MATH 2780 Chapter 10A Worksheet

Summary

Repeated observations on a variable y_t at times $t=1,\ldots,n$ produce a **time series**. This time series is decomposed into a long-term secular trend (T_t), a cyclical business cycle wavelike fluctuation (C_t), a seasonal variation (S_t), and a residual effect (R_t), the additive regression model is $y_t = T_t + C_t + S_t + R_t$.
Residuals one time period apart (at times t and $t+1$) are correlated is called **first-order autocorrelation**.
Model is $y_t=E(y_t)+R_t$, $E(y_t)=\beta_0+\beta_1x_1+\ldots+\beta_kx_k$, residual R_t has $E(R_t)=0$ and $\mathrm{Var}(R_t)=\sigma^2$, but autocorrelated.
Quarterly model $E(y)= \beta_0+\beta_1t+\beta_1Q_1+\beta_1Q_2+\beta_1Q_3$, $Q_1=1$ if quarter 1, $Q_2=1$ if quarter 2, $Q_3=1$, if quarter 3.
Coefficient and Residual Variance Estimation: The ordinary least squares regression coefficients.

$Y = X\beta + E$ $\hat{\beta} = (X'X)^{-1}X'y$ $s^2 = \dfrac{(y-X\hat{\beta})'(y-X\hat{\beta})}{n-k-1}$ $MSE = s^2,\ s=\sqrt{s^2}$	$Y=\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}, X=\begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{kn} \\ 1 & x_{12} & x_{22} & \cdots & x_{k2} \\ 1 & x_{13} & x_{23} & \cdots & x_{k3} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{1n} & x_{2n} & \cdots & x_{kn} \end{bmatrix}, \beta=\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix}, E=\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{bmatrix}$

Regression Residuals: Residuals are $\hat{\varepsilon}_i = y_i - \hat{\beta}_0 - \hat{\beta}_1x_1 - \ldots - \hat{\beta}_kx_k$, $s^2 = \sum (y_i - \hat{y}_i)^2/(n-k-1)$.

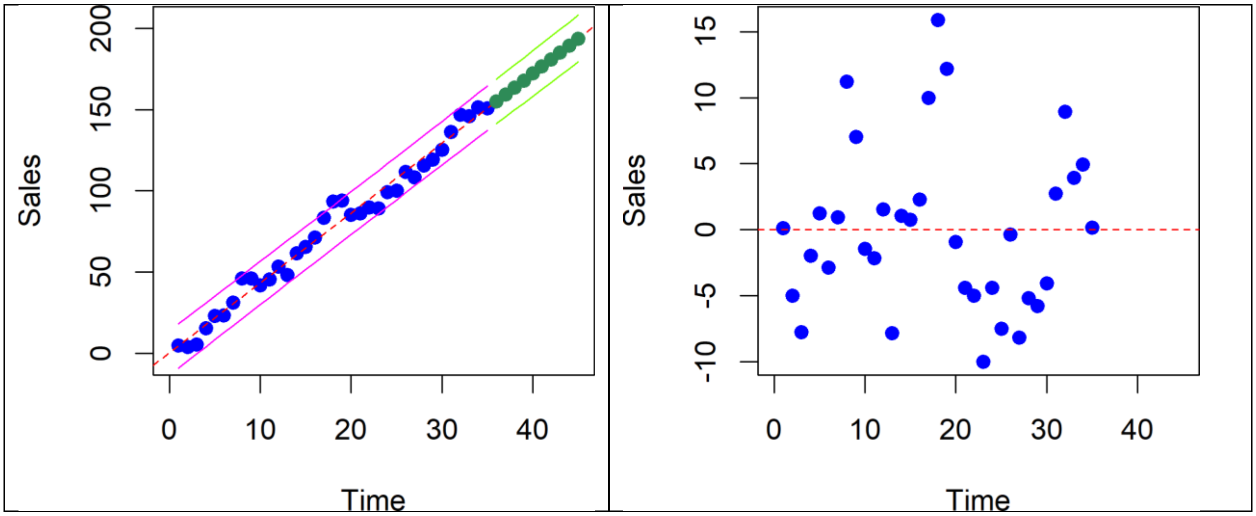
All of the same methods we learned to work with residuals continue to apply.
First-order autoregressive error model: $m=1$, $R_t=\phi R_{t+m}+\varepsilon_t$, $-1<\phi<1$, $AC(R_t,R_{t+m})=\phi^m$.
Forecast limits using AR(1) error model: $\hat{y}_{n+m} \pm 1.96\sqrt{MSE(1+\phi^2+\cdots+\phi^{2(m-1)})}$

Detecting Residual Correlation: The **Durbin-Watson Test**.

