

A Biomed Data Analyst Training Program

Time series

Professor Ron S. Kenett

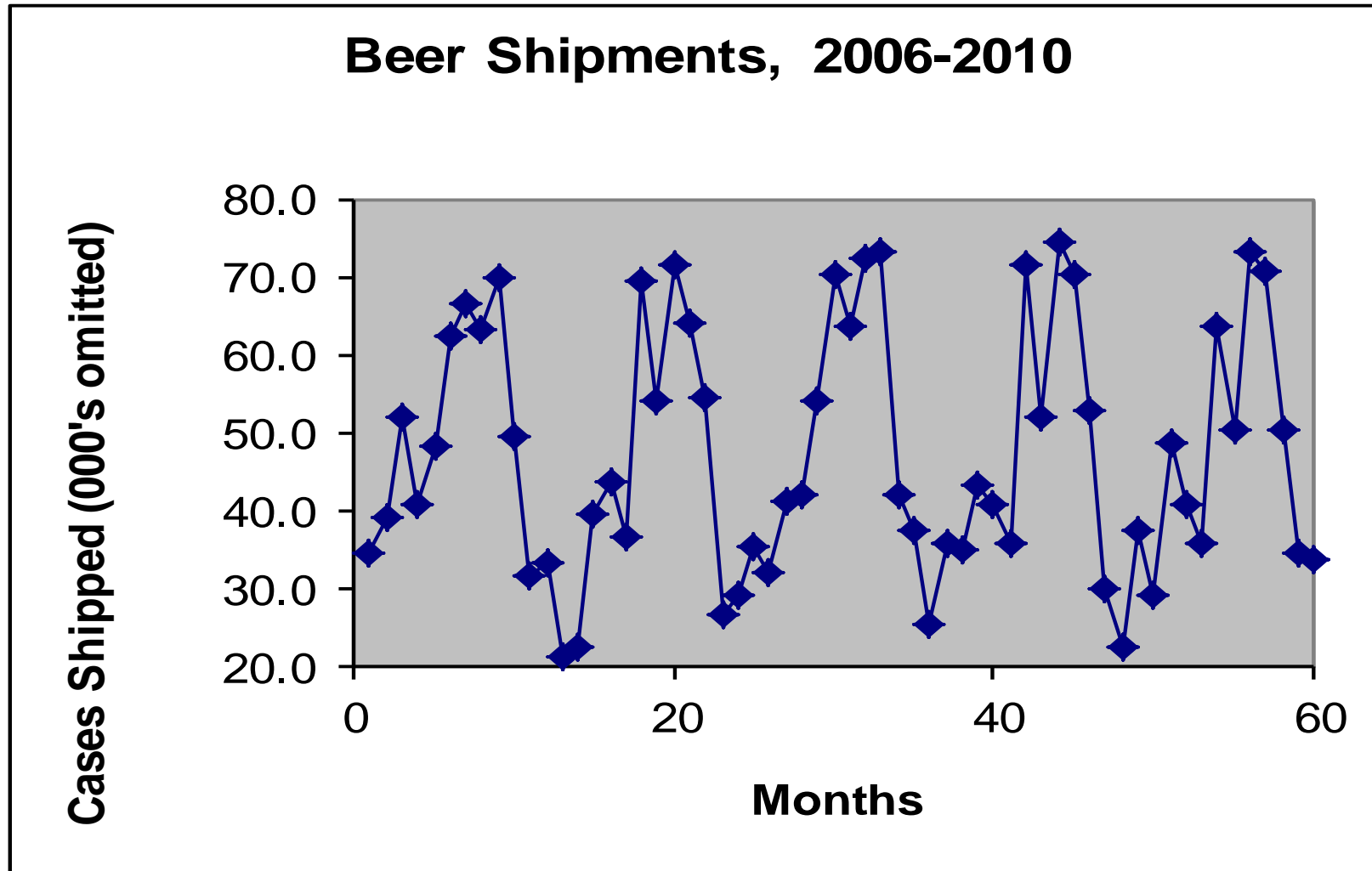
The Beer Deliveries Example

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

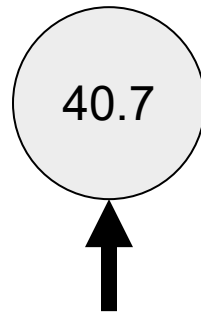
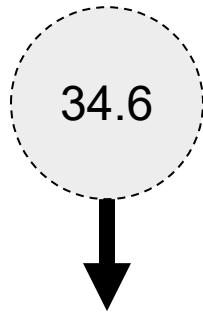
$$\begin{aligned}\bar{X}_t &= \frac{X_{t-1} + X_{t-2} + X_{t-3}}{3} \\ &= \frac{34.6 + 39.3 + 52.4}{3} = \frac{126.3}{3} = 42.1\end{aligned}$$



Three months average

Three months moving average

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

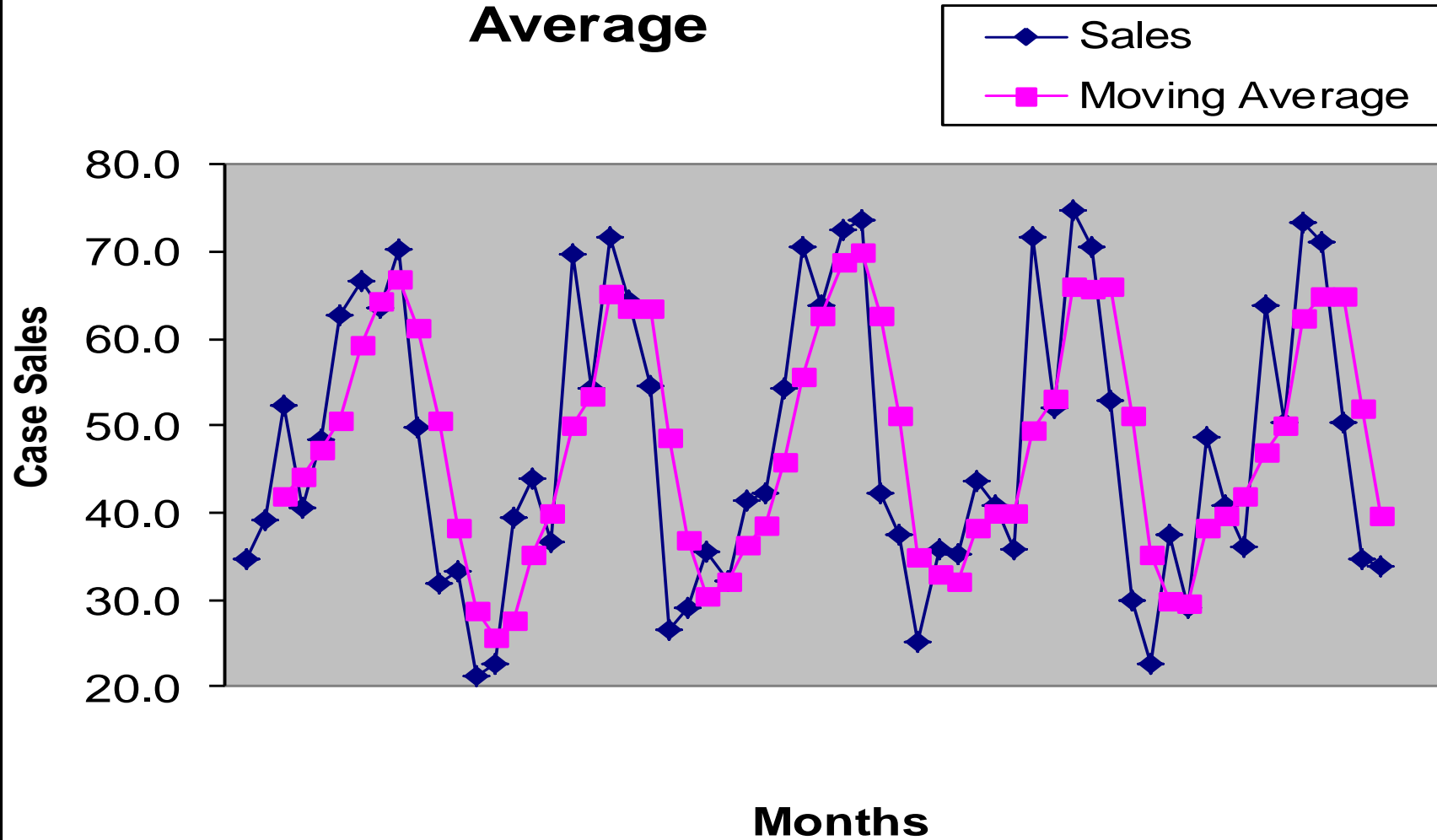


Average of next three months

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

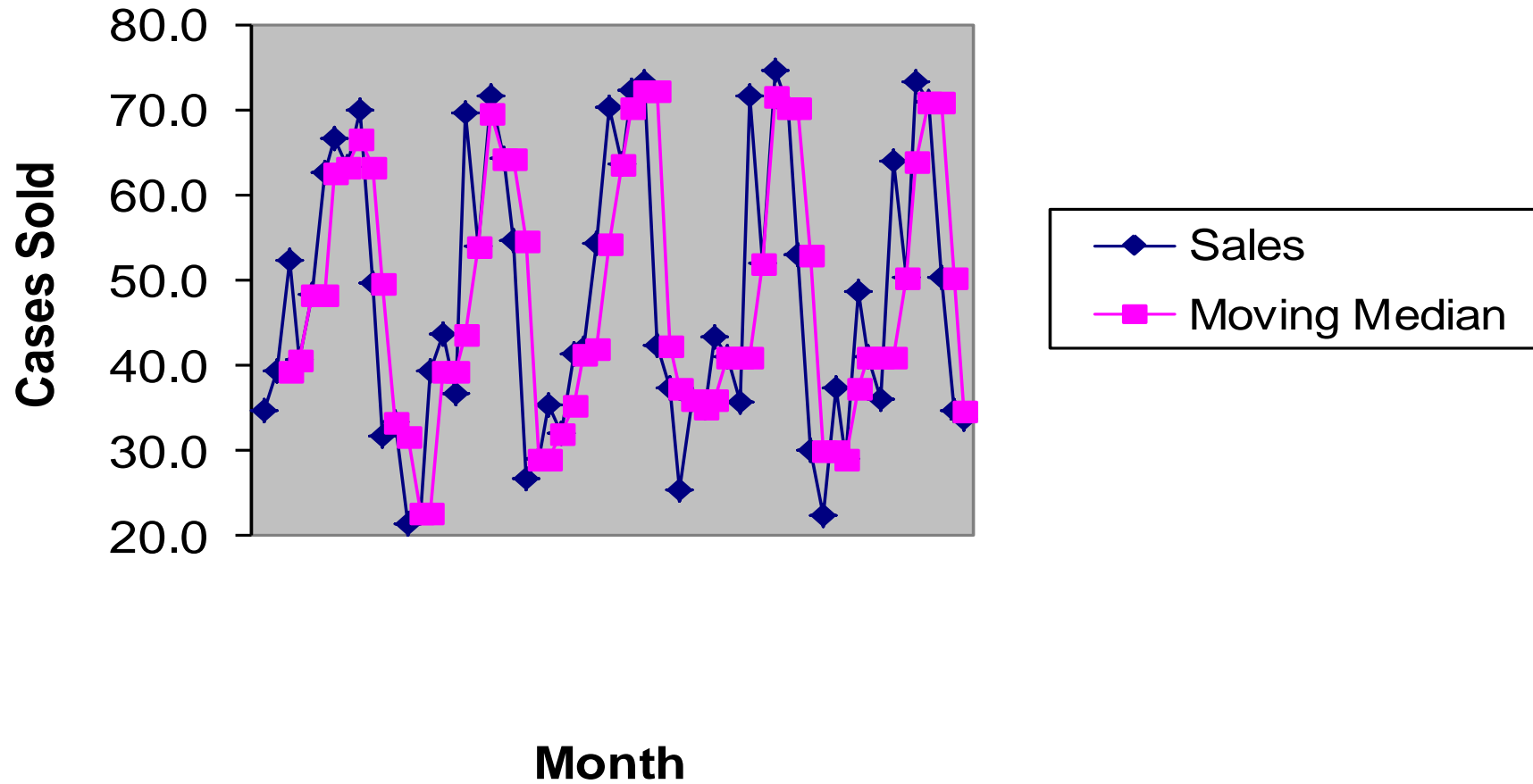
Beer Sales: Actual vs 3-month Moving Average



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

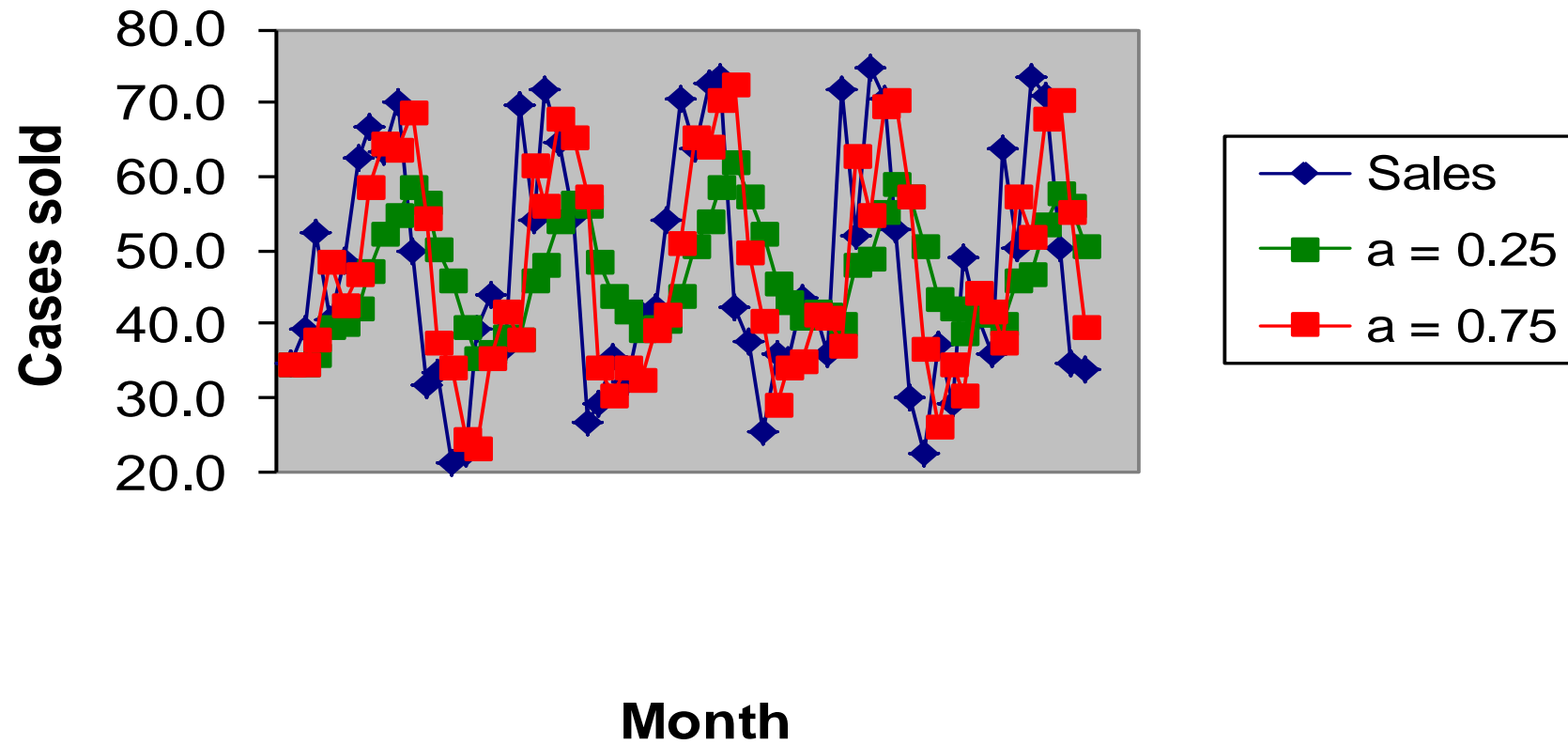


Exponential smoothing

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

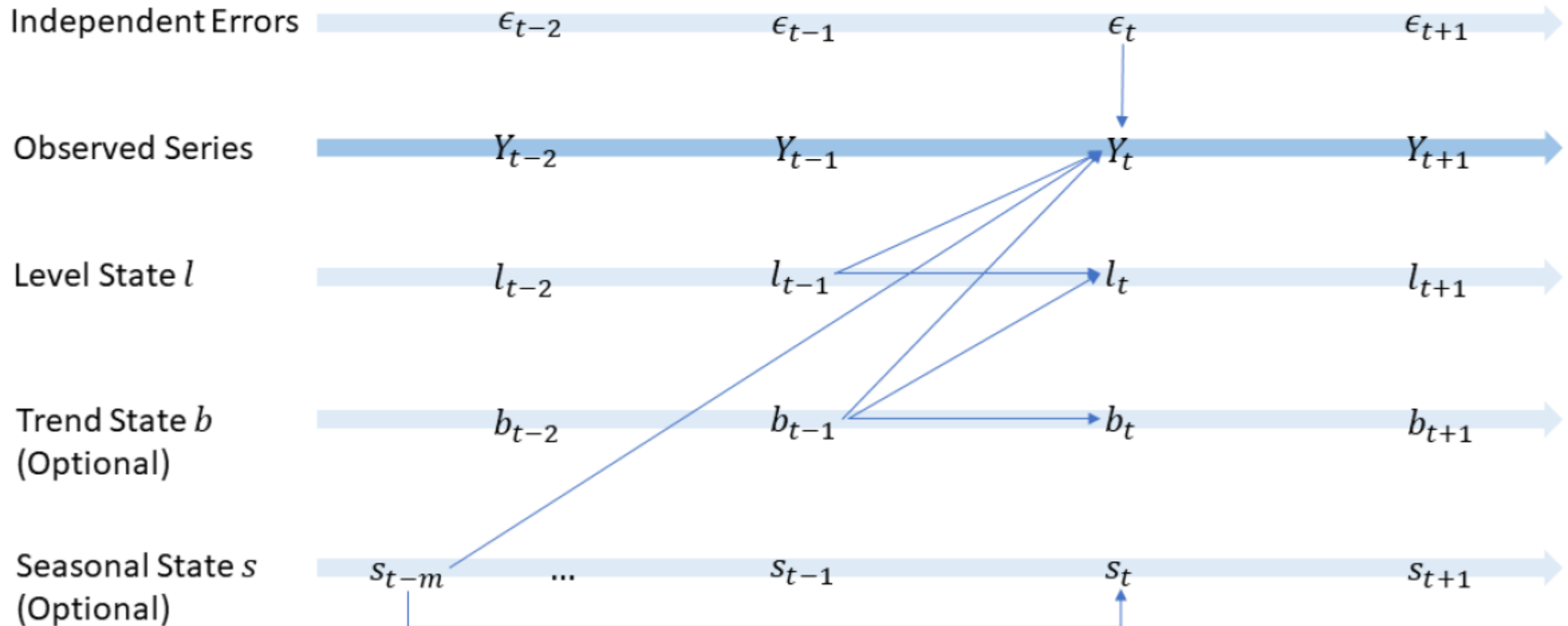
- Generally, for data that is highly variable, a higher **a** is chosen
- In practice, however, **a** seldom exceeds 0.5
- For data which has more stability, a lower value of **a** is chosen
- Typical **a** values range for 0.2 to 0.3

Beer Sales vs. Two Exponential Smoothing values



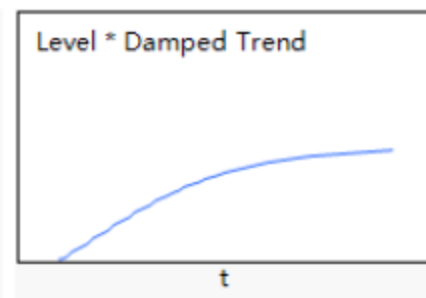
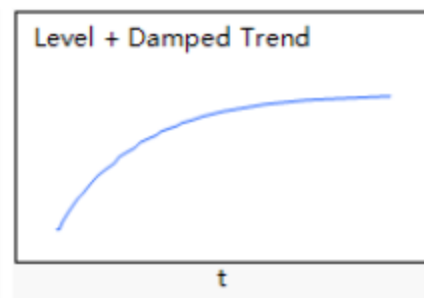
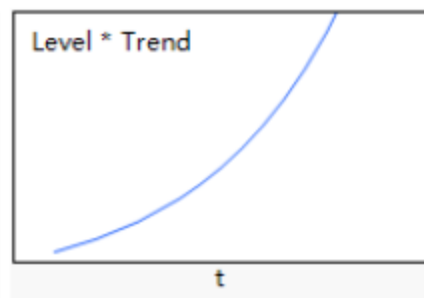
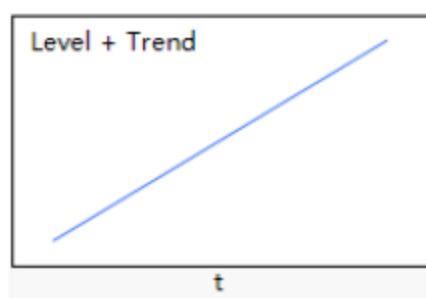
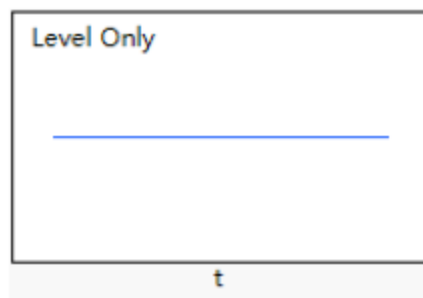
State space model

