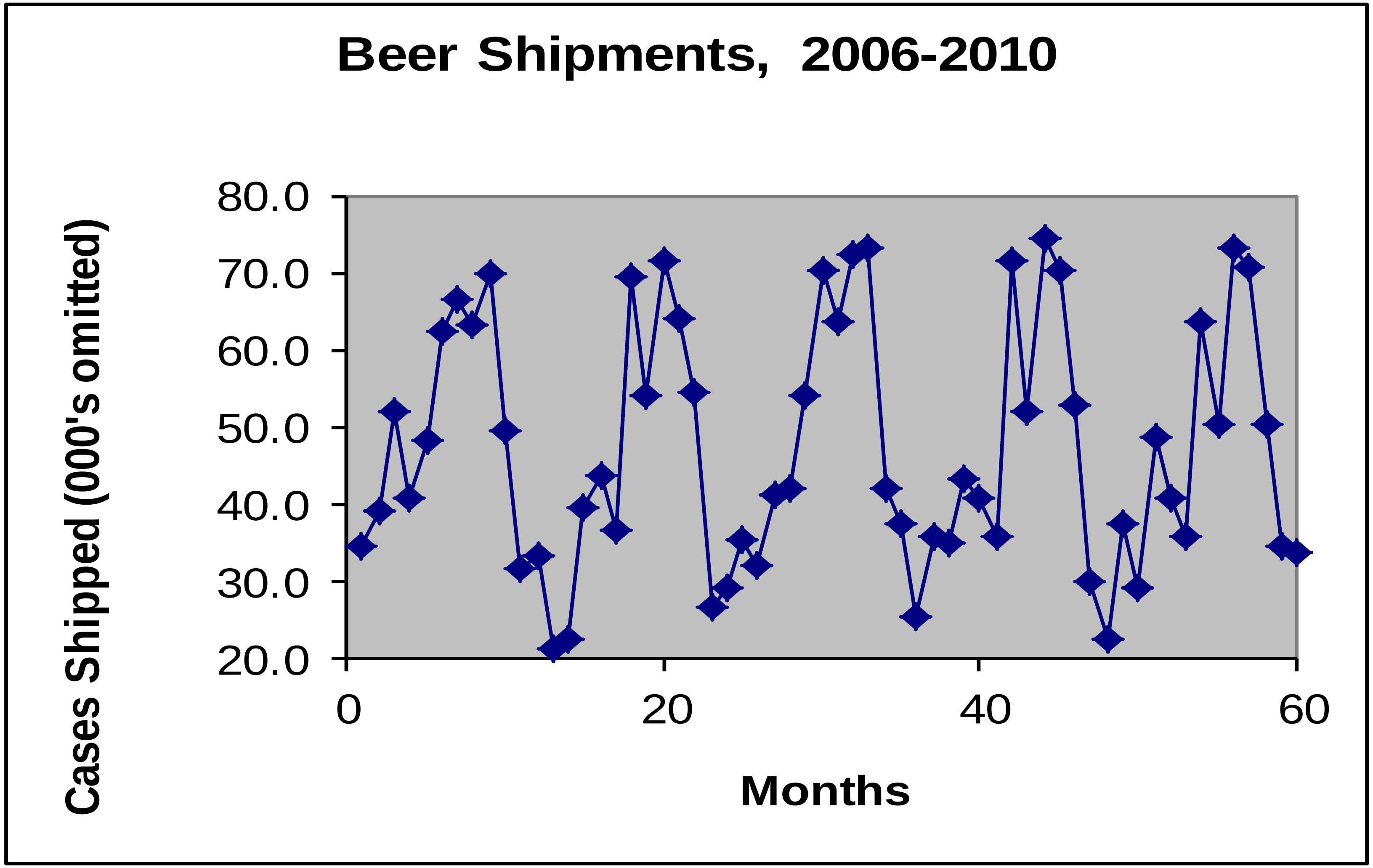
The Beer Deliveries Example

Deliveries of beer by a beer distributor over five years, the sixty months from January 2006 to December 2010

The data is measured as the number of cases distributed (000's omitted)

	January	February	March	April	May	June	July	August	September	October	November	December
2006	34.6	39.3	52.4	40.7	48.5	62.7	66.8	63.5	70.3	49.9	31.8	33.3
2007	21.3	22.6	39.5	43.9	36.7	69.7	54.2	71.8	64.4	54.7	26.7	29.1
2008	35.6	32.2	41.4	42.2	54.3	70.5	63.9	72.6	73.6	42.3	37.5	25.3
2009	35.9	35.2	43.6	41.0	35.8	71.8	52.0	74.7	70.6	53.0	29.9	22.5
2010	37.4	29.1	48.9	40.9	36.1	64.0	50.4	73.4	71.1	50.4	34.8	33.8

Run chart of sales



Three months moving average

$$\bar{X}_t = \frac{X_{t-1} + X_{t-2} + X_{t-3}}{3}$$

$$= \frac{34.6 + 39.3 + 52.4}{3} = \frac{126.3}{3} = 42.1$$

Three months average

Three months moving average

$$\bar{X}_t = \frac{39.3+52.4+40.7}{3} = \frac{132.4}{3} = 44.1$$

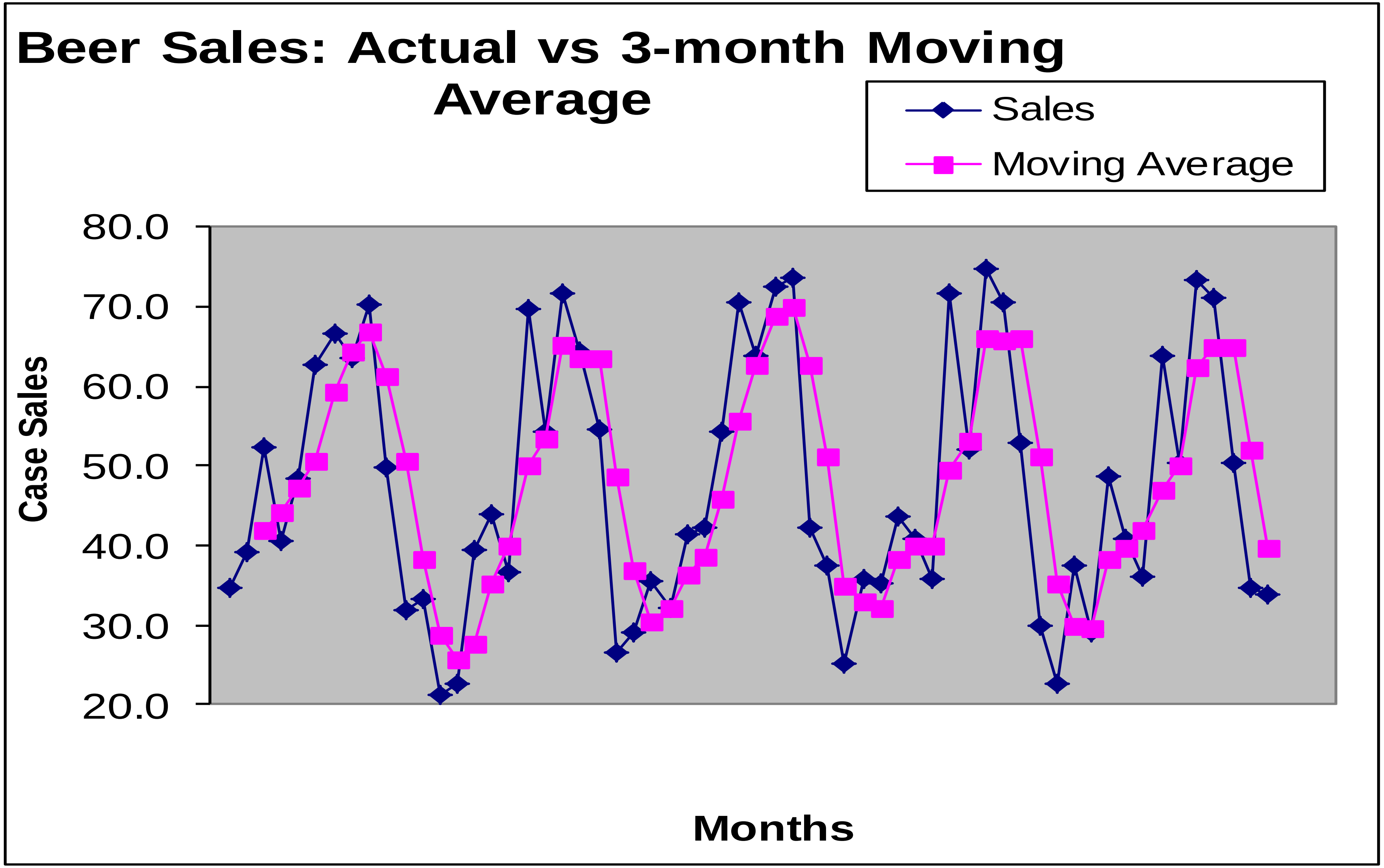
34.6

40.7

Average of next three months

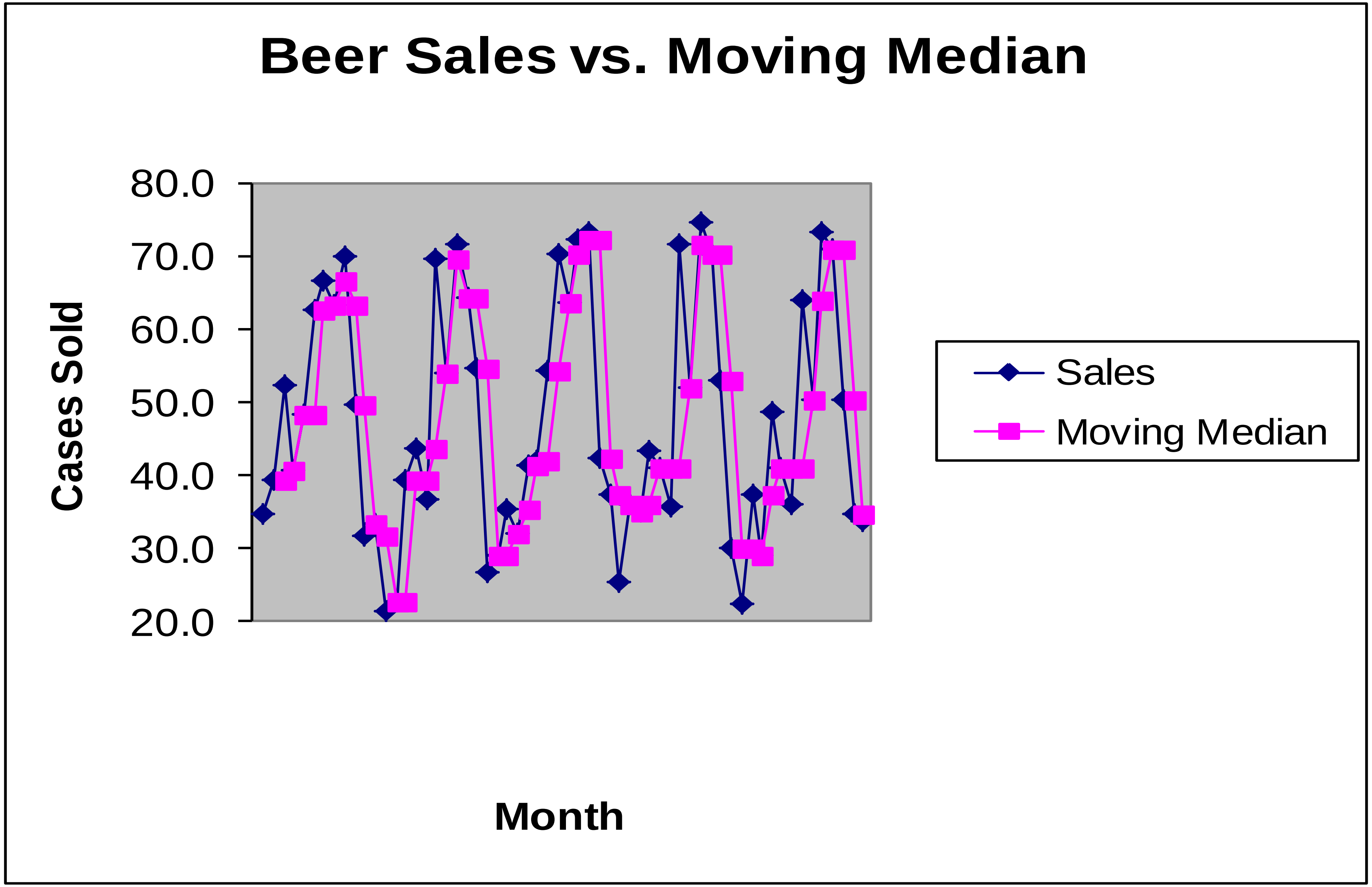
Three months moving average

Month	Sales	Moving Average
January-06	34.6	
February-06	39.3	
March-06	52.4	42.1
April-06	40.7	44.1
May-06	48.5	47.2
June-06	62.7	50.7
July-06	66.8	59.3


Beer Sales: Actual vs 3-month Moving Average
Sales
Moving Average
80.0
70.0
60.0
50.0
40.0
30.0
20.0
Case Sales
Months

Three months moving median

Month	Sales	Moving Median
January-06	34.6	
February-06	39.3	
March-06	52.4	39.3
April-06	40.7	40.7
May-06	48.5	48.5
June-06	62.7	48.5
July-06	66.8	62.7
August-06	63.5	63.5



Beer Sales vs. Moving Median
80.0
70.0
60.0
50.0
40.0
30.0
20.0
Cases Sold
Month
Sales
Moving Median

Exponential smoothing

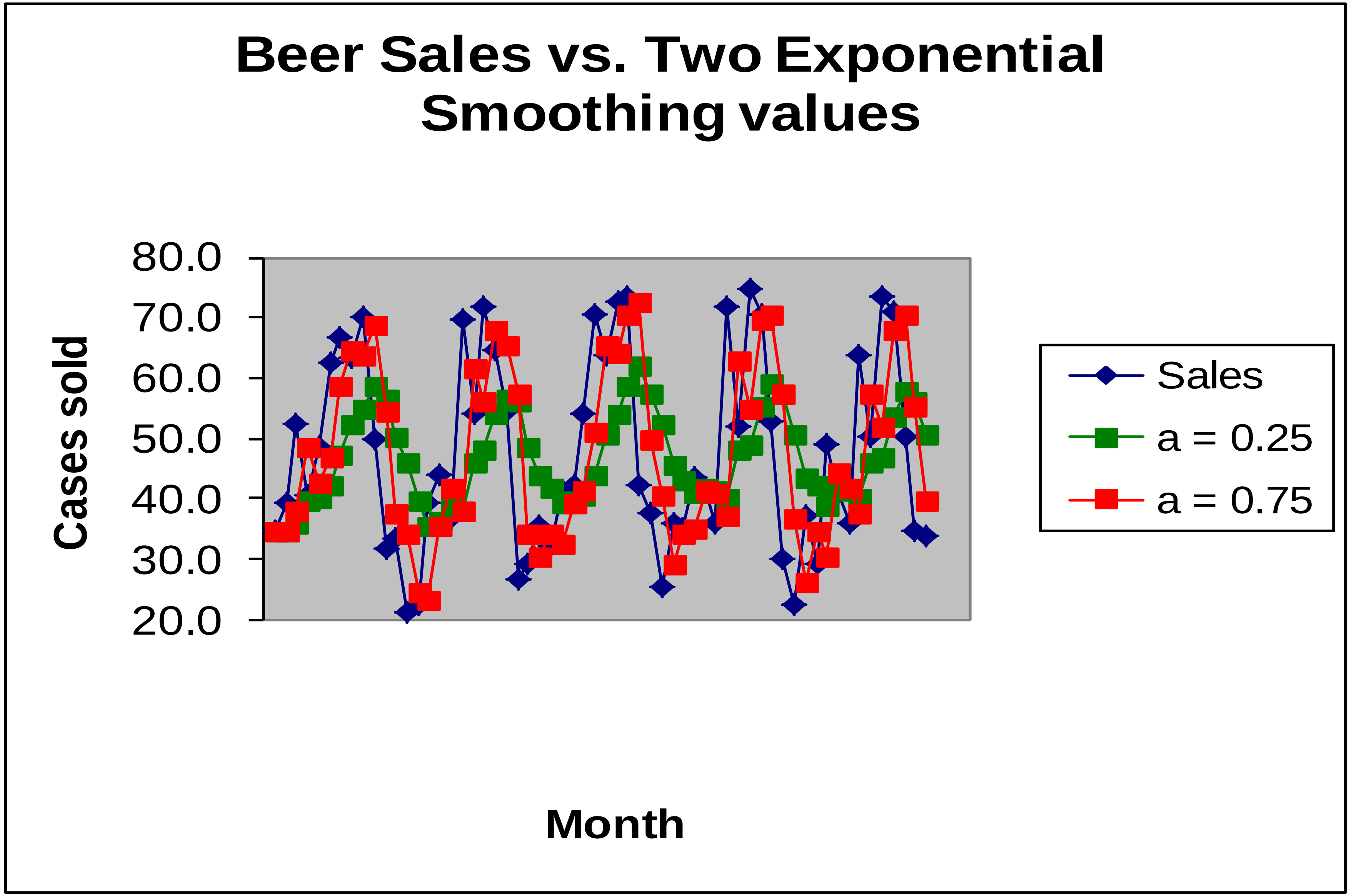
$$\hat{y}_{t+1} = ay_t + a(1-a)y_{t-1} + a(1-a)^2 y_{t-2} + \ldots$$

Generally, for data that is highly variable, a higher **a** is chosen

In practice, however, **a** seldom exceeds 0.5

For data which has more stability, a lower value of **a** is chosen

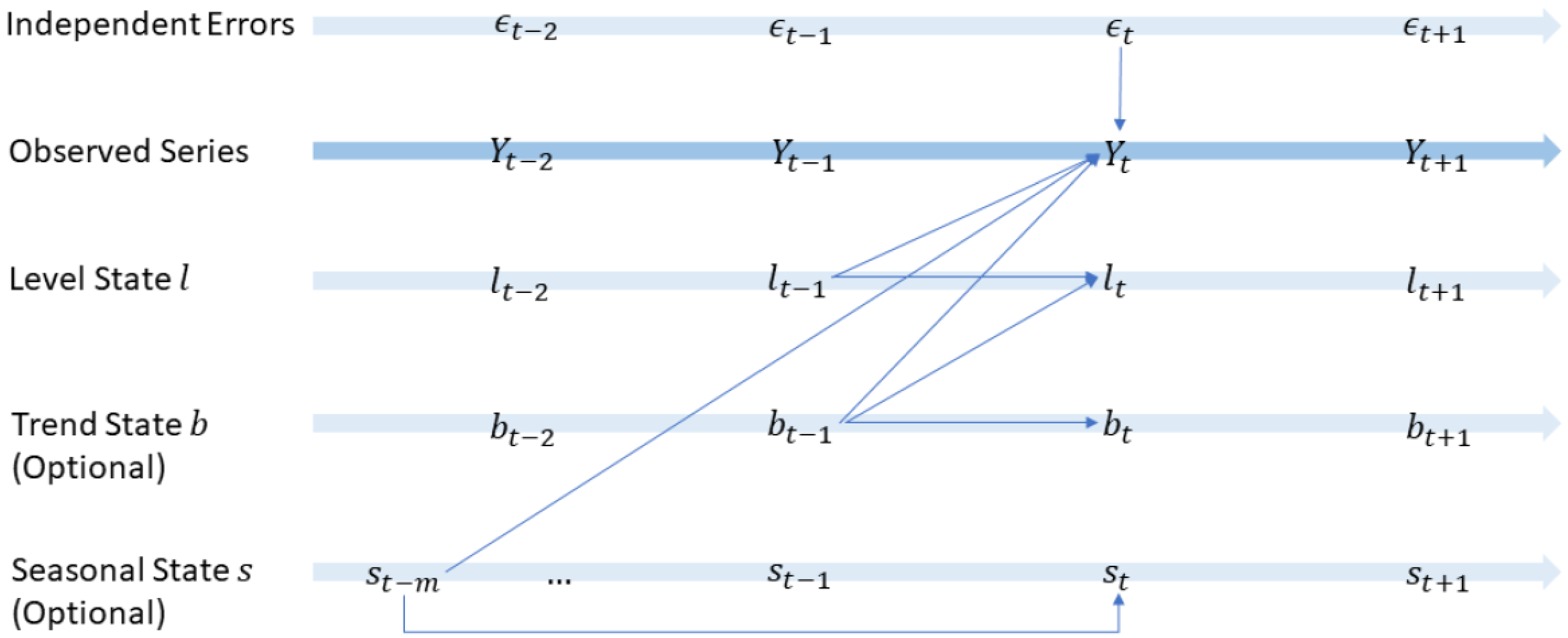
Typical **a** values range for 0.2 to 0.3



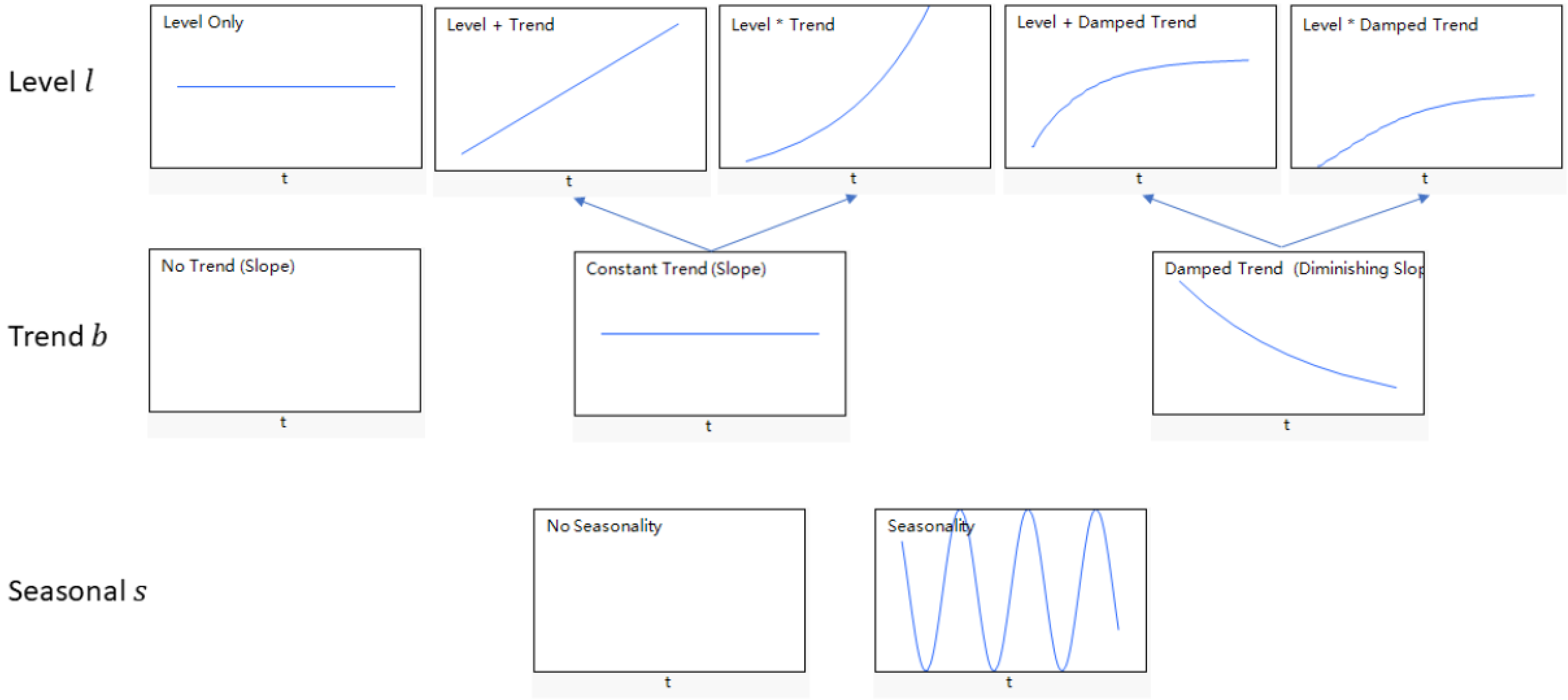
Beer Sales vs. Two Exponential Smoothing values
80.0
70.0
60.0
50.0
40.0
30.0
20.0
Cases sold
Month
Sales
a = 0.25
a = 0.75

State space model

$$Y_t = \left(\left(l_{t-1} \begin{matrix} + \\ \times \end{matrix} b_{t-1} \right) \begin{matrix} + \\ \times \end{matrix} s_{t-m} \right) \begin{matrix} + \epsilon_t \\ \times (1 + \epsilon_t) \end{matrix}$$



$$Y_t = \left(\left(l_{t-1} \begin{matrix} + \\ \times \end{matrix} b_{t-1} \right) \begin{matrix} + \\ \times \end{matrix} s_{t-m} \right) \begin{matrix} + \epsilon_t \\ \times (1 + \epsilon_t) \end{matrix}$$



