Crystal Mosley

Predict 413, Sec 55

Homework #2

**SECTION 8.11**

**Problem 7—**

library(fpp)

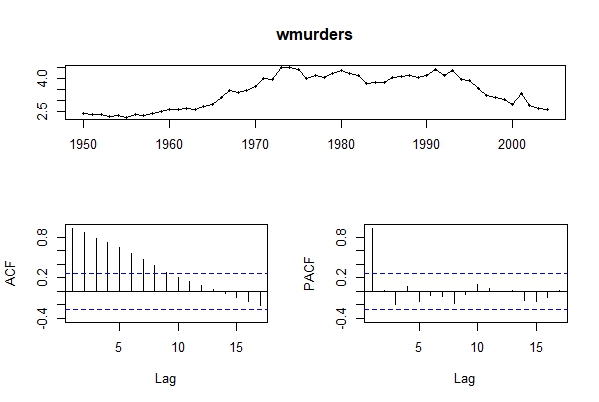
#load data set "wmurder"

data(wmurders)

**7a: By studying appropriate graphs of the series in R, find an appropriate ARIMA(p,d,q) model for these data and 7d: Fit the model using R and examine the residuals. Is the model satisfactory?**

*#plots a ts along with its acf, and either its pacf, lagged scatterplot, or spectrum*

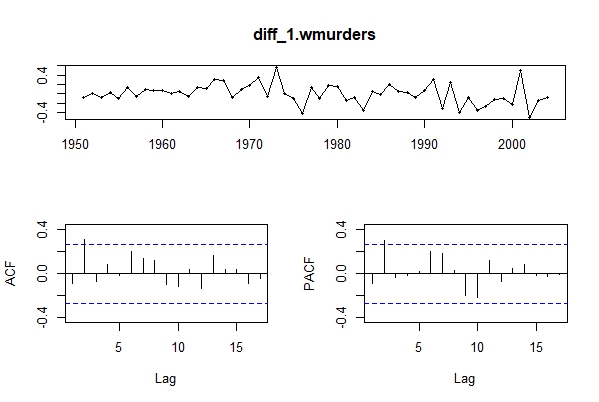
tsdisplay(wmurders)



*#After reviewing the plots, it's not showing as stationary, so we take the first difference.*

diff\_1.wmurders <- diff(wmurders)

tsdisplay(diff\_1.wmurders)



*#The ACF and PACF still show spikes at lag 2 but look better than the previous.*

*#ADF test (augmented dickey-fuller) - checks that 'x' has a unit root; ADF test p-value <0.05 indicates a stationary series*

adf.test(diff\_1.wmurders)

Augmented Dickey-Fuller Test

data: diff\_1.wmurders

Dickey-Fuller = -3.7688, Lag order = 3, p-value = 0.02726

alternative hypothesis: stationary

*#KPSS test - checks whether 'x' is level or trend stationary; KPSS test p-value <0.05 indicates a non-stationary series*

kpss.test(diff\_1.wmurders)

KPSS Test for Level Stationarity

data: diff\_1.wmurders

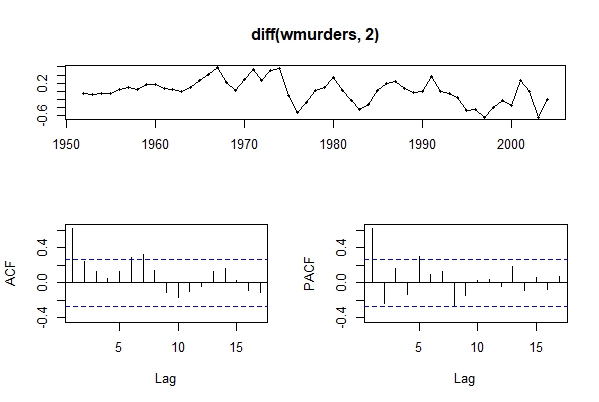
KPSS Level = 0.58729, Truncation lag parameter = 1, p-value = 0.02379

*#Both first difference with ADF and KPSS results show as stationary and non-stationary; therefore, more tests need to be completed.*

*#Take second difference*

diff\_2.wmurders <- diff(diff(wmurders))

tsdisplay(diff(wmurders,2))



adf.test(diff(wmurders,2))

Augmented Dickey-Fuller Test

data: diff(wmurders, 2)