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

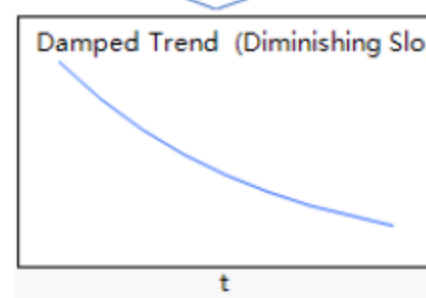
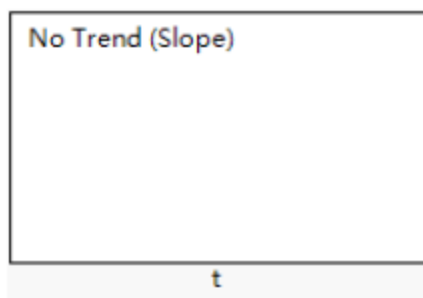


$$Y_t = \left(\left(l_{t-1} \times b_{t-1} \right) \times s_{t-m} \right) \times (1 + \epsilon_t)$$

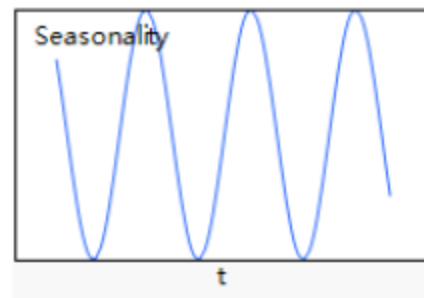
Level l



Trend b



Seasonal s



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:

- the long term trend for a specific product is +1.02
- the cyclic factor at time t is 1.03
- the seasonal factor at time t is 0.96
- by definition, we cannot state the random factor

The Multiplicative Model

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

- This predicts a slight increase ($\sim 1\%$) 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 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

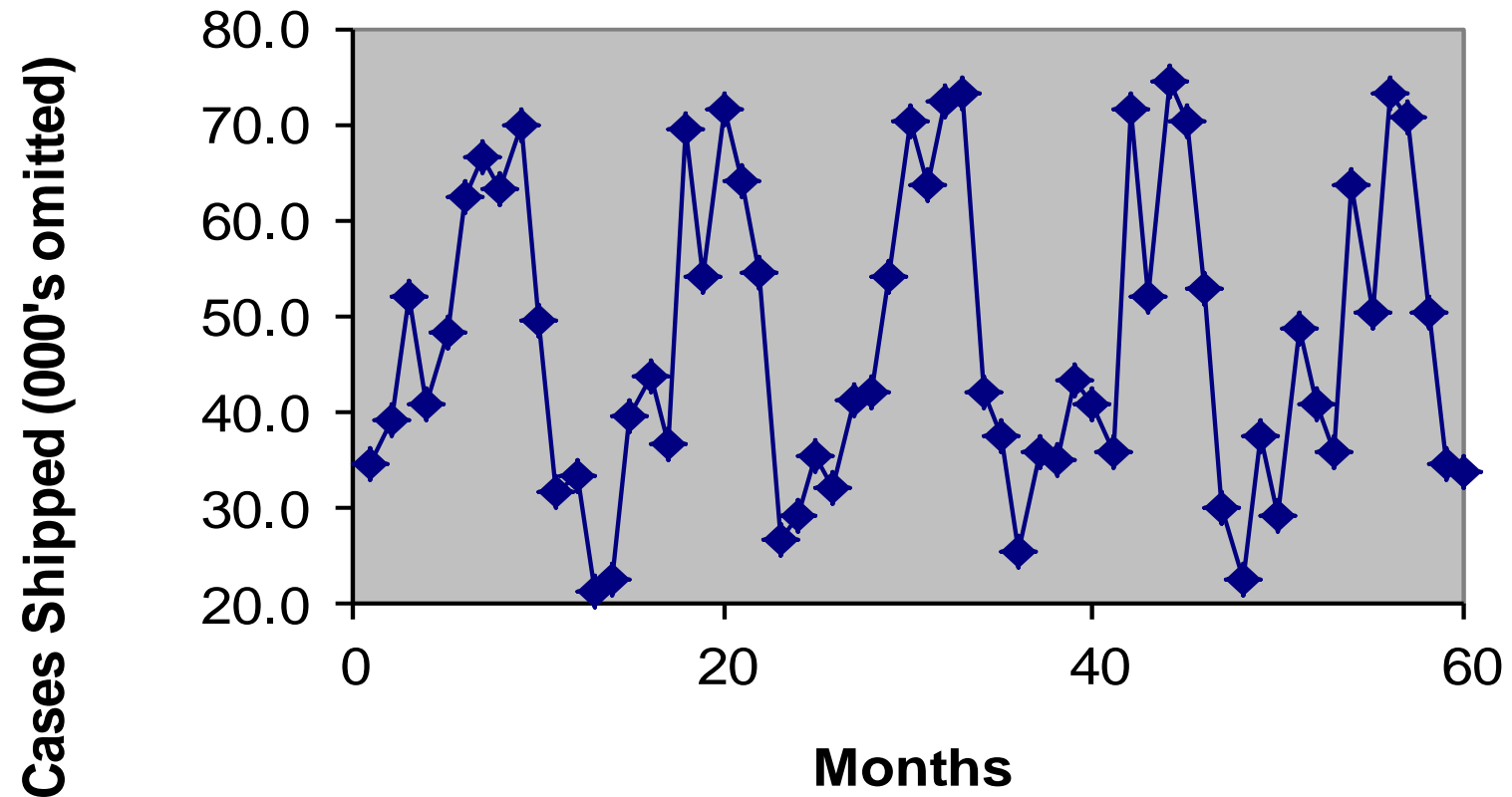
- Say, we know that overtime, we ship 50 additional units per period, and in this business cycle we are shipping an additional 200 units
- However, during this period, we typically ship 175 fewer units:

$$\begin{aligned}\hat{y}_t &= T_t + C_t + S_t \\ &= 50 + 200 - 175 = 75\end{aligned}$$

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



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:

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

- 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 high sales month
- As a result, its seasonal adjustment will bring its value down:

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

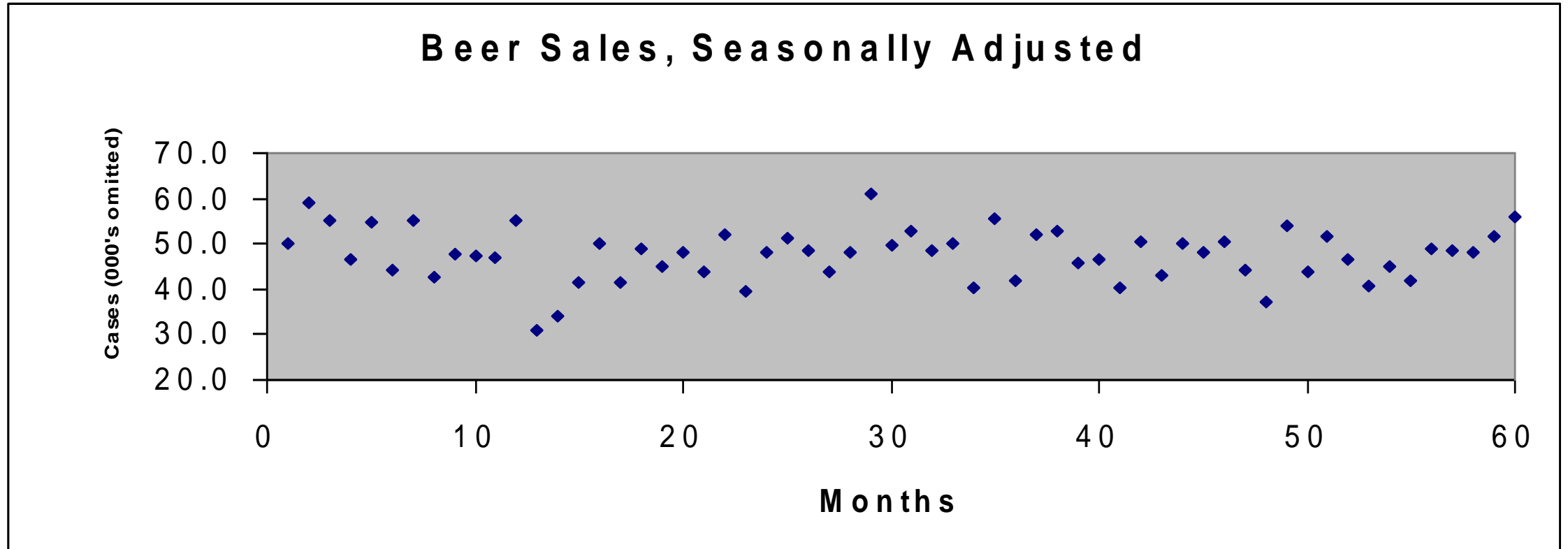
- As a result, to “seasonally adjust”, we will multiple each July value by 0.828

Analysis of seasonal effects

- In the table below, the seasonal adjustment factors are applied to all the original values
- They are “deseasonalized”:

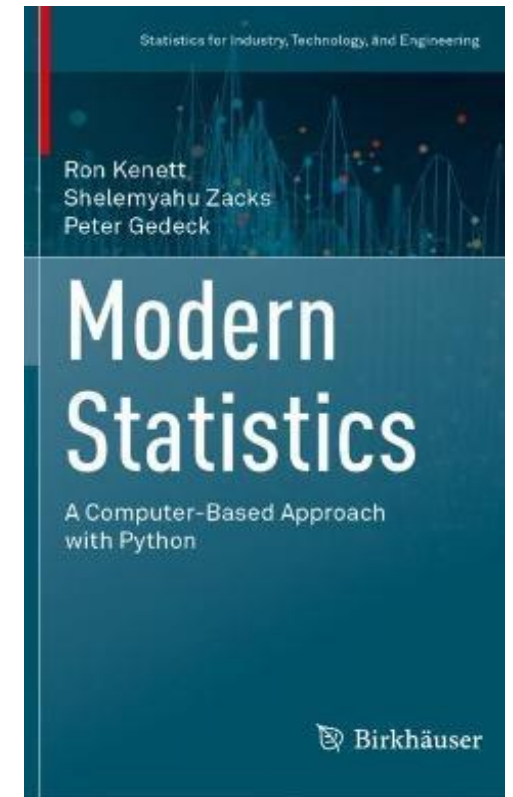
	January	February	March	April	May	June	July	August	September	October	November	December
2006	50.0	59.0	55.2	46.4	54.6	44.1	55.3	42.5	47.8	47.4	47.1	55.1
2007	30.8	34.0	41.6	50.0	41.4	49.0	44.9	48.0	43.8	52.0	39.5	48.1
2008	51.3	48.4	43.7	48.1	61.1	49.5	52.9	48.5	50.0	40.2	55.5	41.8
2009	51.9	52.9	45.9	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

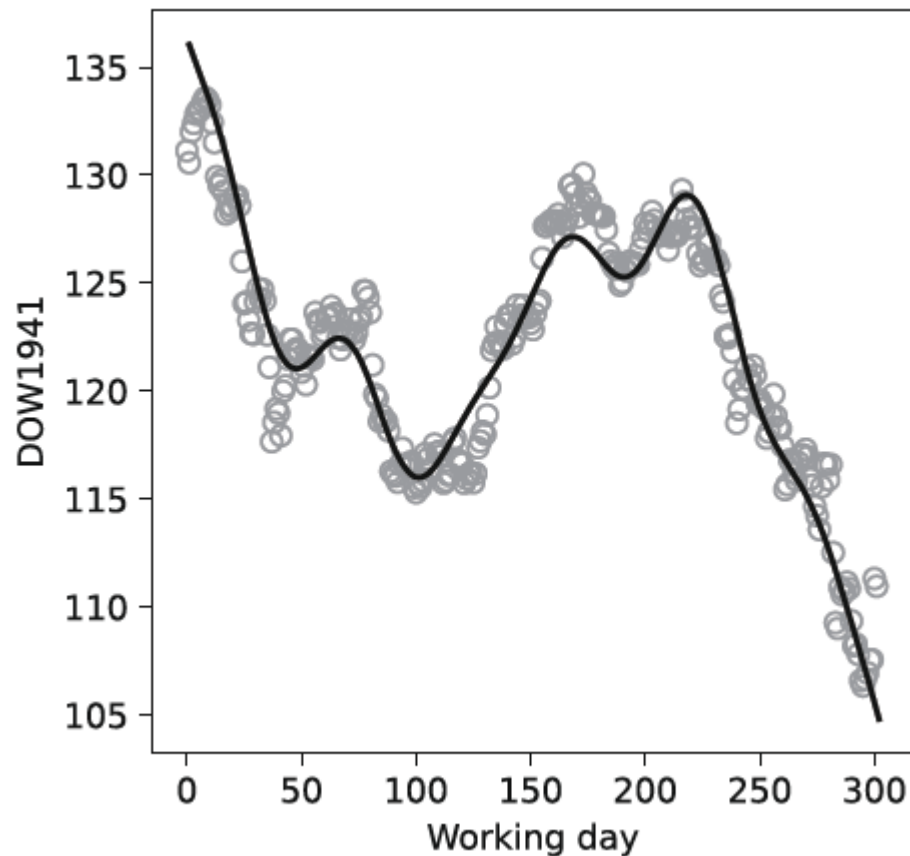
```
In [1]: import os
os.environ['OUTDATED_IGNORE'] = '1'
import warnings
from outdated import OutdatedPackageWarning
warnings.filterwarnings('ignore', category=FutureWarning)
warnings.filterwarnings('ignore', category=OutdatedPackageWarning)
```

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



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

```

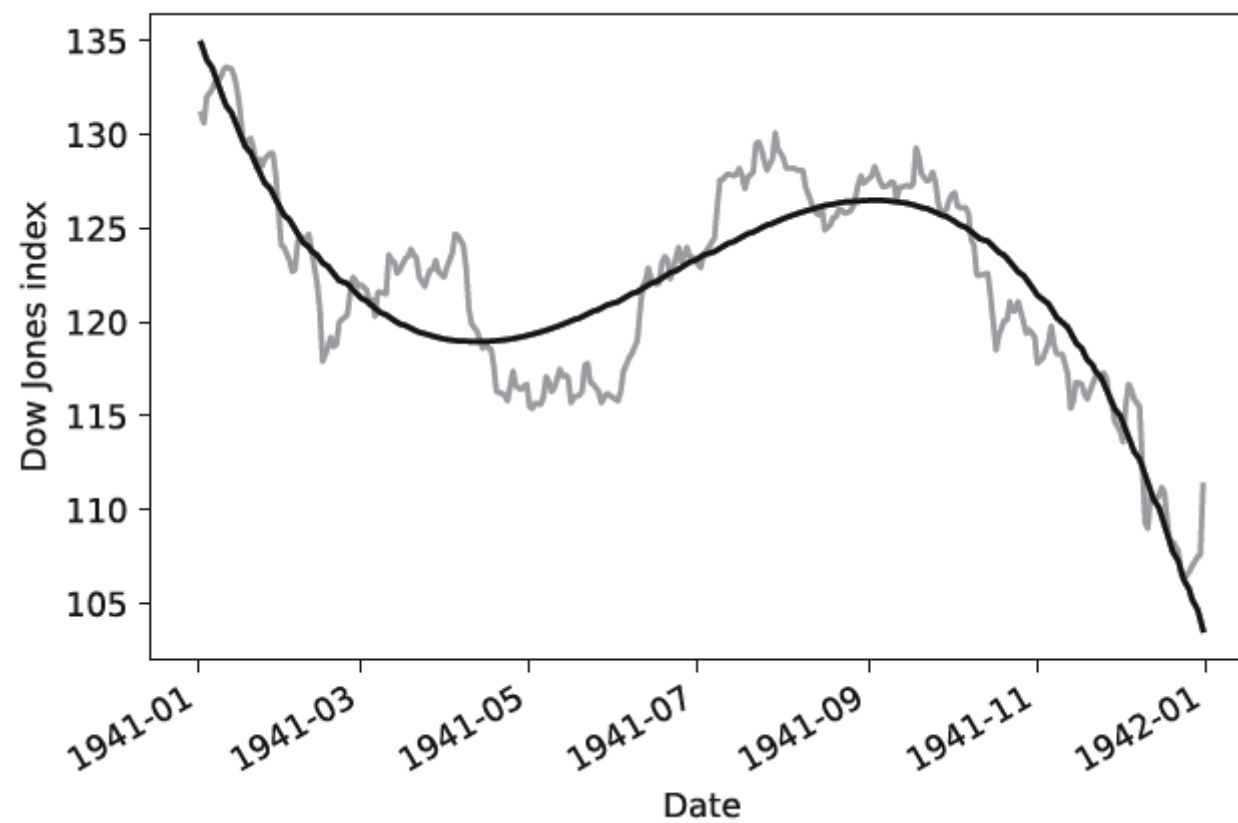
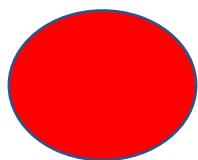
dow1941 = mistat.load_data('DOW1941')
t = np.arange(1, len(dow1941) + 1)
x = (t - 151) / 302
omega = 4 * np.pi * t / 302
ft = (123.34 + 27.73 * x - 15.83 * x ** 2 - 237.00 * x**3
      + 0.1512 * np.cos(omega) + 1.738 * np.sin(omega)
      + 1.770 * np.cos(2 * omega) - 0.208 * np.sin(2 * omega)
    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

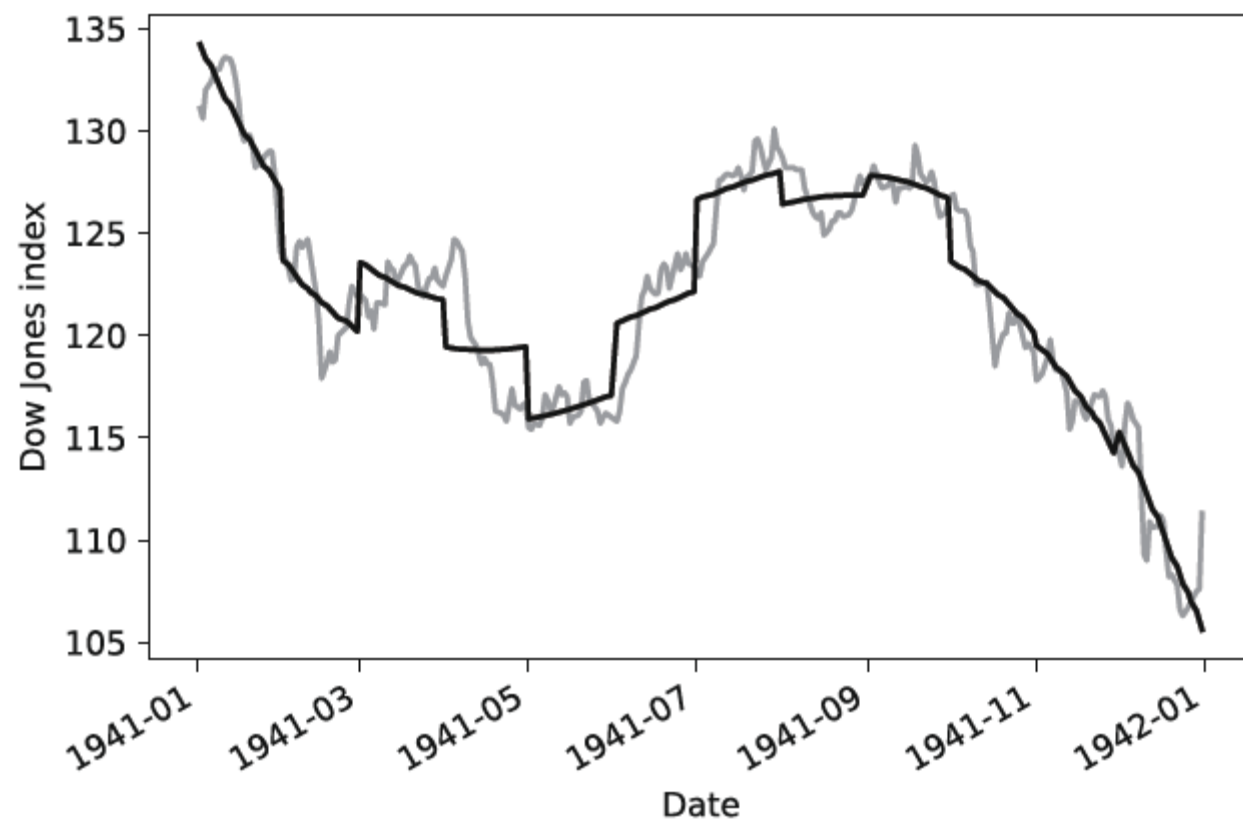
fig, ax = plt.subplots(figsize=[4, 4])
ax.scatter(dow1941.index, dow1941, facecolors='none', edgecolors='grey')
ax.plot(t, ft, color='black')
ax.set_xlabel('Working day')
ax.set_ylabel('DOW1941')
plt.show()

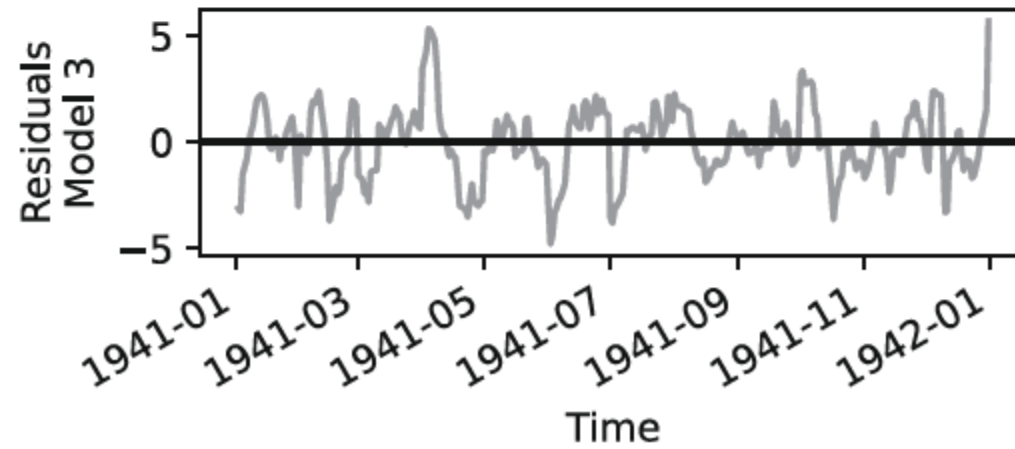
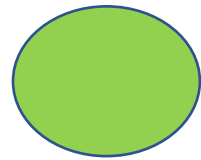
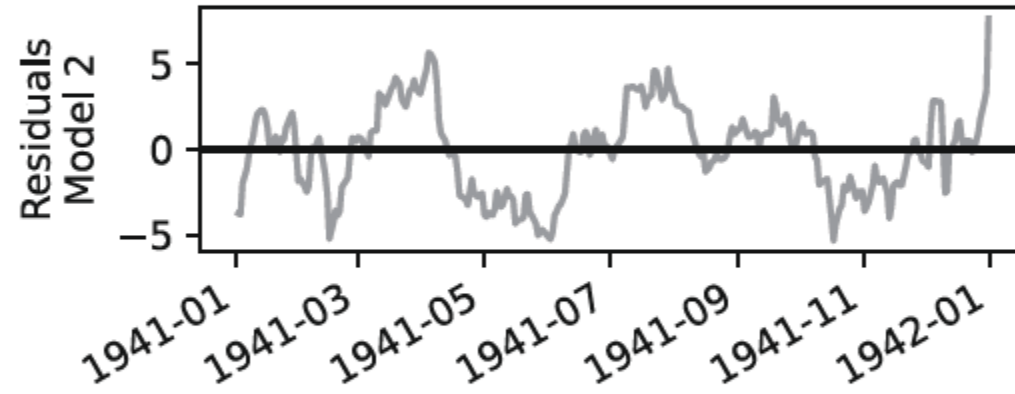
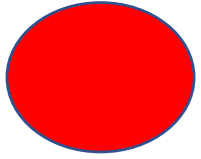
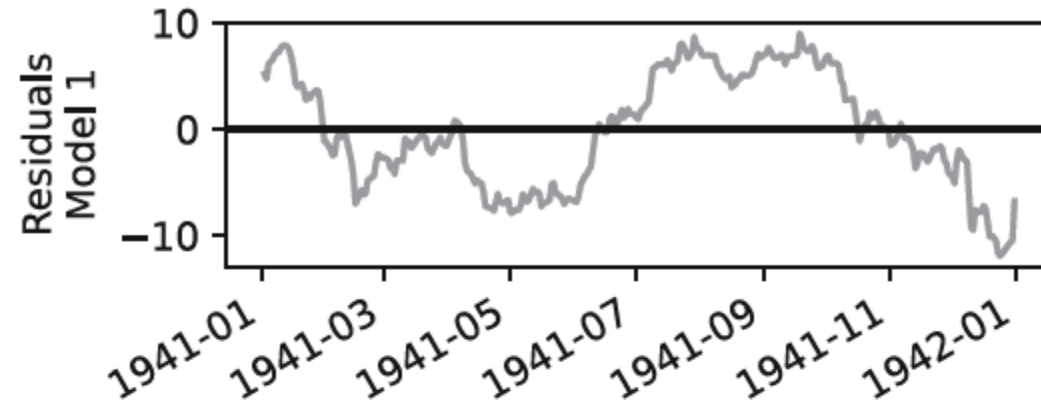
```