$d=\sum_{t=2}^{n}(\hat{\varepsilon}_t-\hat{\varepsilon}_{t-1})^2 / \sum_{t=1}^{n}\hat{\varepsilon}_t^2 \approx \underbrace{2(1-\hat{\phi})}_{\text{Large } n}$	H_0: $\phi\le 0$ vs. H_a: $\phi>0$, Reject $d<d_{L,\alpha}$ H_0: $\phi\ge 0$ vs. H_a: $\phi<0$, Reject $(4-d)<d_{L,\alpha}$ H_0: $\phi=0$ vs. H_a: $\phi\ne 0$, Reject $d<d_{L,\alpha/2}$ or $(4-d)<d_{L,\alpha/2}$

1. Range of d: $0 \le d \le 4$.
2. If residuals are uncorrelated, $d \approx 2$.
3. If residuals are positively correlated, $d < 2$, and if the correlation is very strong, $d\approx 0$.
4. If residuals are negatively correlated, $d > 2$, and if the correlation is very strong, $d\approx 4$.

Example: Data on annual sales *y* for 35 years. The model is $E(y) = \beta_0 + \beta_1 t$. y=annual sales, t=time



Run R code and examine results.
SALES35
read data
parse out variables
fit line model and get statistics
scatter plot with line
residual plot with 0 line
#view model fit regression coefficients
#view model fit ANOVA table
view model s, Rsq and adjRsq
fit and intervals
mean function at x0
scatter plot with line
mean function confidence interval at x0
prediction interval
scatter plot with line and 95% PI
forecasted prediction interval
scatter plot with line and 95% PI