The Multiplicative Model

$$y_t = T_t C_t S_t I_t$$

T_t = trend factor

C_t = cyclic factor

S_t = seasonal factor

I_t = random factor

The Multiplicative Model

Example:

1. The long-term trend for a specific product is +1.02
2. The cyclic factor at time *t* is 1.03
3. The seasonal factor at time *t* is 0.96
4. We cannot state the random factor

The Multiplicative Model

$$\hat{y}_t = T_t C_t S_t = 1.02 \cdot 1.03 \cdot 0.96$$
$$= 1.0086$$

- This predicts a slight increase (~ 1%) for period *t*
- If the actual increase was 1.0125, then I_t = 1.0125 / 1.0086 = 1.0039

The Multiplicative Model

We can also take away the effect of the long-term data

This is termed a *detrended series*

Recall that y_t = 1.0125 and that the trend factor was T_t = 1.02

$$\frac{y_t}{T_t} = \frac{T_t C_t S_t I_t}{T_t} = \frac{1.0125}{1.02} = 0.9927$$

- Thus, without the long-term trend, we would have shipped less units this period

The Additive Model

$$y_t = T_t + C_t + S_t + I_t$$

T_t = trend factor

C_t = cyclic factor

S_t = seasonal factor

I_t = random factor

The Additive Model

Say, we know that overtime, we ship 50 additional units per period, and in this business cycle we are shipping an additional 200 units

However, during this period, we typically ship 175 fewer units:

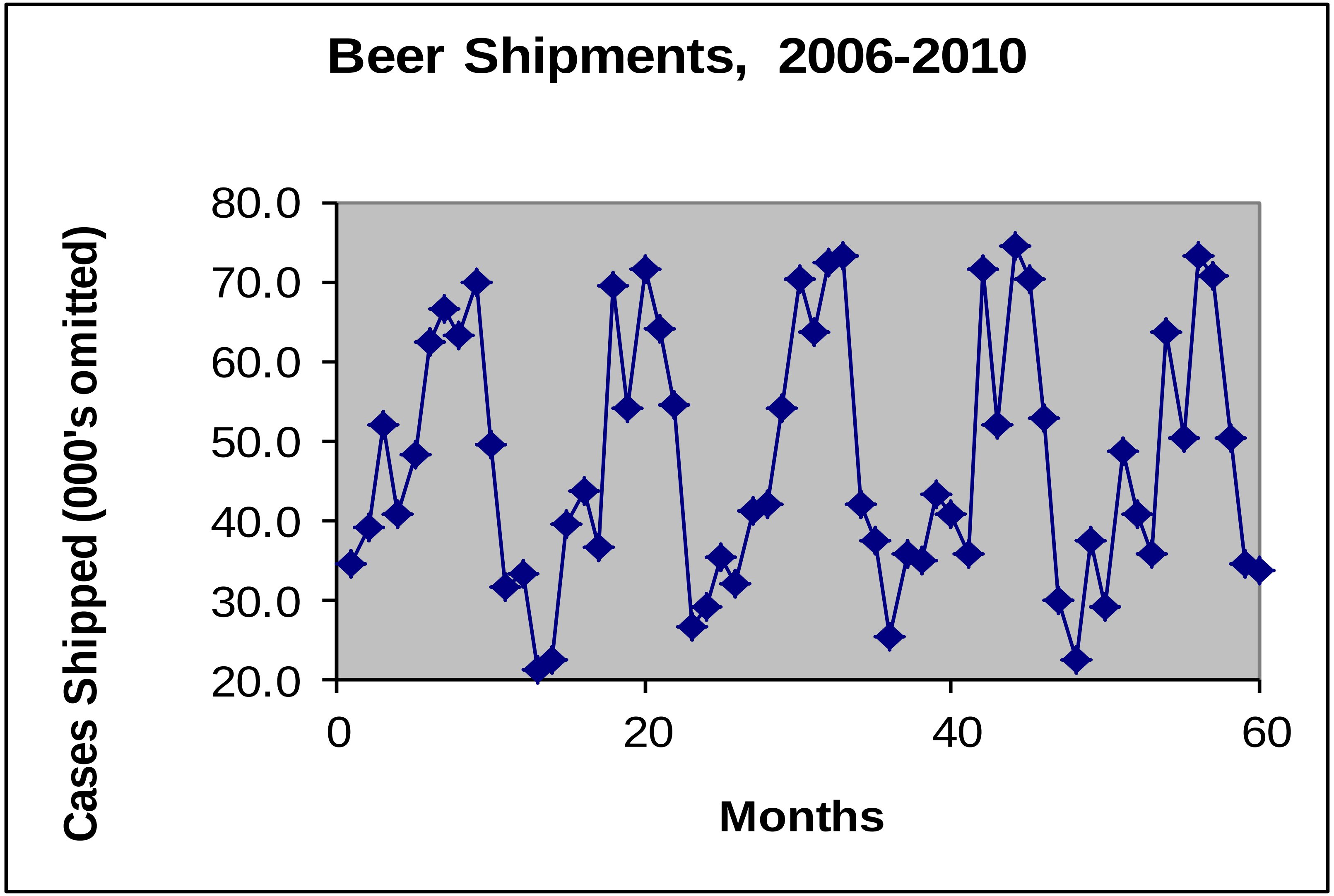
$$\hat{y}_t = T_t + C_t + S_t$$

$$= 50 + 200 - 175 = 75$$

The Additive Model

Thus, we could forecast shipping an additional 75 units this period
If we actually shipped 91 additional units

Then, I_t = (91 - 75) = 16



Beer Shipments, 2006-2010
80.0
70.0
60.0
50.0
40.0
30.0
20.0
Cases Shipped (000's omitted)
0
20
40
60
Months

Analysis of seasonal effects

	January	February	March	April	May	June	July	August	September	October	November	December
2006	34.6	39.3	52.4	40.7	48.5	62.7	66.8	63.5	70.3	49.9	31.8	33.3
2007	21.3	22.6	39.5	43.9	36.7	69.7	54.2	71.8	64.4	54.7	26.7	29.1
2008	35.6	32.2	41.4	42.2	54.3	70.5	63.9	72.6	73.6	42.3	37.5	25.3
2009	35.9	35.2	43.6	41.0	35.8	71.8	52.0	74.7	70.6	53.0	29.9	22.5
2010	37.4	29.1	48.9	40.9	36.1	64.0	50.4	73.4	71.1	50.4	34.8	33.8
Mean:	33.0	31.7	45.1	41.8	42.3	67.7	57.5	71.2	70.0	50.1	32.1	28.8
											Overall Mean:	47.6

Analysis of seasonal effects

For January, the average is 33.0 versus an overall mean of 47.6. The seasonal adjustment is thus:

$$S_{Jan} = \frac{\overline{X}}{\overline{X}_{Jan}} = \frac{47.6}{33.0} = 1.442$$

- As a result, we will multiple actual January sales by 1.442 to adjust for the fact that January is a low sales month

Analysis of seasonal effects

July, however, is a usually high sales month

As a result, its seasonal adjustment will bring its value down:

$$S_{Jul} = \frac{\overline{X}}{\overline{X}_{Jul}} = \frac{47.6}{57.5} = 0.828$$

- As a result, to "seasonally adjust", we multiple each July value by 0.828

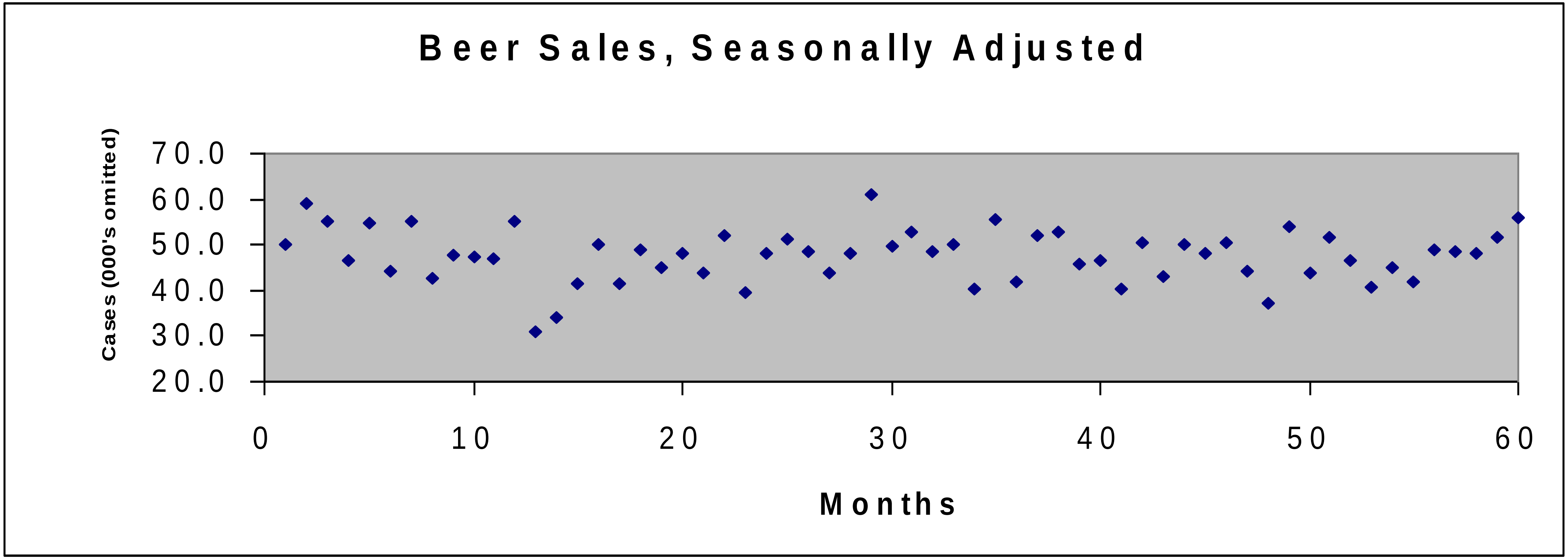
Analysis of seasonal effects

In the table below, the seasonal adjustment factors are applied to all the original values

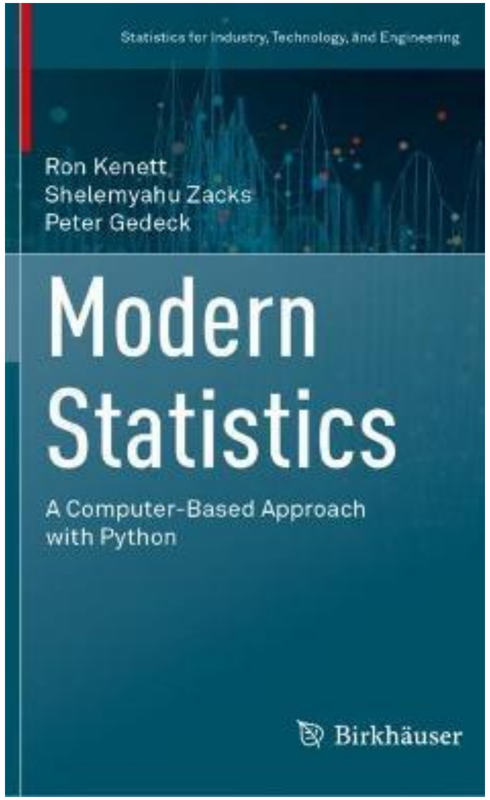
They are "deseasonalized":

	January	February	March	April	May	June	July	August	September	October	November	December
2006	50.0	59.0	55.2	46.4	54.6	44.1	55.3	42.5	47.8	47.4	47.1	55.1
2007	30.8	34.0	41.6	50.0	41.4	49.0	44.9	48.0	43.8	52.0	39.5	48.1
2008	51.3	48.4	43.7	48.1	61.1	49.5	52.9	48.5	50.0	40.2	55.5	41.8
2009	51.9	52.9	45.9	46.7	40.3	50.4	43.1	49.9	48.0	50.4	44.3	37.2
2010	54.0	43.7	51.5	46.7	40.6	45.0	41.8	49.1	48.4	47.9	51.5	55.8

Seasonality Adjusted Series



Chapter 6
Time Series Analysis and Prediction



Preview In this chapter, we present essential parts of time series analysis, with the objective of predicting or forecasting its future development. Predicting future behavior is generally more successful for stationary series, which do not change their stochastic characteristics as time proceeds. We develop and illustrate time series which are of both types, namely, covariance stationary and non-stationary.

Chapter 6

Modern Statistics: A Computer Based Approach with Python
by Ron Kenett, Shelemyahu Zacks, Peter Gedeck

Publisher: Springer International Publishing; 1st edition (September 15, 2022)
ISBN-13: 978-3031075650

(c) 2022 Ron Kenett, Shelemyahu Zacks, Peter Gedeck

The code needs to be executed in sequence.

Time Series Analysis and Prediction

Ron Kenett, Shelemyahu Zacks, Peter Gedeck

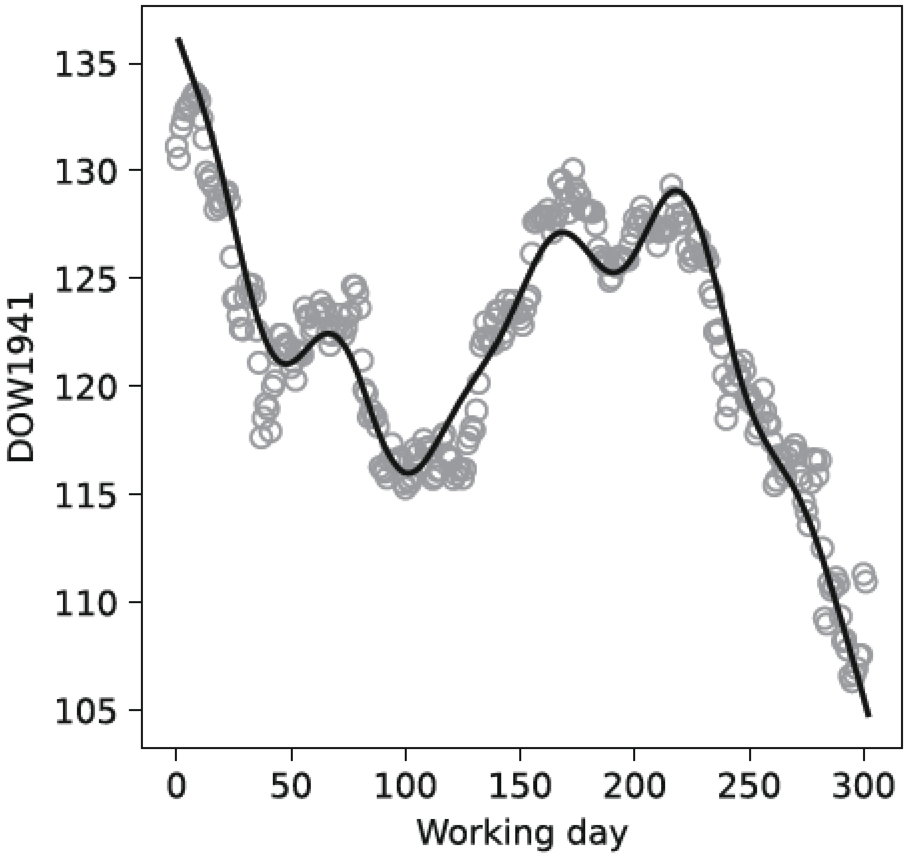
Pages 329-360

In [1]:
```
import os
os.environ['OUTDATED_IGNORE'] = '1'
import warnings
from outdated import OutdatedPackageWarning
warnings.filterwarnings('ignore', category=FutureWarning)
warnings.filterwarnings('ignore', category=OutdatedPackageWarning)
```

Time Series Analysis and Prediction

In [2]:
```
import datetime
import statsmodels.formula.api as smf
from statsmodels.tools.sm_exceptions import ValueWarning
import pandas as pd

import random
import numpy as np
import pingouin as pg
from scipy import stats
import matplotlib.pyplot as plt
import mistat
```

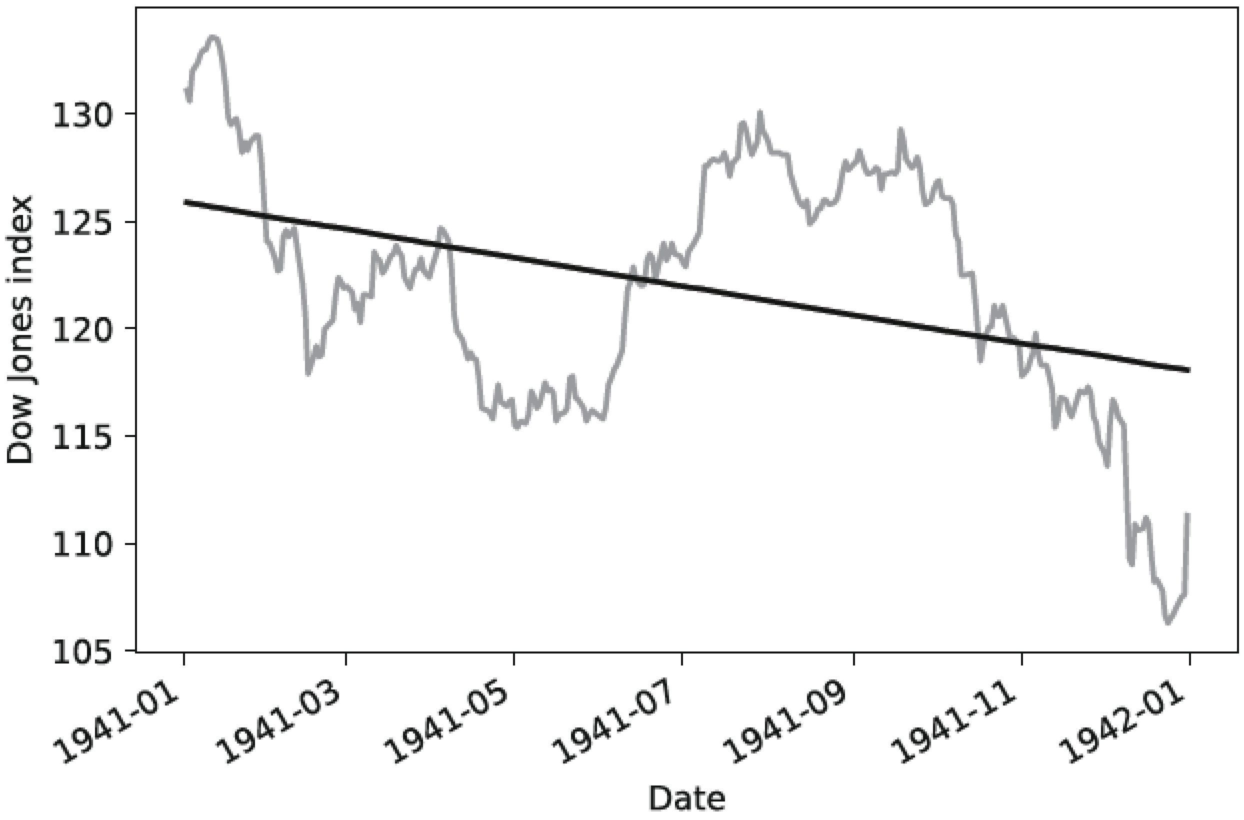
The Components of a Time Series



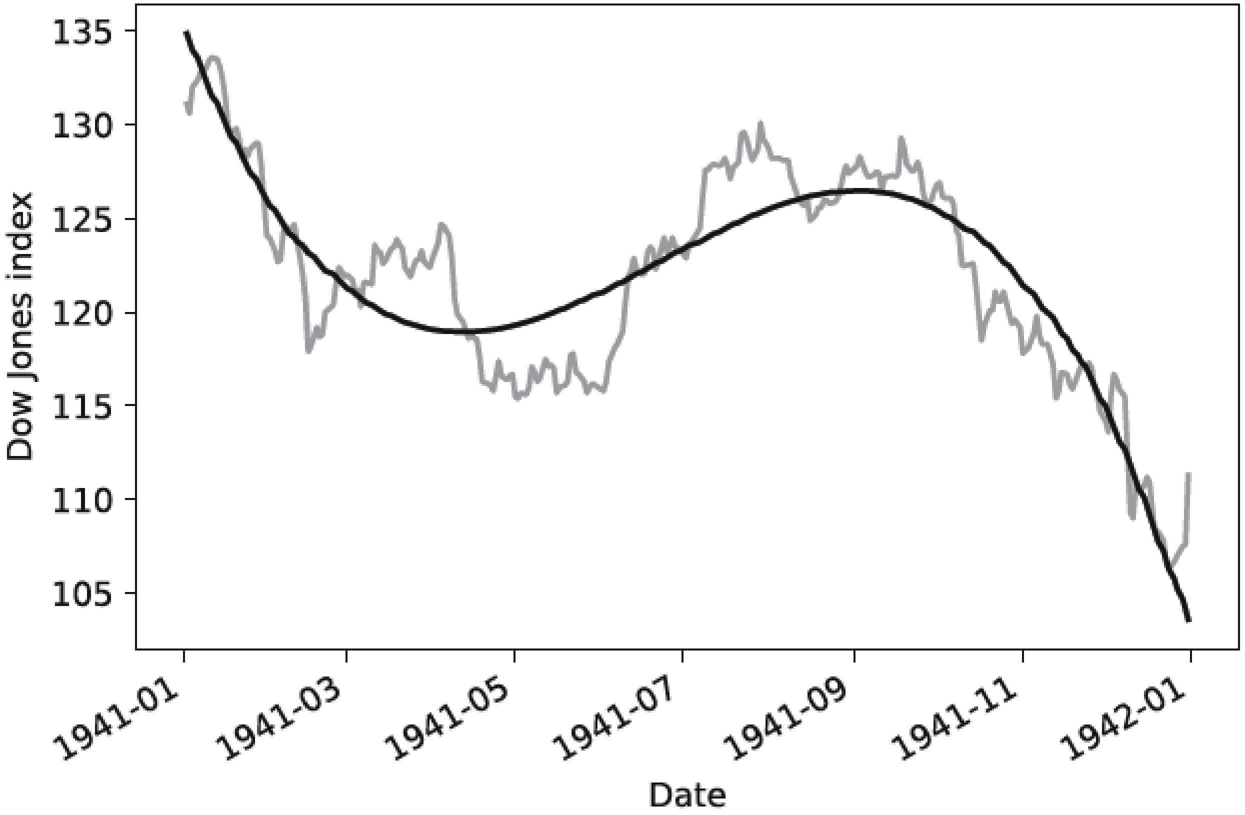
$$
\begin{aligned}
(t) =& 123.34 + 27.73\frac{t-151}{302} - 15.83\left(\frac{t-151}{302}\right)^2 - 237.00\left(\frac{t-151}{302}\right)^3 \\
&+ 0.1512\cos\frac{4\pi t}{302} + 1.738\sin\frac{4\pi t}{302} + 1.770\cos\frac{8\pi t}{302} - 0.208\sin\frac{8\pi t}{302} \\
&- 0.729\cos\frac{12\pi t}{302} + 0.748\sin\frac{12\pi t}{302}.
\end{aligned}
$$