Dickey-Fuller = -3.9819, Lag order = 3, p-value = 0.01705

alternative hypothesis: stationary

kpss.test(diff(wmurders,2))

KPSS Test for Level Stationarity

data: diff(wmurders, 2)

KPSS Level = 0.72551, Truncation lag parameter = 1, p-value = 0.01123

*#Now both p-values are <0.05, meaning it's stationary. The ACF and PACF plots show a large spike at 1 which means either p or q need to be 1.*

arima\_fit<-auto.arima(wmurders)

summary(arima\_fit)

Series: wmurders

ARIMA(1,2,1)

Coefficients:

ar1 ma1

-0.2434 -0.8261

s.e. 0.1553 0.1143

sigma^2 estimated as 0.04632: log likelihood=6.44

AIC=-6.88 AICc=-6.39 BIC=-0.97

Training set error measures:

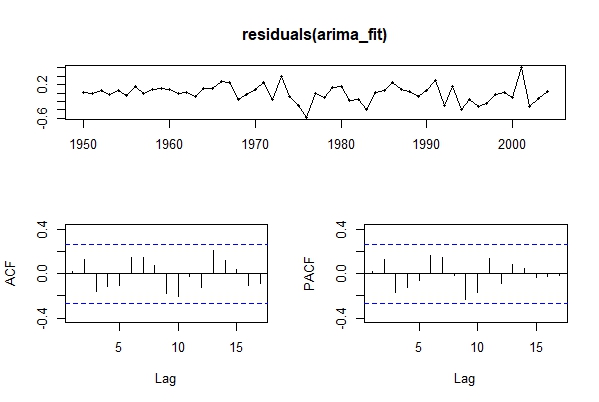
ME RMSE MAE MPE MAPE MASE ACF1

Training set -0.01065956 0.2072523 0.1528734 -0.2149476 4.335214 0.9400996 0.02176343

*#The arima model chose 1,2,1 - which is what was needed for p and/or q*

*#Now review the residuals*

tsdisplay(residuals(arima\_fit))



*#There are no lags for ACF or PACF.*

**7b: Should you include a constant in the model? Explain.**

*#Constants bring a drift into models, so we wouldn't want a constant for this data since d=2.*

**7c: Write this model in terms of the backshift operator.**

*# y^t^=phi x y^(t-2)^ + theta x (e-2)*

**7e: Forecast three times ahead. Check your forecasts by hand to make sure you know how they have been calculated.**

forecast(arima\_fit,h=3)

Point Forecast Lo 80 Hi 80 Lo 95 Hi 95

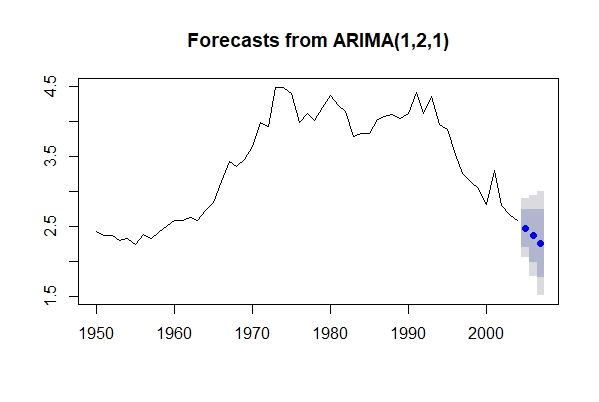
2005 2.470660 2.194836 2.746484 2.048824 2.892496

2006 2.363106 1.986351 2.739862 1.786908 2.939304

2007 2.252833 1.765391 2.740276 1.507354 2.998313

**7f: Create a plot of the series with forecasts and prediction intervals for the next three periods shown.**

plot(forecast(arima\_fit,h=3))



**7g: Does auto.arima give the same model you have chosen? If not, which model do you think is better?**

*# yes*

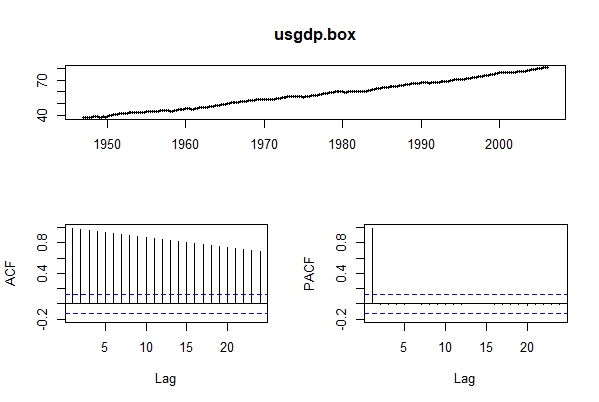
**Problem 9—**

**9a: For the usgdp series: if necessary, find a suitable Box-Cox transformation for the data**

lam <- BoxCox.lambda(usgdp)

usgdp.box <- BoxCox(usgdp, lambda = lam)

tsdisplay(usgdp.box)



