**Jan Malinowski**

Time Series

Final Assessment

**Exercise 1**

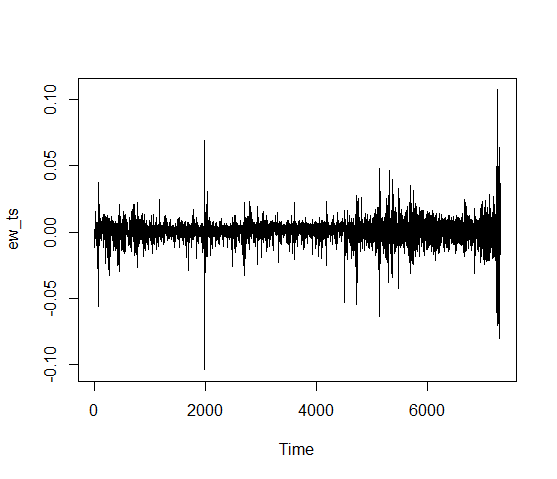
Consider the simple returns of IBM stock, CRSP value-weighted index, CRSP equal-weighted index, and the S&P composite index from January 1980 to December 2008 (understanding the meaning of these terms is not necessary for the exercise). The index returns include dividend distributions. The \_le is d-ibm3dxwkdays8008.txt which has 12 columns. The columns are (year, month, day, IBM, VW, EW, SP, M, T, W, R, F), where M, T, W, R, F denotes indicator variables for Monday to Friday respectively. Use a regression model to study the e\_ects of trading days on the equal-weighted index returns (column EW).

What is the fitted model?

Are the weekday effects significant in the returns at the 5% significance level? Are there correlations in the regression residuals? If so, build a suitable time series model.

Let’s look how does our time series look like.

> plot(ew\_ts)



It looks quit well.

|  |
| --- |
| > taks1\_model1<-lm(ew\_ts~MONDAY+TUESDAY+WEDNESDAY+THURSDAY+FRIDAY)  > taks1\_model1$coefficients  (Intercept) MONDAY1 TUESDAY1 WEDNESDAY1 THURSDAY1 FRIDAY1  0.002238625 -0.003173400 -0.001977828 -0.001018537 -0.001029433 NA |

Variable firday obtain NA so we make our new model without that variable.

> taks1\_model2 <- lm(ew\_ts~MONDAY+TUESDAY+WEDNESDAY+THURSDAY)

> summary(taks1\_model2)

Call:

lm(formula = ew\_ts ~ MONDAY + TUESDAY + WEDNESDAY + THURSDAY)

Residuals:

Min 1Q Median 3Q Max

-0.102962 -0.003094 0.000533 0.003795 0.108319

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.0022386 0.0002155 10.389 < 2e-16 \*\*\*

MONDAY1 -0.0031734 0.0003085 -10.286 < 2e-16 \*\*\*

TUESDAY1 -0.0019778 0.0003028 -6.532 6.94e-11 \*\*\*

WEDNESDAY1 -0.0010185 0.0003027 -3.365 0.000770 \*\*\*

THURSDAY1 -0.0010294 0.0003042 -3.384 0.000719 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.008234 on 7314 degrees of freedom

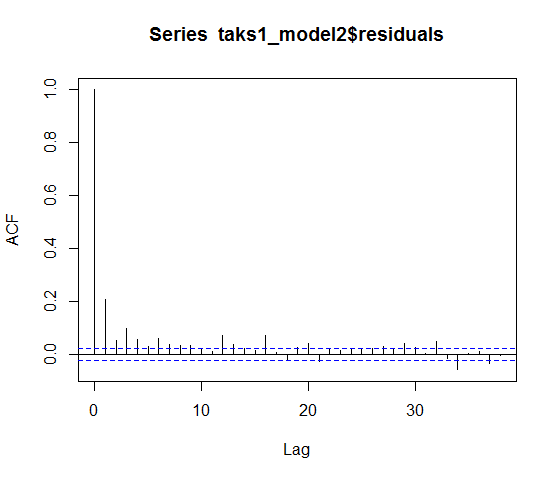
Multiple R-squared: 0.01618, Adjusted R-squared: 0.01564