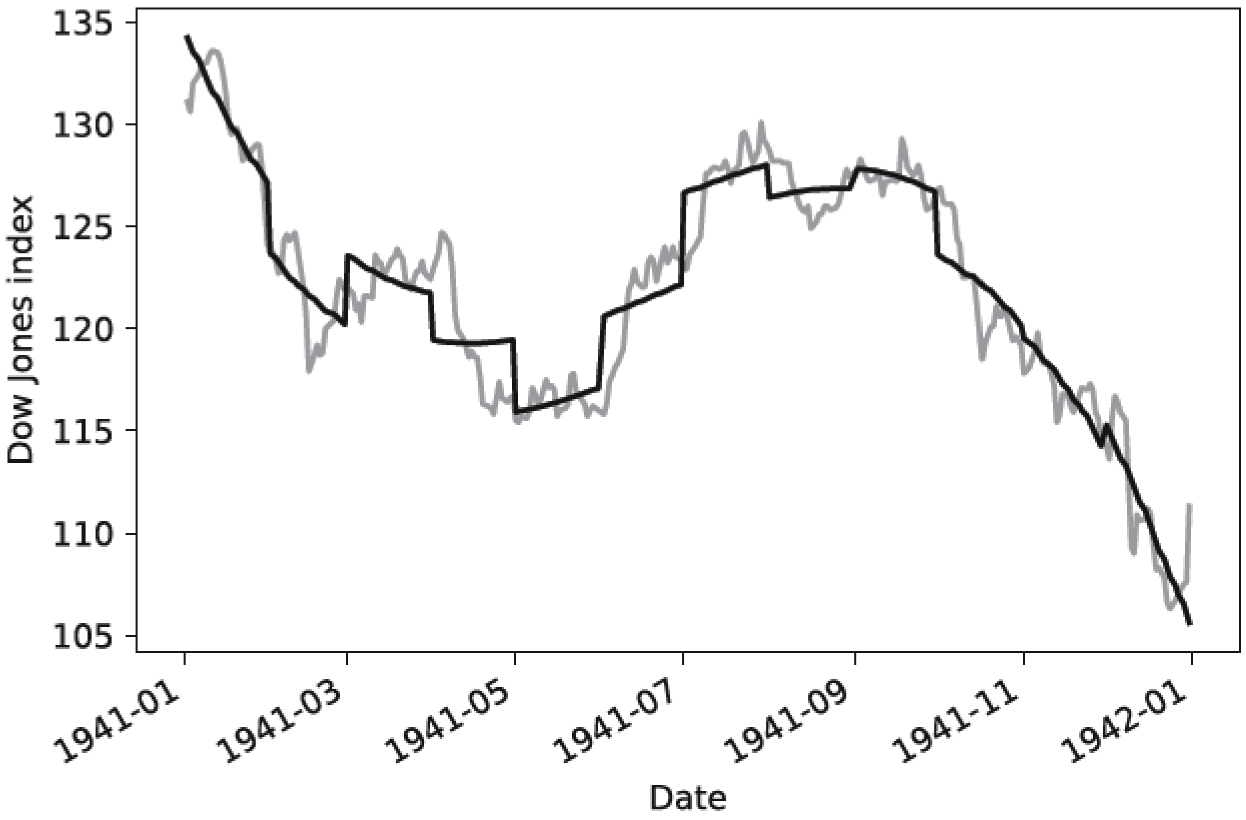
```python
dow1941 = mistat.load_data('DOW1941')
t = np.arange(1, len(dow1941) + 1)
x = (t - 151) / 302
omega = 4 * np.pi * t / 302
ft = (123.34 + 27.73 * x - 15.83* x ** 2 - 237.00 * x**3
      + 0.1512 * np.cos(omega) + 1.738 * np.sin(omega)
      + 1.770 * np.cos(2 * omega) - 0.208 * np.sin(2 * omega)
      - 0.729 * np.cos(3 * omega) + 0.748 * np.sin(3 * omega))

fig, ax = plt.subplots(figsize=[4, 4])
ax.scatter(dow1941.index, dow1941, facecolors='none', edgecolors='grey')
ax.plot(t, ft, color='black')
ax.set_xlabel('Working day')
ax.set_ylabel('DOW1941')
plt.show()
```

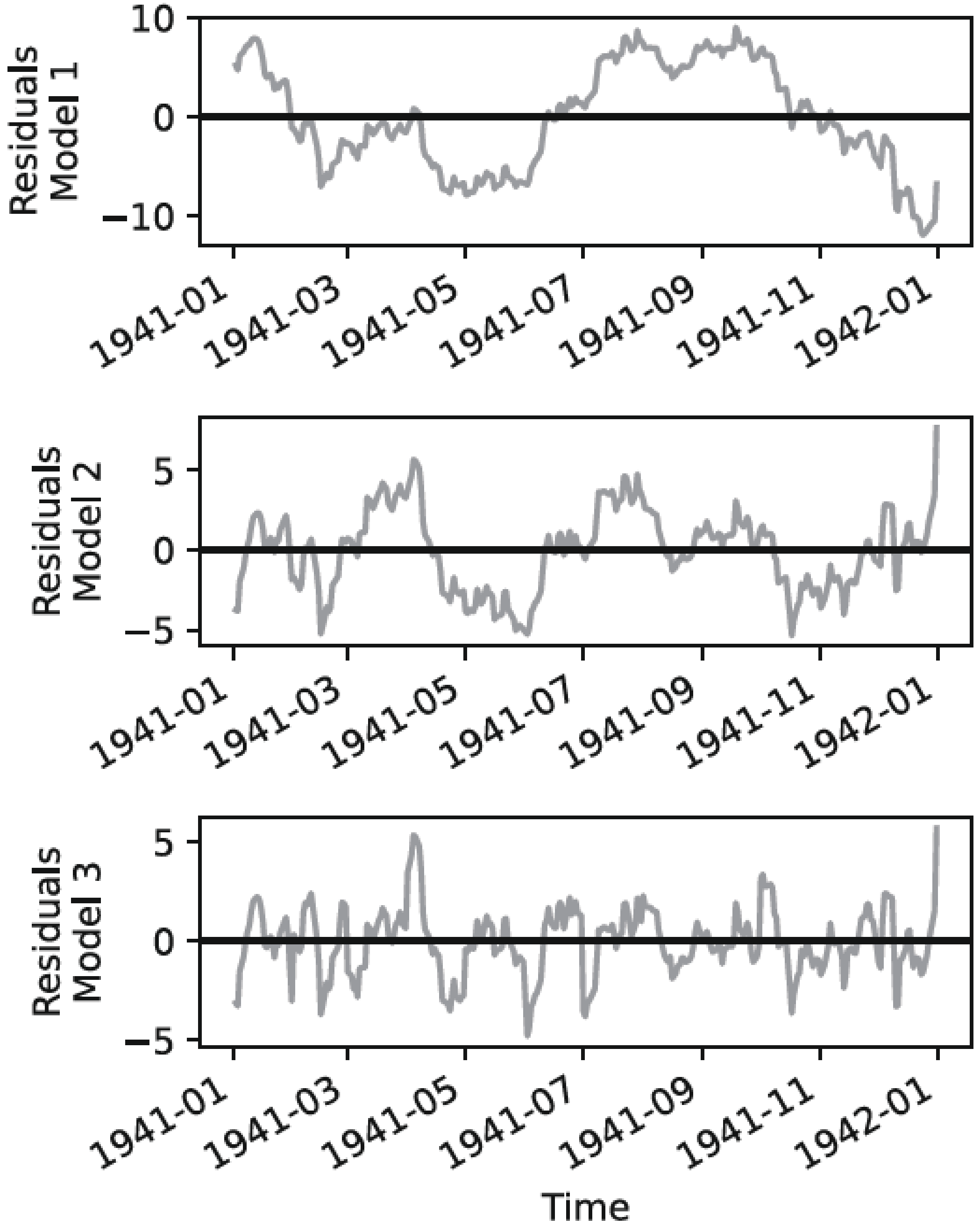
$$
\begin{aligned}
f(t) =& 123.34 + 27.73\frac{t-151}{302} - 15.83\left(\frac{t-151}{302}\right)^2 - 237.00\left(\frac{t-151}{302}\right)^3 \\
& + 0.1512\cos\frac{4\pi t}{302} + 1.738\sin\frac{4\pi t}{302} + 1.770\cos\frac{8\pi t}{302} - 0.208\sin\frac{8\pi t}{302} \\
& - 0.729\cos\frac{12\pi t}{302} + 0.748\sin\frac{12\pi t}{302}.
\end{aligned}
$$



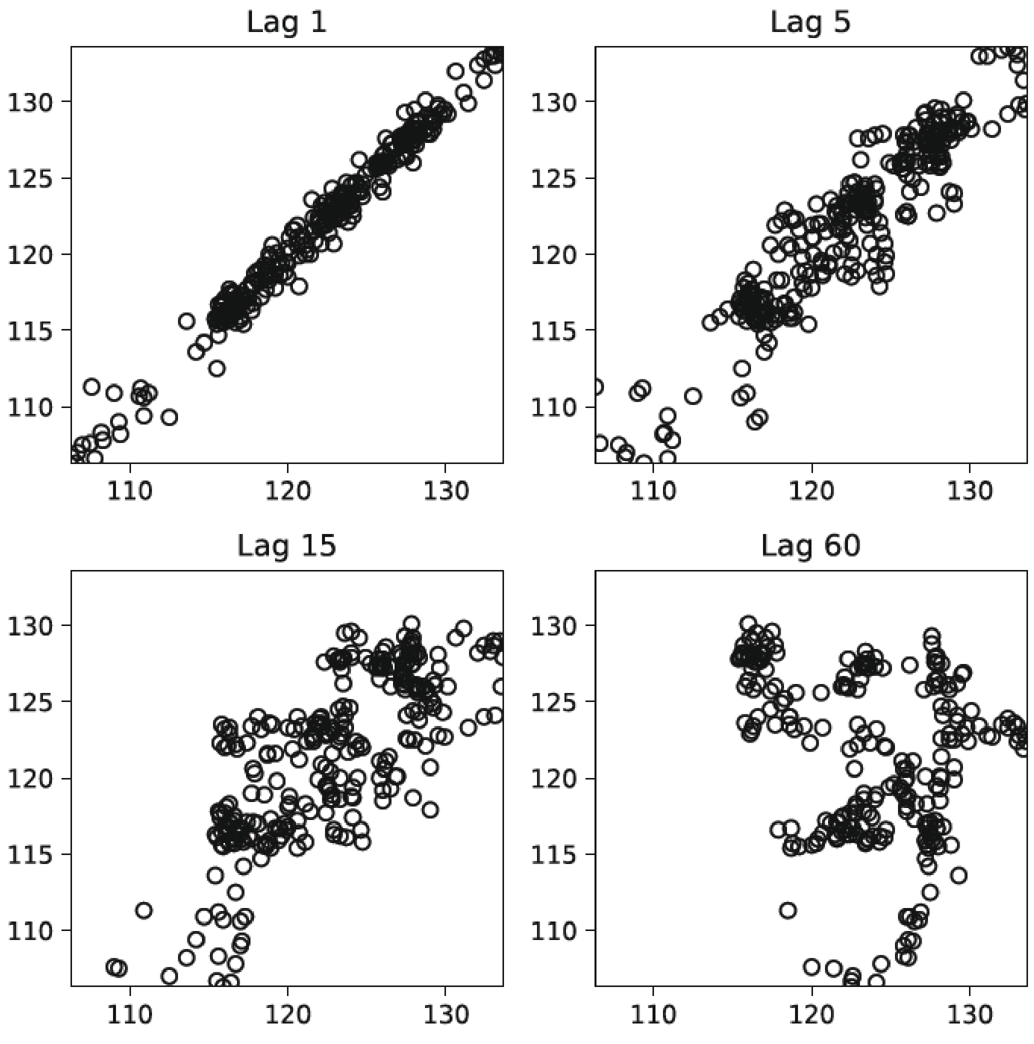
130
125
120
115
110
105
Dow Jones index
1941-01
1941-03
1941-05
1941-07
1941-09
1941-11
1942-01
Date


135
130
125
120
115
110
105
Dow Jones index
1941-01
1941-03
1941-05
1941-07
1941-09
1941-11
1942-01
Date


135
130
125
120
115
110
105
Dow Jones index
1941-01
1941-03
1941-05
1941-07
1941-09
1941-11
1942-01
Date


Residuals Model 1
10
0
−10
Residuals Model 2
5
0
−5
Residuals Model 3
5
0
−5
1941-01
1941-03
1941-05
1941-07
1941-09
1941-11
1942-01
Time

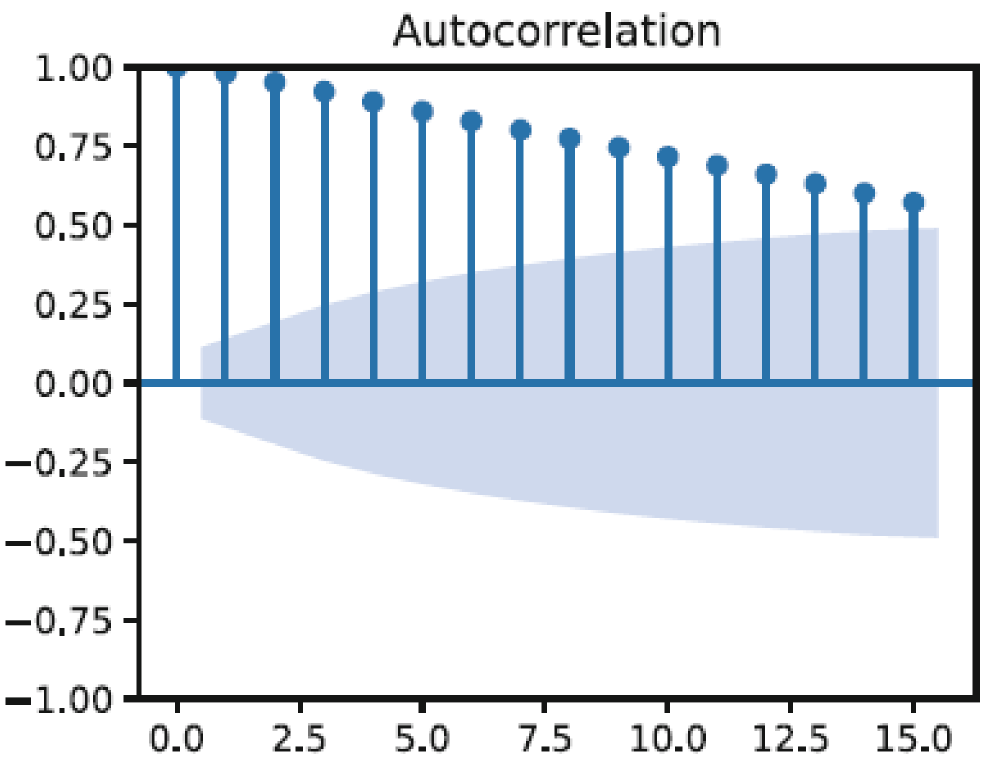
Autocorrelations

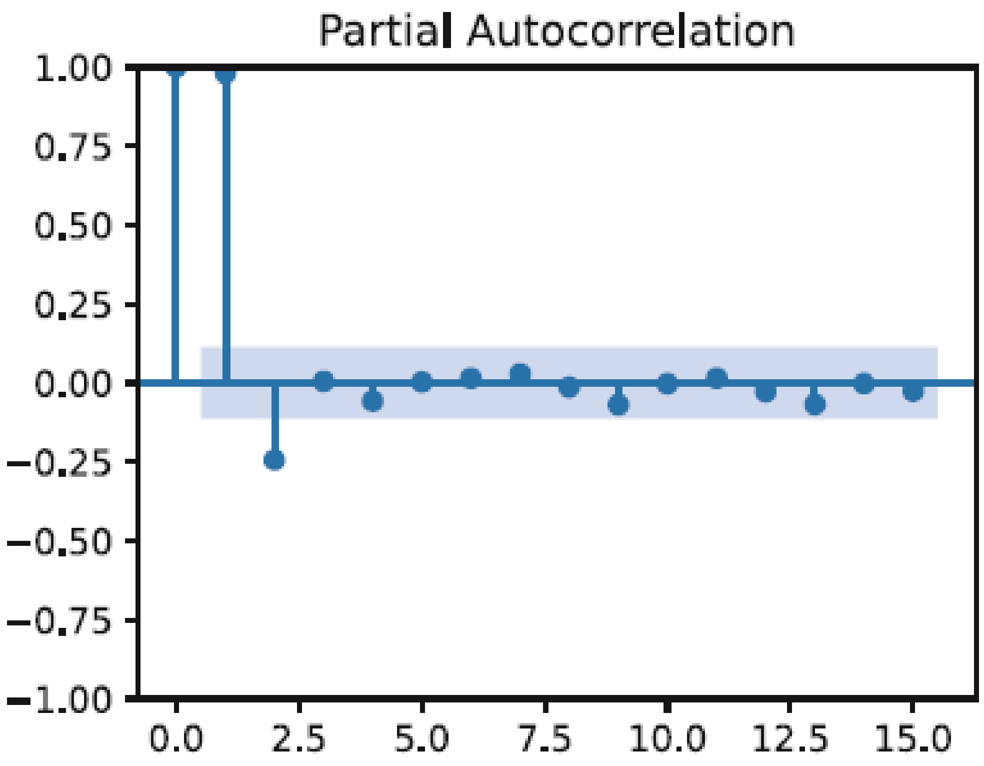


Time Series Basic Diagnostics

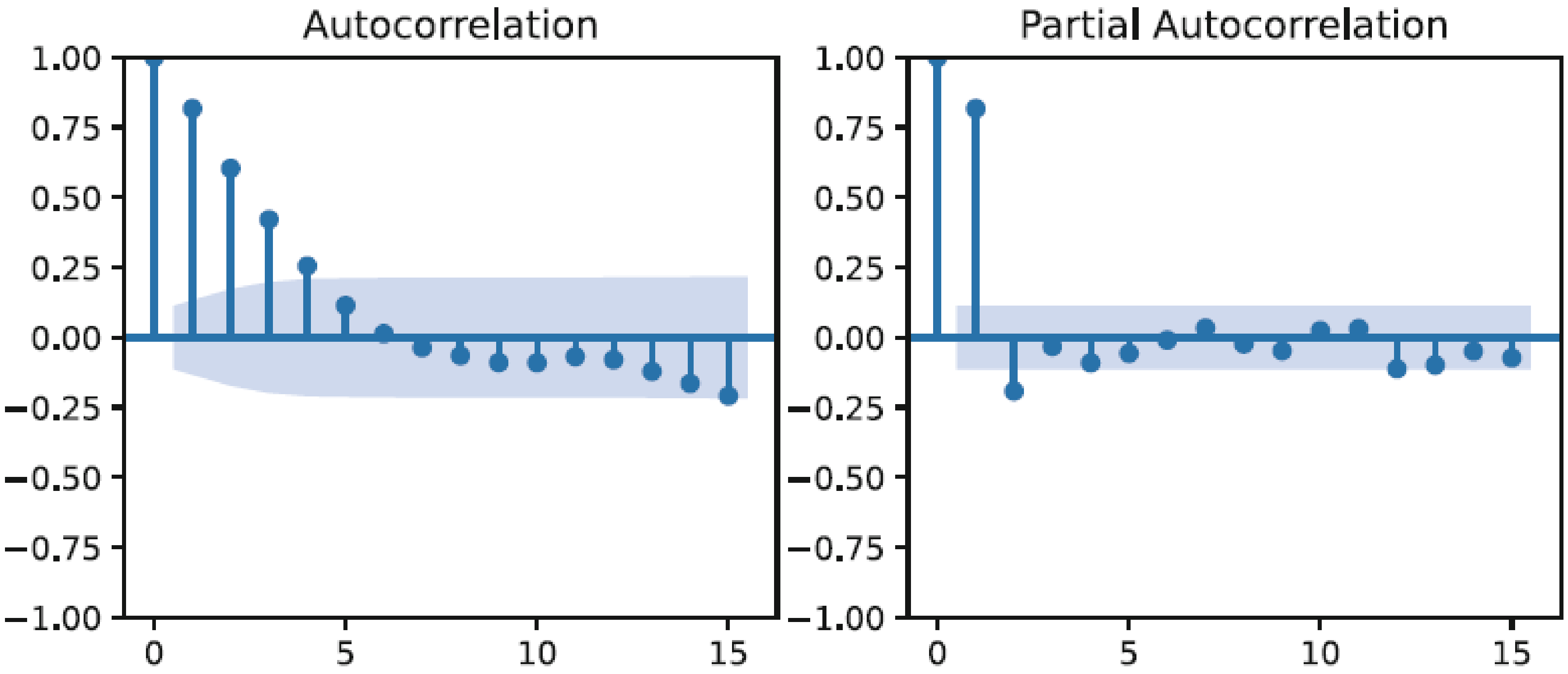
Lag	AutoCorr	Ljung-Box Q	p-Value	Lag	Partial
0	1.0000	.	.	0	1.0000
1	0.9789	292.287	<.0001*	1	0.9789
2	0.9540	570.791	<.0001*	2	-0.1035
3	0.9241	833.020	<.0001*	3	-0.1228
4	0.8936	1079.05	<.0001*	4	-0.0122
5	0.8622	1308.87	<.0001*	5	-0.0254
6	0.8328	1523.98	<.0001*	6	0.0357
7	0.8051	1725.71	<.0001*	7	0.0220
8	0.7774	1914.44	<.0001*	8	-0.0323
9	0.7486	2090.05	<.0001*	9	-0.0504
10	0.7204	2253.20	<.0001*	10	-0.0005
11	0.6919	2404.26	<.0001*	11	-0.0139
12	0.6627	2543.28	<.0001*	12	-0.0357
13	0.6324	2670.32	<.0001*	13	-0.0385
14	0.6027	2786.12	<.0001*	14	0.0012
15	0.5730	2891.13	<.0001*	15	-0.0197
16	0.5418	2985.36	<.0001*	16	-0.0543
17	0.5119	3069.78	<.0001*	17	0.0174
18	0.4824	3145.02	<.0001*	18	-0.0144
19	0.4536	3211.76	<.0001*	19	-0.0104
20	0.4304	3272.07	<.0001*	20	0.1205
21	0.4130	3327.80	<.0001*	21	0.1013
22	0.3937	3378.62	<.0001*	22	-0.1170
23	0.3731	3424.43	<.0001*	23	-0.0653
24	0.3497	3464.83	<.0001*	24	-0.0670
25	0.3280	3500.48	<.0001*	25	0.0480

Autocorrelations and Partial Autocorrelations





Autocorrelations and Partial Autocorrelations

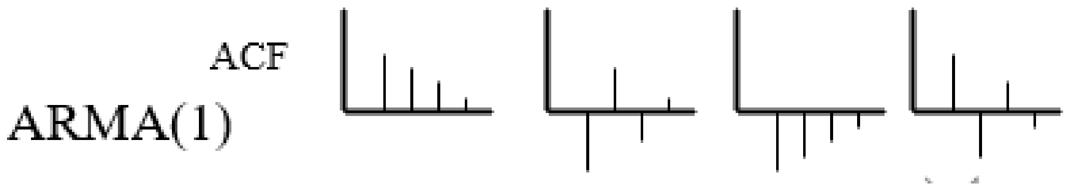


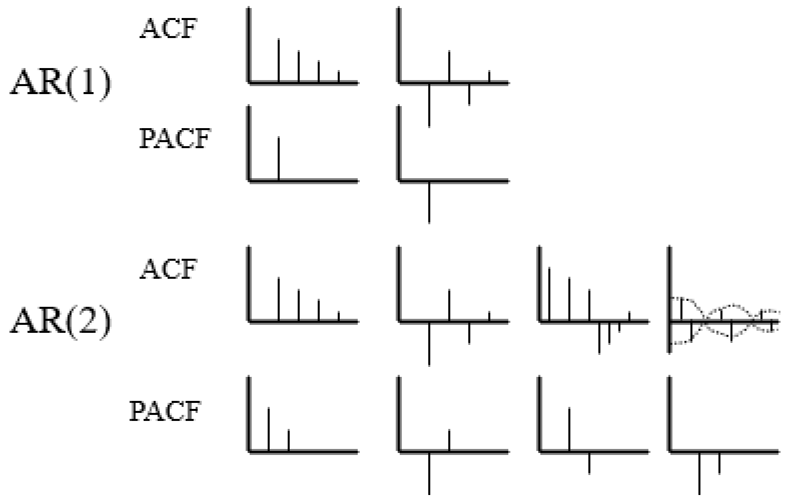
ARIMA Models

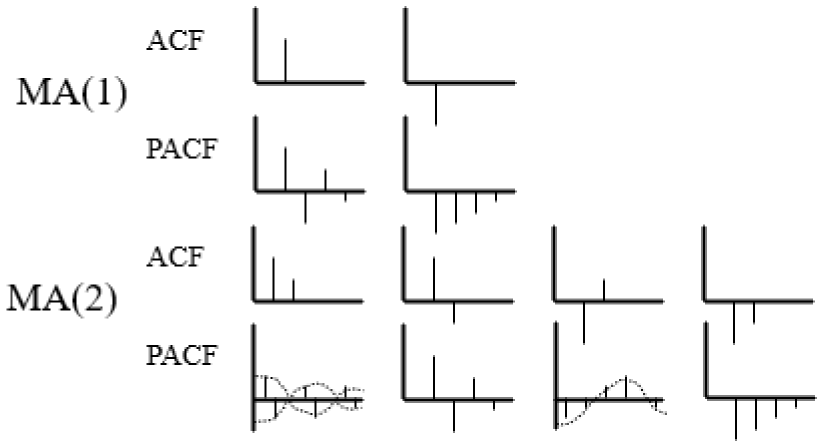
1. Auto Regressive (AR)

2. Moving Average (MA)

3. Auto Regressive Integrated Moving Average (ARIMA)







First-Order Autoregressive Processes, AR(1):

$$y_t = \delta + \theta_1 y_{t-1} + e_t, \quad t = 1, 2, \ldots, T.$$

δ is the intercept.

θ_1 is parameter generally between -1 and +1.

e_t is an uncorrelated random error with mean zero and variance σ_e^2.

Autoregressive Process of order p, AR(p) :

$$y_t = \delta + \theta_1 y_{t-1} + \theta_2 y_{t-2} + \ldots + \theta_p y_{t-p} + e_t$$

δ is the intercept.