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

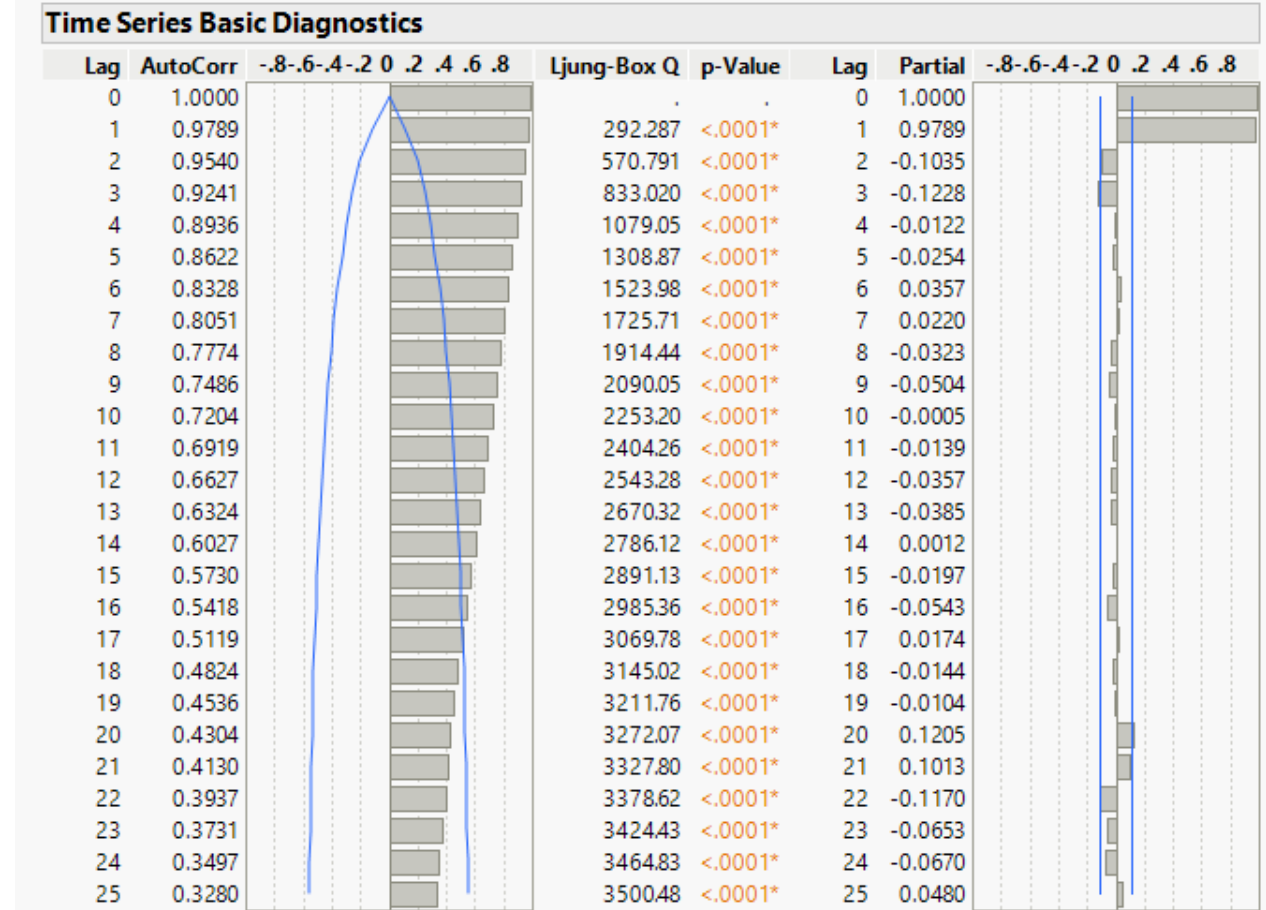
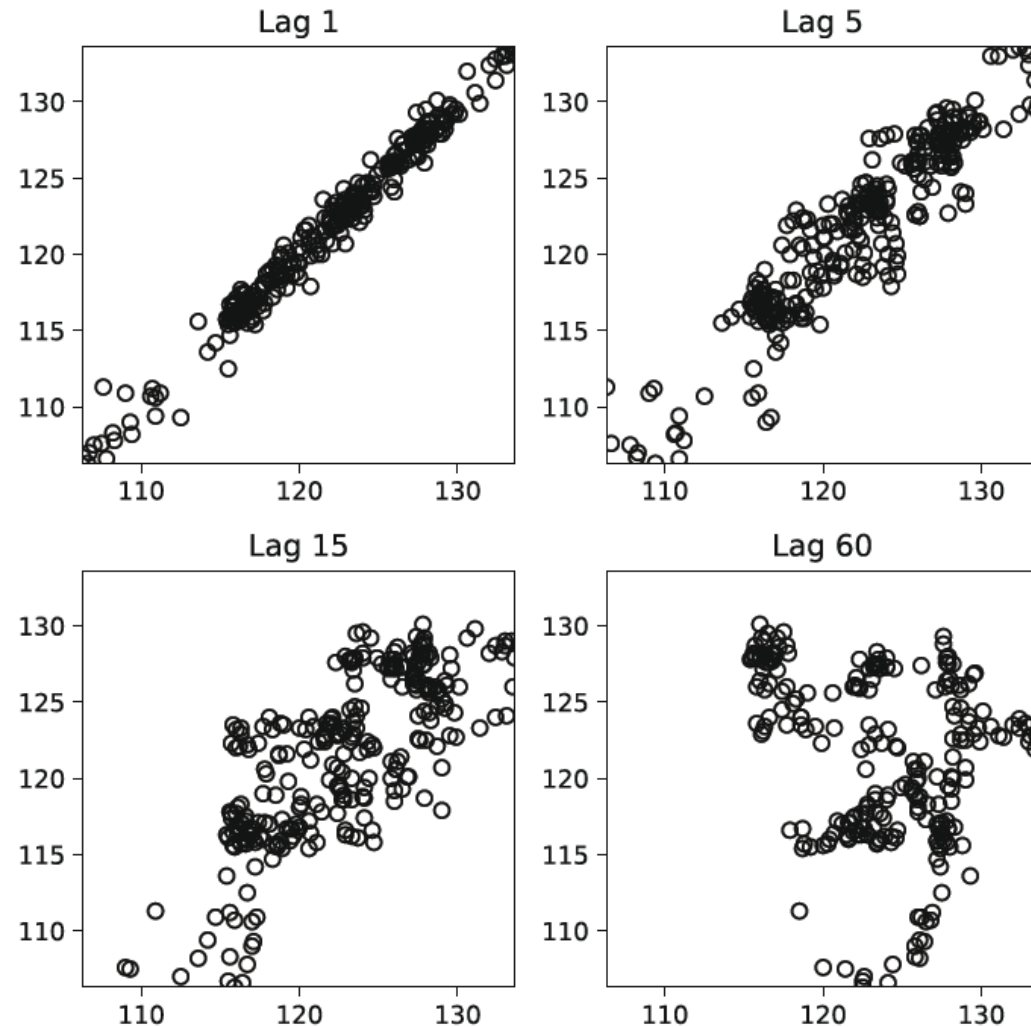




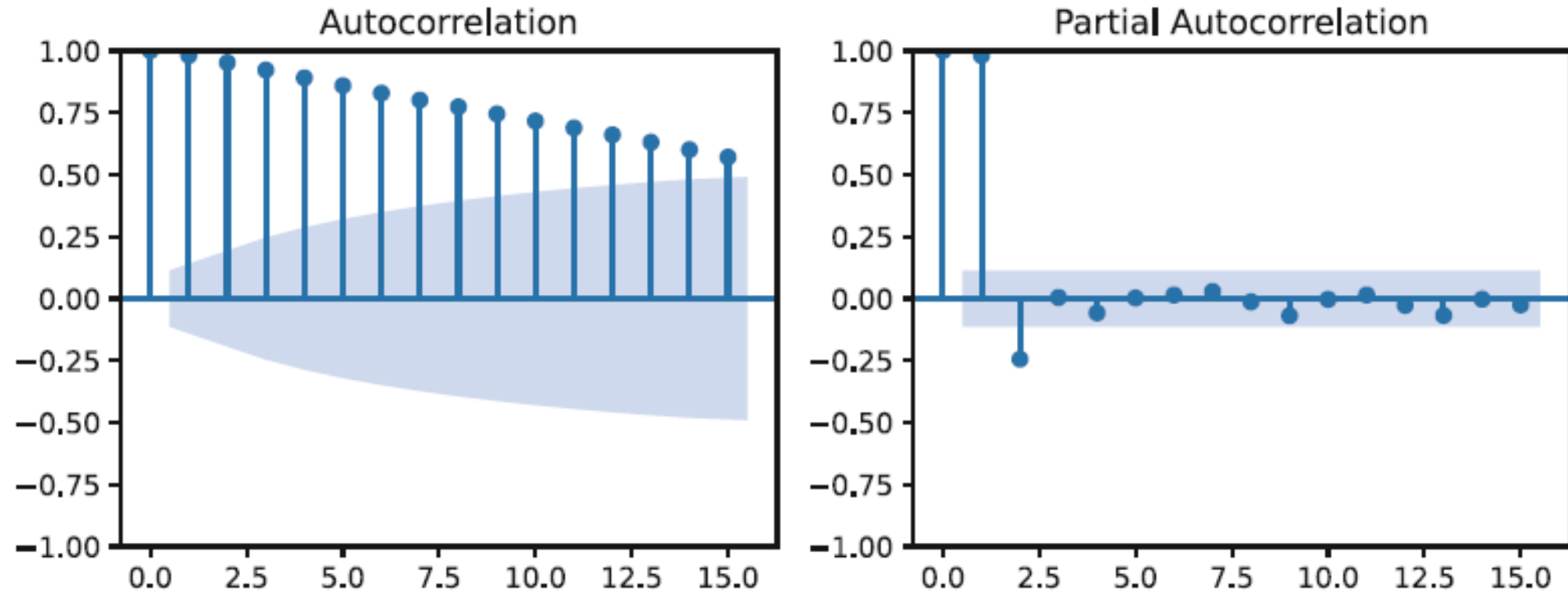




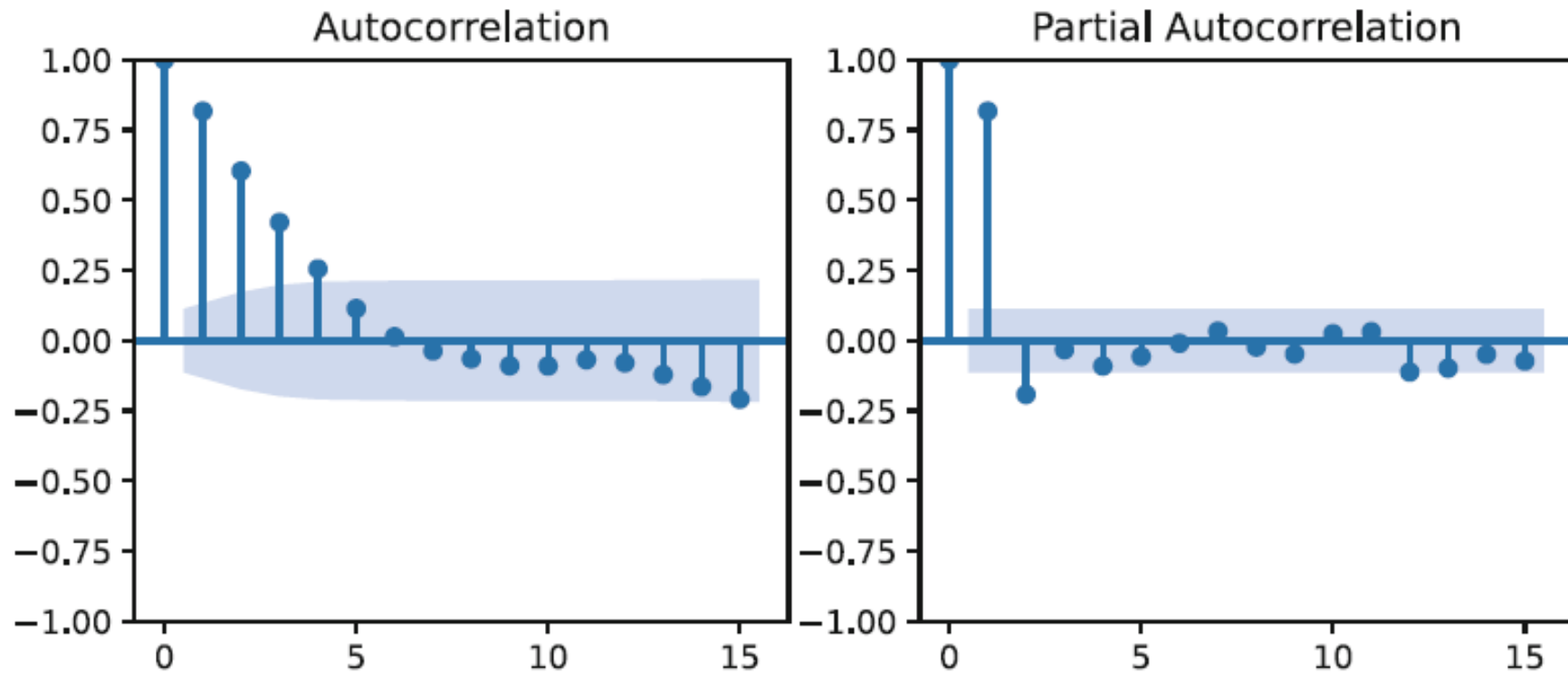
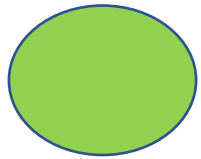
Autocorrelations



Autocorrelations and Partial Autocorrelations



Autocorrelations and Partial Autocorrelations

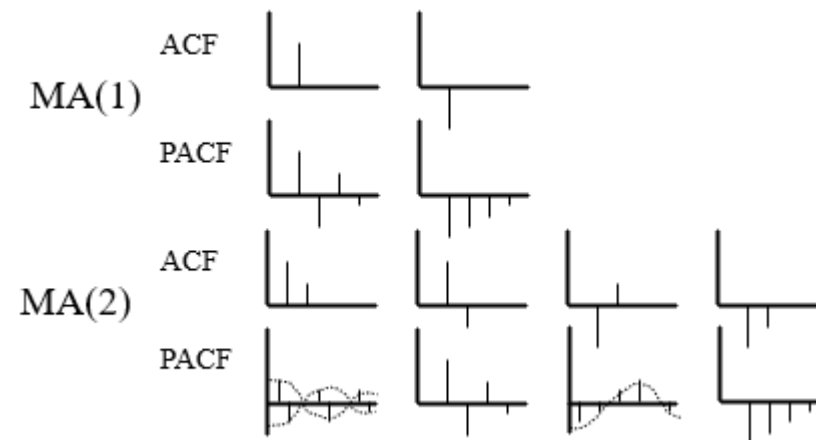
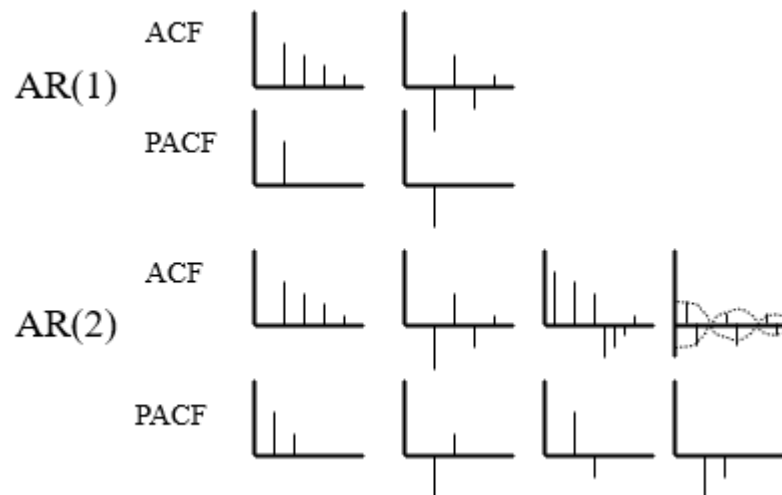


ARIMA Models

1. Auto Regressive (AR)

2. Moving Average (MA)

3. Auto Regressive Integrated Moving Average (ARIMA)



First-Order Autoregressive Processes, AR(1):

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

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) :

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

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

$$y_t = 0.5051 + 1.5537 y_{t-1} - 0.6515 y_{t-2}$$

(0.1267) (0.0707) (0.0708)

positive

negative

The diagram shows the AR(2) model equation: $y_t = 0.5051 + 1.5537 y_{t-1} - 0.6515 y_{t-2}$. Below the coefficients are their standard errors in parentheses: (0.1267) for the intercept, (0.0707) for y_{t-1} , and (0.0708) for y_{t-2} . A blue arrow points from the word 'positive' to the coefficient 1.5537. Another blue arrow points from the word 'negative' to the coefficient -0.6515.