**9b: fit a suitable ARIMA model to the transformed data using auto.arima()**

usgdp.fit <- auto.arima(usgdp, trace = TRUE, ic ="aic", lambda = lam)

usgdp.fit

Series: usgdp

ARIMA(0,1,2) with drift

Box Cox transformation: lambda= 0.366352

Coefficients:

ma1 ma2 drift

0.2727 0.2061 0.1829

s.e. 0.0640 0.0609 0.0179

sigma^2 estimated as 0.03507: log likelihood=61.94

AIC=-115.89 AICc=-115.71 BIC=-102.03

**9c: try some other plausible models by experimenting with the orders chosen**

usgdp.fit2 <- Arima(usgdp, order = c(0,1,1), lambda = lam)

usgdp.fit3 <- Arima(usgdp, order = c(0,1,3), lambda = lam)

accuracy(usgdp.fit)

ME RMSE MAE MPE MAPE MASE ACF1

Training set 1.308821 39.24517 29.27606 -0.01627204 0.6801511 0.1654604 -0.02298287

accuracy(usgdp.fit2)

ME RMSE MAE MPE MAPE MASE ACF1

Training set 29.49404 53.06825 41.9809 0.5892384 0.8990364 0.2372648 -0.1112159

accuracy(usgdp.fit3)

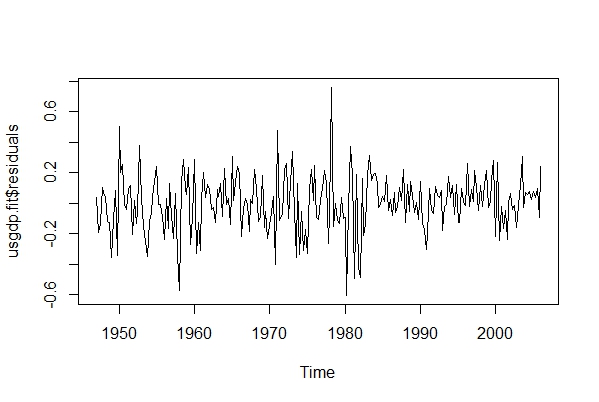
ME RMSE MAE MPE MAPE MASE ACF1

Training set 19.60141 45.81025 34.59863 0.3929712 0.7650733 0.1955422 -0.141552

**9d: Choose what you think is the best model and check the residual diagnostics**

#usgdp.fit had the lowest RMSE/MAE and was considered the best arima model (0,1,2) with drift.

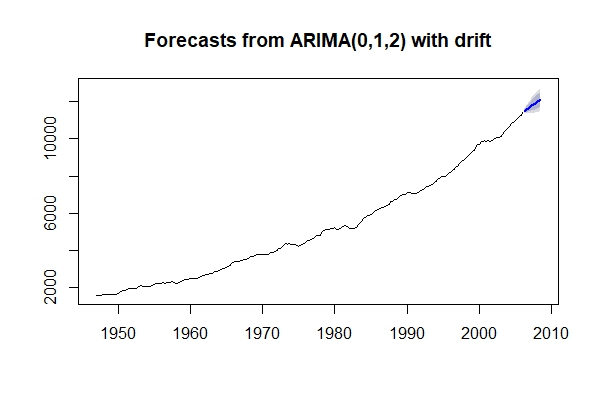
plot(usgdp.fit$residuals)



**9e: produce forecasts of your fitted model. Do the forecasts look reasonable?**

forecast\_1 <- forecast(usgdp.fit, h=10)

plot(forecast\_1)



*# yes this forecast looks reasonable*

**9f: compare the results with what you would obtain using ets() (with no transformation)**

*#fit first*

usgdp.fit4 <- ets(usgdp); usgdp.fit4

ETS(A,A,N)

Call:

ets(y = usgdp)