θ_i's are parameters generally between -1 and +1.

e_t is an uncorrelated random error with mean zero and variance σ_e^2 .

AR(2) model of U.S. unemployment rates

$$y_t = 0.5051 + 1.5537\, y_{t-1} - 0.6515\, y_{t-2}$$
$$\quad (0.1267) \qquad (0.0707) \qquad\qquad (0.0708)$$

positive

negative

Using AR Model for Forecasting:

unemployment rate: $y_{T-1} = 6.63$ and $y_T = 6.20$

$$y_{T+1} = \delta + \theta_1 y_T + \theta_2 y_{T-1} = 0.5051 + (1.5537)(6.2) - (0.6515)(6.63)$$

$$= 5.8186$$

$$y_{T+2} = \delta + \theta_1 y_{T+1} + \theta_2 y_T = 0.5051 + (1.5537)(5.8186) - (0.6515)(6.2)$$

$$= 5.5062$$

Choosing the lag length, p, for AR(p):

The Partial Autocorrelation Function (PAF)

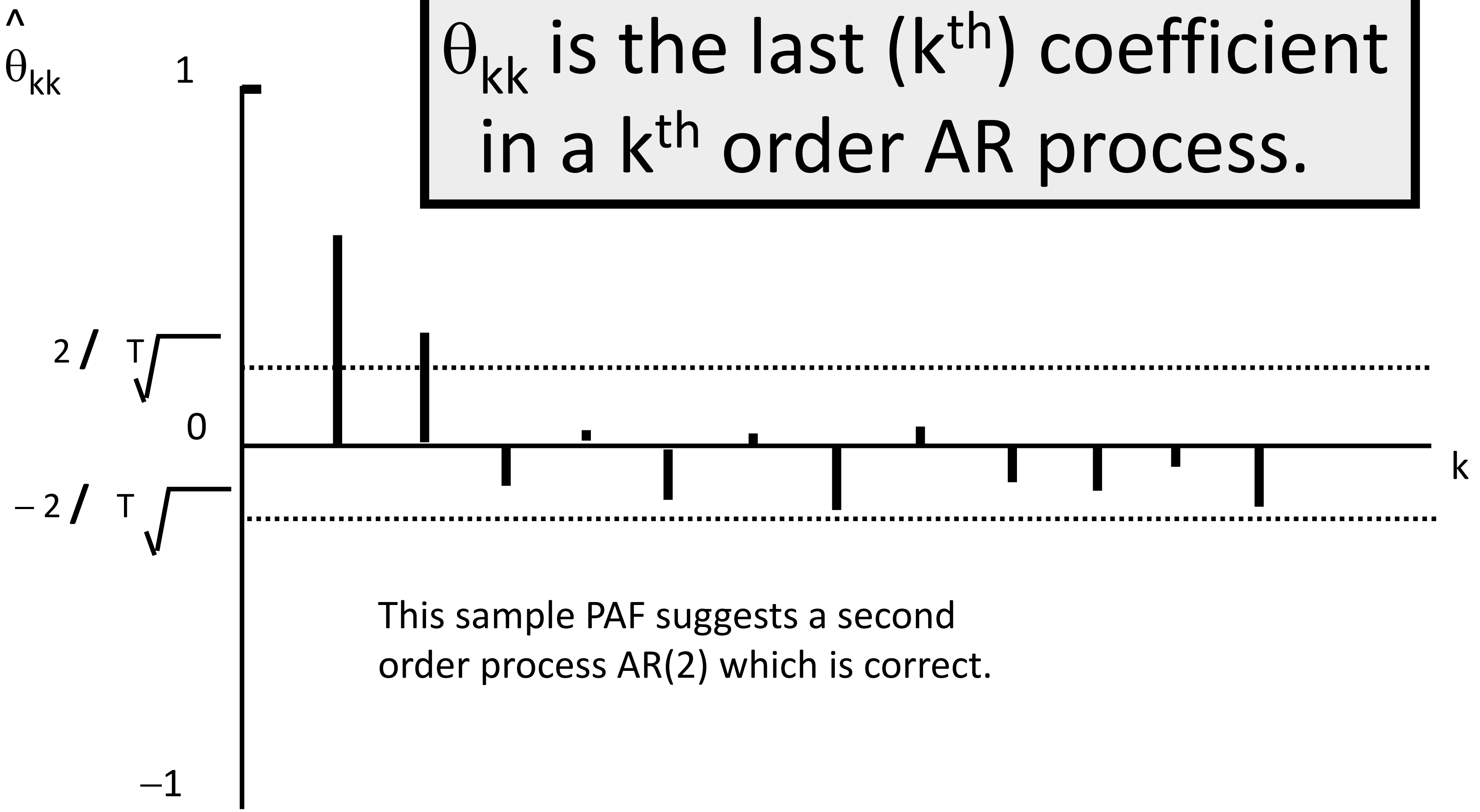
The PAF is the sequence of correlations between (y_t and y_{t-1}), (y_t and y_{t-2}), (y_t and y_{t-3}), and so on, given that the effects of earlier lags on y_t are held constant.

Partial Autocorrelation Function

Data simulated from this model:

$$y_t = 0.5\, y_{t-1} + 0.3\, y_{t-2} + e_t$$

θ_{kk} is the last (k^{th}) coefficient in a k^{th} order AR process.



This sample PAF suggests a second order process AR(2) which is correct.

Moving Average Process of order q, MA(q):

$$y_t = \mu + e_t + \alpha_1 e_{t-1} + \alpha_2 e_{t-2} + \ldots + \alpha_q e_{t-q}$$

μ is the intercept.

α_i's are unknown parameters.

e_t is an uncorrelated random error with mean zero and variance σ_e^2.

An MA(1) process:

$$y_t = \mu + e_t + \alpha_1 e_{t-1}$$

Minimize sum of least squares deviations:

$$S(\mu,\alpha_1) = \sum_{t=1}^{T} e_t^2 = \sum_{t=1}^{T} (y_t - \mu - \alpha_1 e_{t-1})^2$$

Choosing the lag length, q, for MA(q):

The Autocorrelation Function (ACF)

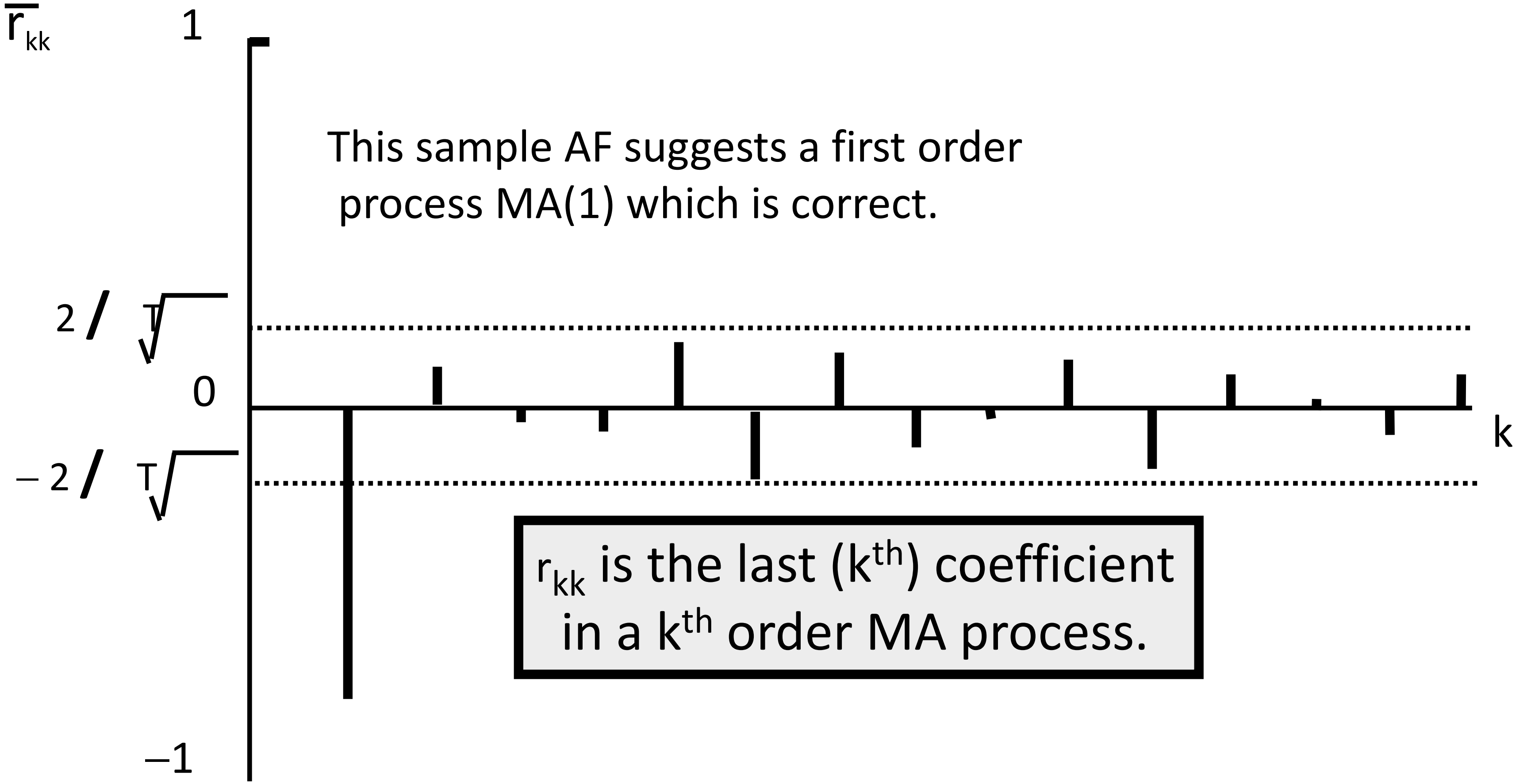
The ACF is the sequence of correlations between (y_t and y_{t-1}), (y_t and y_{t-2}), (y_t and y_{t-3}), and so on, without holding the effects of earlier lags on y_t constant.

The PAF controlled for the effects of previous lags but the ACF does not control for such effects.

Autocorrelation Function

Data simulated
from this model:

$$y_t = e_t - 0.9\, e_{t-1}$$



This sample AF suggests a first order process MA(1) which is correct.

r_{kk} is the last (k^{th}) coefficient in a k^{th} order MA process.

Autoregressive Moving Average ARMA(p,q)

An ARMA(1,2) has one autoregressive lag and two moving average lags:

$$\mathbf{y_t = \delta + \theta_1 y_{t-1} + e_t + \alpha_1 e_{t-1} + \alpha_2 e_{t-2}}$$

Goodness of fit criteria

- Akaike's Information Criterion [AIC]
- Schwartz's Bayesian Criterion [BIC]
- -2LogLikelihood

The smaller the better….

Stationary vs. Nonstationary

Stationary:

A stationary time series is one whose mean, variance, and autocorrelation function do not change over time.

Nonstationary:

A nonstationary time series is one whose mean, variance or autocorrelation function change over time.

First Differencing is often used to transform a nonstationary series into a stationary series:

$$y_t = z_t - z_{t-1}$$

where z_t is the original nonstationary series and y_t is the new stationary series.

Auto Regressive Integrated Moving Average, ARIMA(p,d,q)

An ARIMA(p,d,q) model represents an AR(p) - MA(q) process that has been differenced (integrated, I(d)) d times

$$y_t = \delta + \theta_1 y_{t-1} + \ldots + \theta_p y_{t-p} + e_t + \alpha_1 e_{t-1} + \ldots + \alpha_q e_{t-q}$$

The Box-Jenkins approach:

1. Identification
   determining the values of p, d, and q
2. Estimation
   linear or nonlinear least squares
3. Diagnostic Checking
   model fits well with no autocorrelation?
4. Forecasting
   short-term forecasts of future y_t values

A Case Study: Series F

The series consists of 70 observations on the yield of a batch chemical process.

Series FJMP

A Case Study: Series F

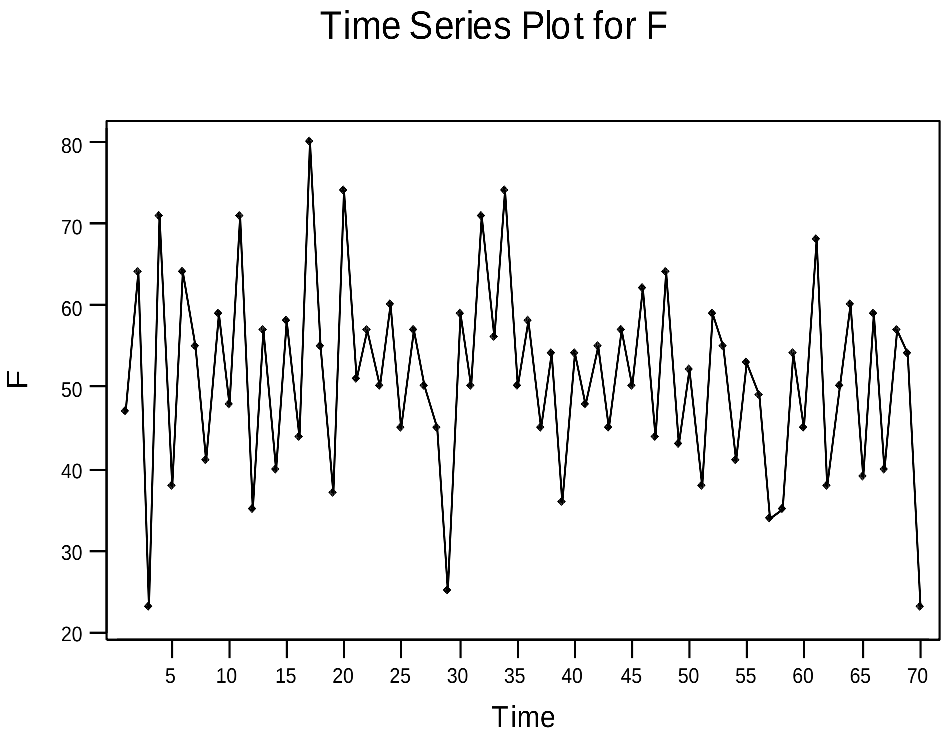
First we plot the series to check for trend, periodicity, etc. which will need the application of differencing.

We inspect the ACF and PAFto help in identifying an ARMA model for the stationary series we obtain.

The Autocorrelation Function (ACF)

The Partial Autocorrelation Function (PAF)

Time Plot of Series F



No obvious non-stationarity in the form of trend or periodic effects.

No apparent need to difference the series.

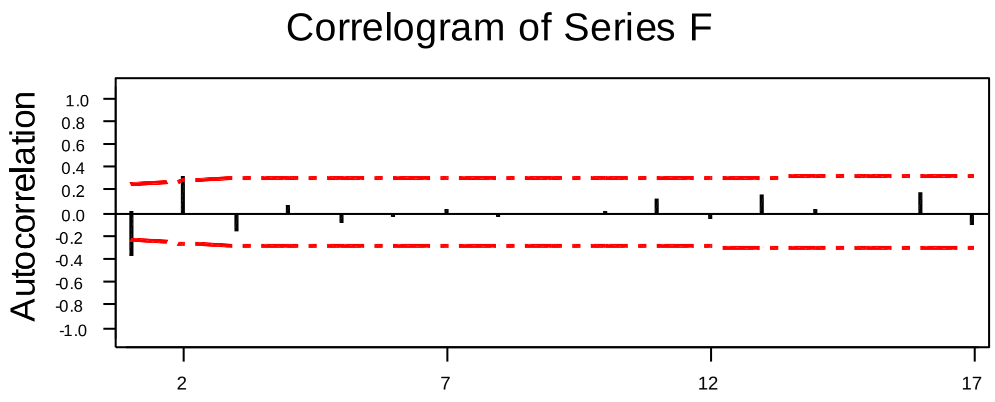
Model Identification.

- The theoretical acf of a MA(q) series shows a cutoff after lag q.
- The ACF of an AR(p) series theoretically shows a geometric decline after lag p.
- The pacf of an AR(p) series theoretically shows a cutoff at lag p.

More Identification.

- Often several models look plausible .
- We can try to identify the order of an AR process by fitting several models of orders in the region of p which we think is plausible. Plotting the residual sum of squares against p may show a “flattening” for values beyond the “true” order.

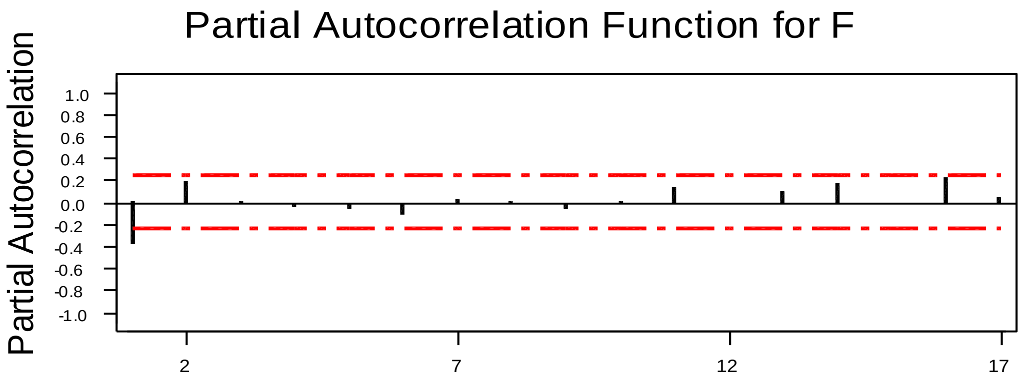
Sample ACF of Series F



Lag	Corr	T	LBQ	Lag	Corr	T	LBQ	Lag	Corr	T	LBQ
1	-0.39	-3.26	11.10	8	-0.04	-0.29	21.57	15	-0.01	-0.04	25.09
2	0.30	2.23	17.97	9	-0.00	-0.03	21.57	16	0.17	1.12	27.89
3	-0.17	-1.14	20.03	10	0.01	0.10	21.59	17	-0.11	-0.71	29.06
4	0.07	0.48	20.41	11	0.11	0.73	22.62				
5	-0.10	-0.65	21.14	12	-0.07	-0.45	23.03				
6	-0.05	-0.31	21.32	13	0.15	0.97	24.97				
7	0.04	0.24	21.42	14	0.04	0.23	25.09				

- Let's look at a correlogram of the series.
- The only large values are at lags 1 and 2.
- Maybe AR(2) or AR(1)?

Partial Correlogram



Lag	PAC	T	Lag	PAC	T	Lag	PAC	T
1	-0.39	-3.26	8	0.00	0.04	15	-0.00	-0.01
2	0.18	1.50	9	-0.06	-0.47	16	0.22	1.85
3	0.00	0.02	10	0.00	0.03	17	0.05	0.44
4	-0.04	-0.37	11	0.14	1.19			
5	-0.07	-0.58	12	-0.01	-0.08			
6	-0.12	-1.01	13	0.09	0.77			
7	0.02	0.16	14	0.17	1.40			

The partial correlogram also only has appreciable values at lags 1 and 2.

Try an AR(2) model.

i.e. p=2, d=0, q=0

Fitting an AR(2) model

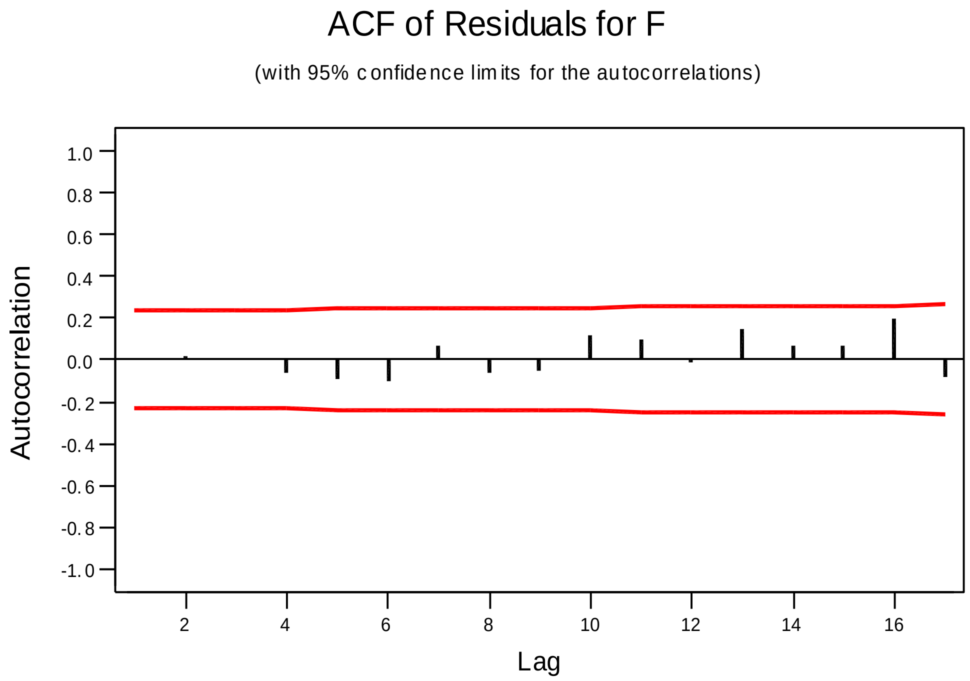
Final Estimates of Parameters

Type	Coef	StDev	T
AR 1	-0.3461	0.1259	-2.75
AR 2	0.1934	0.1259	1.54
Constant	59.047	1.298	45.50
Mean	51.227	1.126	

Model Checking

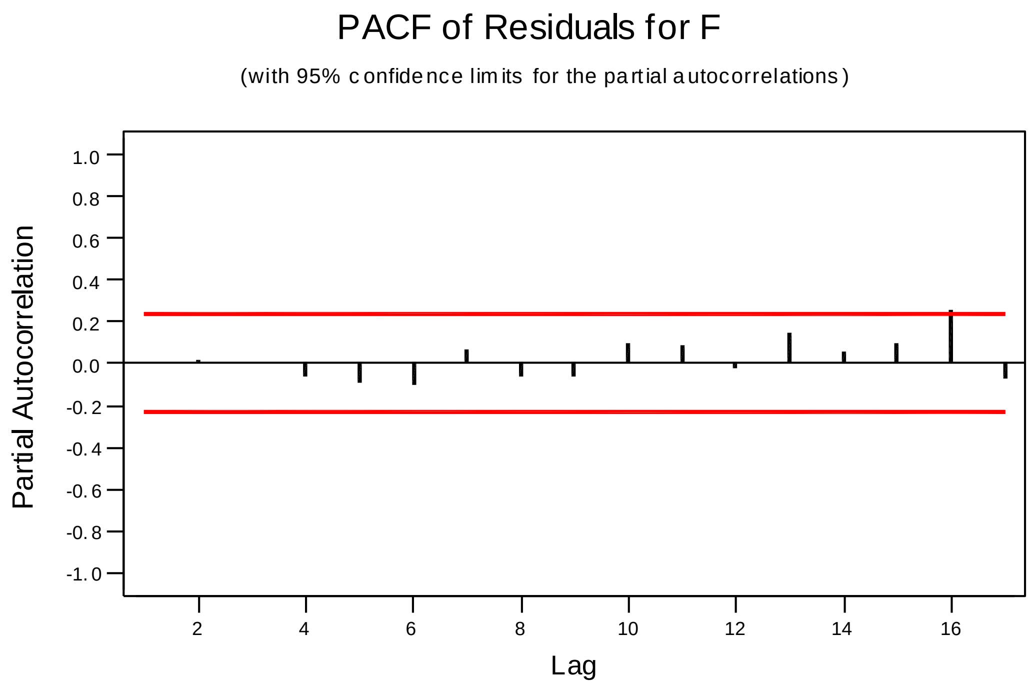
- Examine various aspects of the residuals to evaluate the adequacy of our chosen model.
- Use the acf and pacf to look for any remaining times series structure in the residuals which we have not removed.
- Check for normality and constant variance.

Correlogram of Residuals



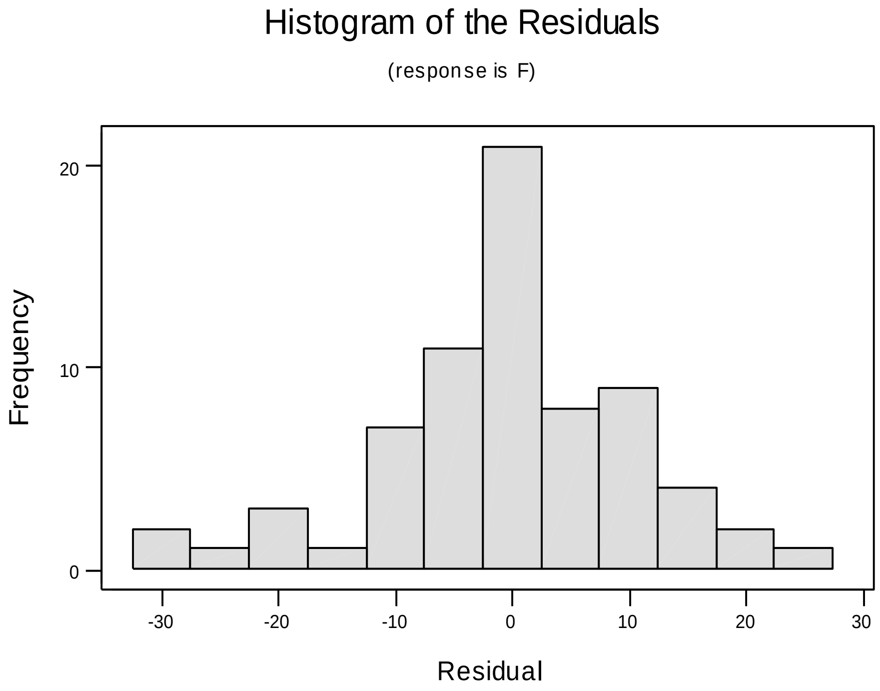
No obvious autocorrelation left after we have fitted our AR(2) model to the original series.

