Forecasting issues

Forcast Padawan 2

November 17, 2016

The goal of this experiment is to design the best model to forcaste the number of issue in the per day in the comming two weeks. We think that sthis could help Open Source organisation to manage there human ressources.

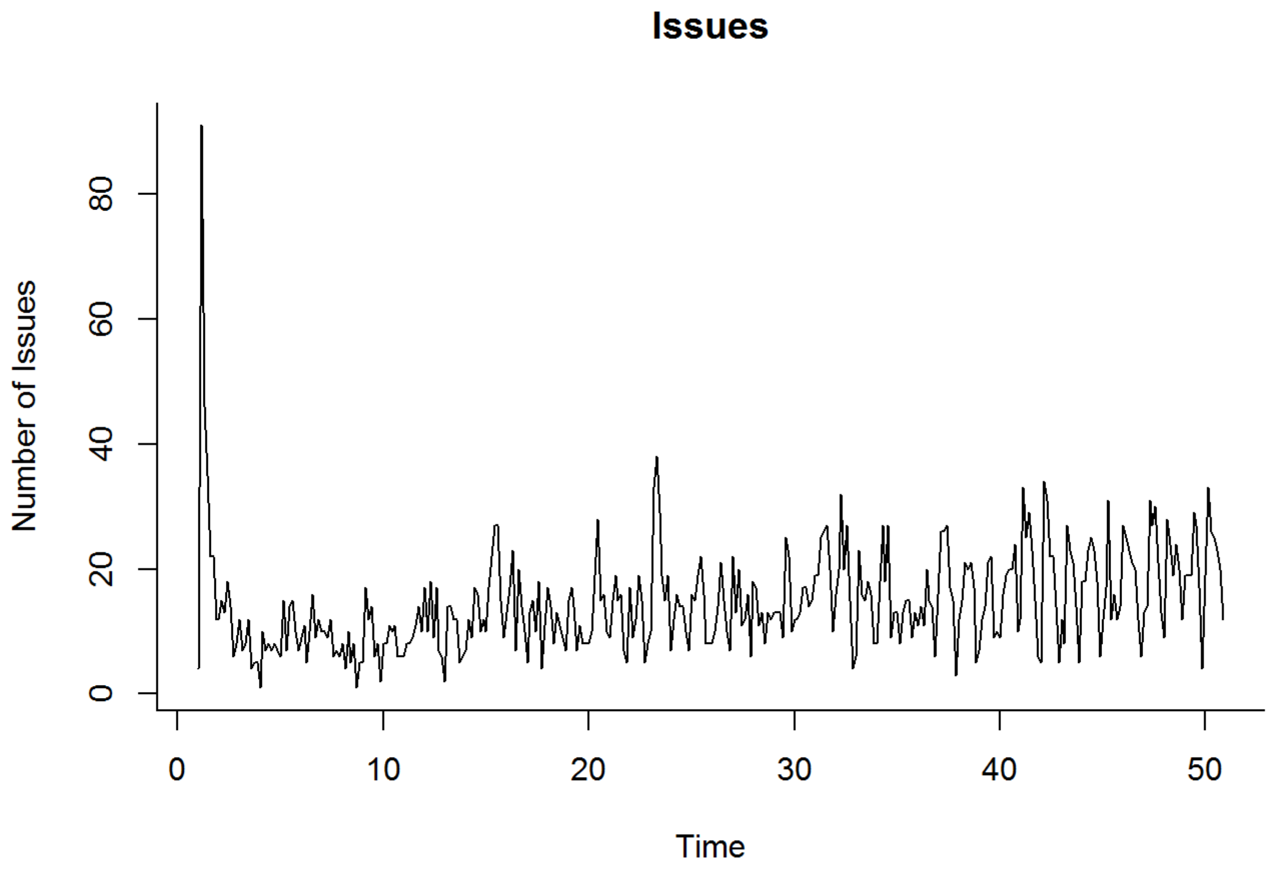
Load the data

```
#install.packages('forecast')
library('forecast')
library(knitr)
#Load the data frame
issues.csv <- read.csv("issues/tensorflow_tensorflow.csv")
commits.csv <- read.csv("commits/tensorflow_tensorflow.csv")

issues.csv$date = as.POSIXlt(as.Date(issues.csv$date,format='%m/%d/%Y'))
commits.csv$date = as.POSIXlt(as.Date(commits.csv$date,format='%m/%d/%Y'))
```