Smoothing parameters:

alpha = 0.9999

beta = 0.278

Initial states:

l = 1557.4589

b = 18.6862

sigma: 41.8895

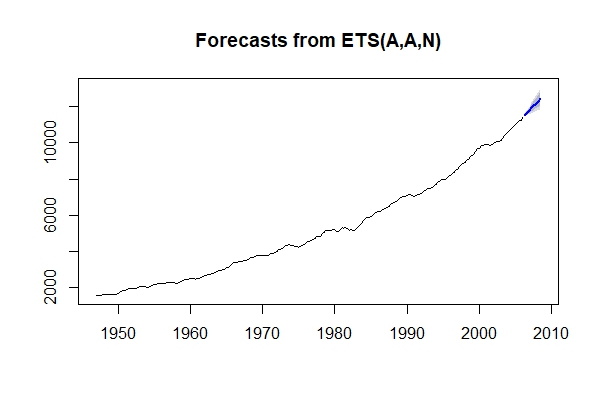
AIC AICc BIC

3072.303 3072.562 3089.643

*#then forecast*

forecast\_2 <- forecast(usgdp.fit4, h=10)

plot(forecast\_2)

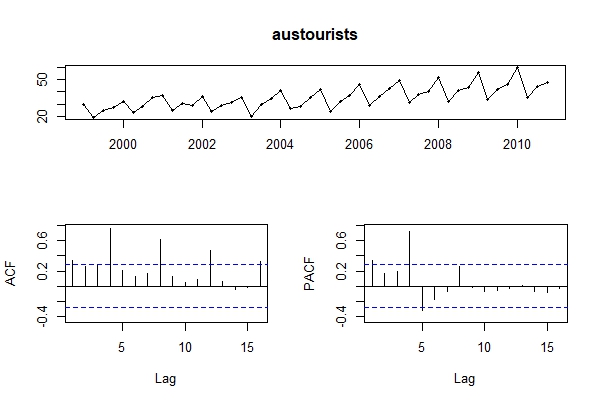


**Problem 10—**

**10a: Consider austourists, the quarterly number of international tourists to Australia for the period 1999–2010. Describe the time plot**

data("austourists")

tsdisplay(austourists)



*#This ts looks multiplicative in seasonality and a growth trend.*

**10b: What can you learn from the ACF graph?**

*#There's autocorrelation in the lagged obs. ACF has 5 lags (1,4,8,12,16) which is 31.25% - well over the 5% threshold.*

**10c: What can you learn from the PACF graph?**

*#PACF has 3 lags (1,4,5) which is 18.75% - also well over the 5% threshold. Seasonal differencing is needed.*

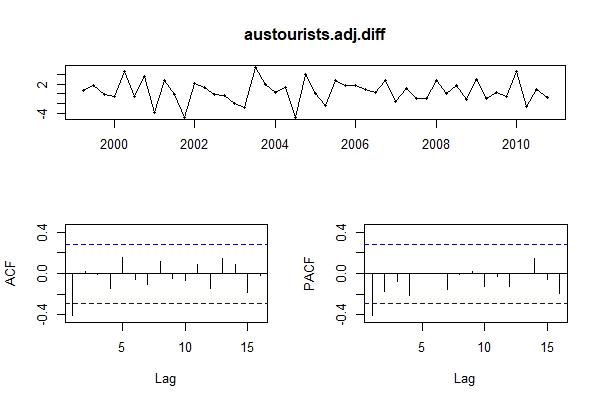
**10d: Produce plots of the seasonally differenced data (equation...). What model do these graphs suggest?**

tourists.decom <- decompose(austourists, "multiplicative")

austourists.adj <- austourists / tourists.decom$seasonal

austourists.adj.diff <- diff(austourists.adj)

tsdisplay(austourists.adj.diff)



*#Lags at 1 suggest that p or q need to be 1.*

**10e: Does auto.arima() give the same model that you chose? If not, which model do you think is better?**

tourist.fit <- auto.arima(austourists)

tourist.fit

Series: austourists

ARIMA(1,0,0)(1,1,0)[4] with drift

Coefficients:

ar1 sar1 drift

0.4493 -0.5012 0.4665

s.e. 0.1368 0.1293 0.1055

sigma^2 estimated as 5.606: log likelihood=-99.47

AIC=206.95 AICc=207.97 BIC=214.09

*#Arima chose (1,0,0)(1,1,0)[4] with drift – this is at least p or q being 1, which is almost in line with what I said previously.*