Using AR Model for Forecasting:

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

$$\begin{aligned} y_{T+1} &= \delta + \theta_1 y_T + \theta_2 y_{T-1} = 0.5051 + (1.5537)(6.2) - (0.6515)(6.63) \\ &= 5.8186 \end{aligned}$$

$$\begin{aligned} y_{T+2} &= \delta + \theta_1 y_{T+1} + \theta_2 y_T = 0.5051 + (1.5537)(5.8186) - (0.6515)(6.2) \\ &= 5.5062 \end{aligned}$$

$$\begin{aligned} y_{T+1} &= \delta + \theta_1 y_T + \theta_2 y_{T-1} = 0.5051 + (1.5537)(5.5062) - (0.6515)(5.8186) \\ &= 5.2693 \end{aligned}$$

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

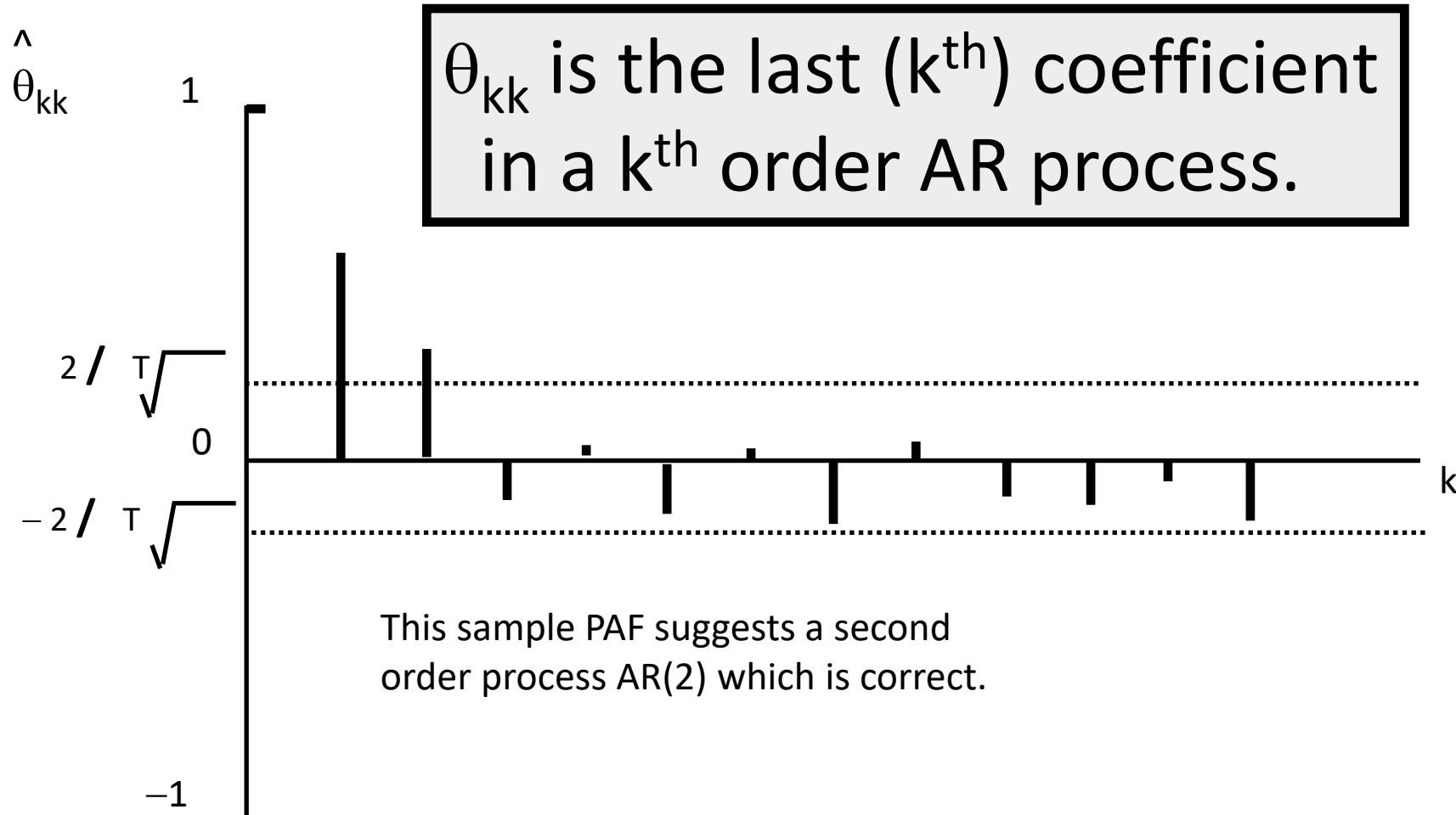
The Partial Autocorrelation Function (PAF)

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

Partial Autocorrelation Function

Data simulated
from this model:

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



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

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

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 (AF)

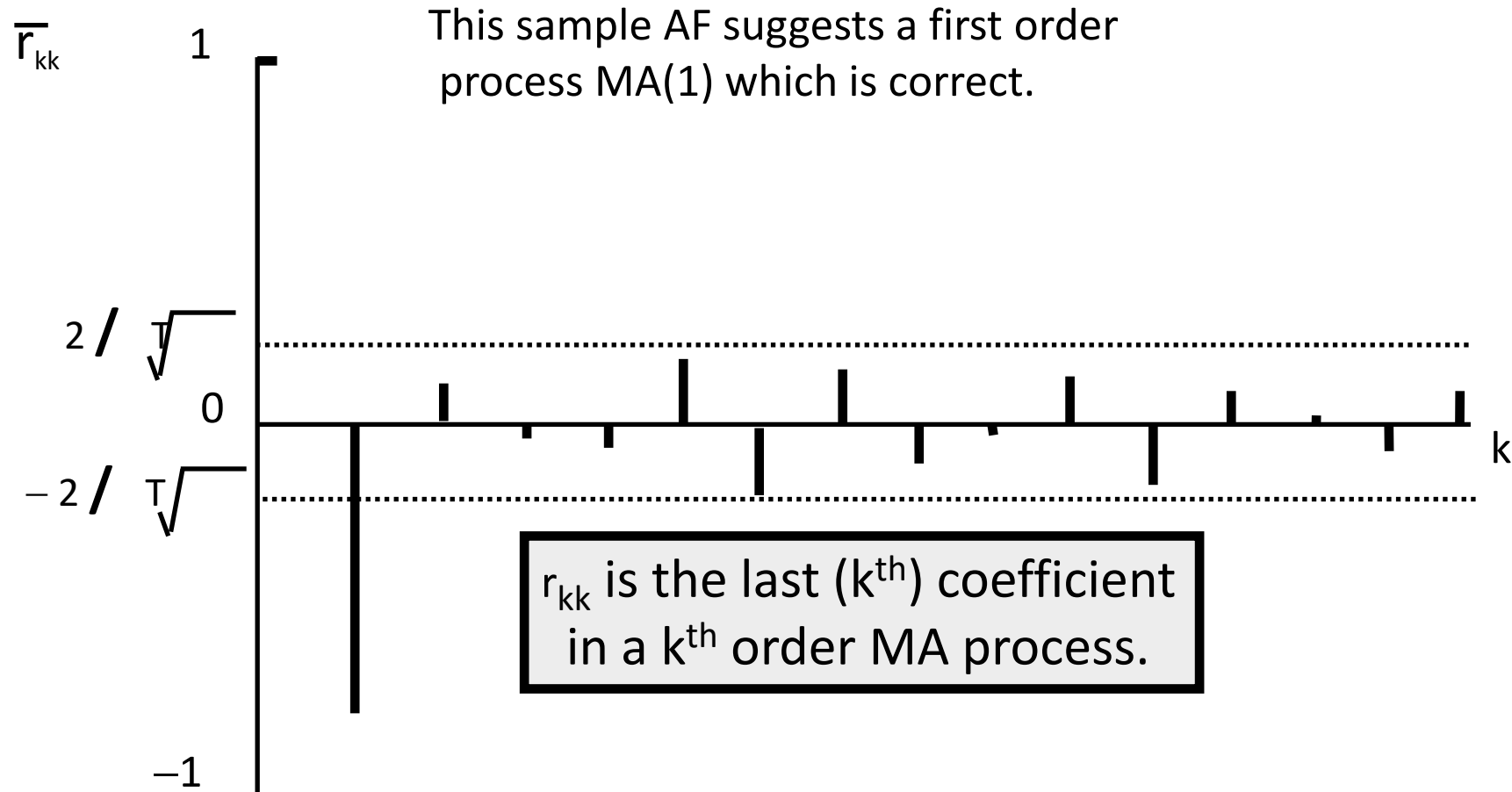
The AF is the sequence of correlations between $(y_t \text{ and } y_{t-1})$, $(y_t \text{ and } y_{t-2})$, $(y_t \text{ and } y_{t-3})$, and so on, without holding the effects of earlier lags on y_t constant.

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

Autocorrelation Function

Data simulated
from this model:

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



Autoregressive Moving Average ARMA(p,q)

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

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

Goodness of fit criteria

- Akaike's Information Criterion [AIC]
- Schwartz's Bayesian Criterion [SBC or 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.

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

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.

Series F

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



Series FJMP

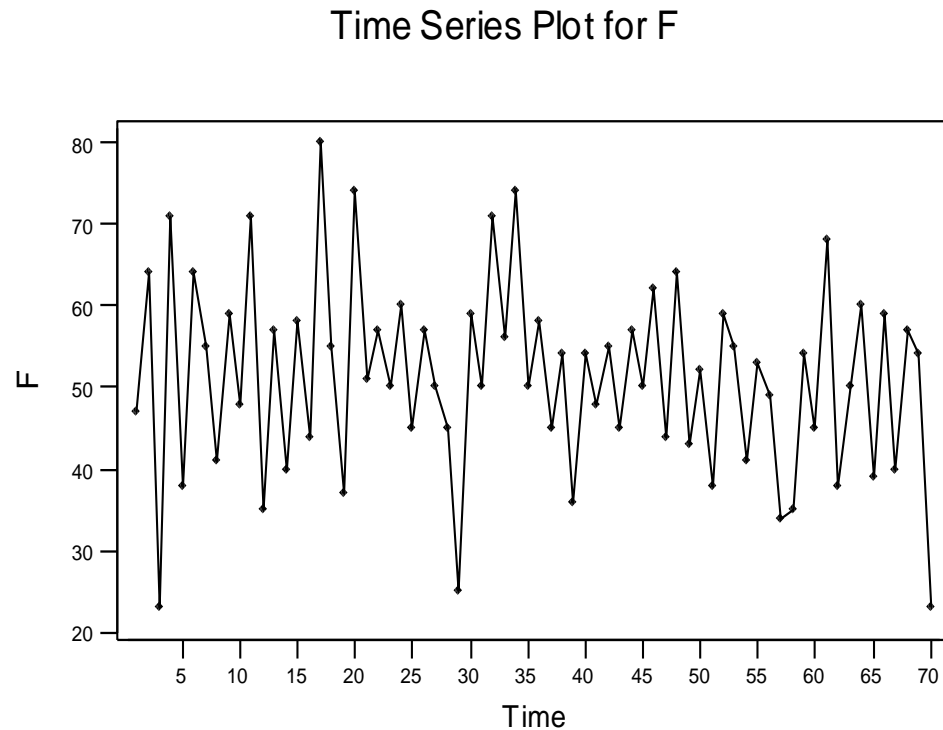
First we plot the series to check for trend, periodicity, etc. which will need the application of differencing.

We inspect the **correlogram** and **partial correlogram** to help in identifying an ARMA model for the stationary series we obtain.

The Autocorrelation Function (AF)

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.

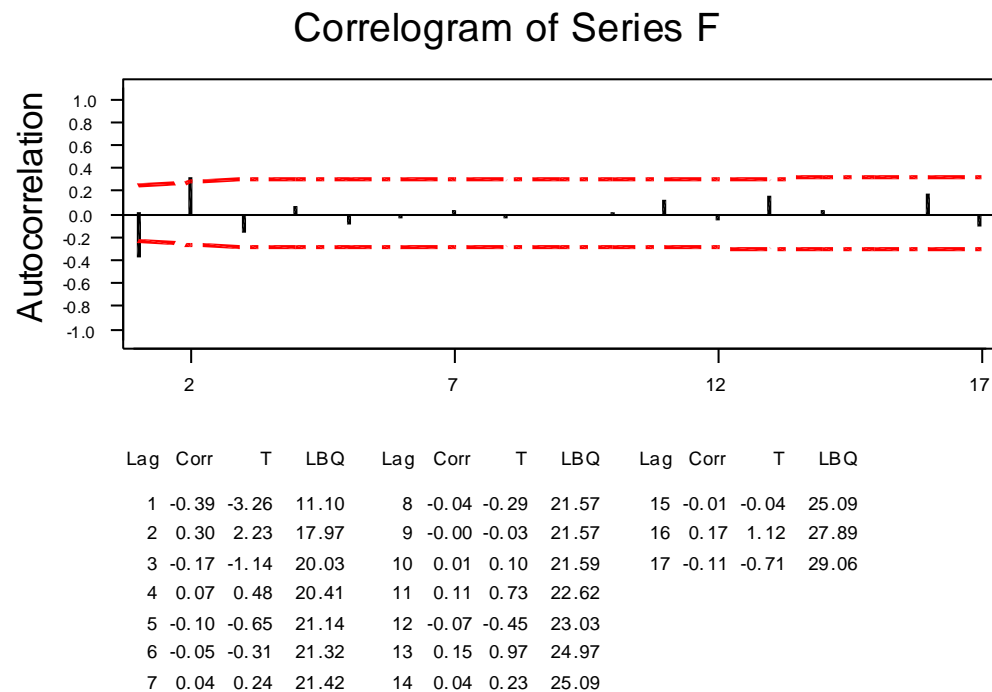
Model Identification.

- 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

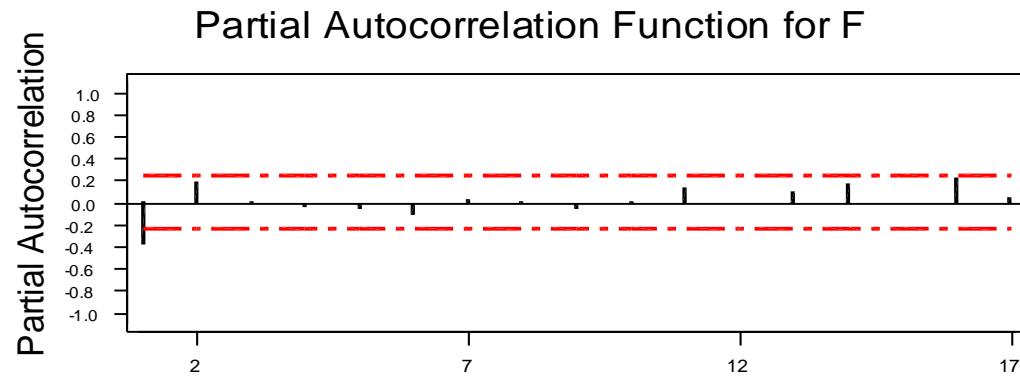


Let's look at a correlogram of the series.

The only large values are at lags 1 and 2.

Maybe AR(2) or AR(1)?

Partial Correlogram



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

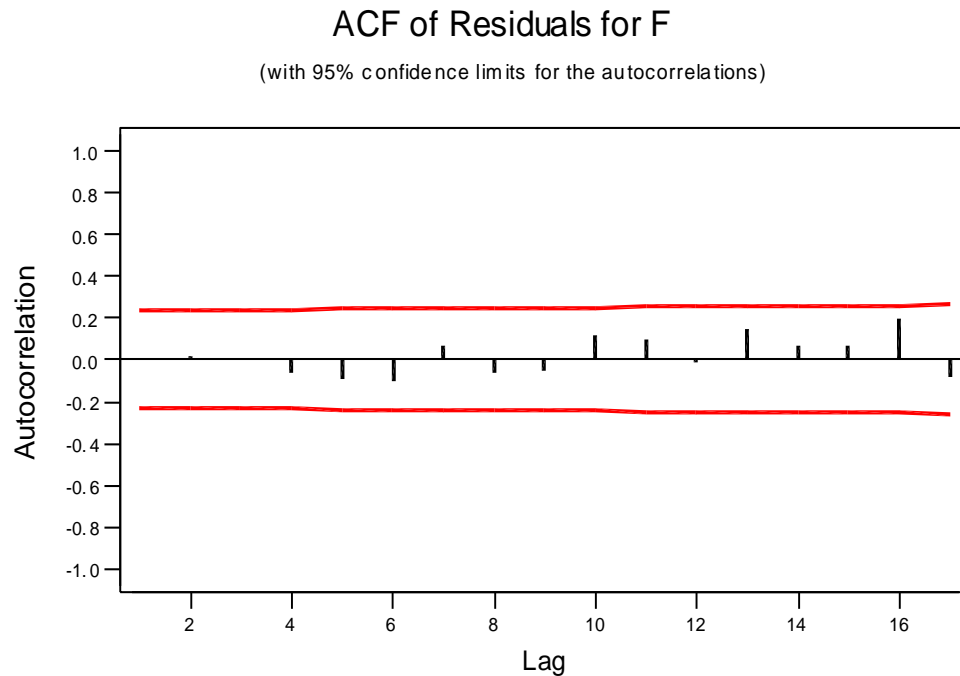
Final Estimates of Parameters

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

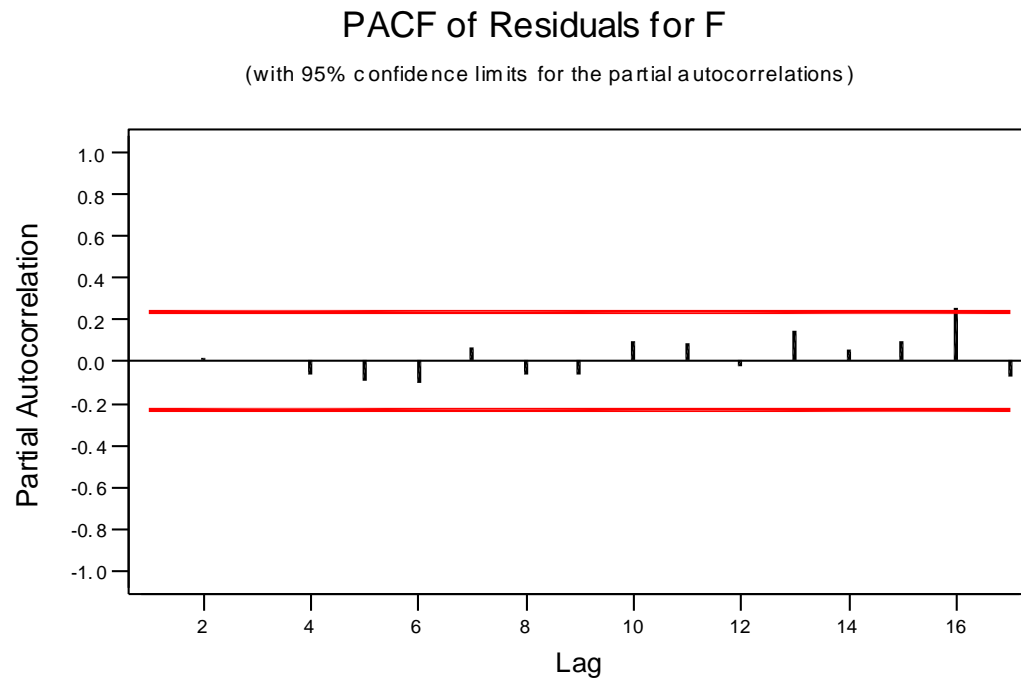
- Examine various aspects of the residuals to evaluate the adequacy of our chosen model.
- Use the acf and pacf to look for any remaining times series structure in the residuals which we have not removed.
- Check for normality and constant variance.