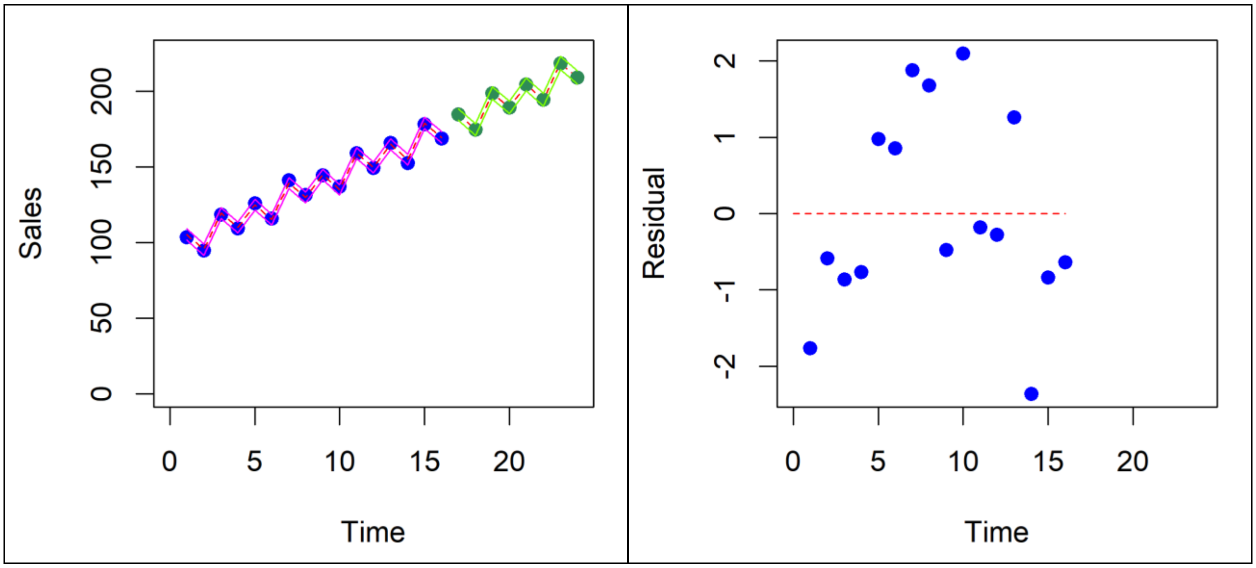
T	SALES	RESIDUAL
1	4.8	0.102857
2	4	-4.99277
3	5.5	-7.7884
4	15.6	-1.98403
5	23.1	1.220336
6	23.3	-2.87529
7	31.4	0.929076
8	46	11.23345
9	46.1	7.037815
10	41.9	-1.45782
11	45.5	-2.15345
12	53.5	1.550924
13	48.4	-7.84471
14	61.6	1.059664
15	65.6	0.764034
16	71.4	2.268403
17	83.4	9.972773
18	93.6	15.87714
19	94.2	12.18151
20	85.4	-0.91412
21	86.2	-4.40975
22	89.9	-5.00538
23	89.2	-10.001
24	99.1	-4.39664
25	100.3	-7.49227
26	111.7	-0.3879
27	108.2	-8.18353
28	115.5	-5.17916
29	119.2	-5.77479
30	125.2	-4.07042
31	136.3	2.73395
32	146.8	8.938319
33	146.1	3.942689
34	151.4	4.947059
35	150.9	0.151429

```
# SALES35
alph <- 0.05;
nf   <-10 # number to forecast

# read data
mydata <- read.de-
lim("SALES35.txt",header=TRUE,sep="",dec=".")

# parse out variables
n <- nrow(mydata)
k <- ncol(mydata)-2
t <- c(mydata[,1]) #T time
y <- c(mydata[,2]) #y sales
r <- c(mydata[,3]) #R residual
tcrit<-qt(1-alph/2,n-k-1)

df      <- data.frame(cbind(t,y,r))
names(df) <- c("t","y","r")

# fit line model and get statistics
mymodel <- lm(y~t)

# scatter plot with line
plot(t,y,pch=19,col="blue",xlab='Time',ylab='Sales',xlim=c(0,n+
nf),
ylim=c(-5,205))
abline(mymodel,lty=2,col='red')

# residual plot with 0 line
plot(t,r,pch=19,col="blue",xlab='Time',ylab='Sales',
xlim=c(0,n+nf))
abline(lm(r~t),r,lty=2,col='red')

#view model fit regression coefficients
summary(mymodel)$coefficients[,]

#view model fit ANOVA table
temp<-anova(mymodel)
out <- temp
m <- nrow(temp)
out$Df <- with(temp,c(sum(Df[1:(m-1)]),Df[m],rep(NA_real_,
m-2)))
out$`Sum Sq` <- with(temp,c(sum(`Sum Sq`[1:(m-1)]),
                     `Sum Sq`[m],rep(NA_real_,m-2)))
out$`Mean Sq` <- with(out,out$`Sum Sq`/out$Df)
out$`F value` <- c(out$`Mean Sq`[1]/out$`Mean
Sq`[2],rep(NA_real_,m-1))
out$`Pr(>F)` <- c(pf(out$`F value`[1],out$Df[1],out$Df[2],
                lower.tail = FALSE),rep(NA_real_,m-1))
out <- out[1:2,]
rownames(out) <- c("Model","Residuals")
out
```

```
# view model s, Rsq and adjRsq
print('s,R-squared,adj R-squared')
c(summary(mymodel)$s,sum-
mary(mymodel)$r.squared,
 summary(mymodel)$adj.r.squared)

# fit and intervals
c  <-rep(1,n)     #Ones
X  <-matrix(cbind(c,t),nrow=n,ncol=2) #design matrix
W  <-solve(t(X)%*%X)
b  <-W%*%t(X)%*%y
yhat<-X%*%b

# mean function at x0
xnew  <-seq(n+1,n+nf,by=1)
X0   <-cbind(rep(1,nf),xnew)
yhatx0<-X0%*%b
fcast <-data.frame(cbind(xnew,yhatx0))
names(fcast) <- c("xnew","yhat")

# scatter plot with line
plot(t,y,pch=19,col="blue",xlab='Time',ylab='Sales',
    xlim=c(0,n+nf),ylim=c(-5,205))
abline(lm(y~t),lty=2,col='red')
points(fcast$xnew,fcast$yhat,pch=19,col='seagreen')