Partial Correlogram of Residuals



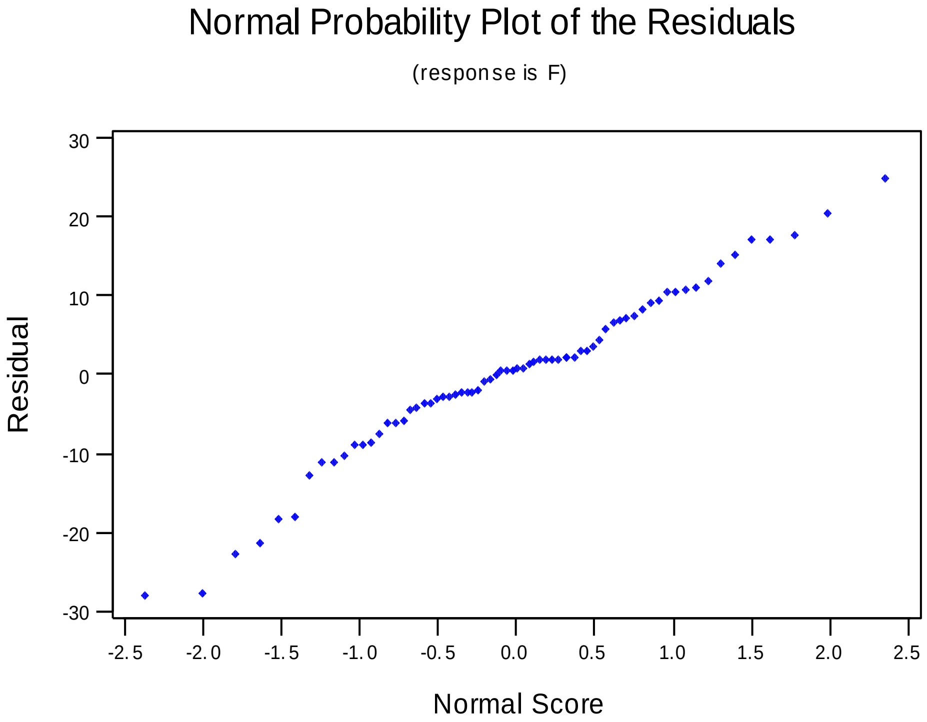
As we expect from the **PAF** and **ACF**, there is no indication of residual partial autocorrelation.

Histogram of Residuals



The residuals are symmetric about zero and die away fairly rapidly. A normal distribution with mean zero looks plausible.

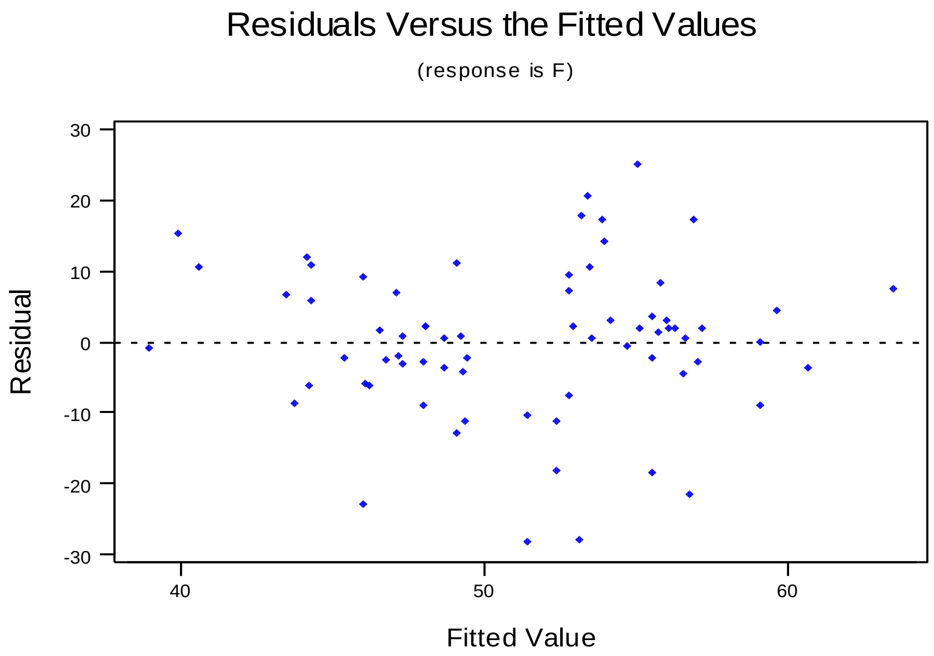
Normal Probability Plot



The graph should approximate a straight line if the residuals are normally distributed.

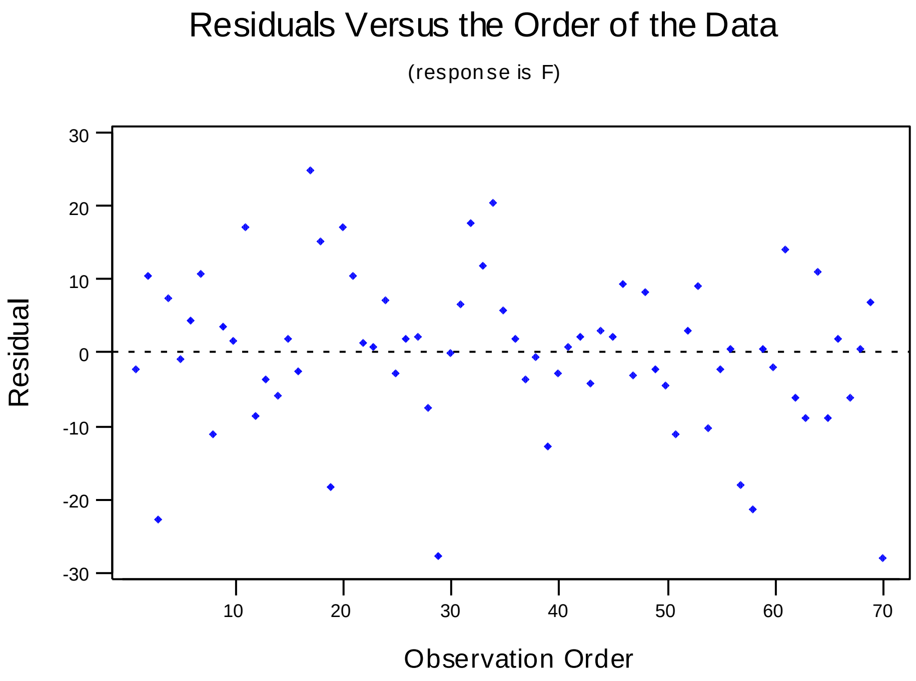
A normal distribution looks OK.

Residuals v. Fitted Values



No obvious problems such as the variance depending on the size of the observation.

Residuals in Time Order



No obvious failures to model dependence of residuals over time .

Which Model ?

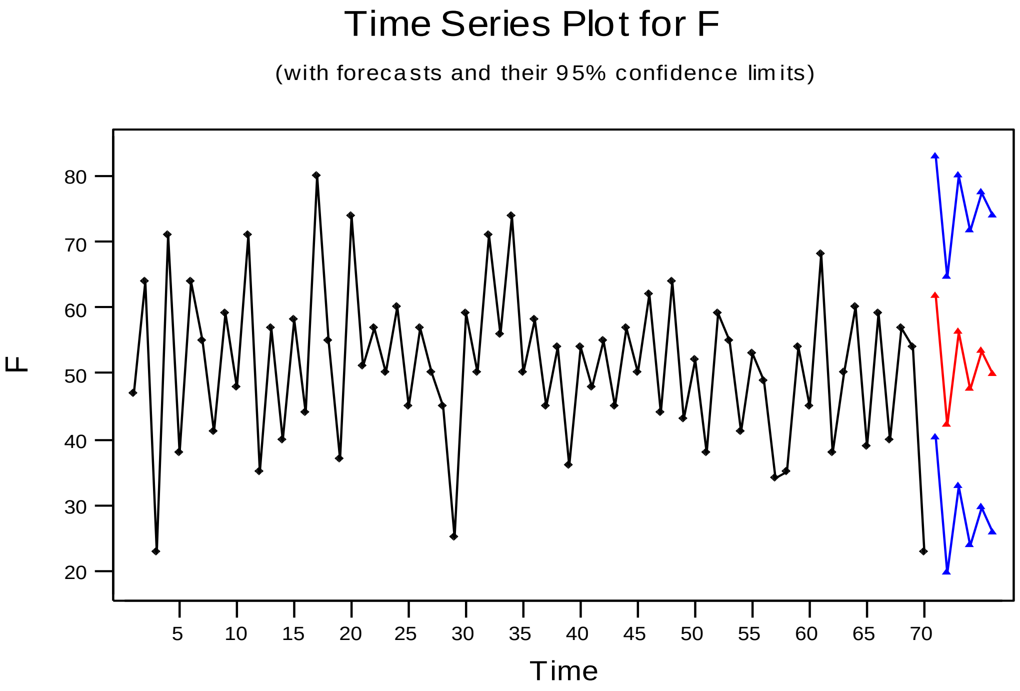
Time Series Yield

Mean	51.128571
Std	11.82361
N	70

Model Comparison

Model	DF	Variance	AIC	SBC	RSquare	-2LogLH
MA(2) No Constrain	67	10.895539	173.18474	179.93023	0.179	332.05415
AR(1) No Constrain	68	120.03093	339.14246	343.63945	0.166	333.30647
AR(2) No Constrain	67	117.76326	339.80735	346.55283	0.193	331.00592
ARMA(1, 1) No Constrain	67	118.6094	340.3085	347.05399	0.188	331.48621

Forecasting 6 periods ahead



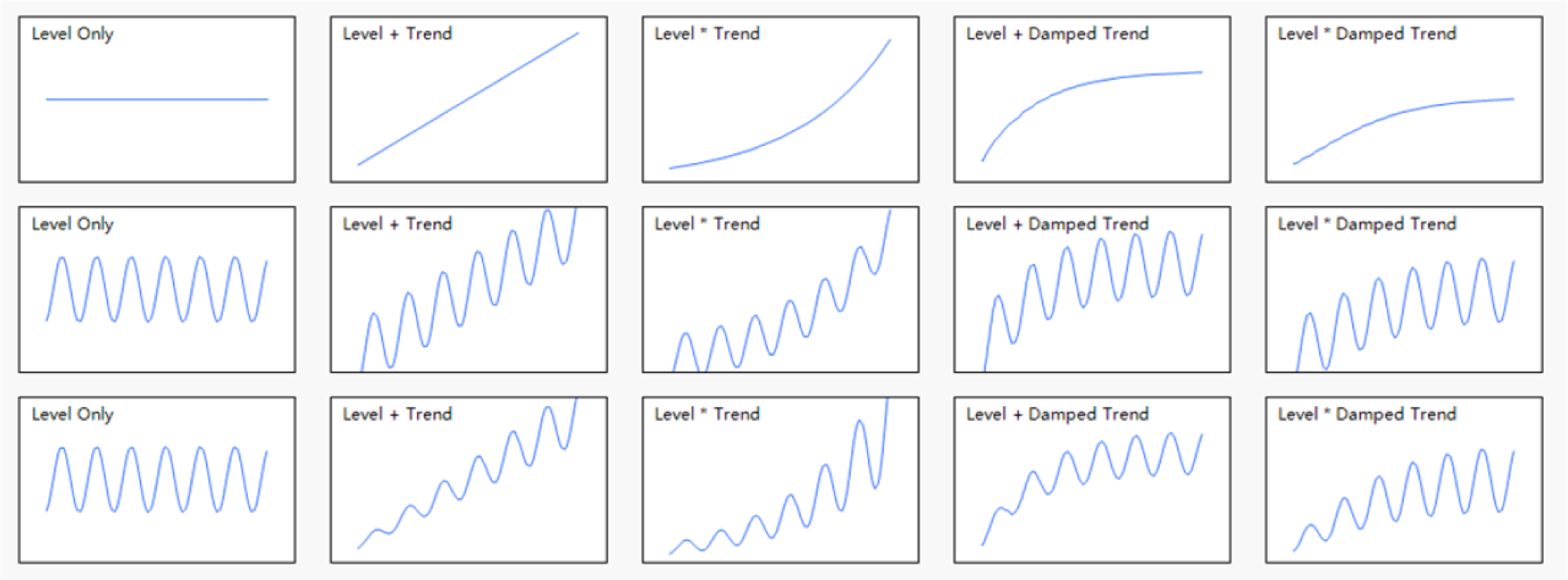
Opposite is a time series plot with forecasts up to 6 periods ahead added.

You can see that the interval estimates in blue are quite wide.

No Season

Additive Season

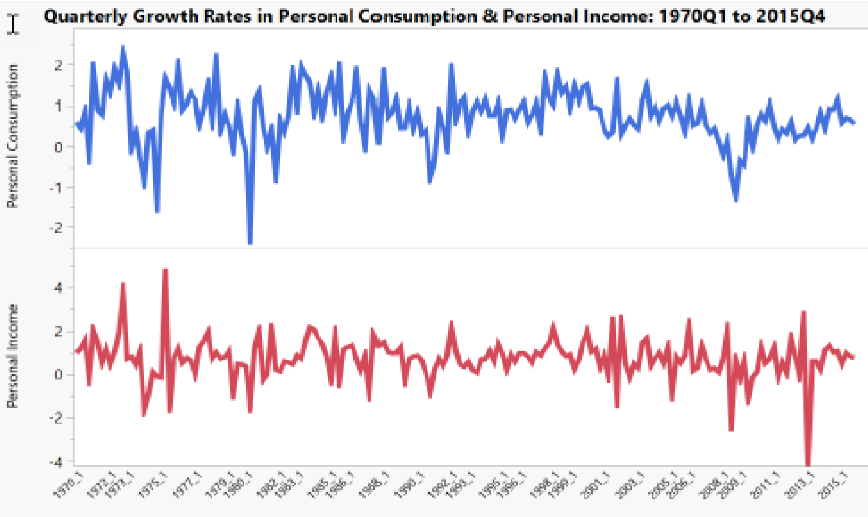
Multiplicative Season

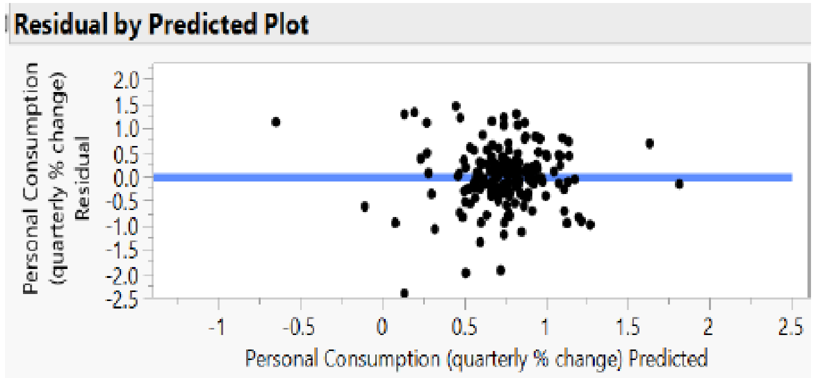


Transfer Functions

y_t: **Quarterly US Personal Consumption Growth Rate**

x_t: **Quarterly US Personal Income Growth Rate**





Durbin-Watson

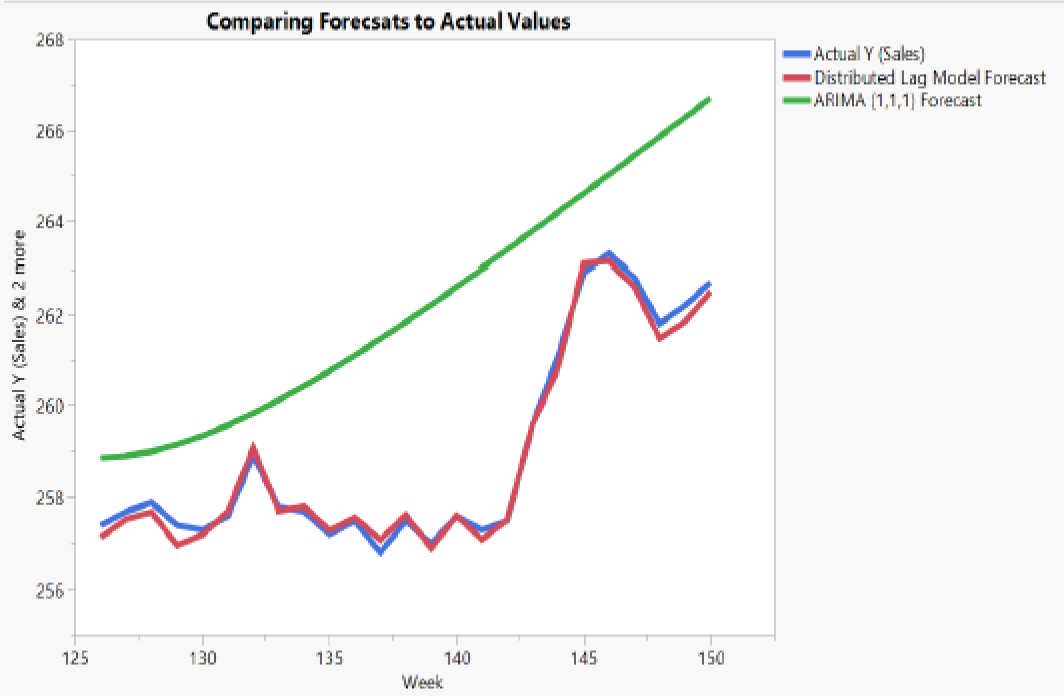
Durbin-Watson	Number of Obs.	AutoCorrelation	Prob<DW
1.692595	184	0.1531	0.0182*

- *"Can I just run a standard least squares regression model to predict y(t)?"* ($\widehat{y_t} = 0.55 + 0.28 * x_t$)

  --Key assumption that the regression errors are independent is likely violated => Misleading inferences.

Transfer Functions

--A pure ARIMA model uses the past values of an variable to predict its future values. If there is a good predictor bring it in.



Blue line: Actual Sales
Green Line: Forecast from Pure ARIMA
Red Line: Forecast from Transfer Function Model

Transfer Functions

$$y_t = \mu + x_t\beta + e_t \qquad \text{(OLS)}$$

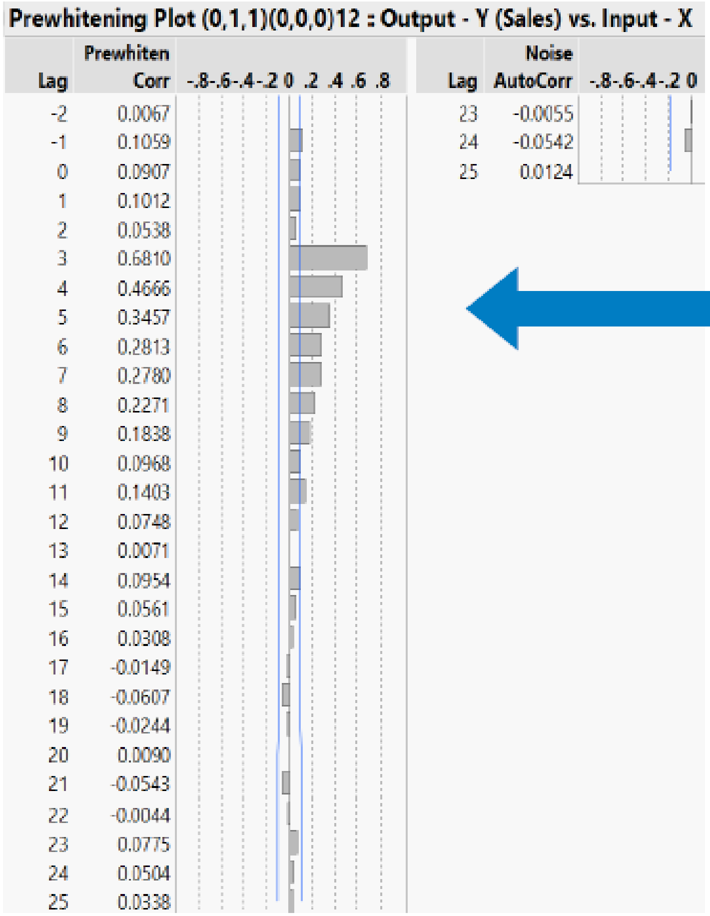
$$y_t = \mu + \frac{\theta(B)}{\varphi(B)} e_t \qquad \text{(Pure ARIMA)}$$

Transfer Function ARIMA Error Term

$$y_t = \mu + \frac{\omega(B)}{\delta(B)} x_{t-l} + \frac{\theta(B)}{\varphi(B)} e_t$$

Prewhitening

Identifying *when* the input effect takes place, *how long* it lasts and in *what* shape it decays.



Prewhitening Plot (0,1,1)(0,0,0)12 : Output - Y (Sales) vs. Input - X

Lag	Prewhiten Corr
-2	0.0067
-1	0.1059
0	0.0907
1	0.1012
2	0.0538
3	0.6810
4	0.4666
5	0.3457
6	0.2813
7	0.2780
8	0.2271
9	0.1838
10	0.0968
11	0.1403
12	0.0748
13	0.0071
14	0.0954
15	0.0561
16	0.0308
17	-0.0149
18	-0.0607
19	-0.0244
20	0.0090
21	-0.0543
22	-0.0044
23	0.0775
24	0.0504
25	0.0338

Lag	Noise AutoCorr
23	-0.0055
24	-0.0542
25	0.0124

Ex2: Forecasting Sales using Lagged Predictors

Input series x_t is delayed by 3 lags and then exponentially decreasing:

$$\omega_0(x_{t-3} + \delta x_{t-4} + \delta^2 x_{t-5} + \delta^3 x_{t-6} + \delta^4 x_{t-7} + \cdots)$$

try the transfer function $\frac{\boldsymbol{\omega_0}}{\boldsymbol{1-\delta B}}\boldsymbol{x_{t-3}}$ to approximate the input-output relationships

Forecasting: Principles and Practice (2nd ed)

Rob J Hyndman and George Athanasopoulos

Monash University, Australia

https://otexts.com/fpp2/

QUALITY ENGINEERING
https://doi.org/10.1080/08982112.2022.2153696

Taylor & Francis
Taylor & Francis Group

Check for updates

Functional data analysis and nonlinear regression models: an information quality perspective

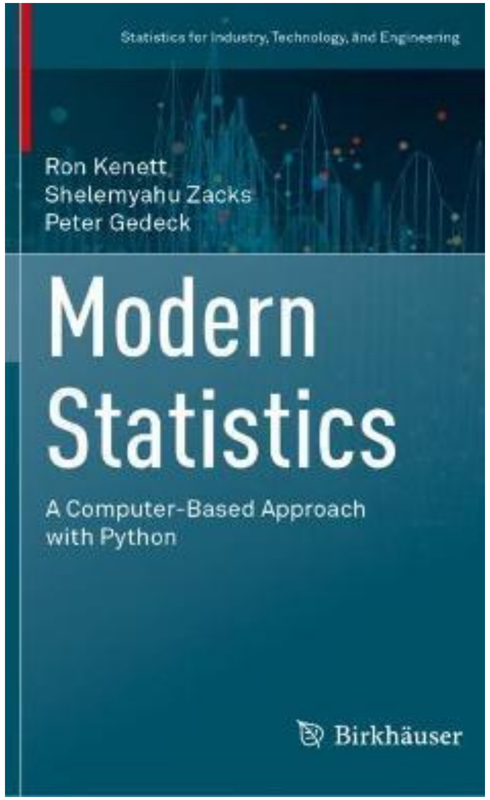
Ron S. Kenett[a] and Chris Gotwalt[b]

[a]The KPA Group and the Samuel Neaman Institute, Technion, Israel; [b]JMP Statistical Discovery, LLC, Research Triangle, North Carolina, USA

ABSTRACT
Data from measurements over time can be analyzed in different ways. In this article, we compare functional data analysis and nonlinear regression models using, among others, eight information quality dimensions. We present two case studies. The first case study introduces functional data analysis and nonlinear regression models in analyzing dissolution profiles of drug tablets where profiles of tablets under test are compared to reference tablets. A second case study involves statistically designed mixture experiments used in optimization tablet formulation. Python and JMP features are used to demonstrate the methods used in the two case studies.