Correlogram of Residuals



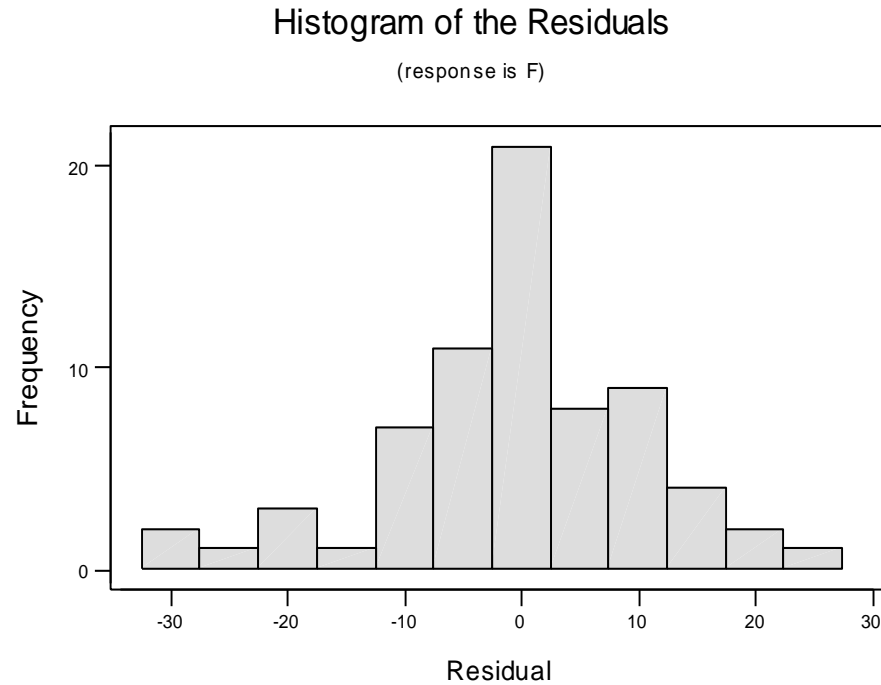
No obvious autocorrelation left after we have fitted our AR(2) model to the original series.

Partial Correlogram of Residuals



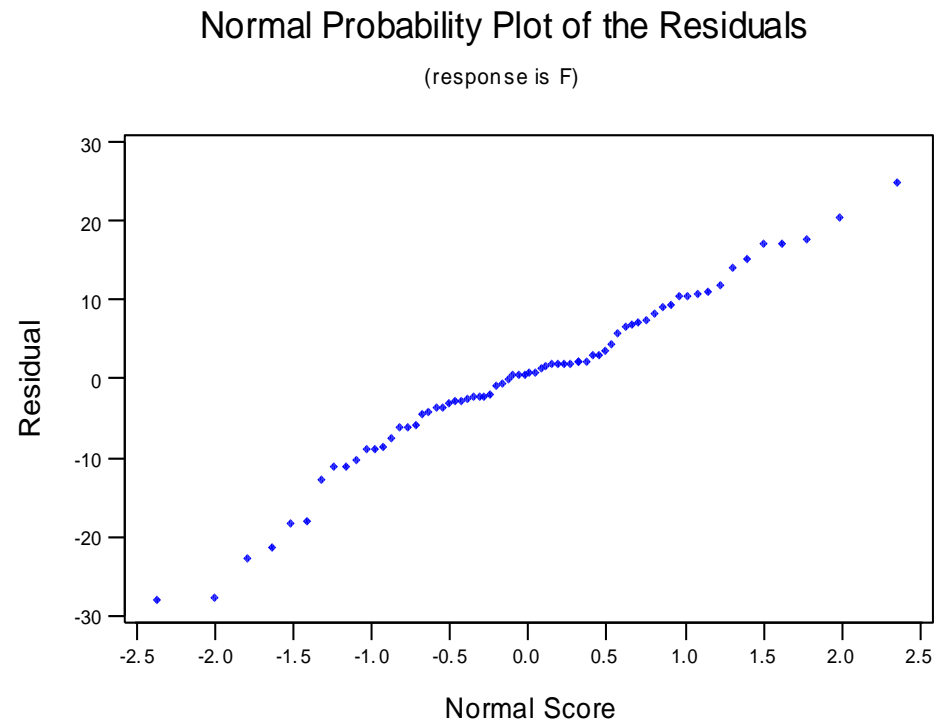
As we expect from the **correlogram**, there is no indication of residual partial autocorrelation.

Histogram of Residuals



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

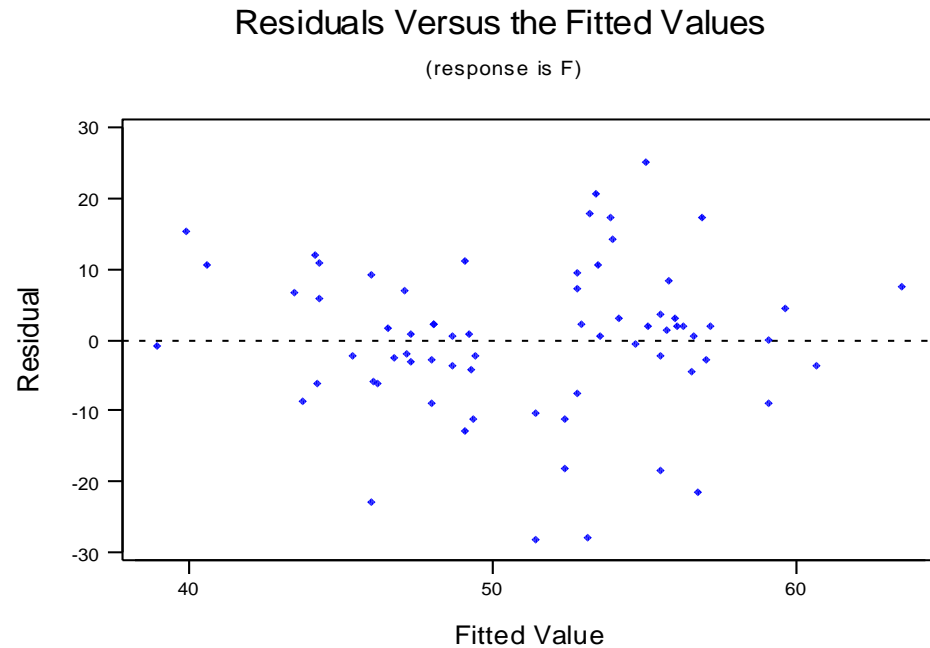
Normal Probability Plot



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

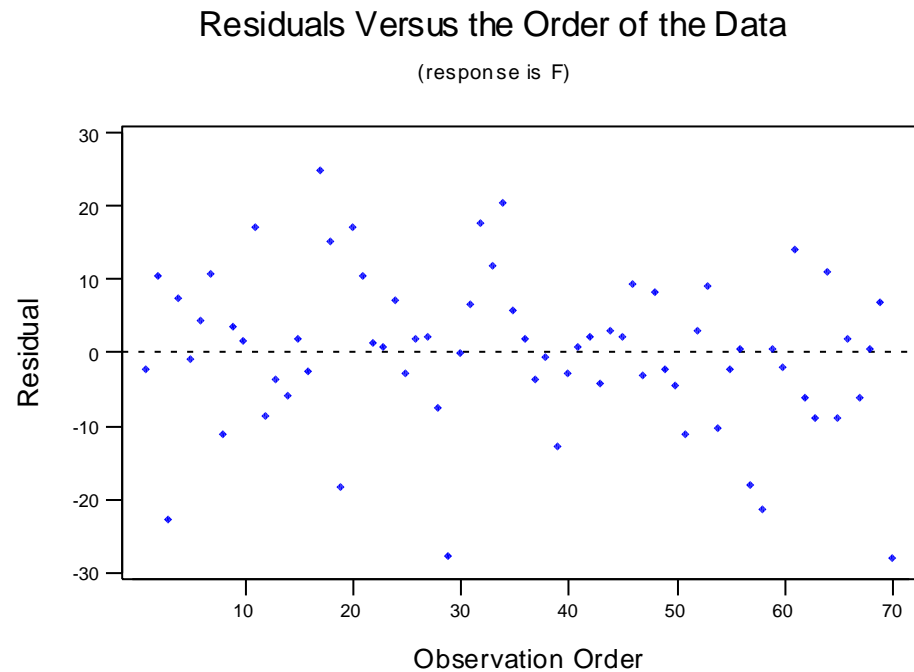
A normal distribution looks OK.

Residuals v. Fitted Values

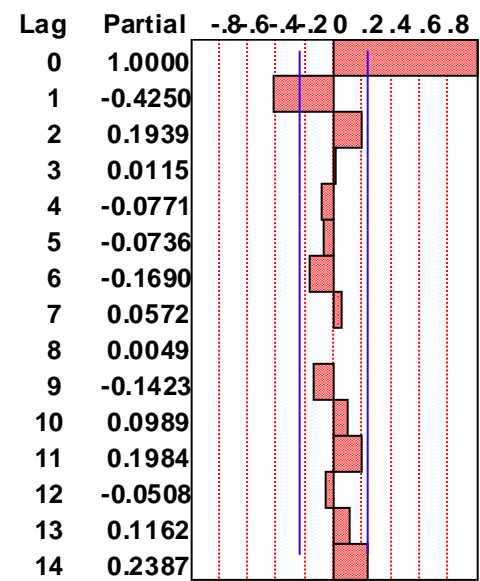
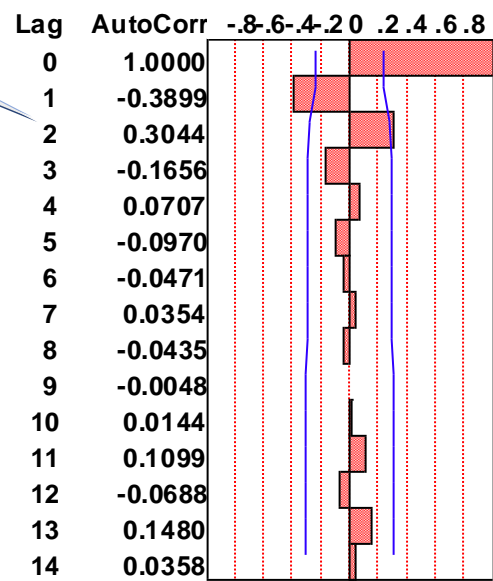
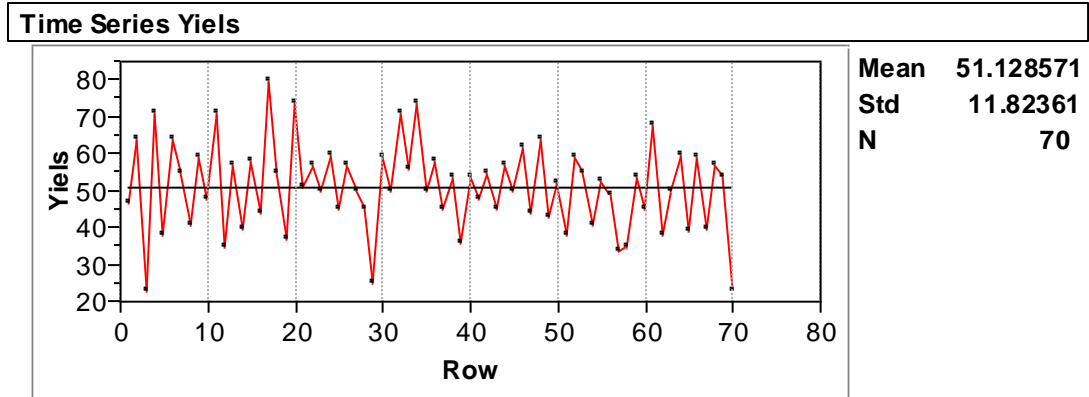


No obvious problems such as the variance depending on the size of the observation.

Residuals in Time Order



No obvious failures to model
dependence of residuals
over time .



Time Series Yield

Mean 51.128571
Std 11.82361
N 70

Model: AR(2) No Constrain

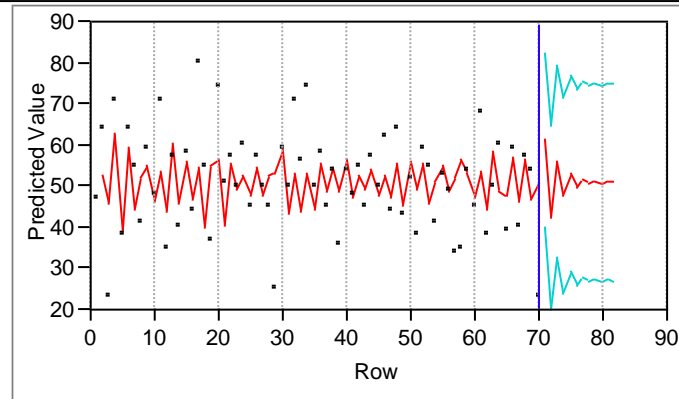
Model Summary

DF	67	Stable	Yes
Sum of Squared Errors	7890.13866	Invertible	Yes
Variance Estimate	117.763264		
Standard Deviation	10.8518783		
Akaike's 'A' Information Criterion	339.807346		
Schwarz's Bayesian Criterion	346.552832		
RSquare	0.19331113		
RSquare Adj	0.16886601		
-2LogLikelihood	331.005918		

Parameter Estimates

Term	Lag	Estimate	Std Error	t Ratio	Prob> t
Constant		59.0828904			
AR1	1	-0.3407072	0.1217622	-2.80	0.0067
AR2	2	0.1873369	0.1223016	1.53	0.1303
Intercept	0	51.2262958	1.100755	46.54	<.0001

Forecast



AR (2)

Time Series Yield

Mean 51.128571
Std 11.82361
N 70

Model: AR(1) No Constrain

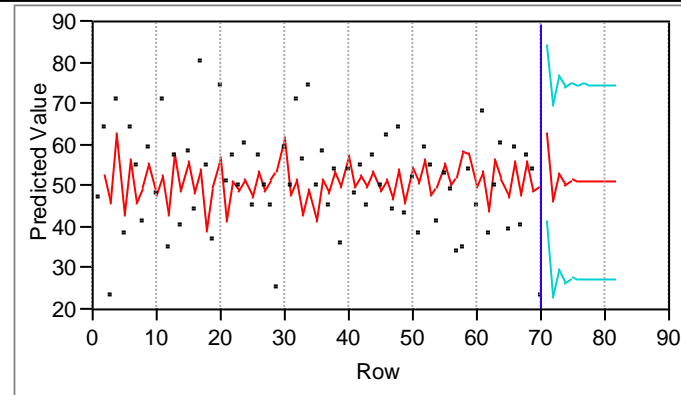
Model Summary

DF	68	Stable	Yes
Sum of Squared Errors	8162.10344	Invertible	Yes
Variance Estimate	120.030933		
Standard Deviation	10.9558629		
Akaike's 'A' Information Criterion	339.142464		
Schwarz's Bayesian Criterion	343.639454		
RSquare	0.1659866		
RSquare Adj	0.15353864		
-2LogLikelihood	333.306469		

Parameter Estimates

Term	Lag	Estimate	Std Error	t Ratio	Prob> t
Constant		Estimate 72.7502314			
AR1	1	-0.4190788	0.1128024	-3.72	0.0004
Intercept	0	51.2658165	0.9137256	56.11	<.0001

Forecast



AR (1)

Time Series Yield

Mean 51.128571
Std 11.82361
N 70

Model: MA(2) No Constrain

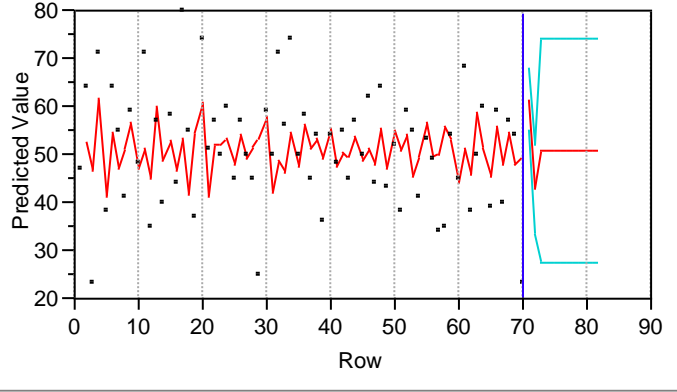
Model Summary

DF	67	Stable	Yes
Sum of Squared Errors	730.00111	Invertible	No
Variance Estimate	10.895539		
Standard Deviation	3.30083913		
Akaike's 'A' Information Criterion	173.184741		
Schwarz's Bayesian Criterion	179.930226		
RSquare	0.1794338		
RSquare Adj	0.15456816		
-2LogLikelihood	332.054151		

Parameter Estimates

Term	Lag	Estimate	Std Error	t Ratio	Prob> t	Constant Estimate
MA1	1	1.0581298	0.553345	1.91	0.0601	51.1694704
MA2	2	-3.3125935	1.3512071	-2.45	0.0168	
Intercept	0	51.1694704	1.2515956	40.88	<.0001	

Forecast



MA (2)

Time Series Yield

Mean 51.128571
Std 11.82361
N 70

Model: ARMA(1, 1) No Constrain

Model Summary

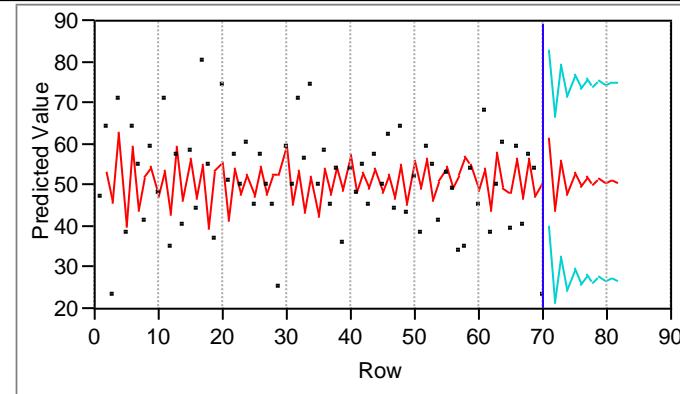
DF	67	Stable	Yes
Sum of Squared Errors	7946.82957	Invertible	Yes
Variance Estimate	118.609397		
Standard Deviation	10.8907941		
Akaike's 'A' Information Criterion	340.3085		
Schwarz's Bayesian Criterion	347.053986		
RSquare	0.18760671		
RSquare Adj	0.16298873		
-2LogLikelihood	331.486205		

Parameter Estimates

Term	Lag	Estimate	Std Error	t Ratio	Prob> t
AR1	1	-0.6856451	0.1731017	-3.96	0.0002
MA1	1	-0.3236699	0.2175862	-1.49	0.1416
Intercept	0	51.2519228	1.002723	51.11	<.0001

Constant Estimate 86.3925517

Forecast



ARMA (1,1)

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

Time Series Yield

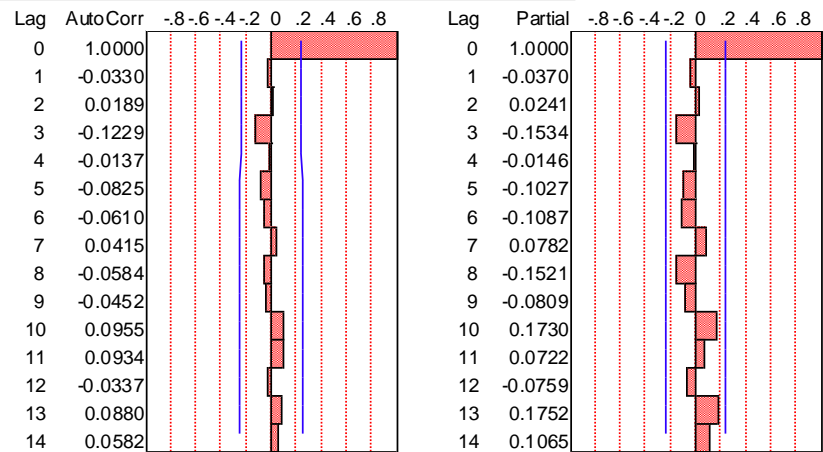
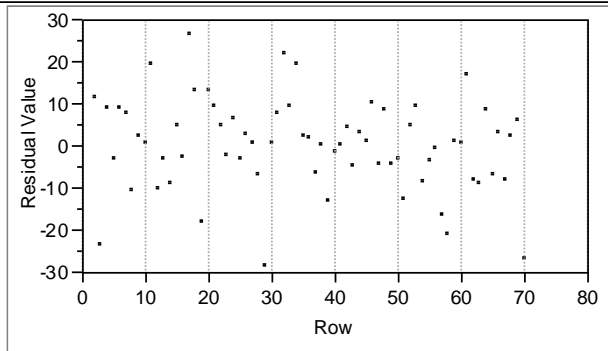
Mean 51.128571
Std 11.82361
N 70