# mean function confidence interval at x0
xbar<-mean(t)
SSxx<-sum(t^2)-(sum(t)^2/n)
s<- summary(mymodel)$sigma

# prediction interval
XCI<-cbind(rep(1,n),t)
tXCI <- matrix(t(XCI),2,n)
yCIx0<-XCI%*%b

yCIL <- matrix(rep(0,n),nrow=n,ncol=1) # preallocate
yCIU <- matrix(rep(0,n),nrow=n,ncol=1) # preallocate
yPIL <- matrix(rep(0,n),nrow=n,ncol=1) # preallocate
yPIU <- matrix(rep(0,n),nrow=n,ncol=1) # preallocate
counter<-0
for (i in (1):(n)){
 counter<-counter+1
 yCIL[counter]<-yCIx0[counter]-tcrit*s*sqrt(
1/n+(t[counter]-xbar)^2/SSxx)
 yCIU[counter]<-yCIx0[counter]+tcrit*s*sqrt(
1/n+(t[counter]-xbar)^2/SSxx)
 yPIL[counter]<-yCIx0[counter]-
tcrit*s*sqrt(1+1/n+(t[counter]-xbar)^2/SSxx)
 yPIU[counter]<-yCIx0[coun-
ter]+tcrit*s*sqrt(1+1/n+(t[counter]-xbar)^2/SSxx)
}
withinPI<-cbind(yhat,yPIL,yPIU)
withinPI
```

```
# scatter plot with line and 95% PI
plot(t,y,pch=19,col="blue",xlab='Time',ylab='Sales',
    xlim=c(0,n+nf),ylim=c(-5,205))
abline(lm(y~t),r,lty=2,col='red')
points(fcast$xnew,fcast$yhat,pch=19,col='seagreen')
#lines(t,yCIL, type = "l", col = "deepskyblue3")
#lines(t,yCIU, type = "l", col = "deepskyblue3")
lines(t,yPIL, type = "l", col = "magenta")
lines(t,yPIU, type = "l", col = "magenta")

# forecasted prediction interval
tF <- seq(n+1,n+nf)
XCIF<-cbind(rep(1,nf),tF)
tXCIF <- matrix(t(XCIF),2,nf)
yCIx0F<-XCIF%*%b
yCILF <- matrix(rep(0,nf),nrow=nf,ncol=1) # preallocate
yCIUF <- matrix(rep(0,nf),nrow=nf,ncol=1) # preallocate
yPILF <- matrix(rep(0,nf),nrow=nf,ncol=1) # preallocate
yPIUF <- matrix(rep(0,nf),nrow=nf,ncol=1) # preallocate
counter<-0
for (i in (1):(nf)){
  counter<-counter+1
  yCILF[counter]<-yCIx0F[counter]-tcrit*s*sqrt(  1/n+(tF[counter]-xbar)^2/SSxx)
  yCIUF[counter]<-yCIx0F[counter]+tcrit*s*sqrt(  1/n+(tF[counter]-xbar)^2/SSxx)
  yPILF[counter]<-yCIx0F[counter]-tcrit*s*sqrt(1+1/n+(tF[counter]-xbar)^2/SSxx)
  yPIUF[counter]<-yCIx0F[counter]+tcrit*s*sqrt(1+1/n+(tF[counter]-xbar)^2/SSxx)
}
outsidePI<- cbind(yhatx0,yPILF,yPIUF)
outsidePI

# scatter plot with line and 95% PI
plot(t,y,pch=19,col="blue",xlab='Time',ylab='Sales',
    xlim=c(0,n+nf),ylim=c(-5,205))
abline(lm(y~t),r,lty=2,col='red')
points(fcast$xnew,fcast$yhat,pch=19,col='seagreen')
#lines(t,yCIL, type = "l", col = "deepskyblue3")
#lines(t,yCIU, type = "l", col = "deepskyblue3")
lines(t,yPIL, type = "l", col = "magenta")
lines(t,yPIU, type = "l", col = "magenta")
lines(tF,yPILF, type = "l", col = "chartreuse1")
lines(tF,yPIUF, type = "l", col = "chartreuse1")

results<-data.frame(rbind(withinPI,outsidePI))
names(results) <- c("yhat","PI_L","PI_U")

write.csv(round(results,digits=3),file="forecasteandpiSales35.csv")
```