**10f: Write the model in terms of the backshift operator, then without using the backshift operator.**

*# (1−B4)Yt=BY(t−1)+et*

**Problem 18—**

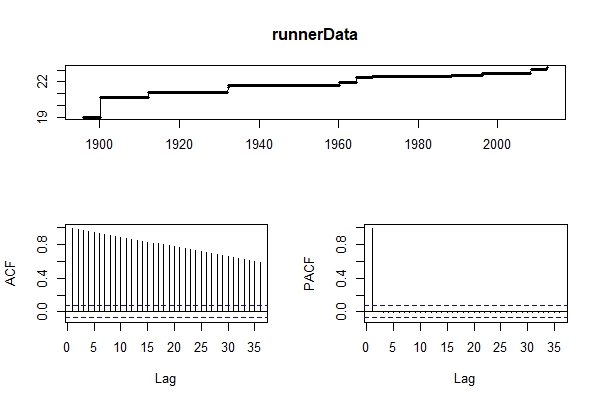
**18a: install rdatamarket; pick a ts and import**

library(rdatamarket)

runnerData <- ts(rdatamarket::dmseries("https://datamarket.com/data/set/248g/mens-olympic-running-records-by-year#!ds=248g!2ltf&display=table")[,1], start=1896, frequency=12)

**18b: Plot graphs of the data, and try to identify an appropriate ARIMA model**

tsdisplay(runnerData)



*#The data shows an upwards trend in the top plot. The ACF looks just like that of a white noise series. There are no autocorrelations lying outside the 95% limits.*

*#I’m not sure on which Arima model is happening, but maybe a (0,0,0)?*

#Let autoarima choose best fit

runnerData.fit <- auto.arima(runnerData)

runnerData.fit

Series: runnerData

ARIMA(0,1,0) with drift

Coefficients:

drift

0.0031

s.e. 0.0014

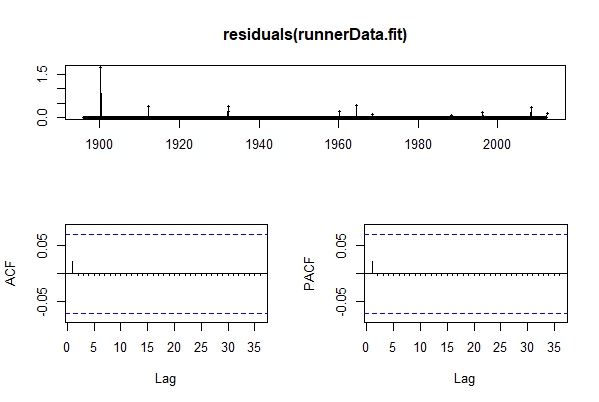
sigma^2 estimated as 0.002769: log likelihood=2128.11

AIC=-4252.22 AICc=-4252.21 BIC=-4241.73

*#The arima chose (0,1,0) – which is a random walk with a drift.*

**18c: Do residual diagnostic checking of your ARIMA model. Are the residuals white noise?**

tsdisplay(residuals(runnerData.fit))



*# These residuals do look like white noise.*

**18d: Use your chosen ARIMA model to forecast the next four years.**

forecast(runnerData.fit,h=4)