KEYWORDS
dissolution curve; functional data analysis; information quality; mixture experiments; nonlinear regression

Chapter 8
Modern Analytic Methods: Part II

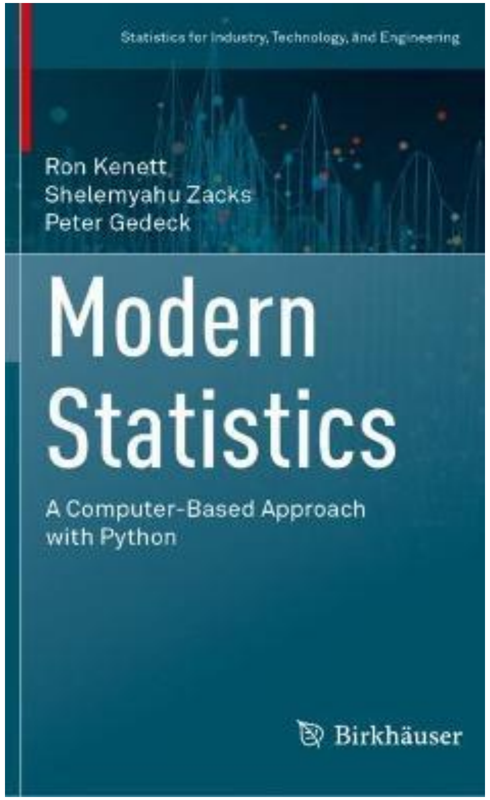


Preview Chapter 8 includes the tip of the iceberg examples with what we thought were interesting insights, not always available in standard texts. The chapter covers functional data analysis, text analytics, reinforcement learning, Bayesian networks, and causality models.

Chapter 8
Modern Analytic Methods: Part II

8.1 Functional Data Analysis

When you collect data from tests or measurements over time or other dimensions, we might want to focus on the functional structure of the data. Examples can be chromatograms from high-performance liquid chromatography (HPLC) systems, dissolution profiles of drug tablets over time, distribution of particle sizes, or measurement of sensors. Functional data is different using individual measurements recorded at different sets of time points. It views functional observations as continuously defined so that an observation is the entire function. With functional data consisting of a set of curves representing repeated measurements, we characterize the main features of the data, for example, with a functional version of principal component analysis (FPCA). The regular version of principal component analysis (PCA) is presented in detail in Chap. 4 (Industrial Statistics book) on Multivariate Statistical Process Control. With this background, let us see an example of functional data analysis (FDA).



Chapter 8

Modern Statistics: A Computer Based Approach with Python
by Ron Kenett, Shelemyahu Zacks, Peter Gedeck

Publisher: Springer International Publishing; 1st edition (September 15, 2022)
ISBN-13: 978-3031075650

(c) 2022 Ron Kenett, Shelemyahu Zacks, Peter Gedeck

The code needs to be executed in sequence.

In [1]:
```
import os
os.environ['OUTDATED_IGNORE'] = '1'
import warnings
from outdated import OutdatedPackageWarning
warnings.filterwarnings('ignore', category=FutureWarning)
warnings.filterwarnings('ignore', category=OutdatedPackageWarning)
```

Modern analytic methods: Part II

In [2]:
```
import networkx as nx

import statsmodels.api as sm
from statsmodels.tsa.stattools import grangercausalitytests
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import mistat
```

Modern Analytic Methods: Part II

Ron Kenett, Shelemyahu Zacks, Peter Gedeck

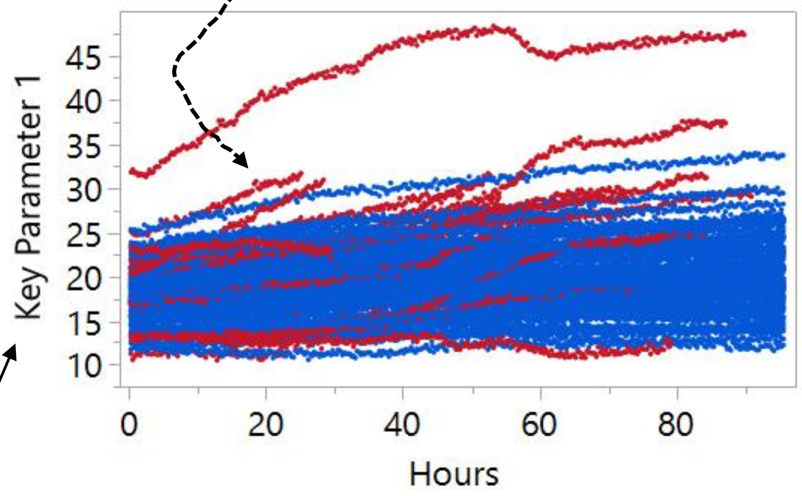
Pages 395-419

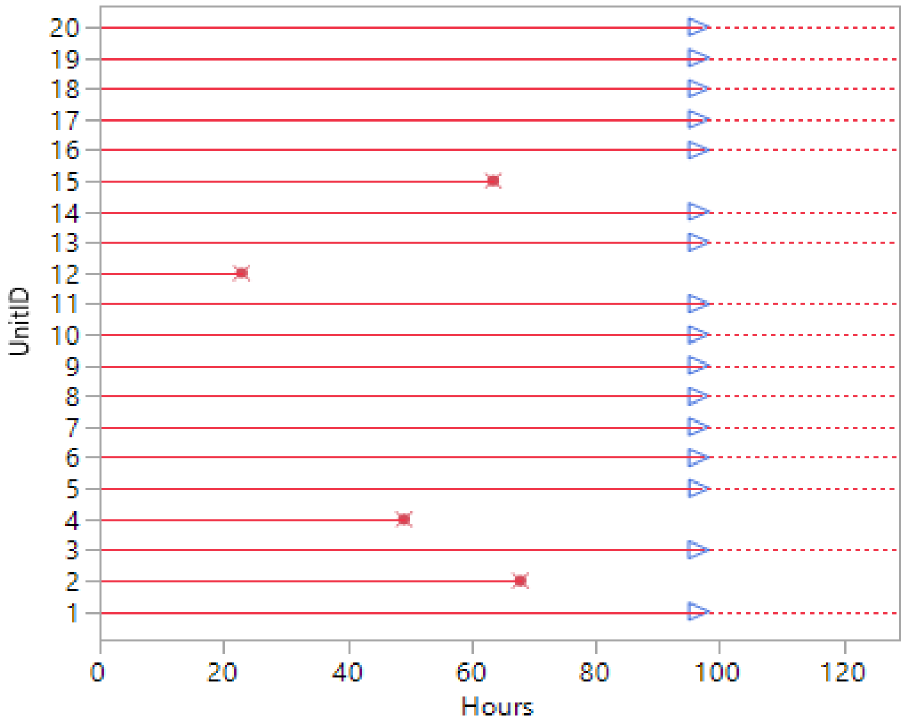
A Case Study

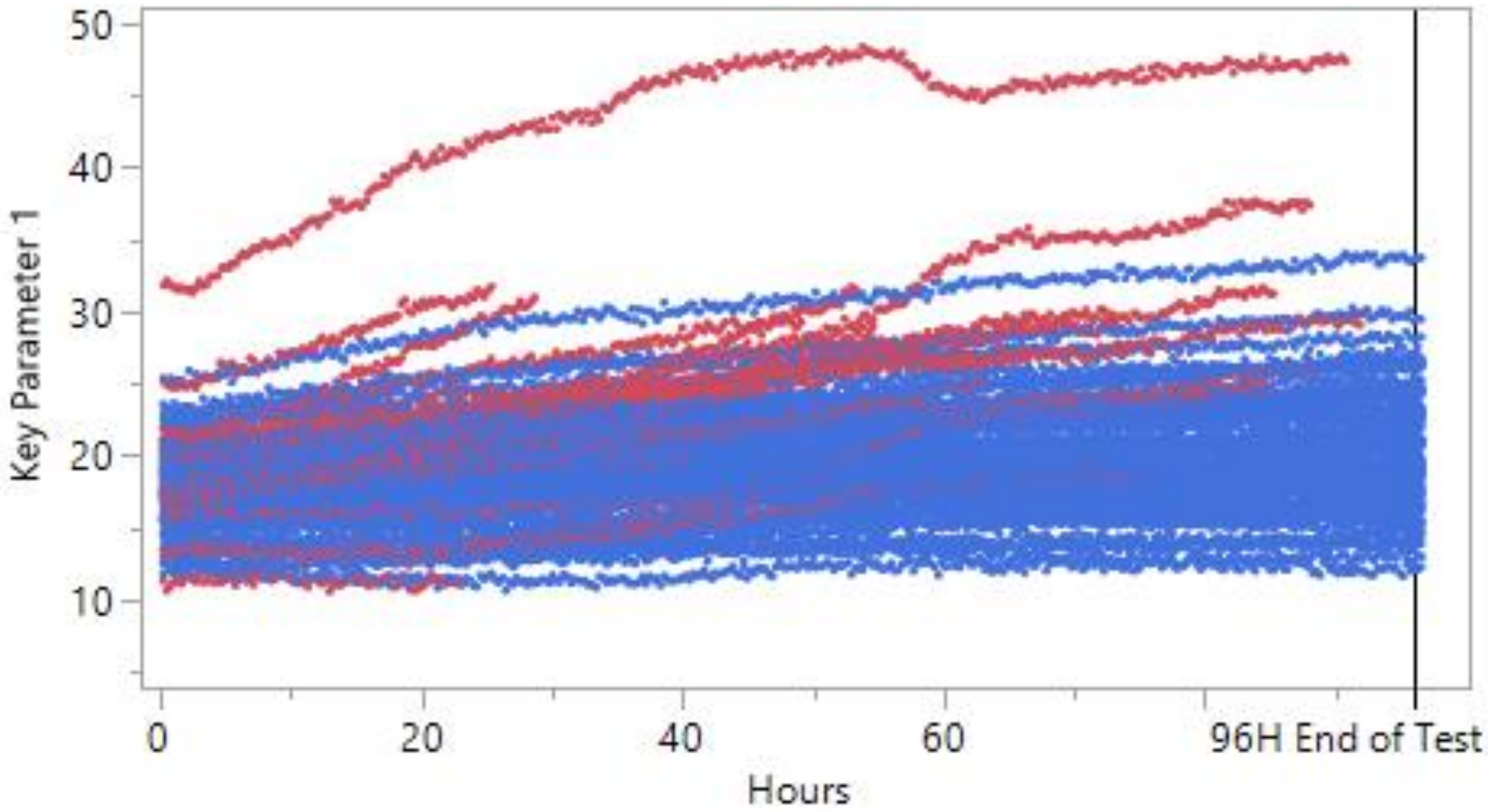
- A manufacturer of electro-mechanical devices
- 22% of units fail early
- The company screens units via burn-in under accelerated conditions
- 23 measurements are collected in real time (2 key parameters)
- Can we use these measurements to predict which units will fail in less time than the current protocol?

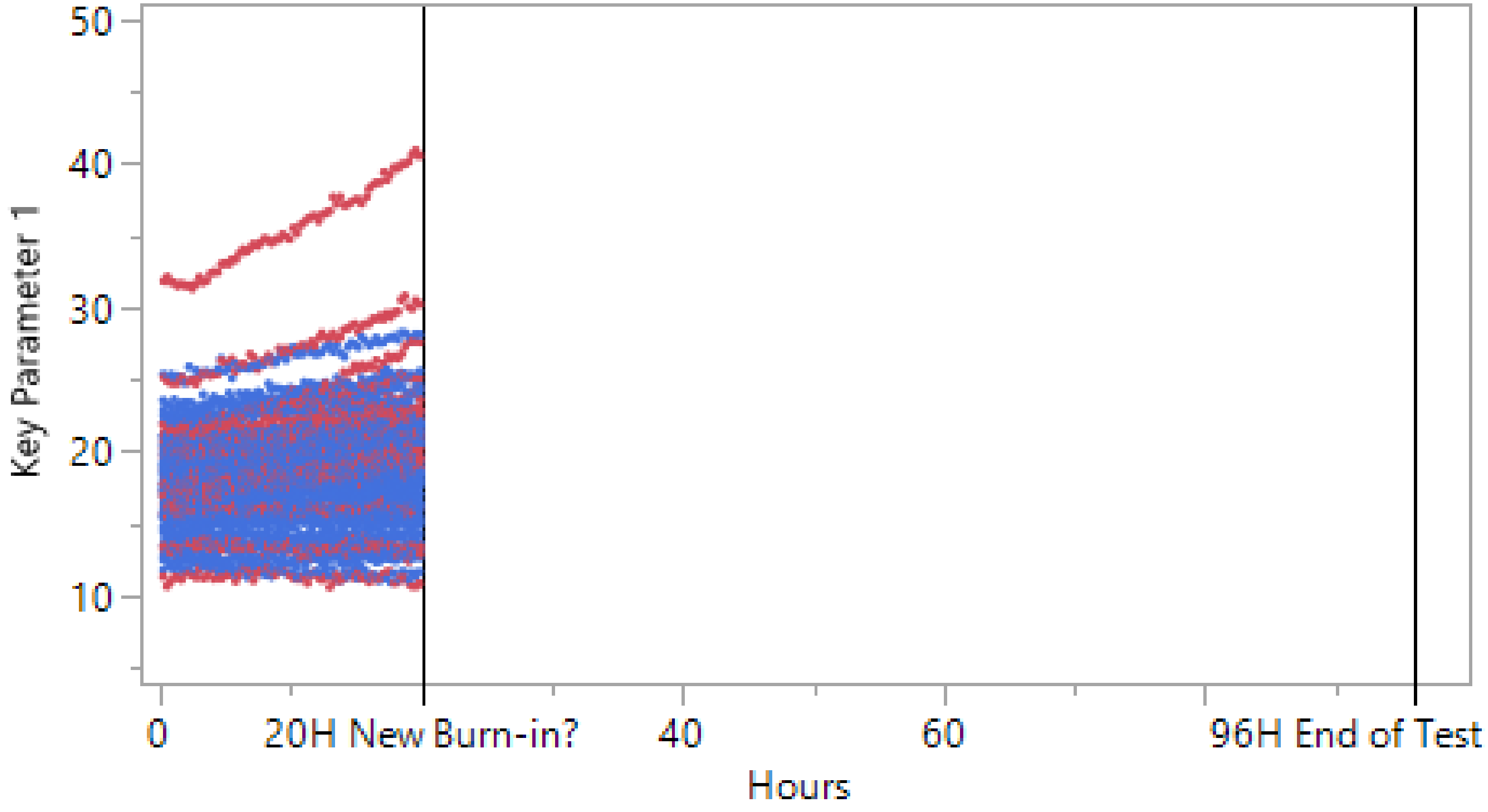
Key parameter 1

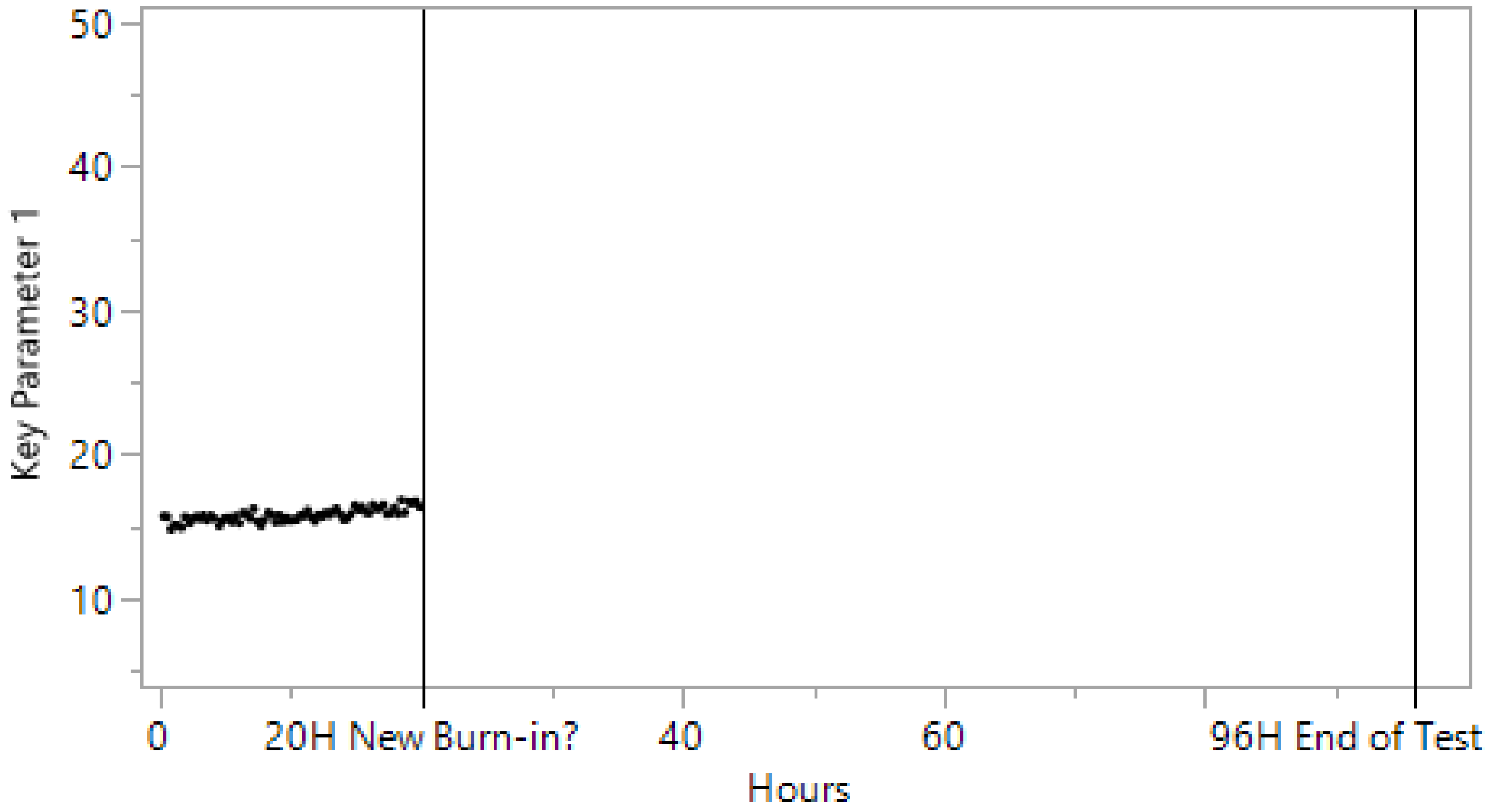
Data consists of 92 units. If a unit didn't fail, there would be around 25,000 measurements. Failed series can be much shorter