Model: MA(2) No Constrain

Parameter Estimates

Term	Lag	Estimate	Std Error	t Ratio	Prob> t	Constant Estimate
MA1	1	1.0581298	0.553345	1.91	0.0601	51.1694704
MA2	2	-3.3125935	1.3512071	-2.45	0.0168	
Intercept	0	51.1694704	1.2515956	40.88	<.0001	

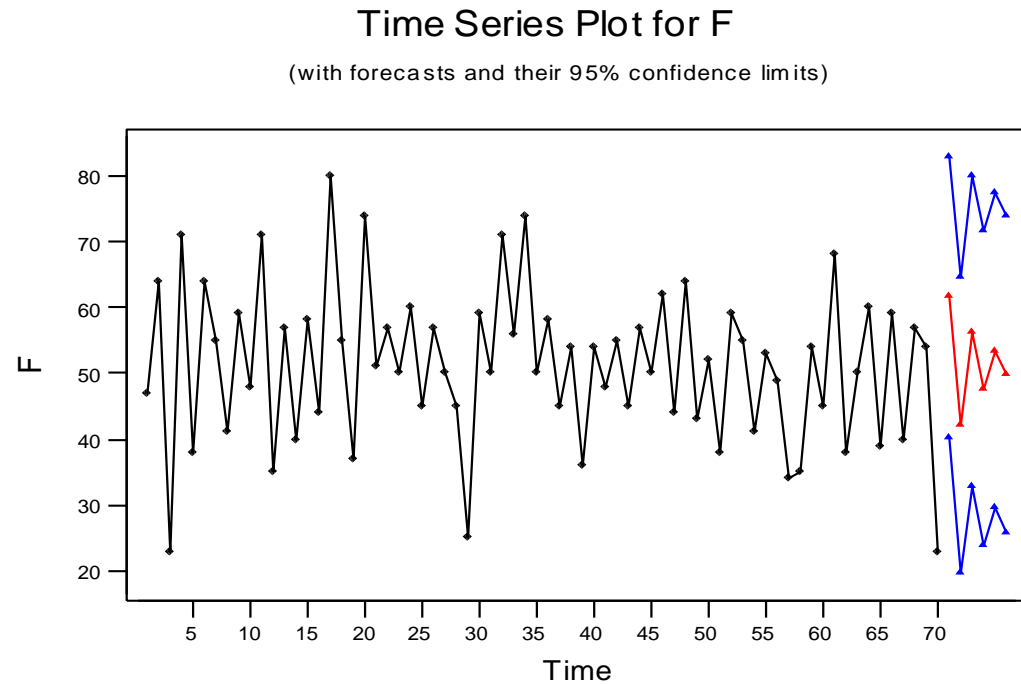
Residuals



MA(2)

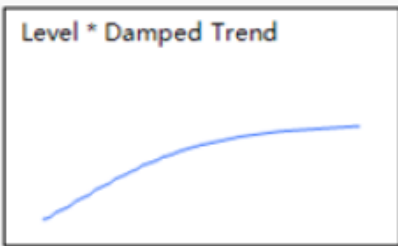
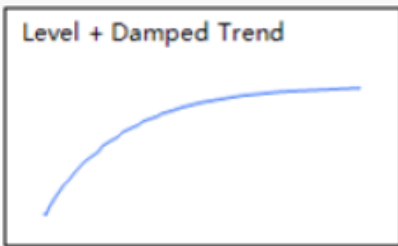
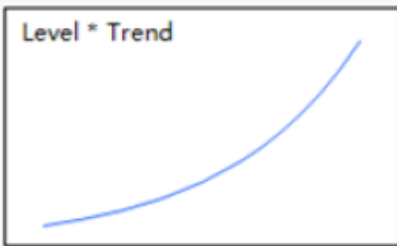
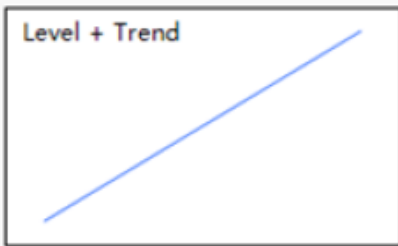
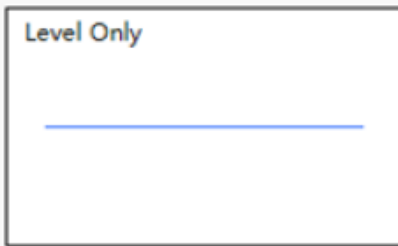
$$y_t = 51.17 + 1.06e_{t-1} - 3.31e_{t-2}$$

Forecasting 6 periods ahead

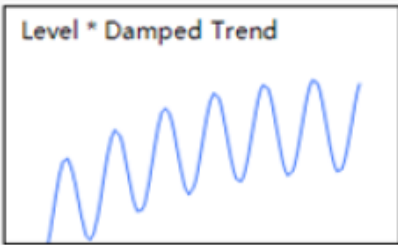
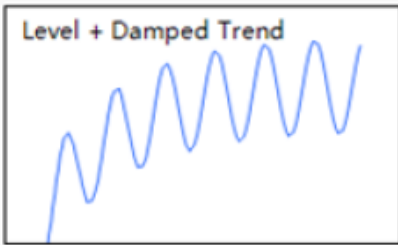
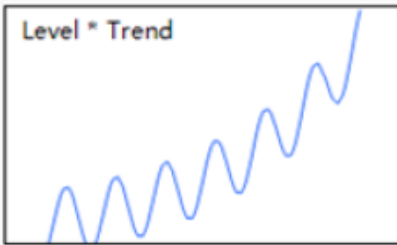
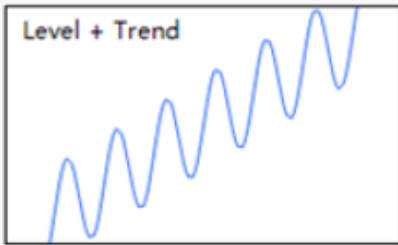
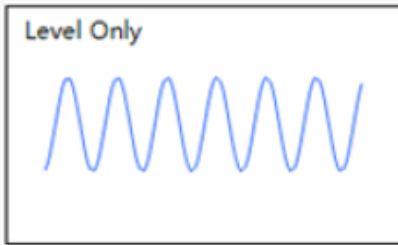


Opposite is a time series plot with forecasts up to 6 periods ahead added. You can see that the interval estimates in blue are quite wide.

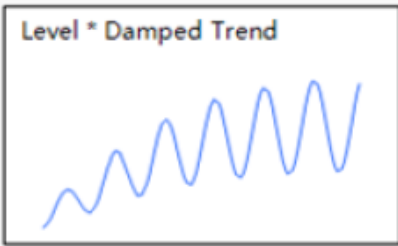
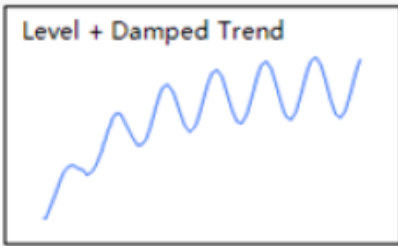
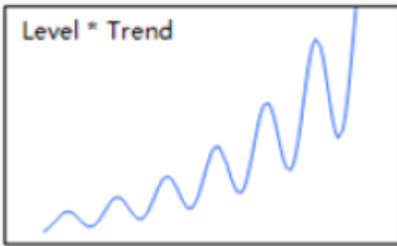
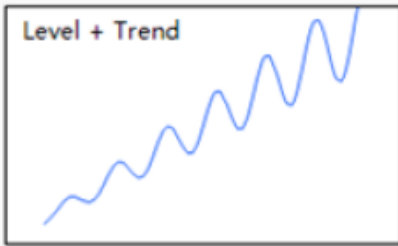
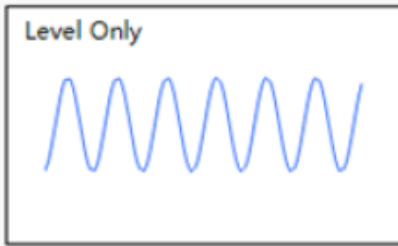
No
Season



Additive
Season



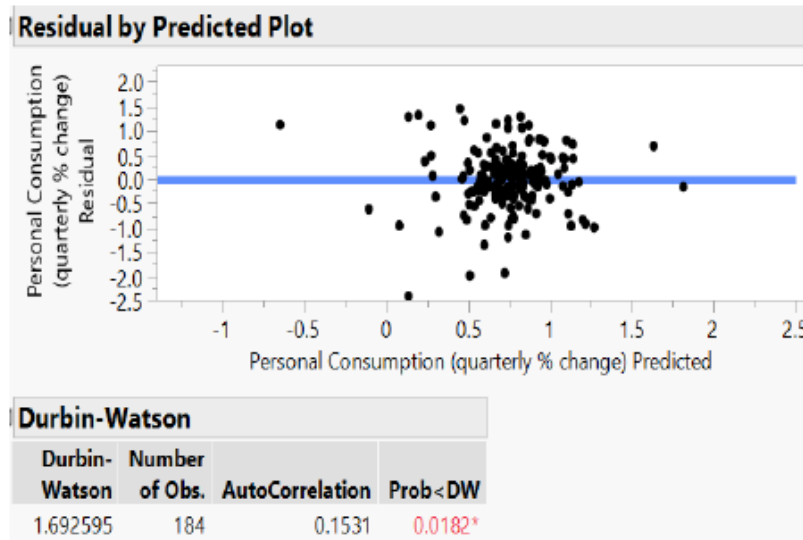
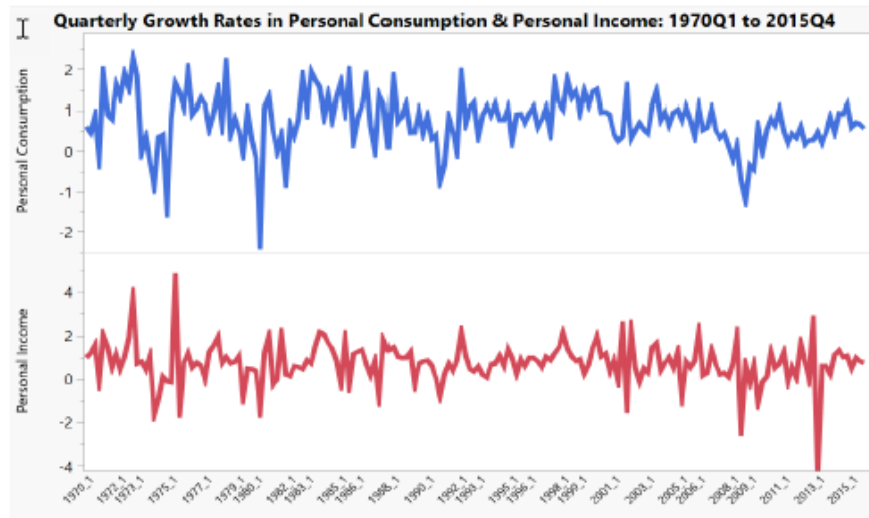
Multiplicative
Season



Transfer Functions

y_t : Quarterly US Personal Consumption Growth Rate

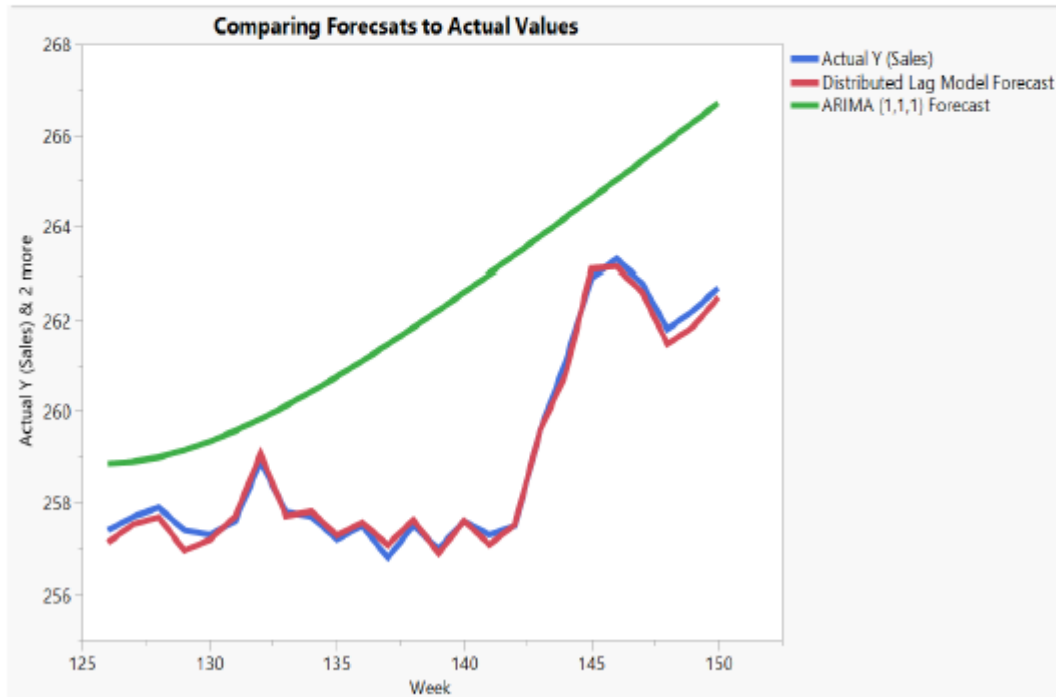
x_t : Quarterly US Personal Income Growth Rate



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

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

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

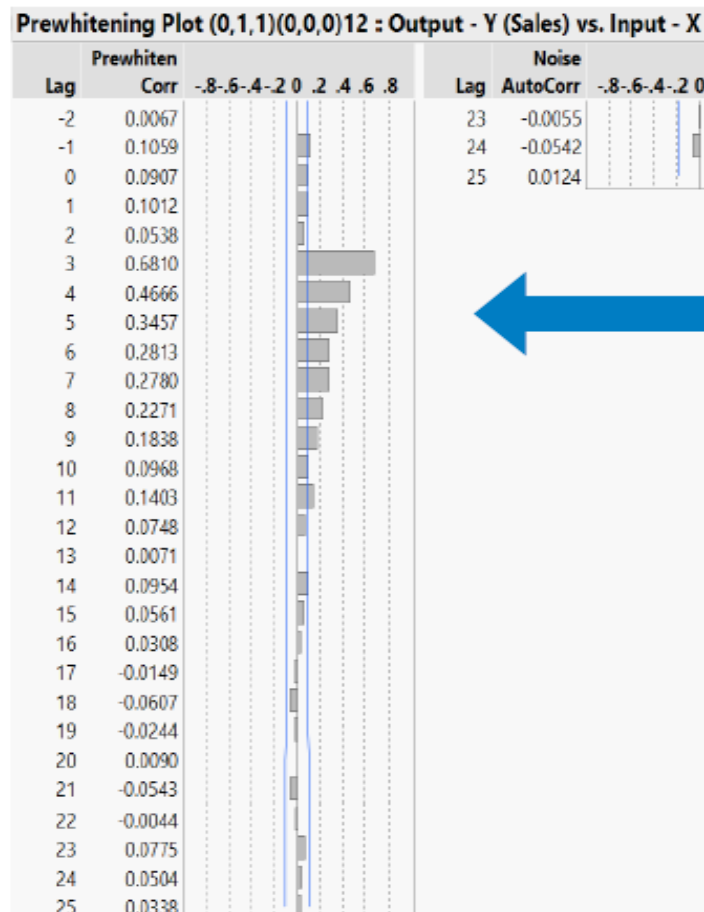
Transfer Function ARIMA Error Term



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

Prewhitening

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



Ex2: Forecasting Sales using Lagged Predictors

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

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

try the transfer function

$\frac{\omega_0}{1-\delta B} x_{t-3}$ 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/>





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

Ron S. Kenett^a  and Chris Gotwalt^b

^aThe KPA Group and the Samuel Neaman Institute, Technion, Israel; ^bJMP Statistical Discovery, LLC, Research Triangle, North Carolina, USA

ABSTRACT

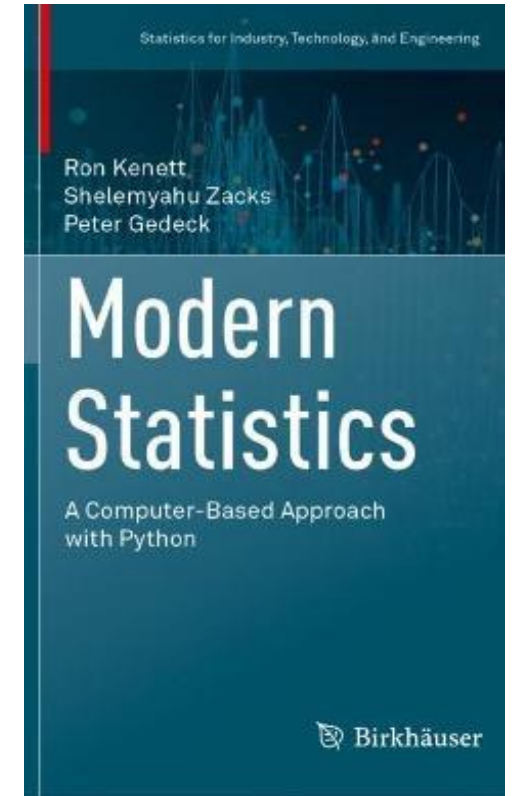
Data from measurements over time can be analyzed in different ways. In this article, we compare functional data analysis and nonlinear regression models using, among others, eight information quality dimensions. We present two case studies. The first case study introduces functional data analysis and 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



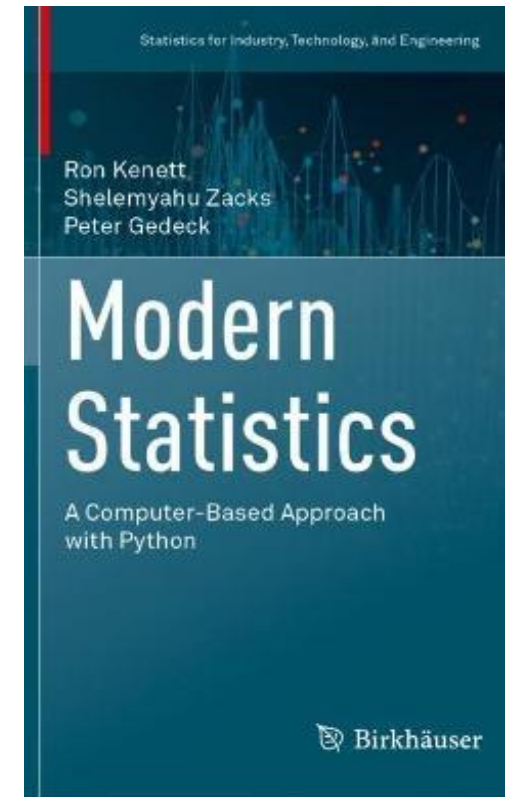
Preview Chapter 8 includes the tip of the iceberg examples with what we thought were interesting insights, not always available in standard texts. The chapter covers functional data analysis, text analytics, reinforcement learning, Bayesian networks, and causality models.