Point Forecast Lo 80 Hi 80 Lo 95 Hi 95

Jun 2012 23.23186 23.16443 23.29930 23.12873 23.33500

Jul 2012 23.23493 23.13956 23.33030 23.08908 23.38078

Aug 2012 23.23799 23.12119 23.35479 23.05936 23.41662

Sep 2012 23.24106 23.10619 23.37593 23.03479 23.44732

Oct 2012 23.24412 23.09333 23.39491 23.01351 23.47473

Nov 2012 23.24718 23.08200 23.41237 22.99456 23.49981

Dec 2012 23.25025 23.07183 23.42866 22.97738 23.52311

Jan 2013 23.25331 23.06258 23.44405 22.96161 23.54502

Feb 2013 23.25638 23.05407 23.45868 22.94698 23.56577

Mar 2013 23.25944 23.04619 23.47269 22.93331 23.58557

Apr 2013 23.26250 23.03885 23.48616 22.92045 23.60456

May 2013 23.26557 23.03197 23.49917 22.90831 23.62283

Jun 2013 23.26863 23.02549 23.51177 22.89678 23.64048

Jul 2013 23.27170 23.01938 23.52401 22.88581 23.65758

Aug 2013 23.27476 23.01359 23.53593 22.87533 23.67419

Sep 2013 23.27782 23.00809 23.54756 22.86529 23.69035

Oct 2013 23.28089 23.00285 23.55893 22.85566 23.70612

Nov 2013 23.28395 22.99785 23.57005 22.84640 23.72151

Dec 2013 23.28702 22.99308 23.58096 22.83747 23.73656

Jan 2014 23.29008 22.98850 23.59166 22.82886 23.75130

Feb 2014 23.29314 22.98412 23.60217 22.82053 23.76576

Mar 2014 23.29621 22.97991 23.61251 22.81247 23.77994

Apr 2014 23.29927 22.97587 23.62268 22.80467 23.79388

May 2014 23.30234 22.97198 23.63270 22.79709 23.80758

Jun 2014 23.30540 22.96823 23.64257 22.78974 23.82106

Jul 2014 23.30846 22.96461 23.65232 22.78259 23.83434

Aug 2014 23.31153 22.96113 23.66193 22.77564 23.84742

Sep 2014 23.31459 22.95776 23.67142 22.76887 23.86032

Oct 2014 23.31766 22.95451 23.68080 22.76227 23.87304

Nov 2014 23.32072 22.95137 23.69008 22.75584 23.88560

Dec 2014 23.32378 22.94832 23.69925 22.74957 23.89800

Jan 2015 23.32685 22.94538 23.70832 22.74344 23.91026

Feb 2015 23.32991 22.94253 23.71730 22.73746 23.92236

Mar 2015 23.33298 22.93977 23.72619 22.73162 23.93434

Apr 2015 23.33604 22.93709 23.73499 22.72590 23.94618

May 2015 23.33911 22.93450 23.74371 22.72031 23.95790

Jun 2015 23.34217 22.93198 23.75236 22.71484 23.96950

Jul 2015 23.34523 22.92954 23.76093 22.70948 23.98099

Aug 2015 23.34830 22.92717 23.76943 22.70423 23.99236

Sep 2015 23.35136 22.92487 23.77786 22.69909 24.00363

Oct 2015 23.35443 22.92263 23.78622 22.69405 24.01480

Nov 2015 23.35749 22.92046 23.79452 22.68911 24.02587

Dec 2015 23.36055 22.91835 23.80275 22.68427 24.03684

Jan 2016 23.36362 22.91631 23.81093 22.67951 24.04772

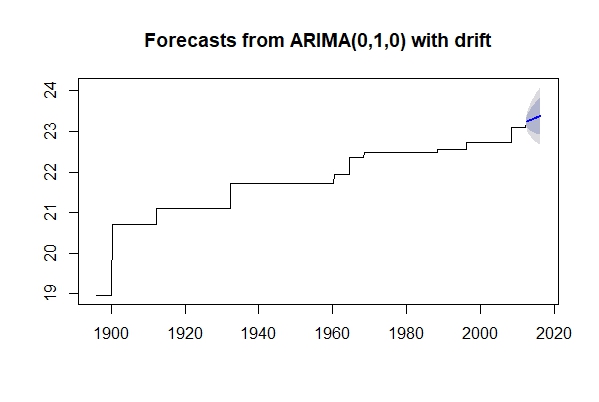
Feb 2016 23.36668 22.91432 23.81905 22.67485 24.05852

Mar 2016 23.36975 22.91238 23.82711 22.67027 24.06922

Apr 2016 23.37281 22.91050 23.83512 22.66577 24.07985

May 2016 23.37587 22.90867 23.84308 22.66135 24.09040

plot(forecast(runnerData.fit,h=4))



**18e: Now try to identify an appropriate ETS model.**

*#fit first*

runnerData.fit2 <- ets(runnerData); runnerData.fit2

ETS(A,N,N)

Call:

ets(y = runnerData)

Smoothing parameters:

alpha = 0.9999

Initial states:

l = 18.9572

sigma: 0.0673

AIC AICc BIC

971.6018 971.6330 985.5526

*#find accuracy*

accuracy(runnerData.fit2)