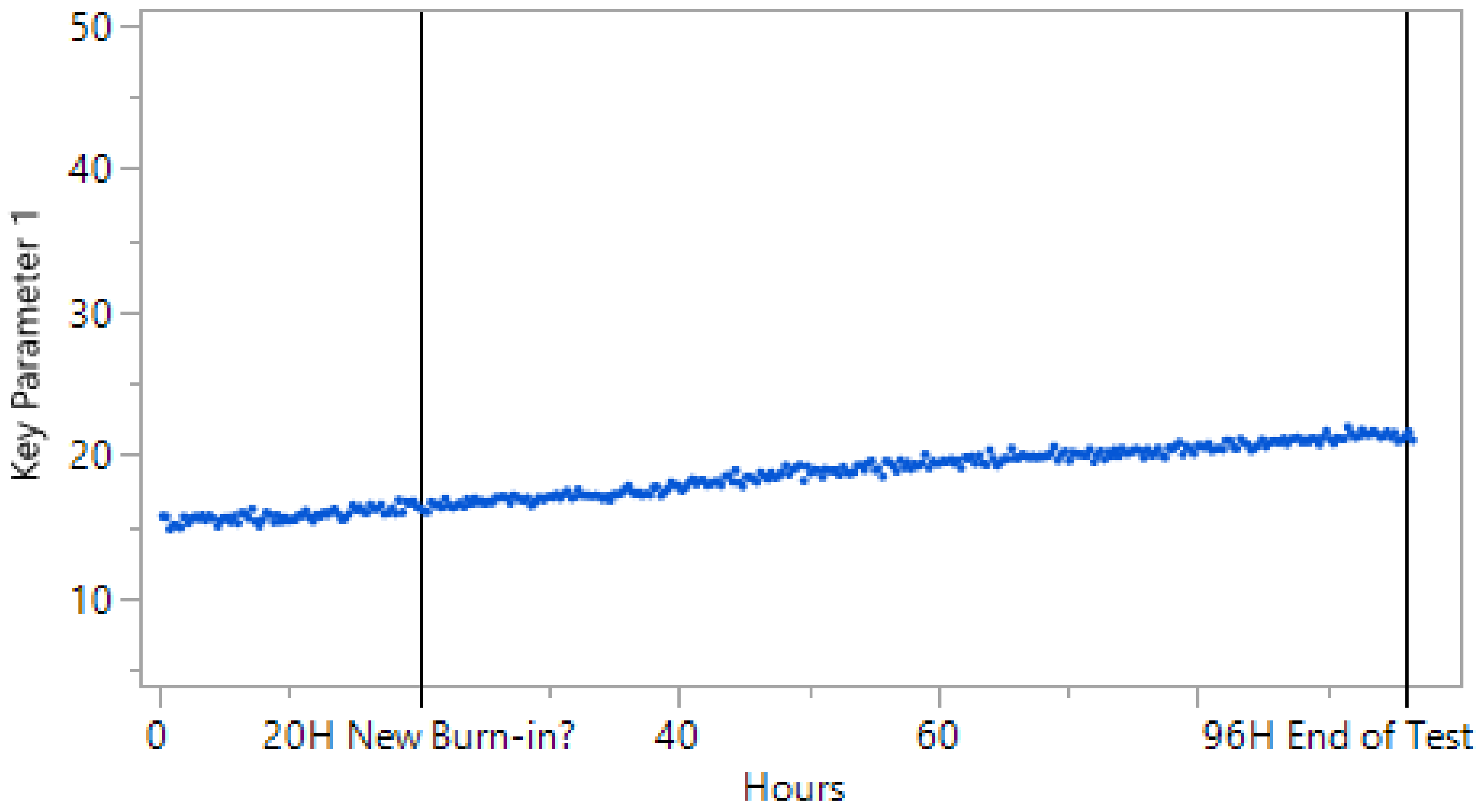


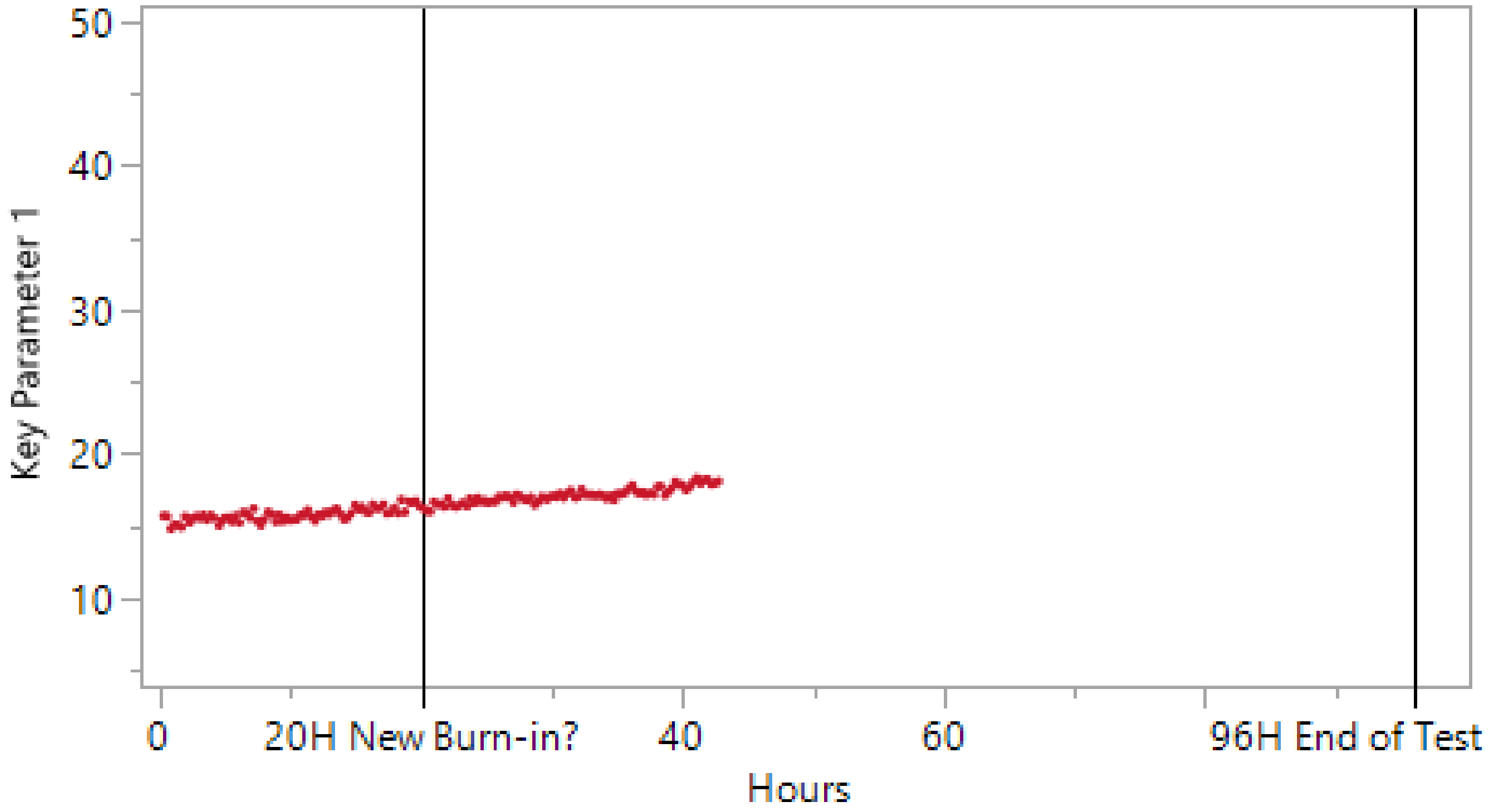

50
40
30
20
10
Key Parameter 1
0
20
40
60
96H End of Test
Hours

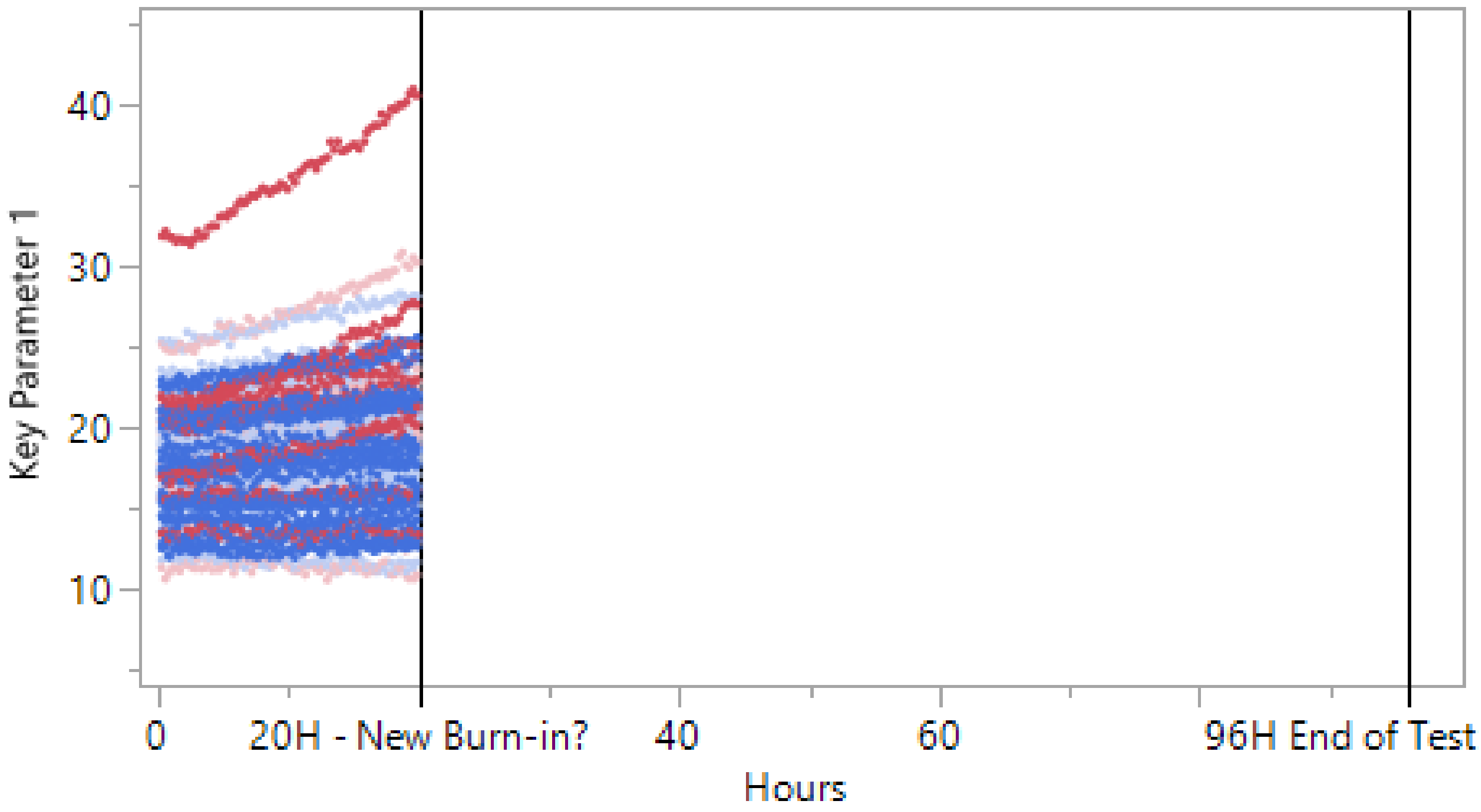

50
40
30
20
10
Key Parameter 1
0
20H New Burn-in?
40
60
96H End of Test
Hours



50
40
30
20
10
Key Parameter 1
0
20H New Burn-in?
40
60
96H End of Test
Hours


Key Parameter 1
50
40
30
20
10
0
20H New Burn-in?
40
60
96H End of Test
Hours


50
40
30
20
10
Key Parameter 1
0
20H New Burn-in?
40
60
96H End of Test
Hours



Key Parameter 1
40
30
20
10
0
20H - New Burn-in?
40
60
96H End of Test
Hours

Functional Data Analysis models

- B-Splines
  - Piecewise polynomials with an underlying mean model and variance components on the spline coefficients.
  - These often work the best.
  - Try these first and customize the #Knots as needed.

- P-Splines
  - P is for "Penalized". These tend to have lots of knots and are often slower to fit but similar in properties to B Splines.
  - Worth trying if B Spline do not fit well.

- Fourier Basis
  - Uses a sine/cosine expansion as the basis.
  - Good for periodic data (like vibration/sound signals).
  - Usually the spline models work better on other types of functional data.

P-Spline Functional Model Fit

$$y_i(t_{i,j}) = \sum_k \beta_k b(t_{i,j}) + \sum_k \gamma_{i,k} b(t_{i,j}) + \varepsilon_{i,j,k}$$

- $b(t_{i,j})$: basis functions, these form X, and Z.
- β_k: mean function coefficients, fixed effects
- $\gamma_{i,k} \sim N(0, \sigma_k^2)$: random effects
- $\varepsilon_{i,j,k} \sim N(0, \sigma_\varepsilon^2)$: errors

Scoring New P-Spline Functions

Training

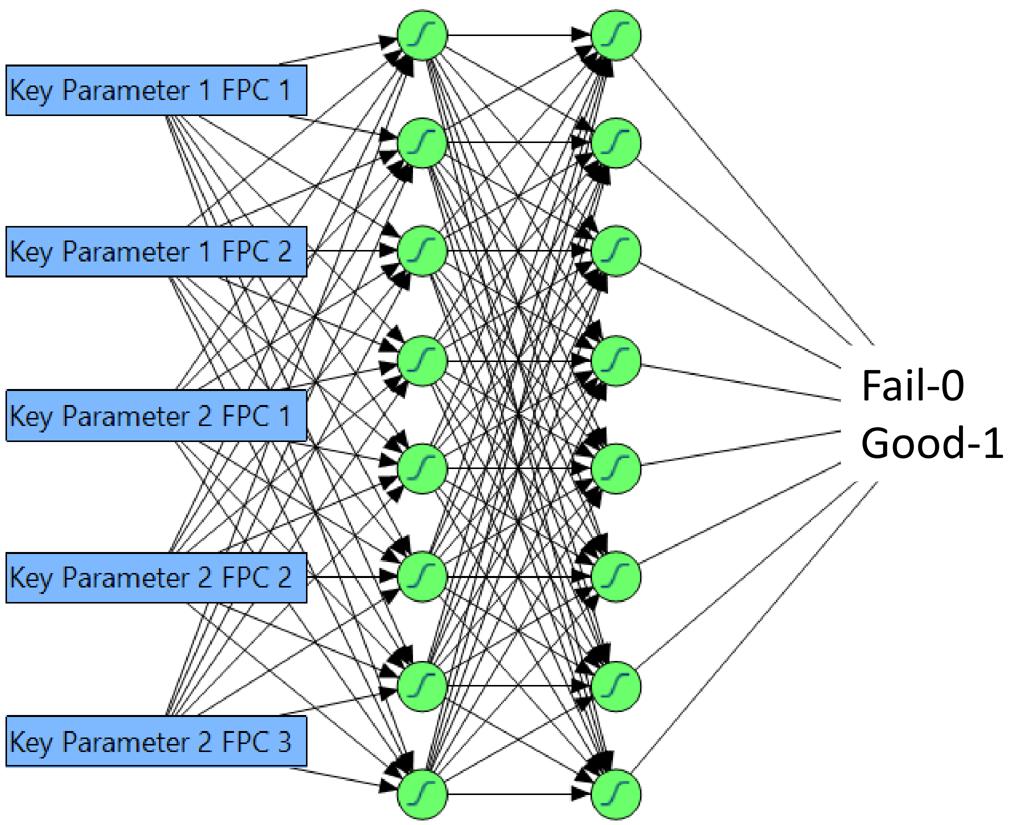
$$\hat{\gamma}_{Val} = \left(Z_{Val}^{T} Z_{Val} + G(\hat{\sigma}_{Tr}^{2})\right)^{-1} Z_{Val}^{T}(y_{Val} - X_{Val}\hat{\beta}_{Tr})$$

- $\hat{\beta}_{Tr}, \hat{\sigma}_{Tr}^{2}$ estimated from training data
- Use BLUP formula to score $\hat{\gamma}_{Val}$ or any new units

A best linear unbiased prediction (BLUP) estimate of realized values of a random variable are linear in the sense that they are linear functions of the data. They are unbiased in the sense that the average value of the estimate is equal to the average value of the quantity being estimated and best in the sense that they have minimum sum of squared error within the class of linear unbiased estimators. Estimators of random effects are called **predictors**, to distinguish them from **estimators** of fixed effects called estimators. BLUP estimates are solutions to mixed model equations and are usually different from generalized linear regression estimates used for fixed effects.

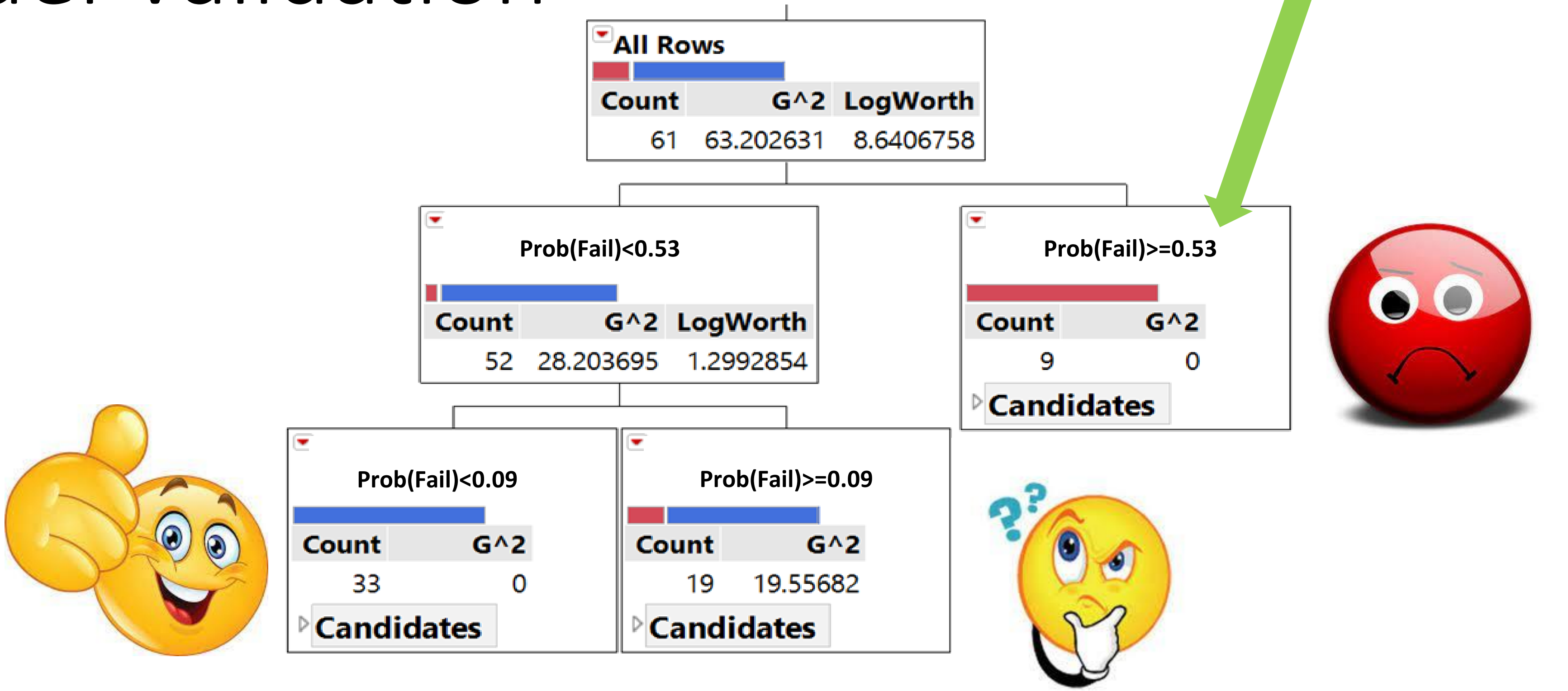
	UnitID	Validation	Failed	Key Parameter 1 FPC 1	Key Parameter 1 FPC 2
1	1	Training	0	-0.137461119	0.1518411781
2	2	Training	1	1.2865813341	0.2245665174
3	3	Training	0	1.7696429741	-0.097194514
4	4	Validation	1	2.2470498512	0.3152274003
5	5	Training	0	1.0745237905	0.0063042898
6	6	Training	0	-2.743836032	-0.00195229
7	7	Training	0	0.4589908625	0.1970255259
8	8	Training	0	-0.81562604	0.0974489975
9	9	Training	0	-0.547114624	-0.129940706
10	10	Training	0	-0.633798753	0.0691610259
11	11	Validation	0	1.2816670105	0.059283723
12	12	Training	1	-3.588716661	0.0642051917

- Data summarized into one row per unit
- Model failure probability



A two-layer Neural Network predicting failure probability

Model Validation

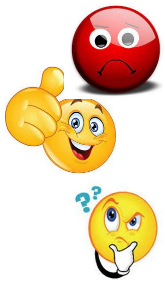


A Regression Tree was used to predict “Censor” at 20 hours using the Neural probability as input, ***using only the Training subset of the data***.

Model Validation

Fail

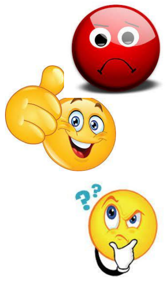
Predicted Category: Count / Row %	0	1	Total
BAD	9 100.00	0 0.00	9
GOOD	0 0.00	33 100.00	33
MAYBE GOOD	4 21.05	15 78.95	19
Total	13	48	61



Training

Fail

Predicted Category: Count / Row %	0	1	Total
BAD	6 100.00	0 0.00	6
GOOD	1 5.00	19 95.00	20
MAYBE GOOD	0 0.00	5 100.00	5
Total	7	24	31



Validation

Yes/No/Maybe decision rule was developed from the neural network prediction

Dissolution Curves of 12 tablets. Test and Reference

	Time	Label R	Data R	Label T	Data T
1	5	T1R	72.7	T1	46.6
2	5	T2R	78.8	T2	10.5
3	5	T3R	32.3	T3	10
4	5	T4R	38.8	T4	42.9
5	5	T5R	18.9	T5	61
6	5	T6R	52.1	T6	36.3
7	5	T7R	14.3	T7	6.4
8	5	T8R	67.8	T8	4.4
9	5	T9R	7.5	T9	5.4
10	5	T10R	8.5	T10	3.6
11	5	T11R	26.5	T11	6.4
12	5	T12R	10.2	T12	35
13	10	T1R	89.1	T1	74
14	10	T2R	94.4	T2	38.1
15	10	T3R	60.5	T3	30.9
16	10	T4R	63.7	T4	88.1
17	10	T5R	31.3	T5	84.3
18	10	T6R	79.6	T6	71.9
19	10	T7R	44.3	T7	39.4

Test

Level	Count
5	12
10	12
15	12
20	12
30	12
45	12
Total	72

Reference

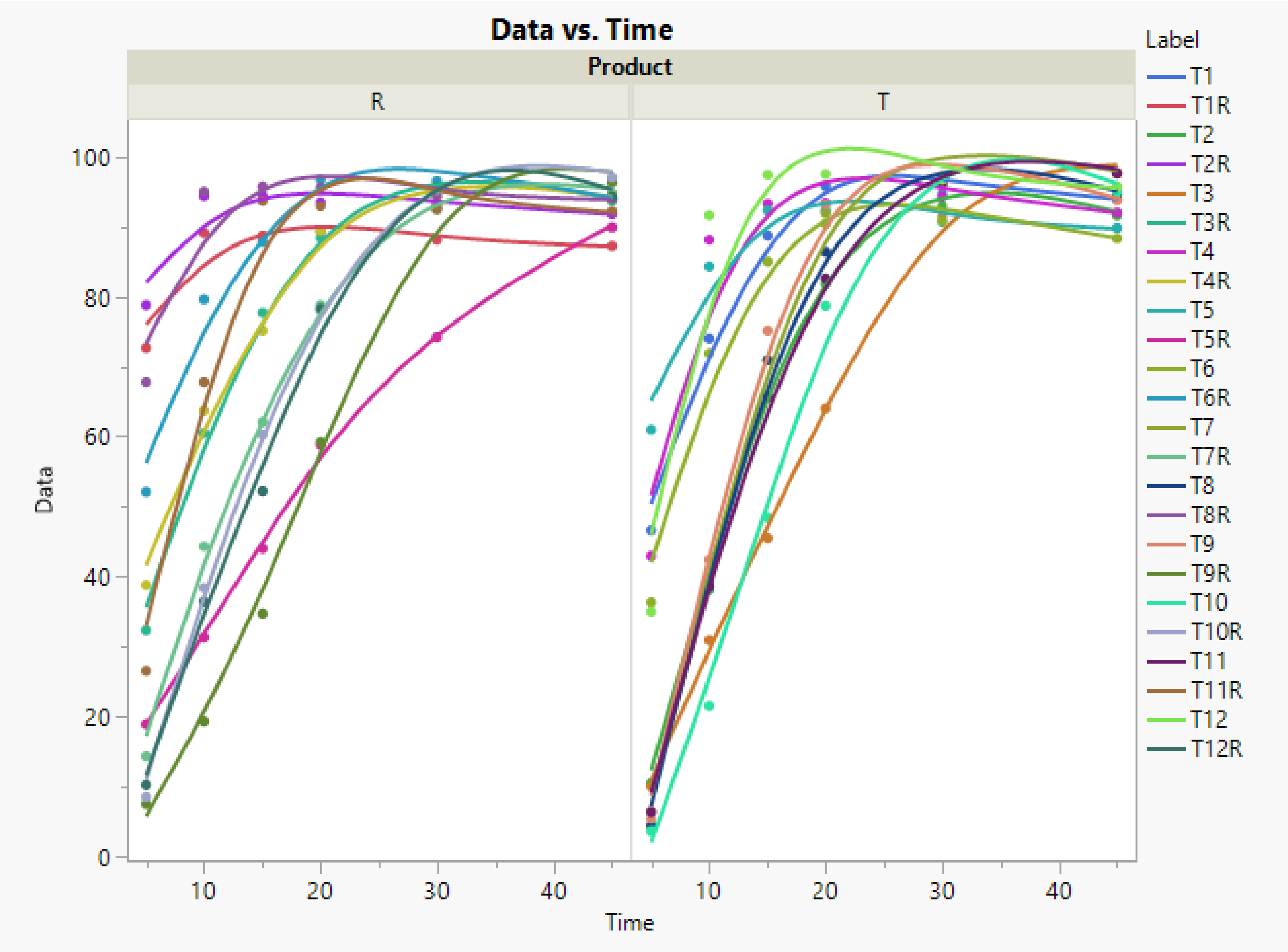
Level	Count
5	12
10	12
15	12
20	12
30	12
45	12
Total	72

FDA

NLR

https://www.youtube.com/watch?v=g4gxLG2IQeo

Dissolution Curves of 12 tablets. Test and Reference



Dissolution

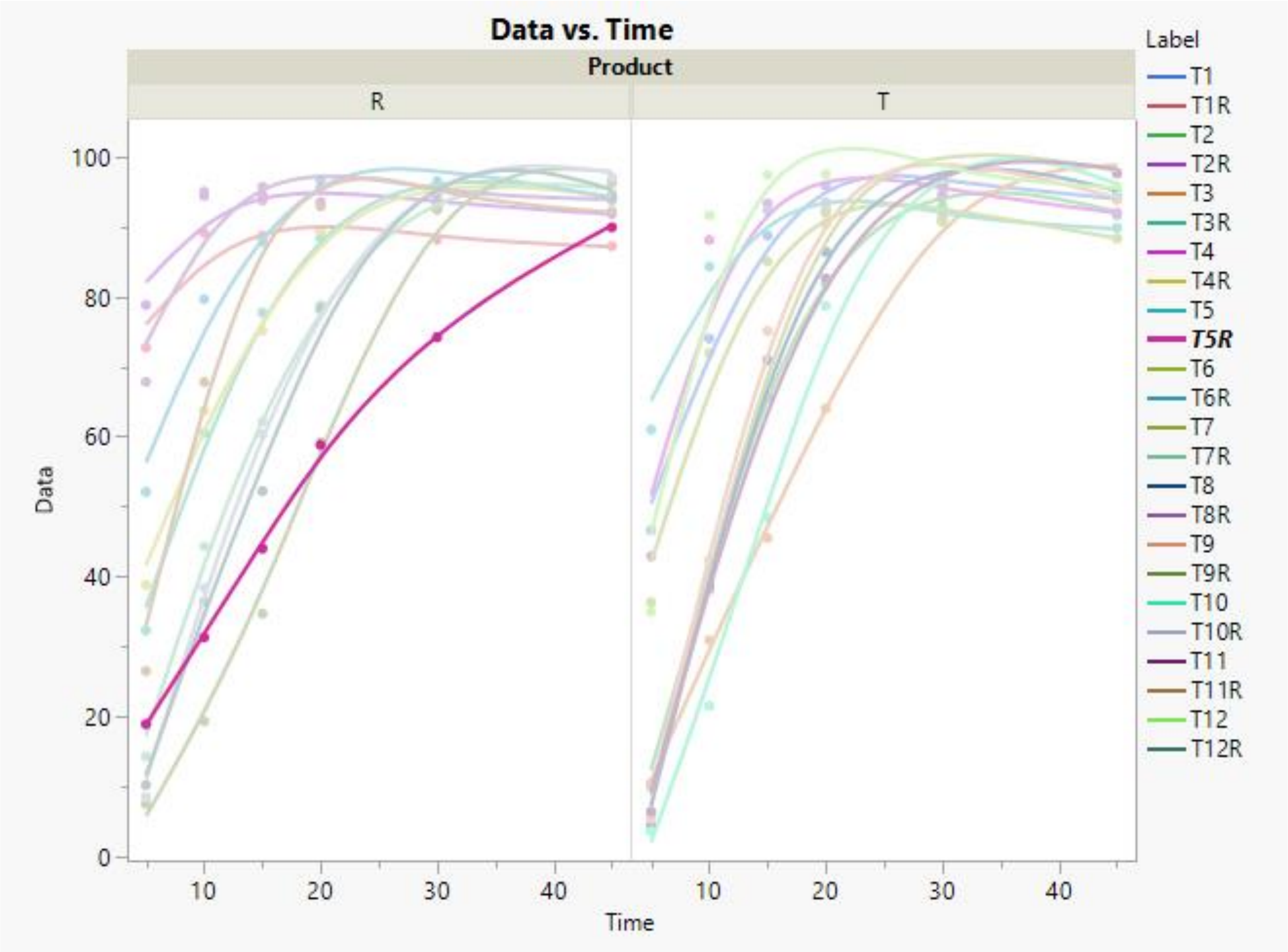
Curves of

12 tablets.