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

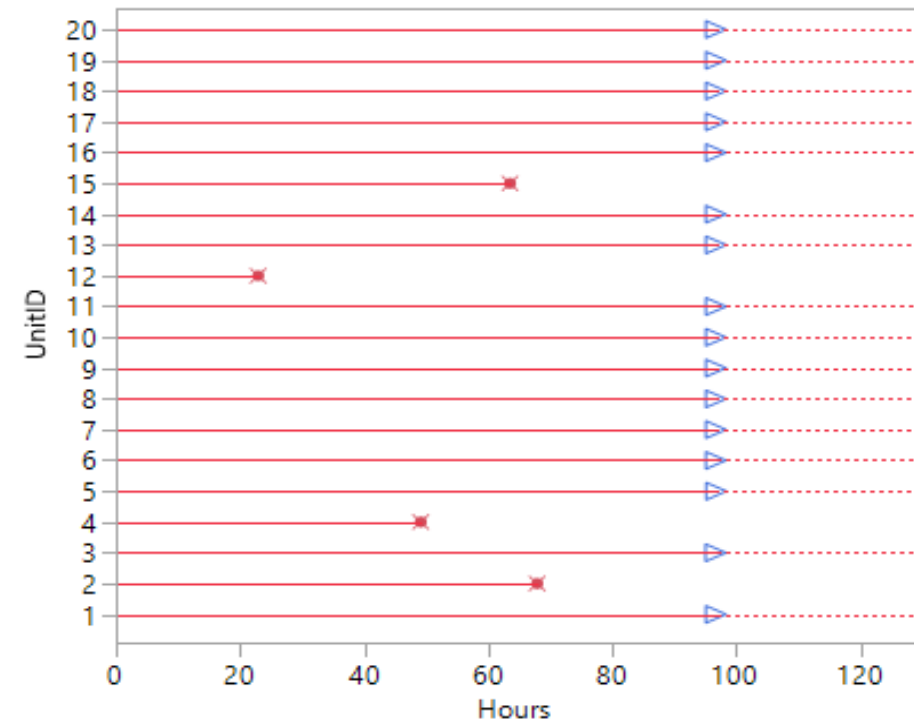
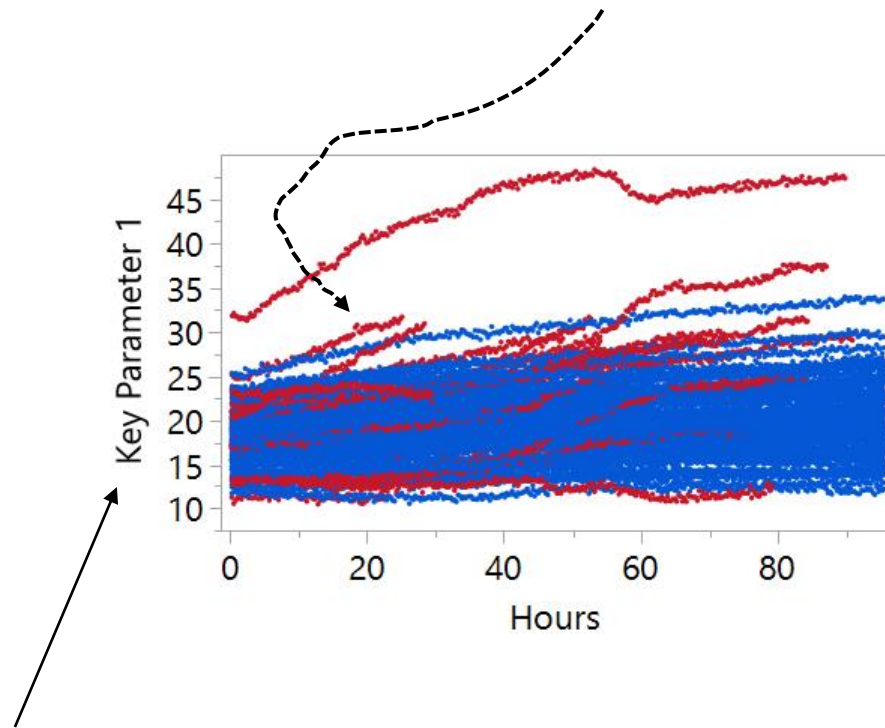
A Case Study

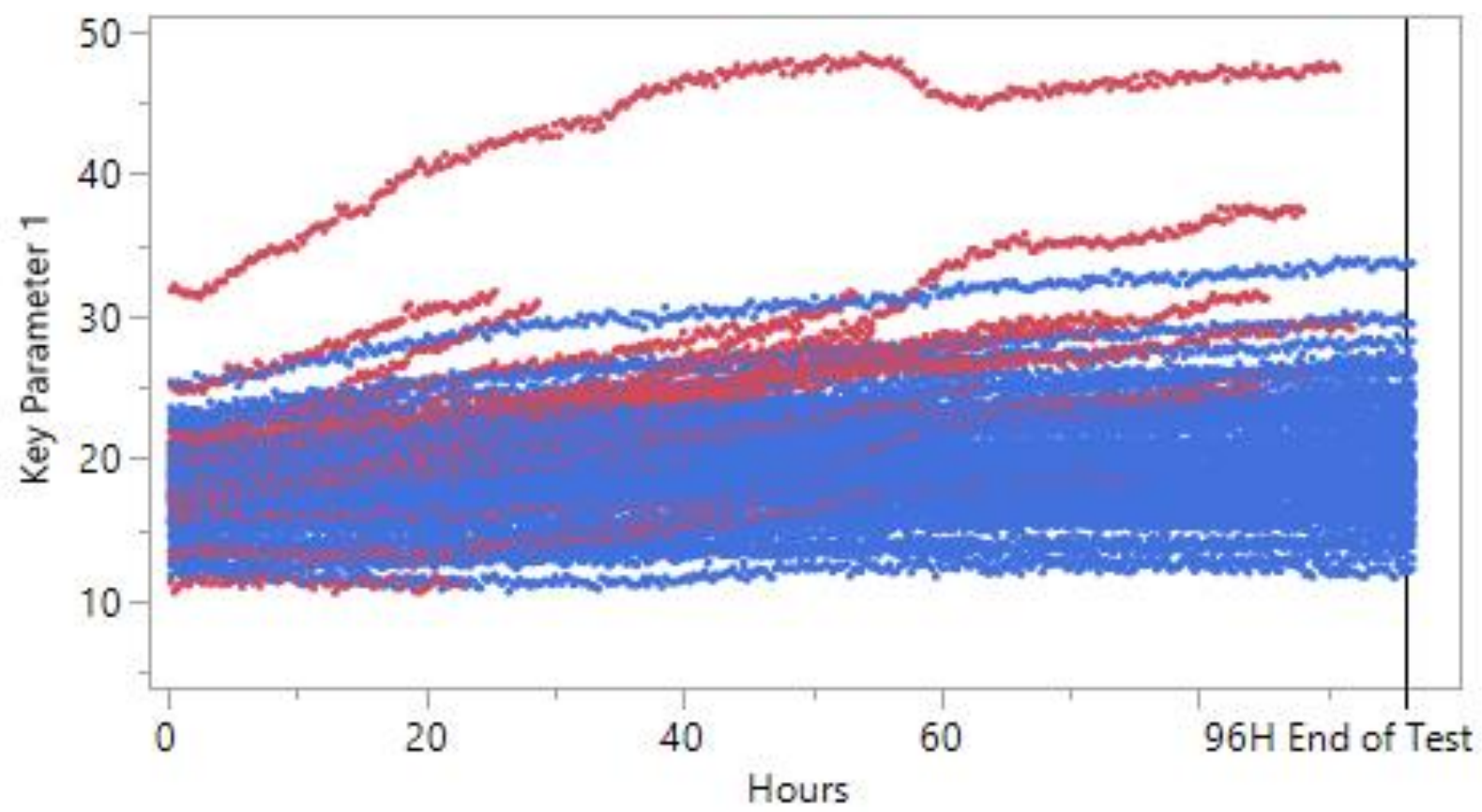


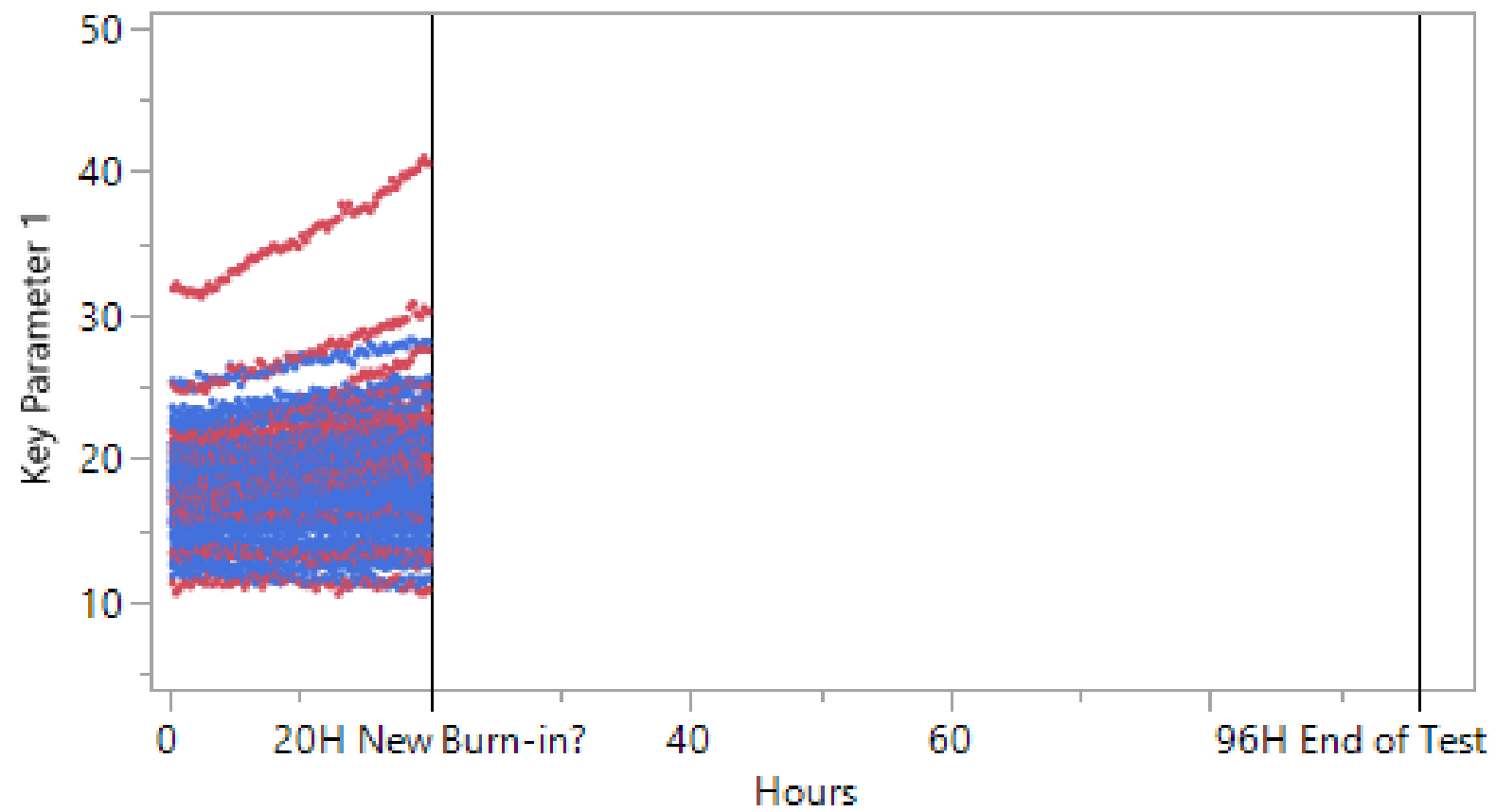
- A manufacturer of electro-mechanical devices
- 22% of units fail early
- The company screens units via burn-in under accelerated conditions
- 23 measurements are collected in real time (2 key parameters)
- Can we use these measurements to predict which units will fail in less time than the current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