Example: The model is $E(y) = \beta_0 + \beta_1 t + \beta_1 Q_1 + \beta_1 Q_2 + \beta_1 Q_3$.



YEAR	T	POWLOAD
2015	1	103.5
2015	2	94.7
2015	3	118.6
2015	4	109.3
2016	5	126.1
2016	6	116
2016	7	141.2
2016	8	131.6
2017	9	144.5
2017	10	137.1
2017	11	159
2017	12	149.5
2018	13	166.1
2018	14	152.5
2018	15	178.2
2018	16	169
2019	17	*

t	yhat	PI_L	PI_U
1	105.264	101.526	109.001
2	95.289	91.614	98.963
3	119.464	115.844	123.083
4	110.064	106.491	113.636
5	125.121	121.586	128.656
6	115.146	111.64	118.653
7	139.321	135.834	142.809
8	129.921	126.443	133.399
9	144.979	141.501	148.457
10	135.004	131.516	138.491
11	159.179	155.672	162.685
12	149.779	146.244	153.314
13	164.836	161.264	168.409
14	154.861	151.242	158.481
15	179.036	175.362	182.711
16	169.636	165.899	173.374
17	184.694	180.885	188.502
18	174.719	170.832	178.605
19	198.894	194.922	202.866
20	189.494	185.43	193.557