Test

T5R

Is different



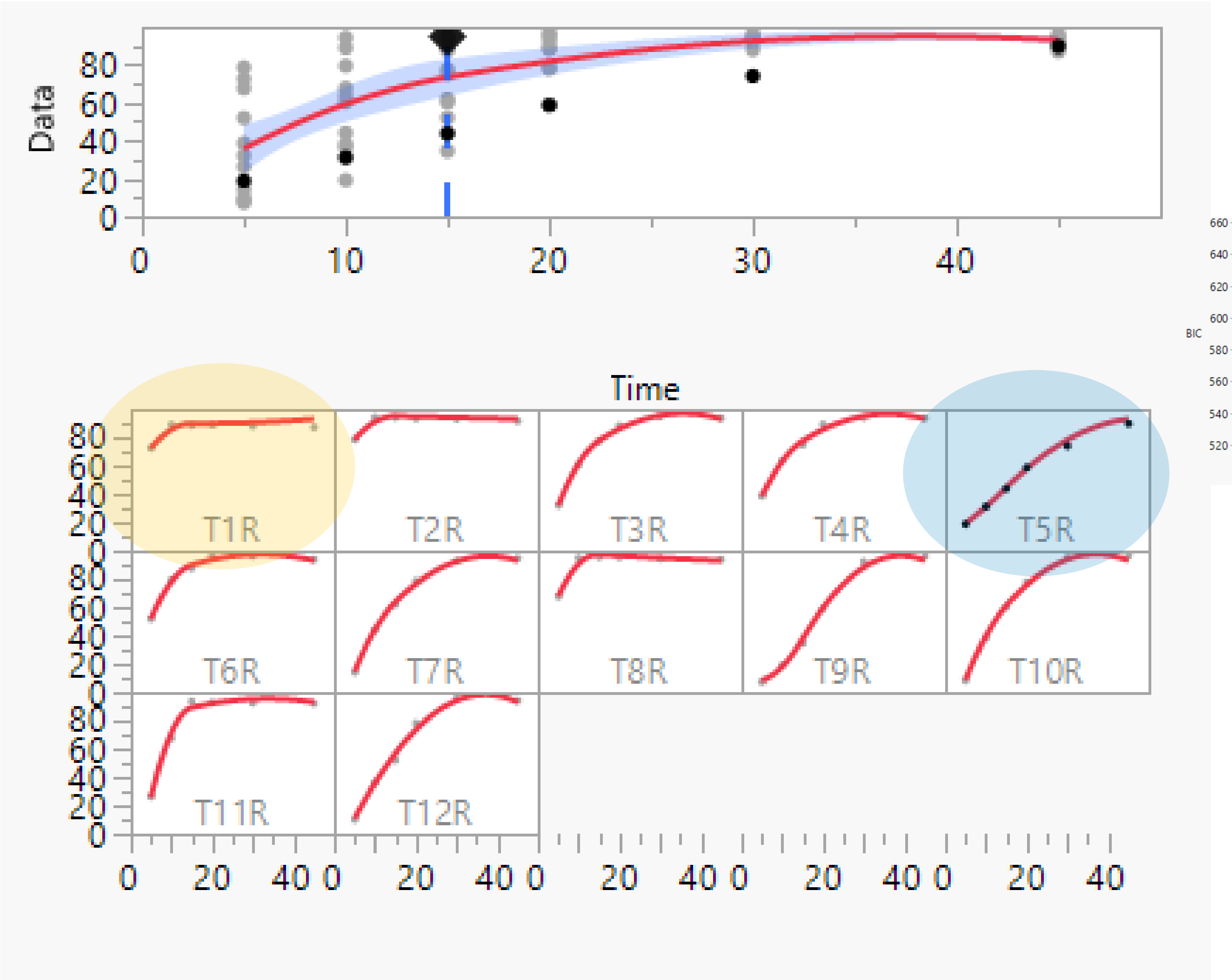
Dissolution

Curves of

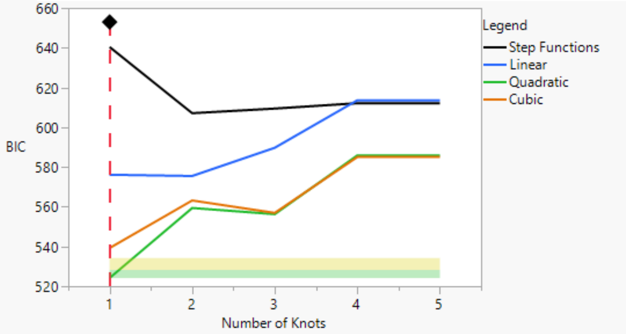
12 tablets.

Test T5R

Is different



FDA



Quadratic B-spline with 1 knot

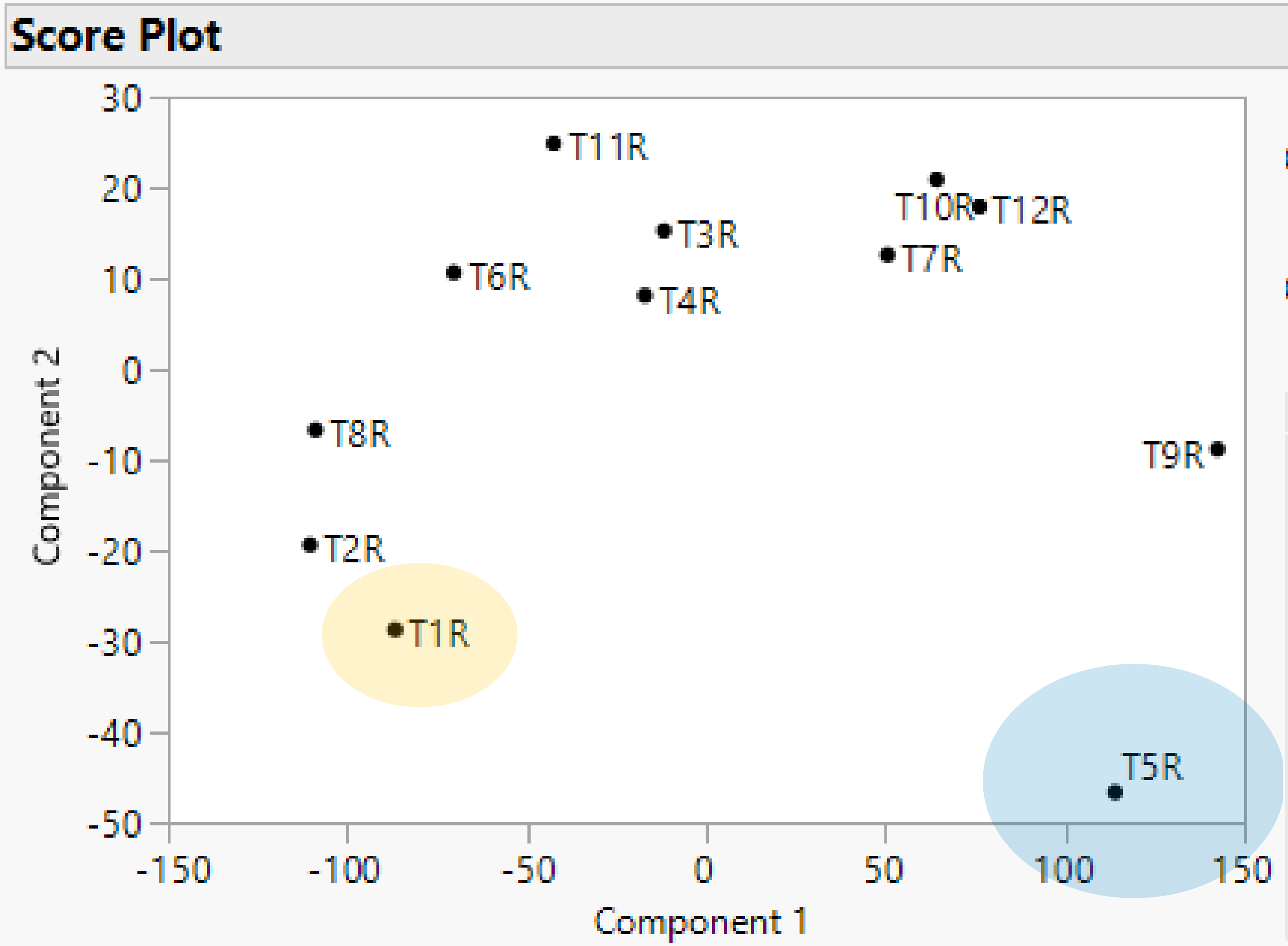
Dissolution

Curves of

12 tablets.

Test T5R

Is different



Label variables

Component 1

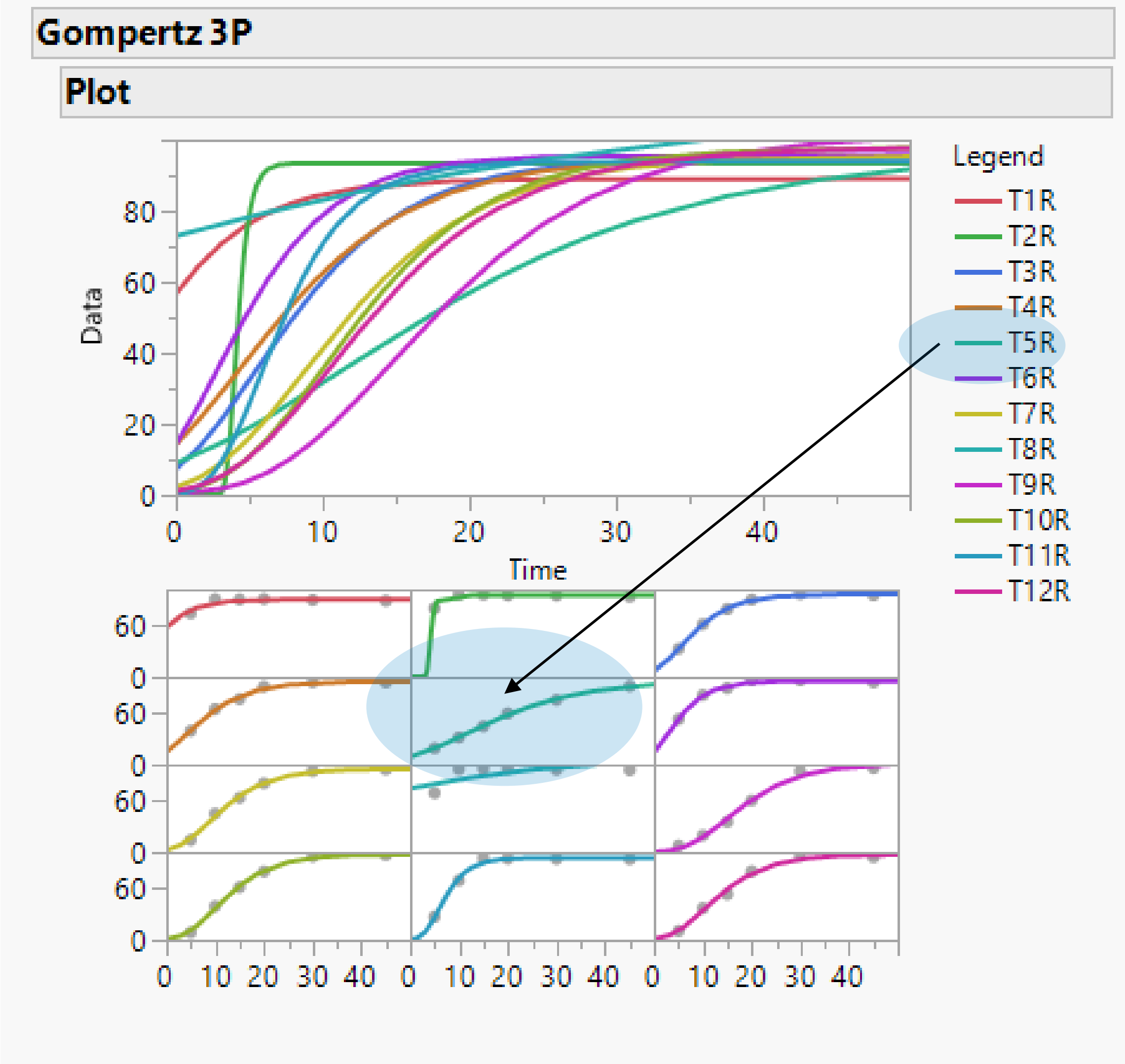
FPC 1

Component 2

FPC 2

Label	FPC 1	FPC 2	FPC 3
T1R	-86.63056	-28.70704	0.996566
T2R	-110.4346	-19.38626	7.7488408
T3R	-11.7261	15.242747	0.0425637
T4R	-17.00577	8.1173121	3.7777815
T5R	114.01378	-46.64264	-12.92808
T6R	-70.34096	10.648925	3.9284304
T7R	50.636525	12.627046	-3.211241
T8R	-108.9026	-6.710816	0.0497409
T9R	142.48324	-8.840222	11.976841
T10R	64.240312	20.878974	-0.66174
T11R	-42.52142	24.890182	-18.14564
T12R	76.188141	17.881793	6.4259358

FDA

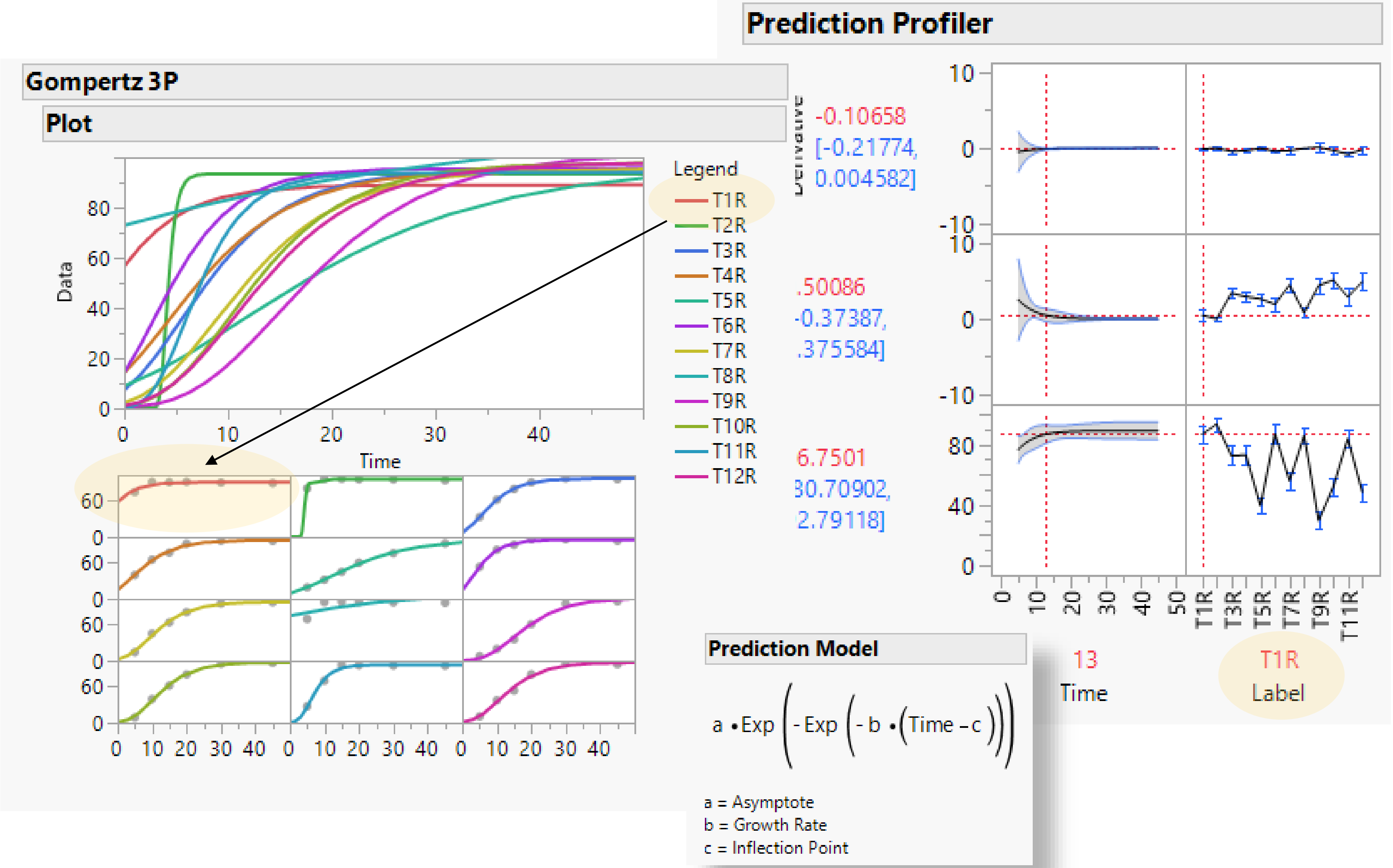


NLR

Prediction Model

$$a \cdot \mathrm{Exp}\left(-\mathrm{Exp}\left(-b \cdot \left(\mathrm{Time} - c\right)\right)\right)$$

a = Asymptote
b = Growth Rate
c = Inflection Point


Prediction Profiler
Gompertz 3P
Plot
Legend
T1R
T2R
T3R
T4R
T5R
T6R
T7R
T8R
T9R
T10R
T11R
T12R
Data
Time
-0.10658
[-0.21774, 0.004582]
.50086
-0.37387,
.375584]
6.7501
30.70902,
2.79118]
13
Time
T1R
Label
Prediction Model
a • Exp (- Exp (- b • (Time – c)))
a = Asymptote
b = Growth Rate
c = Inflection Point

NLR

NLR

	Label	Asymptote	Growth Rate	Inflection Point
T1R 1	T1R	89.072244404	0.2185624809	-3.625806907
2	T2R	93.480399791	1.76758908	4.0002987548
3	T3R	95.117117858	0.1732544061	5.4556204689
4	T4R	95.393703545	0.1508168903	4.2794474042
T5R 5	T5R	97.047132531	0.0750862269	11.579937352
6	T6R	95.886344295	0.2282099484	2.8239229453
7	T7R	95.608682945	0.1500953986	8.8540204547
8	T8R	113.26922091	0.0355126872	-23.11022674
9	T9R	102.16502758	0.1201635618	14.766362121
10	T10R	97.965019617	0.1562304451	10.087517474
11	T11R	94.032980681	0.3037771891	5.8648755174
12	T12R	97.966870258	0.1439240958	10.549169714

Prediction Model

$$a \cdot \mathrm{Exp}\left(-\mathrm{Exp}\left(-b \cdot (\mathrm{Time} - c)\right)\right)$$

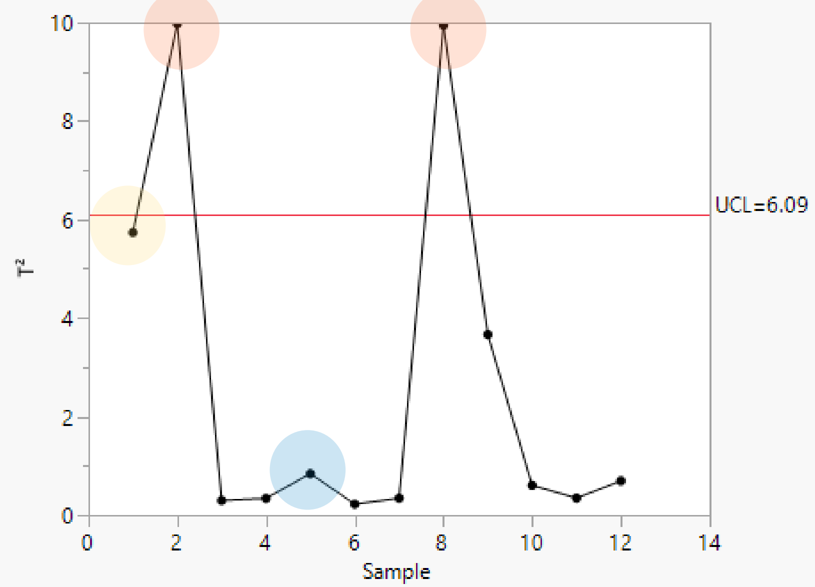
a = Asymptote
b = Growth Rate
c = Inflection Point

multivariate statistical distance (MSD)

NLR

Multivariate Control Chart

T Square with All Principal Components



Note: UCL is calculated based on Alpha=0.05

Principal Components: on Covariances

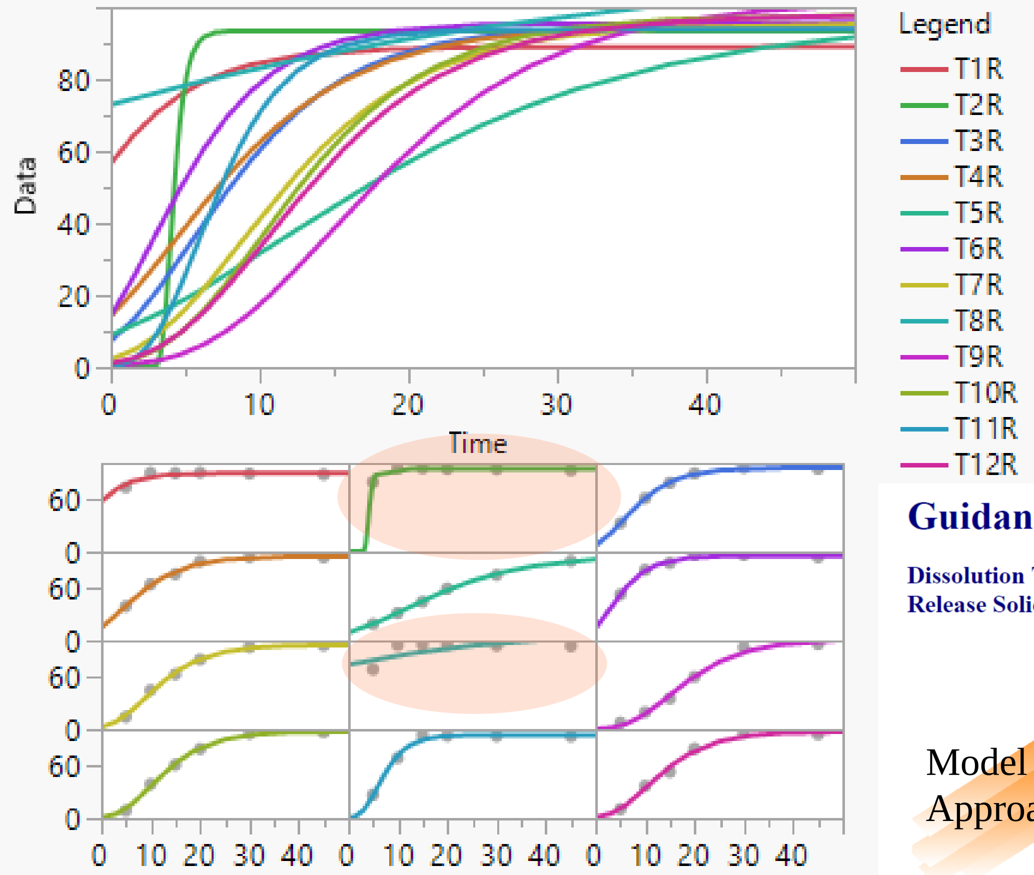
Eigenvalue	Percent	20 40 60 80	Cum Percent	ChiSquare	DF	Prob>ChiSq
109.6992	82.409		82.409	56.861	5.000	<.0001*
23.2208	17.444		99.853	37.506	2.000	<.0001*
0.1951	0.147		100.000	0.000	0.000	.

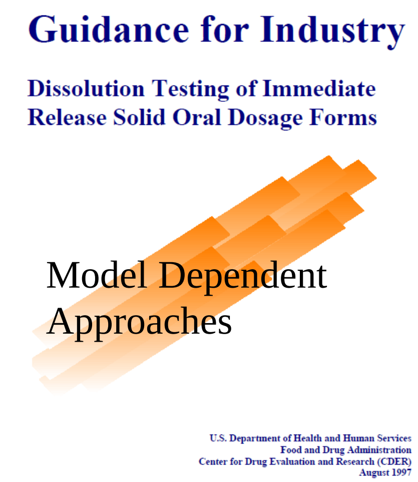
Eigenvectors			
Asymptote	-0.03538	0.19264	0.07053
Growth Rate	0.00038	-0.00663	2.26272
Inflection Point	0.08868	0.07689	0.01853

Note: Eigenvectors were divided by square root of eigenvalues.

Gompertz 3P

Plot




Guidance for Industry

Dissolution Testing of Immediate Release Solid Oral Dosage Forms

Model Dependent Approaches

U.S. Department of Health and Human Services
Food and Drug Administration
Center for Drug Evaluation and Research (CDER)
August 1997

Thank you for your attention