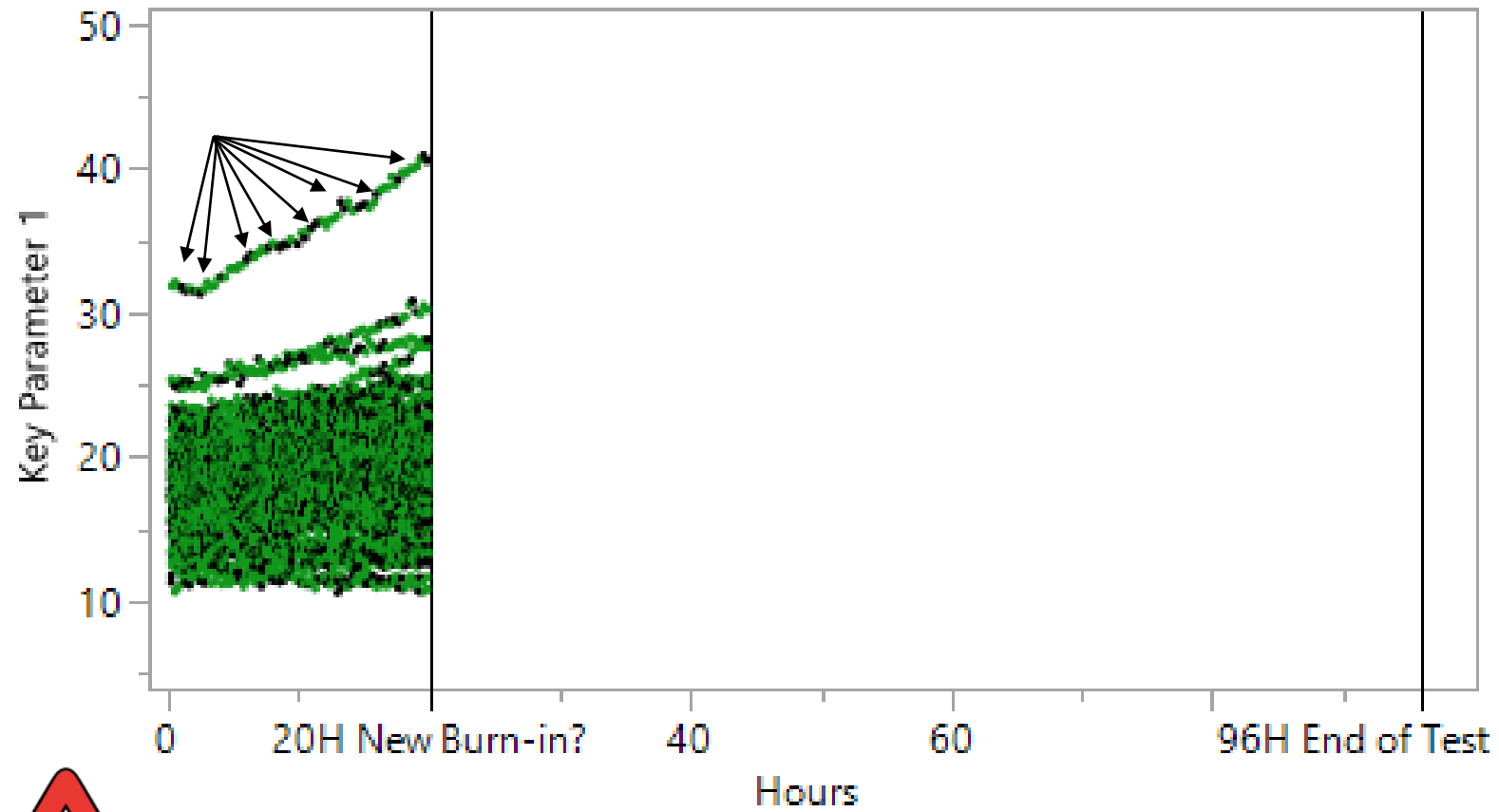
Key parameter 1

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

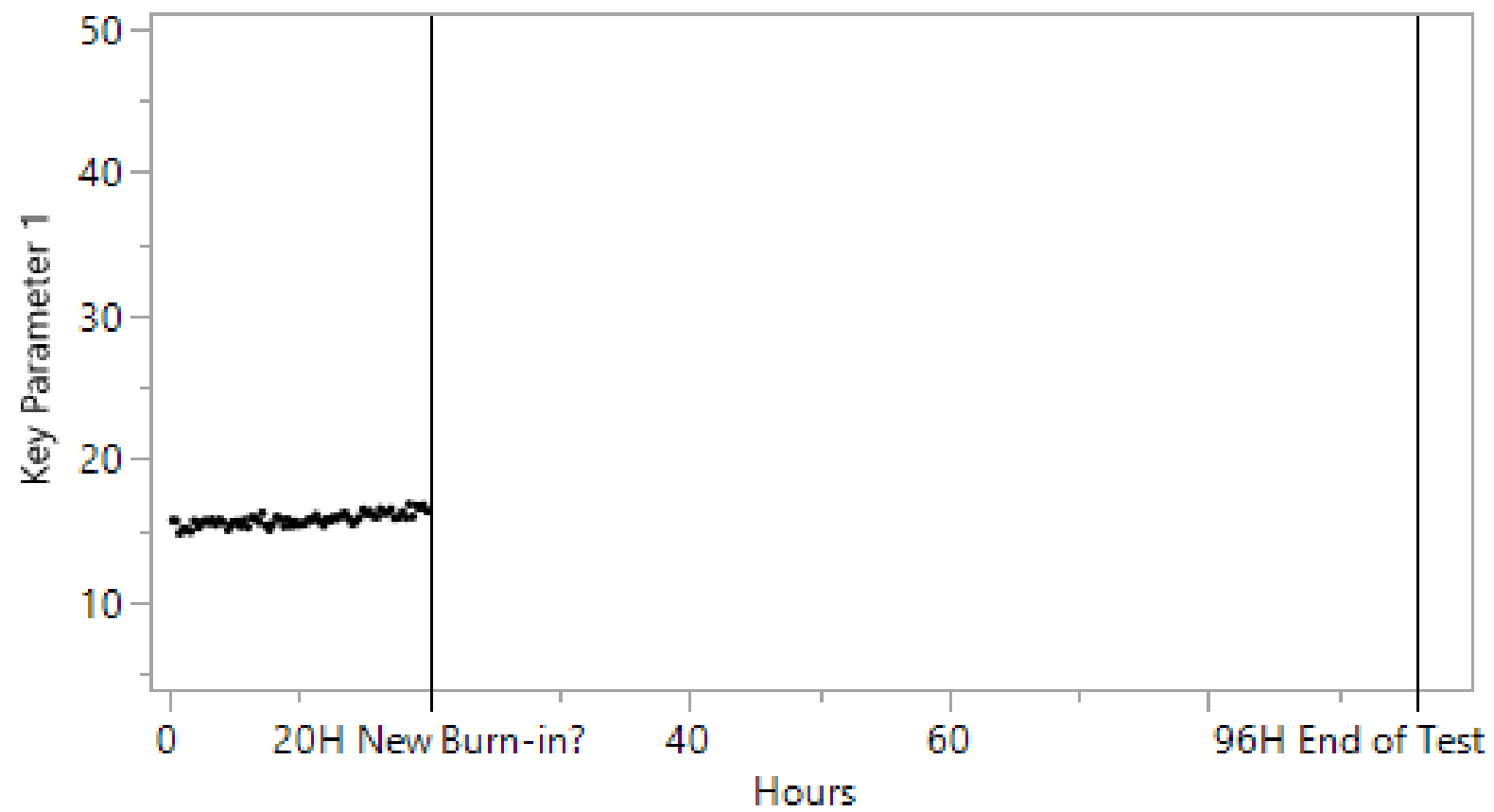


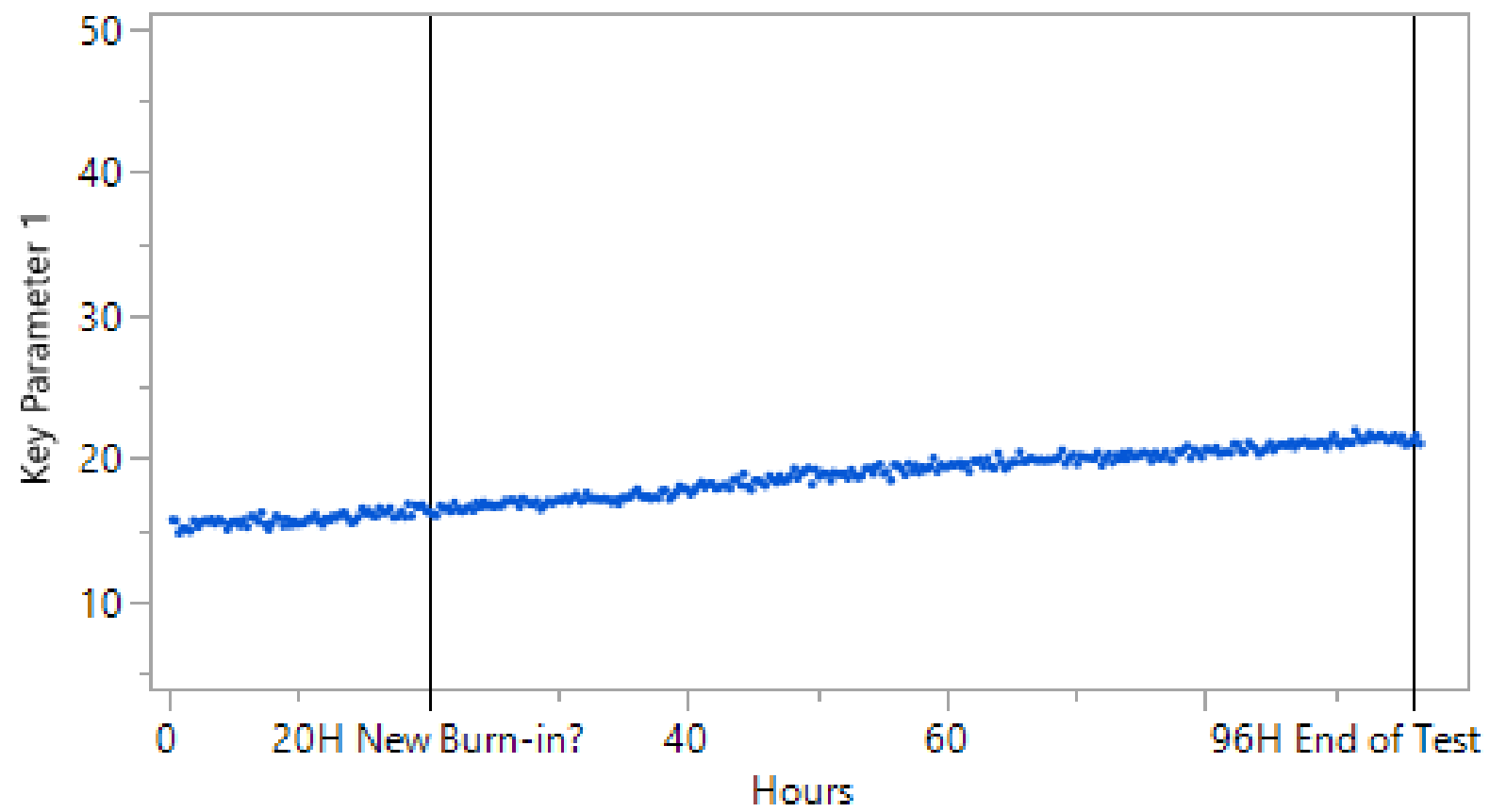


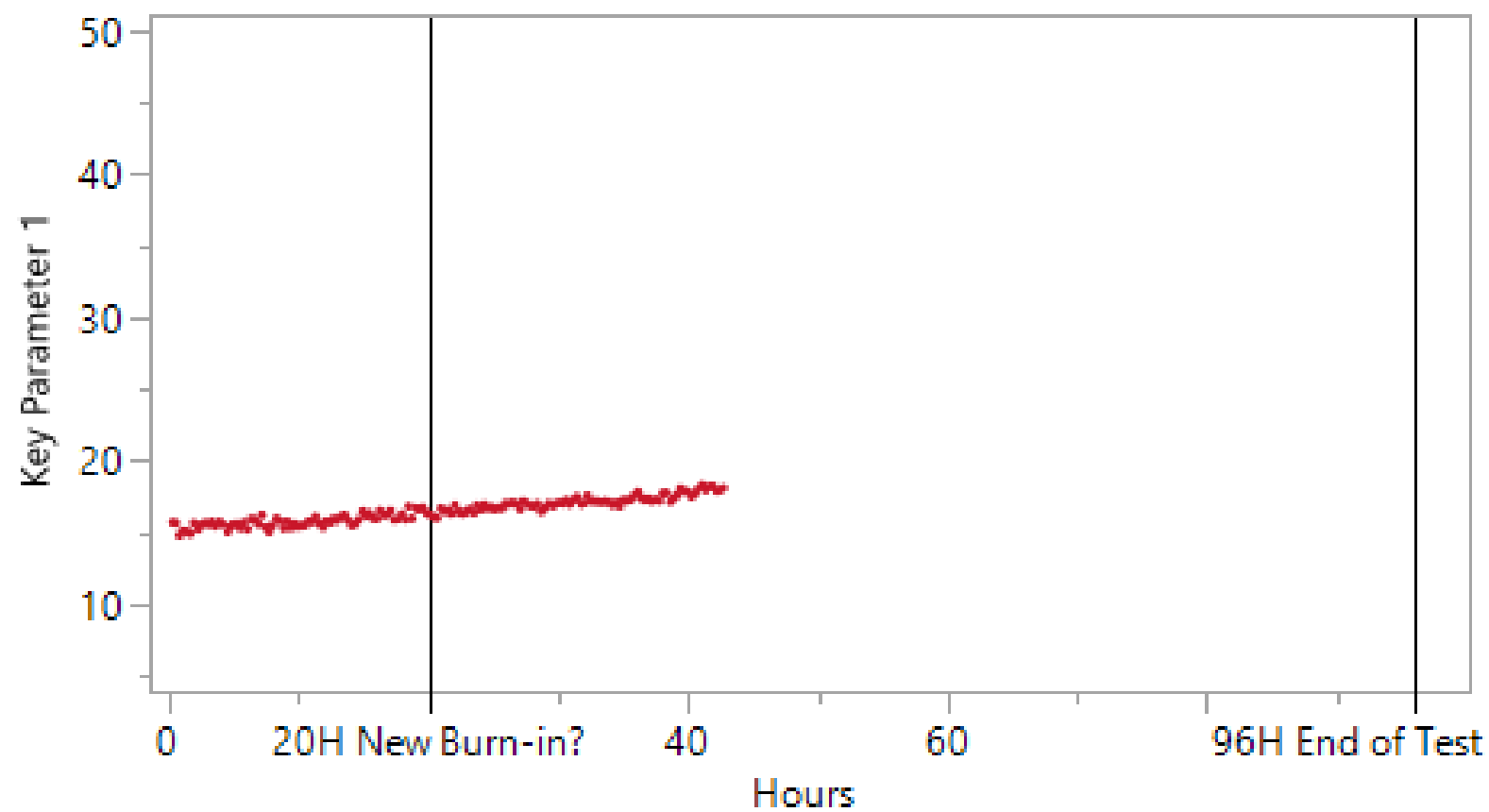




Without BCV







	Unit ID	Hours	Failed	Key Parameter 1	Key Parameter 2
1	1	0.274	0	17.62	-2.25
2	1	0.541	0	17.94	-2.24
3	1	0.819	0	18.06	-2.26
4	1	1.100	0	17.64	-2.37
5	1	1.367	0	18.33	-2.31
6	1	1.670	0	18.11	-2.32
7	1	1.920	0	18.18	-2.26
8	1	2.197	0	18.15	-2.27
9	1	2.477	0	18.14	-2.31
10	1	2.745	0	17.96	-2.30
11	1	3.021	0	18.37	-2.31
12	1	3.294	0	17.80	-2.33
13	1	3.577	0	17.71	-2.33
14	1	3.855	0	18.14	-2.33
15	1	4.167	0	18.07	-2.30
16	1	4.407	0	17.68	-2.29

Make Validation Column - JMP

Makes a column used to divide the data into training, validation, and test sets.

Select Columns

5 Columns

- UnitID
- Hours
- Failed
- Key Parameter 1
- Key Parameter 2

Selected Method

Make Validation Column

Stratify By Group Validation Column

Cast Selected Columns into Roles

Stratification Columns: Failed

Grouping Columns: UnitID

Action

OK

Cancel

Remove

Recall

Help

	Unit ID	Hours	Failed	Key Parameter 1	Key Parameter 2
1	1	0.274	0	17.62	-2.25
2	1	0.541	0	17.94	-2.24
3	1	0.819	0	18.06	-2.26
4	1	1.100	0	17.64	-2.37
5	1	1.367	0	18.33	-2.31
6	1	1.670	0	18.11	-2.32
7	1	1.920	0	18.18	-2.26
8	1	2.197	0	18.15	-2.27
9	1	2.477	0	18.14	-2.31
10	1	2.745	0	17.96	-2.30
11	1	3.021	0	18.37	-2.31
12	1	3.294	0	17.80	-2.33
13	1	3.577	0	17.71	-2.33
14	1	3.855	0	18.14	-2.33
15	1	4.167	0	18.07	-2.30
16	1	4.407	0	17.68	-2.29

Make Validation Column - JMP

Makes a column used to divide the data into training, validation, and test sets.

Select Columns

5 Columns

- UnitID
- Hours
- Failed
- Key Parameter 1
- Key Parameter 2

Cast Selected Columns into Roles

Stratification Columns: Failed

Grouping Columns: UnitID

Selected Method

Make Validation Column

Stratify By Group Validation Column

Action

OK

Cancel

Remove

Recall

Help

	Unit ID	Hours	Failed	Key Parameter 1	Key Parameter 2
1	1	0.274	0	17.62	-2.25
2	1	0.541	0	17.94	-2.24
3	1	0.819	0	18.06	-2.26
4	1	1.100	0	17.64	-2.37
5	1	1.367	0	18.33	-2.31
6	1	1.670	0	18.11	-2.32
7	1	1.920	0	18.18	-2.26
8	1	2.197	0	18.15	-2.27
9	1	2.477	0	18.14	-2.31
10	1	2.745	0	17.96	-2.30
11	1	3.021	0	18.37	-2.31
12	1	3.294	0	17.80	-2.33
13	1	3.577	0	17.71	-2.33
14	1	3.855	0	18.14	-2.33
15	1	4.167	0	18.07	-2.30
16	1	4.407	0	17.68	-2.29

Make Validation Column - JMP

Makes a column used to divide the data into training, validation, and test sets

Select Columns

5 Columns

- UnitID
- Hours
- Failed
- Key Parameter 1
- Key Parameter 2

Selected Method

Make Validation Column

Stratify By Group Validation Column

Cast Selected Columns into Role

Stratification Columns

Failed

Grouping Columns

UnitID

Action

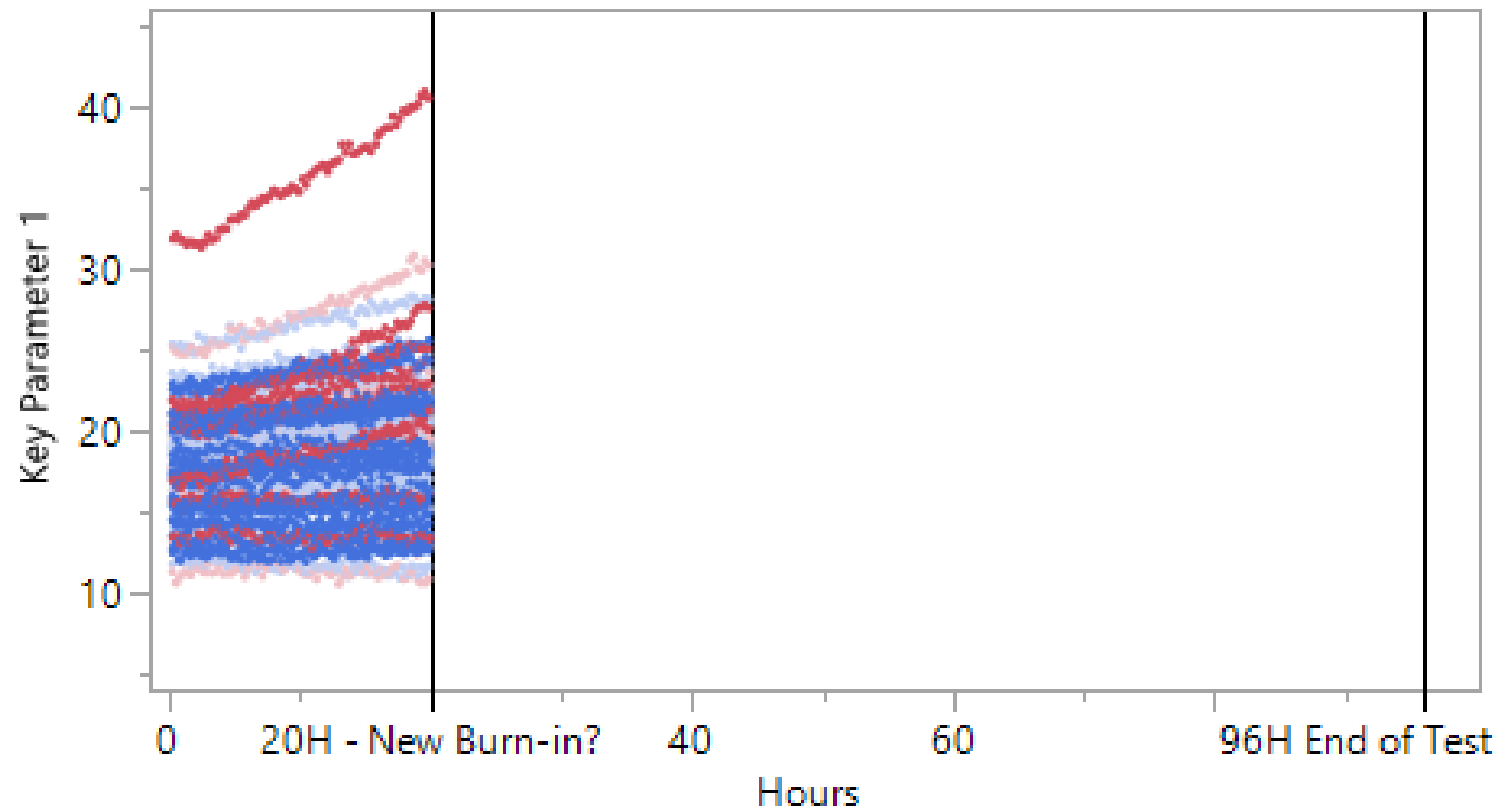
OK

Cancel

Remove

Recall

Help



With BCV

JMP FDE platform

- 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

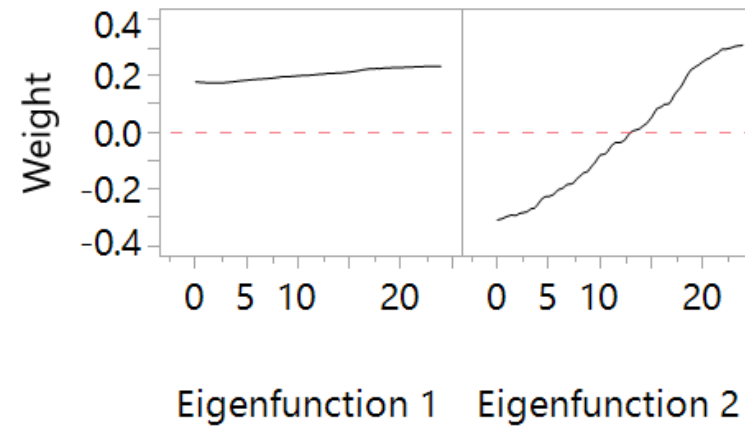

$$\hat{y}_{Val} = \left(Z_{Val}^T Z_{Val} + G(\hat{\sigma}_{Tr}^2) \right)^{-1} Z_{Val}^T (y_{Val} - X_{Val} \hat{\beta}_{Tr})$$

- $\hat{\beta}_{Tr}, \hat{\sigma}_{Tr}^2$ estimated from training data
- Use BLUP formula to score \hat{y}_{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.

Functional PCA

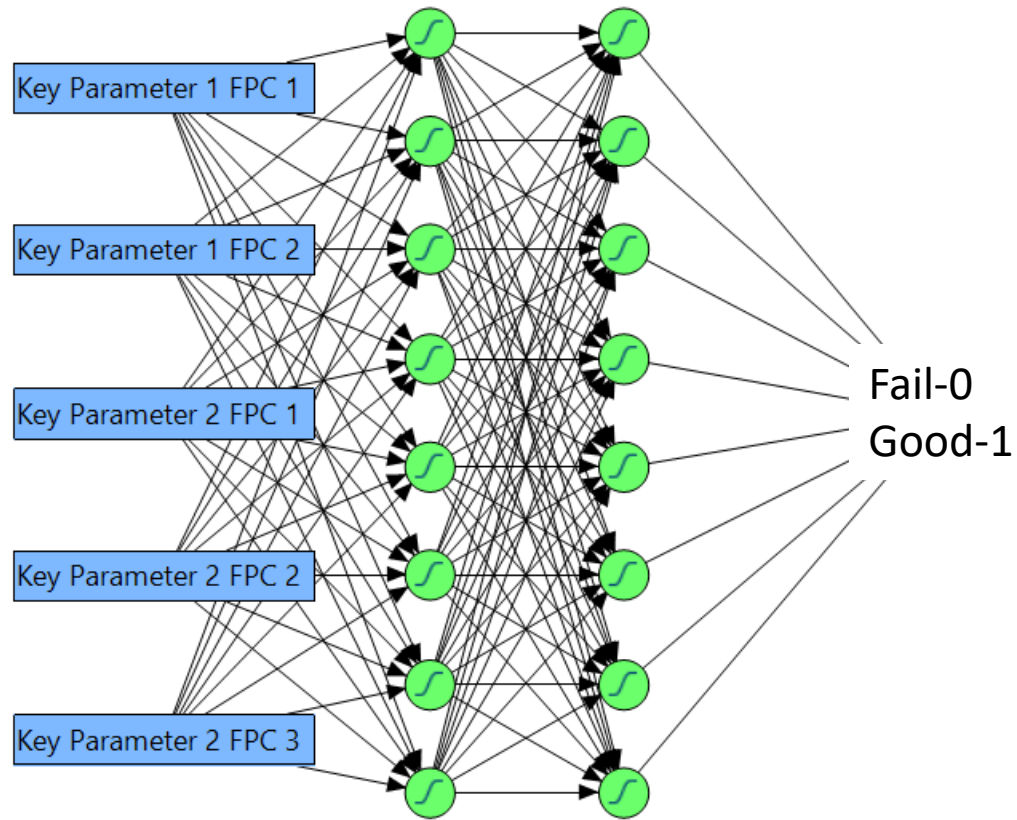
FPC	Eigenvalue	20406080	Percent	Cumulative
1	349.55		97.6%	97.6%
2	5.90		1.65%	99.3%



- Functional PCA amounts to rotating the P Spline basis functions
- FPC Scores are the induced rotation of the coefficients
- Operationally, FPC is derived from a weighted SVD of the training $\hat{\gamma}_{Tr}$

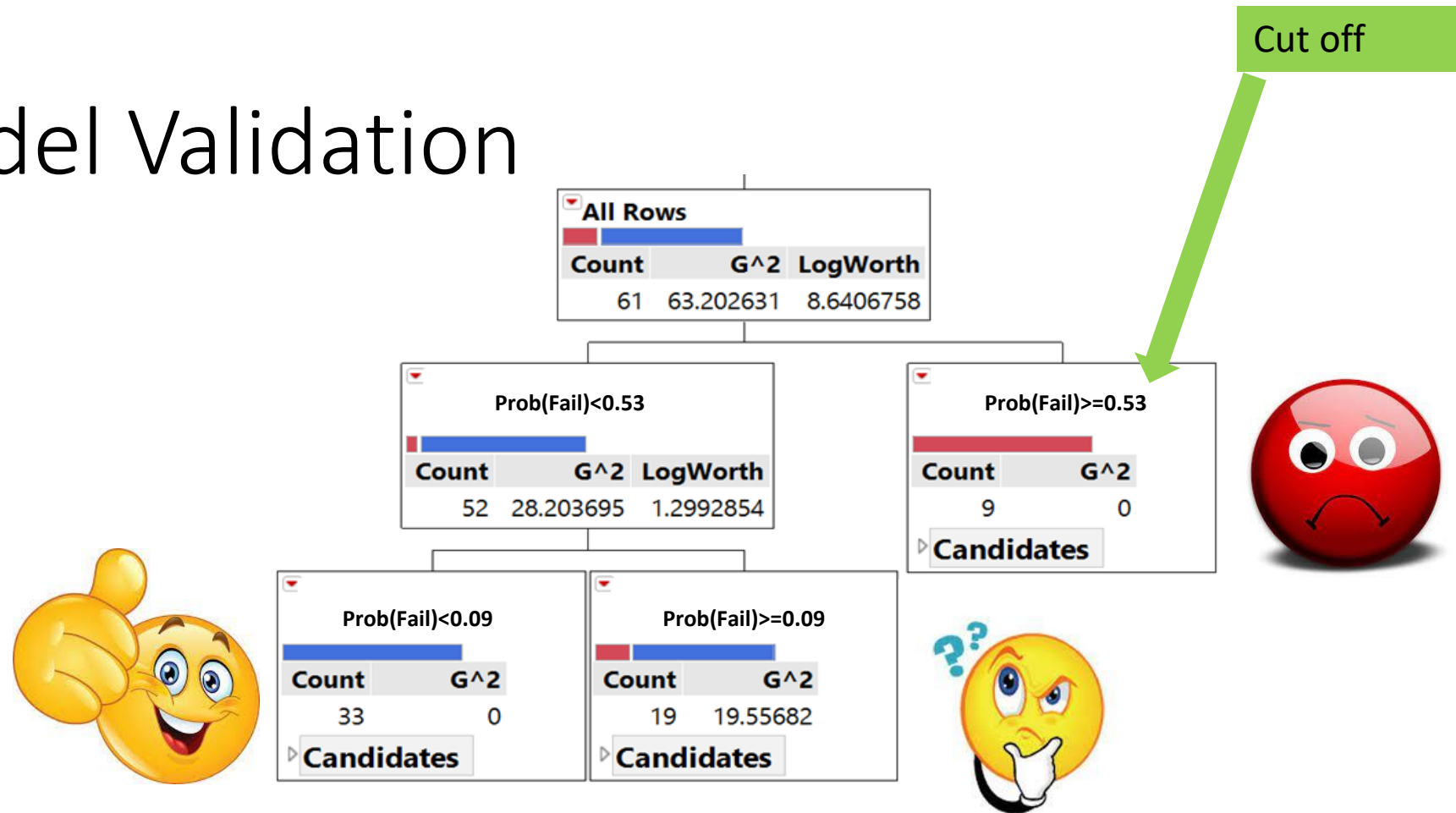
	UnitID	Validation	Failed	Key Parameter 1 FPC 1	Key Parameter 1 FPC 2
1	1	Training	0	-0.137461119	0.1518411781
2	2	Training	1	1.2865813341	0.2245665174
3	3	Training	0	1.7696429741	-0.097194514
4	4	Validation	1	2.2470498512	0.3152274003
5	5	Training	0	1.0745237905	0.0063042898
6	6	Training	0	-2.743836032	-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		Total
	0	1	
Count			
Row %			
Predicted Category BAD	9	0	9
	100.00	0.00	
Predicted Category GOOD	0	33	33
	0.00	100.00	
Predicted Category MAYBE GOOD	4	15	19
	21.05	78.95	
Total	13	48	61

Training

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

Validation

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

Multiple Time Series