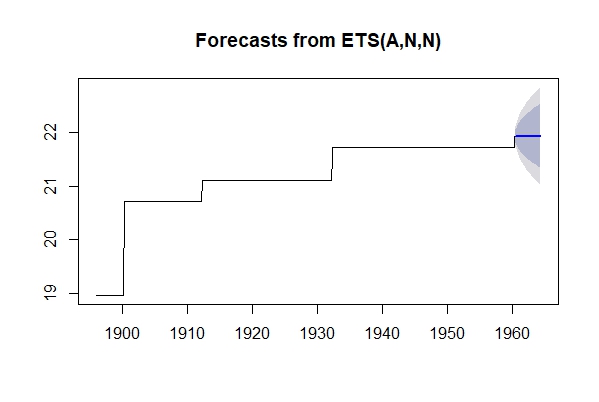
ME RMSE MAE MPE MAPE MASE ACF1

Training set 0.003846998 0.06716797 0.003847378 0.01829999 0.01830199 0.08781466 0.02141542

*#then forecast*

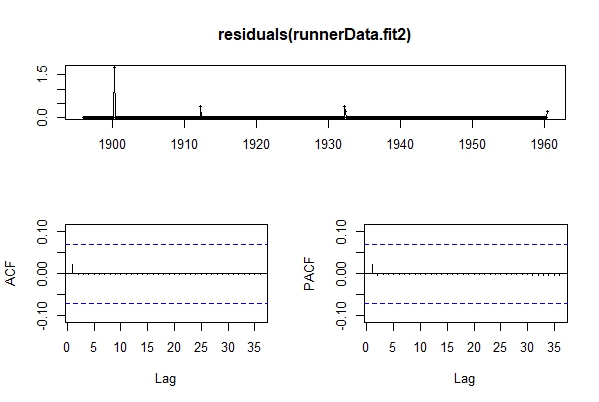
forecast.runnerData <- forecast(runnerData.fit2, h=48)

plot(forecast.runnerData)



**18f: Do residual diagnostic checking of your ETS model. Are the residuals white noise?**

tsdisplay(residuals(runnerData.fit2))

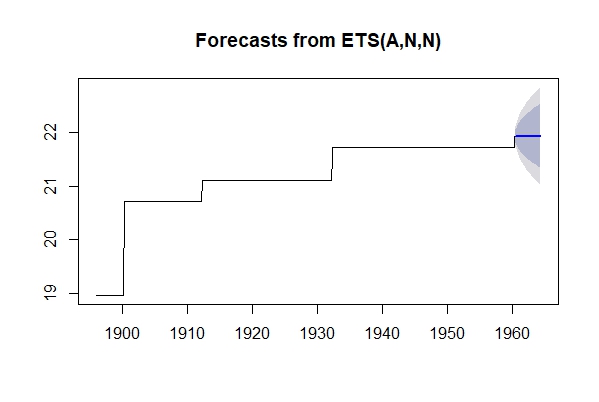


*# These residuals do look like white noise.*

**18g: Use your chosen ETS model to forecast the next four years.**

forecast.runnerData <- forecast(runnerData.fit2, h=48)

plot(forecast.runnerData)



**18h: Which of the two models do you prefer?**

*#I think the arima model did better than the ets, so I'd go with the arima (0,1,0) with drift.*

**SECTION 9.7**

**Problem 1—**

library(ggplot2)

data("advert")

**1a: Plot the data using autoplot. Why is it useful to set facets=TRUE?**

autoplot(advert[,1:2], facets = TRUE)

*#If facets = FALSE, the ts will be merged together with a color assigned to each variable.*

**1b: Fit a standard regression model yt=a+bxt+ηtyt=a+bxt+ηt where ytyt denotes sales and xtxtdenotes advertising using the tslm() function.**

advert %>%

as.data.frame() %>%

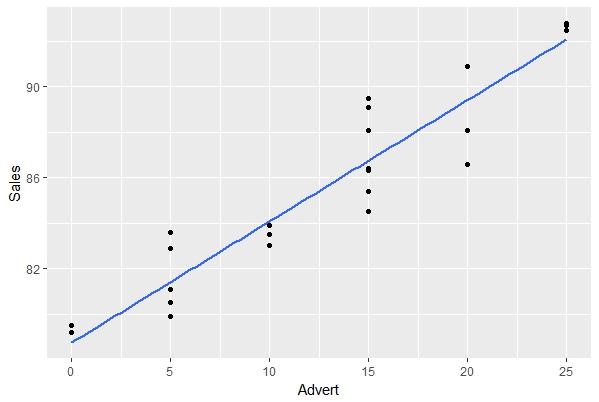
ggplot(aes(x=advert, y=sales)) +

ylab("Sales") +

xlab("Advert") +

geom\_point() +

geom\_smooth(method="lm", se=FALSE)



tslm(sales ~ advert, data=advert)