MATH 2780 Chapter 10A Worksheet

```
# QTRPOWER #
alph <- 0.05;
nf   <-8 # number to forecast
# read data
mydata <- read.delim("QTRPOWER.txt",
header=TRUE, sep="",dec=".")

# parse out variables
n <- nrow(mydata[1:16,])
k <- 4
y    <- c(mydata[1:16,3]) #y power
year <- c(mydata[1:16,1]) #year
t    <- c(mydata[1:16,2]) #t time
Q1 <- c(1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0)
Q2 <- c(0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0)
Q3 <- c(0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0)
tcrit<-qt(1-alph/2,n-k-1)
df       <- data.frame(ma-
trix(cbind(y,t,Q1,Q2,Q3),n,5))
names(df) <- c("y","t","Q1","Q2","Q3")
# fit line model and get statistics
mymodel <- lm(y~t+Q1+Q2+Q3)
b<-matrix(summary(mymodel)$coeffi-
cients[,1],k+1,1)
#view model fit regression coefficients
summary(mymodel)$coefficients[,]
# scatter plot with line
c   <-rep(1,n)      #Ones
X   <-cbind(c,t,Q1,Q2,Q3) #design matrix
plot(t,y,pch=19,col="blue",xlab='Time',ylab='Power',
xlim=c(0,n+nf),ylim=c(0,225))
lines(t,X%*%b,type="l",lty=2,col='red')
# residual plot with 0 line
r<-mymodel$residuals
plot(t,mymodel$residu-
als,pch=19,col="blue",xlab='Time',
ylab='Residual',xlim=c(0,n+nf))
lines(c(0,n),c(0,0),type="l",lty=2,col='red')
#view model fit ANOVA table
temp<-anova(mymodel)
out <- temp
m <- nrow(temp)
out$Df <- with(temp,c(sum(Df[1:(m-
1)]),Df[m],rep(NA_real_,m-2)))
out$`Sum Sq` <- with(temp,c(sum(`Sum Sq`[1:(m-
1)]),
                  `Sum Sq`[m],rep(NA_real_,m-2)))
out$`Mean Sq` <- with(out,out$`Sum Sq`/out$Df)
out$`F value` <- c(out$`Mean Sq`[1]/out$`Mean
Sq`[2],rep(NA_real_,m-1))
out$`Pr(>F)` <- c(pf(out$`F
value`[1],out$Df[1],out$Df[2],lower.tail = FALSE),
rep(NA_real_,m-1))
out <- out[1:2,]
rownames(out) <- c("Model","Residuals")
out
```

```
# view model s, Rsq and adjRsq
print('s,R-squared,adj R-squared')
c(summary(mymodel)$s,summary(mymodel)$r.squared,
  summary(mymodel)$adj.r.squared)

# fit and intervals
yhat<-X%*%b

# mean function at x0
Q1 <- c(1,0,0,0)
Q2 <- c(0,1,0,0)
Q3 <- c(0,0,1,0)
xnew  <-cbind(seq(n+1,n+nf,by=1),Q1,Q2,Q3)
X0    <-cbind(rep(1,nf),xnew)
yhatx0<-X0%*%b
fcast <-data.frame(cbind(xnew,yhatx0))
names(fcast) <- c("xnew","yhat")

# scatter plot with line
plot(t,y,pch=19,col="blue",xlab='Time',ylab='Sales',
   xlim=c(0,n+nf),ylim=c(0,225))
lines(t,X%*%b,type="l",lty=2,col='red')
lines(seq(n+1,n+nf,by=1),yhatx0,type="l",lty=2,col='red')
points(seq(n+1,n+nf,by=1),yhatx0,pch=19,col='seagreen')

# mean function confidence interval at x0
xbar<-mean(t)
SSxx<-sum(t^2)-(sum(t)^2/n)
s<- summary(mymodel)$sigma

# prediction interval
XCI  <- X
tXCI <- matrix(t(XCI),5,n)
yCIx0<-XCI%*%b

yCIL <- matrix(rep(0,n),nrow=n,ncol=1) # preallocate
yCIU <- matrix(rep(0,n),nrow=n,ncol=1) # preallocate
yPIL <- matrix(rep(0,n),nrow=n,ncol=1) # preallocate
yPIU <- matrix(rep(0,n),nrow=n,ncol=1) # preallocate
counter<-0
for (i in (1):(n)){
  counter<-counter+1
  yCIL[counter]<-yCIx0[counter]-tcrit*s*sqrt(
1/n+(t[counter]-xbar)^2/SSxx)
  yCIU[counter]<-yCIx0[counter]+tcrit*s*sqrt(
1/n+(t[counter]-xbar)^2/SSxx)
  yPIL[counter]<-yCIx0[counter]-
tcrit*s*sqrt(1+1/n+(t[counter]-xbar)^2/SSxx)
  yPIU[counter]<-yCIx0[coun-
ter]+tcrit*s*sqrt(1+1/n+(t[counter]-xbar)^2/SSxx)
}
withinPI<-cbind(yhat,yPIL,yPIU)
withinPI
```

```
# scatter plot with line and 95% PI
plot(t,y,pch=19,col="blue",xlab='Time',ylab='Sales',
    xlim=c(0,n+nf),ylim=c(0,225))
lines(t,X%*%b,type="l",lty=2,col='red')
lines(seq(n+1,n+nf,by=1),yhatx0,type="l",lty=2,col='red')
points(seq(n+1,n+nf,by=1),yhatx0,pch=19,col='seagreen')
#lines(t,yCIL, type = "l", col = "deepskyblue3")
#lines(t,yCIU, type = "l", col = "deepskyblue3")
lines(t,yPIL, type = "l", col = "magenta")
lines(t,yPIU, type = "l", col = "magenta")

# forcasted prediction interval
tF <- seq(n+1,n+nf)
XCIF<-cbind(rep(1,nf),xnew)
tXCIF <- matrix(t(XCIF),2,nf)
yCIx0F<-XCIF%*%b
yCILF <- matrix(rep(0,nf),nrow=nf,ncol=1) # preallocate
yCIUF <- matrix(rep(0,nf),nrow=nf,ncol=1) # preallocate
yPILF <- matrix(rep(0,nf),nrow=nf,ncol=1) # preallocate
yPIUF <- matrix(rep(0,nf),nrow=nf,ncol=1) # preallcate
counter<-0
for (i in (1):(nf)){
  counter<-counter+1
  yCILF[counter]<-yCIx0F[counter]-tcrit*s*sqrt(
1/n+(tF[counter]-xbar)^2/SSxx)
  yCIUF[counter]<-yCIx0F[counter]+tcrit*s*sqrt(
1/n+(tF[counter]-xbar)^2/SSxx)
  yPILF[counter]<-yCIx0F[counter]-
tcrit*s*sqrt(1+1/n+(tF[counter]-xbar)^2/SSxx)
  yPIUF[counter]<-yCIx0F[coun-
ter]+tcrit*s*sqrt(1+1/n+(tF[counter]-xbar)^2/SSxx)
}
outsidePI<- cbind(yhatx0,yPILF,yPIUF)
outsidePI
```

```
# scatter plot with line and 95% PI
plot(t,y,pch=19,col="blue",xlab='Time',ylab='Sales',
    xlim=c(0,n+nf),ylim=c(0,225))
lines(t,X%*%b,type="l",lty=2,col='red')
lines(seq(n+1,n+nf,by=1),yhatx0,type="l",lty=2,
col='red')
points(seq(n+1,n+nf,by=1),yhatx0,pch=19,
col='seagreen')
#lines(t,yCIL, type = "l", col = "deepskyblue3")
#lines(t,yCIU, type = "l", col = "deepskyblue3")
lines(t,yPIL, type = "l", col = "magenta")
lines(t,yPIU, type = "l", col = "magenta")
lines(tF,yPILF, type = "l", col = "chartreuse1")
lines(tF,yPIUF, type = "l", col = "chartreuse1")

results<-data.frame(rbind(withinPI,outsidePI))
names(results) <- c("yhat","PI_L","PI_U")

write.csv(round(results,digits=3),
file="forecatandpiPower.csv")
```