F-statistic: 30.06 on 4 and 7314 DF, p-value: < 2.2e-16

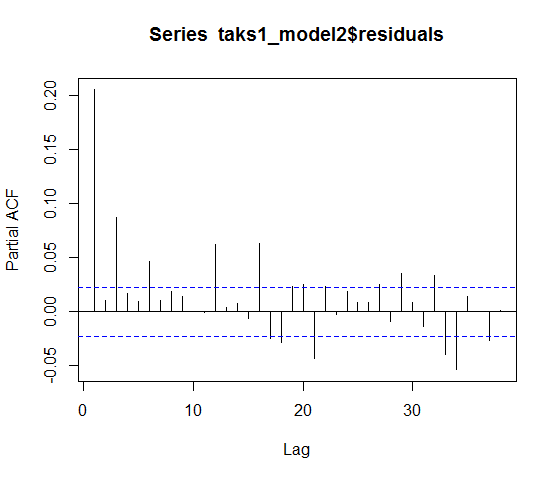
Are the weekday effects significant in the returns at the 5% significance level?

Every weekday has p-Value less then 5% so they are signicant.

> acf(taks1\_model2$residuals)



> pacf(taks1\_model2$residuals)



Model for residuals

> arima\_residuals<-auto.arima(taks1\_model2$residuals)

> summary(arima\_residuals)

Series: taks1\_model2$residuals

ARIMA(5,0,2) with zero mean

Coefficients:

ar1 ar2 ar3 ar4 ar5 ma1 ma2

0.1652 0.8701 -0.0962 0.0440 -0.0675 0.0332 -0.8757

s.e. 0.0372 0.0342 0.0165 0.0127 0.0132 0.0353 0.0318

sigma^2 estimated as 6.408e-05: log likelihood=24952.19

AIC=-49888.38 AICc=-49888.36 BIC=-49833.2

Training set error measures:

ME RMSE MAE MPE MAPE MASE ACF1

Training set 2.887735e-06 0.008001052 0.004993911 86.38142 203.9048 0.7839033 0.001381969

**Exercise 2**

Consider the monthly simple returns of GE stock from January 1926 to December 2008. They are found in m-ge2608.txt in the course directory. Use the last three years of data for forecast evaluation.

1. Using lagged returns r(t􀀀1), r(t􀀀2), r(t􀀀3) as input, build a 3-2-1 feed forward neural

network to forecast 1-step-ahead returns. Calculate the mean squared error of forecasts.

I transformed our data to data frame , and I added lagged raturns. After that I build a 3-2-1 feed forward neural

network to forecast 1-step-ahead returns.

> model\_nnet<-nnet(rtn~., data\_task2[961:993,], size = 2)

# weights: 17

initial value 8.296081

final value 0.156111

converged

Calculate the mean squared error of forecasts.

> mean(( predict(model\_nnet, data\_task2[961:993,], type = "raw") - data\_task2$rtn[961:993])^2)

[1] 0.004730637

I think it is a very good, satisfacting prediction.

2. Again, using r(t 􀀀 1), r(t 􀀀 2), r(t 􀀀 3) and also their signs, build a 6-5-1 feed forward

neural network to forecast the 1-step-ahead GE stock price movement (that is, we're only

interested in whether the price goes up or down; not by how much), with 1 denoting

upward movement. Calculate the mean squared error of the forecasts.

I ussed that guidline

If rtn denotes return, you can create a direction variable by:

drtn = ifelse(rtn>0,1,0)

to create 3 new columns containing price movement

Creating a model

> model\_nnet2<-nnet(rtn~., data\_task2[961:993,], size = 2)

# weights: 17

initial value 12.407595

final value 0.156090

converged

Calculate the mean squared error of forecasts.

> mean(( predict(model\_nnet2, data\_task2[961:993,], type = "raw") - data\_task2$rtn[961:993])^2)

[1] 0.004730009

The mean squered errors are even smaller than in the first predictions.