Call:

tslm(formula = sales ~ advert, data = advert)

Coefficients:

(Intercept) advert

78.7343 0.5343

**1c: Show that the residuals have significant autocorrelation.**

*#Since this dataset has multiple entries for time, it's a multivariate ts and therefore auto.arima cannot be used unless the data is converted into a ts.*

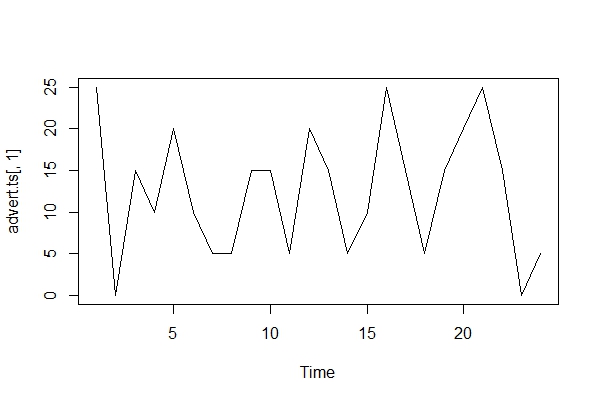
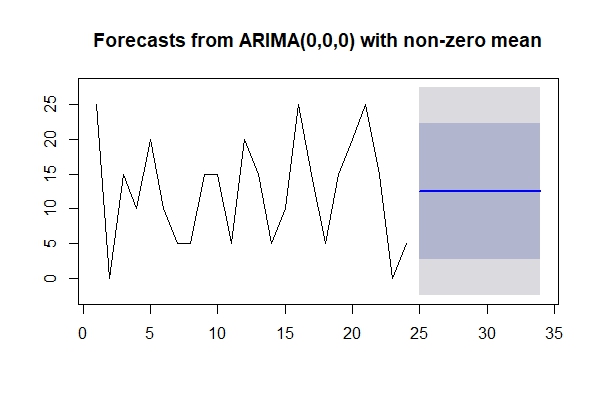
advert.ts = ts(advert)

arima\_fit = auto.arima(advert.ts[,1])

plot(advert.ts[,1]) *#first plot*

arima\_forecast = forecast(arima\_fit, h = 10)

plot(arima\_forecast) *#second plot*

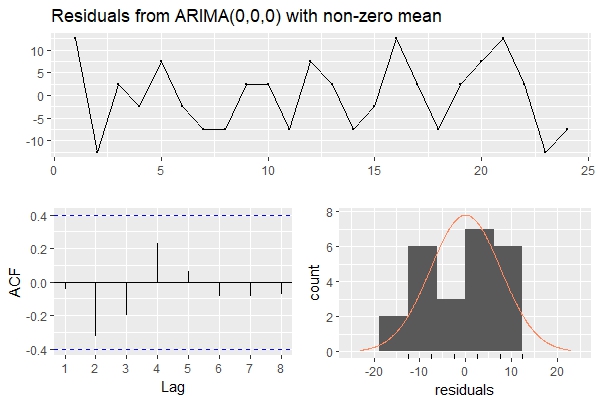
checkresiduals(arima\_fit, plot = TRUE)

Ljung-Box test

data: Residuals from ARIMA(0,0,0) with non-zero mean

Q\* = 5.8754, df = 3.8, p-value = 0.1887

Model df: 1. Total lags used: 4.8



**1d: What difference does it make you use the Arima function instead: Arima(advert[,"sales"], xreg=advert[,"advert"], order=c(0,0,0))**

Series: advert[, "sales"]

Regression with ARIMA(0,0,0) errors

Coefficients:

intercept advert[, "advert"]

78.7343 0.5343

s.e. 0.5719 0.0392

sigma^2 estimated as 2.267: log likelihood=-42.83

AIC=91.66 AICc=92.86 BIC=95.2

**1e: Refit the model using auto.arima(). How much difference does the error model make to the estimated parameters? What ARIMA model for the errors is selected?**

**Problem 2—**

**Problem 3—**

**Problem 4—**

**Problem 5—**

**Problem 6—**