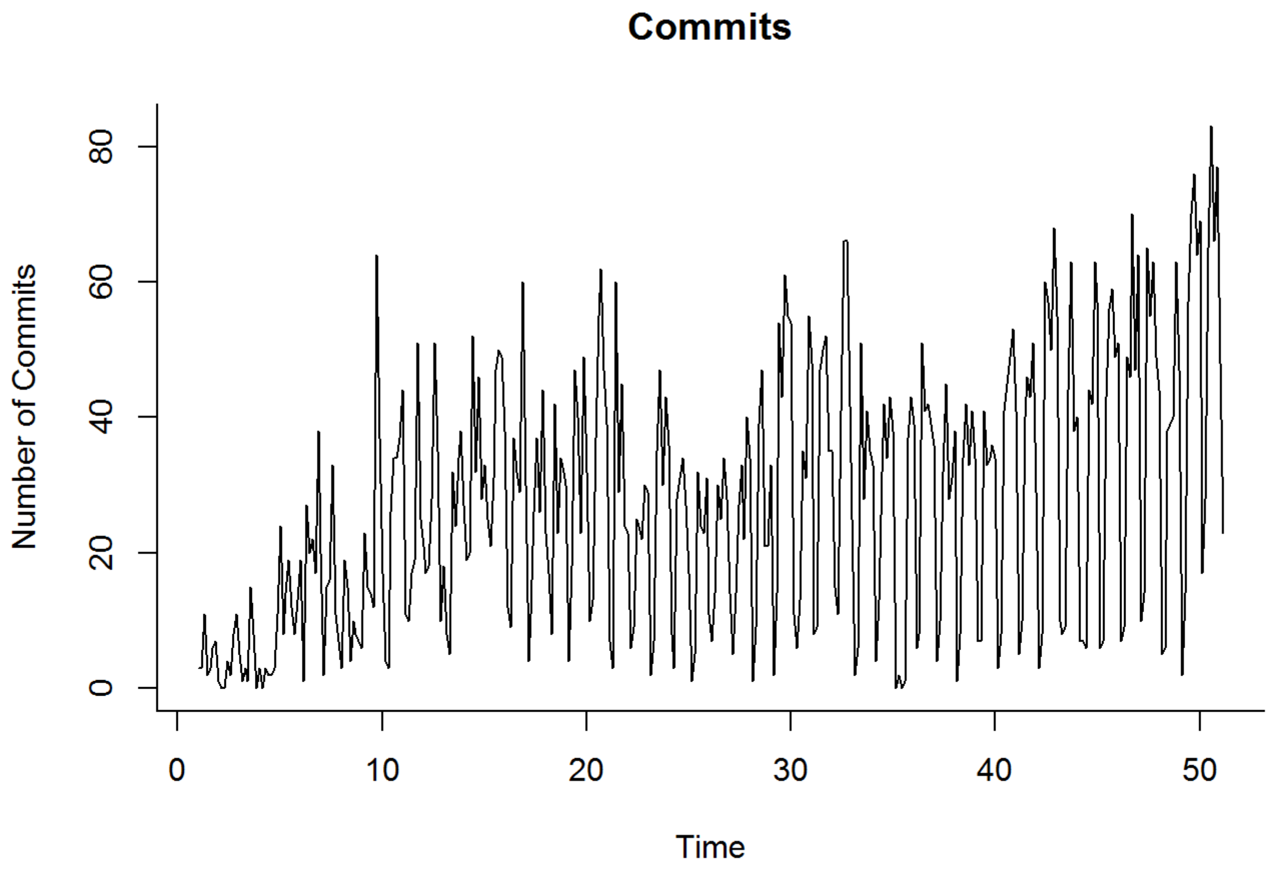
keep the last 12 months

```
to_date <- issues.csv$date[length(issues.csv$date)]
from_date <- to_date
from_date$year <- from_date$year - 1

issues.csv <- subset(issues.csv, date <= to_date & date >= from_date)
commits.csv <- subset(commits.csv, date <= to_date & date >= from_date)
```

```
#loading issues and commits into a ts object
issues.ts <- ts(issues.csv$number_of_issues, frequency = 7)

commits.ts <- ts(commits.csv$number_of_commits, frequency = 7)
plot(issues.ts, main = 'Issues', bty = 'l', ylab = 'Number of Issues')
```




```
plot(commits.ts, main = 'Commits', bty = 'l', ylab = 'Number of Commits')
```



```
time <- time(issues.ts)

n.valid <- 21
n.train <- length(issues.ts) - n.valid

train.issues.ts <- window(issues.ts, start=time[1], end=time[n.train])
valid.issues.ts <- window(issues.ts,
                    start=time[n.train+1],
                    end=time[n.train+n.valid])

train.commits.ts <- window(commits.ts, start=time[1], end=time[n.train])
valid.commits.ts <- window(commits.ts,
                      start=time[n.train+1],
                      end=time[n.train+n.valid])
```

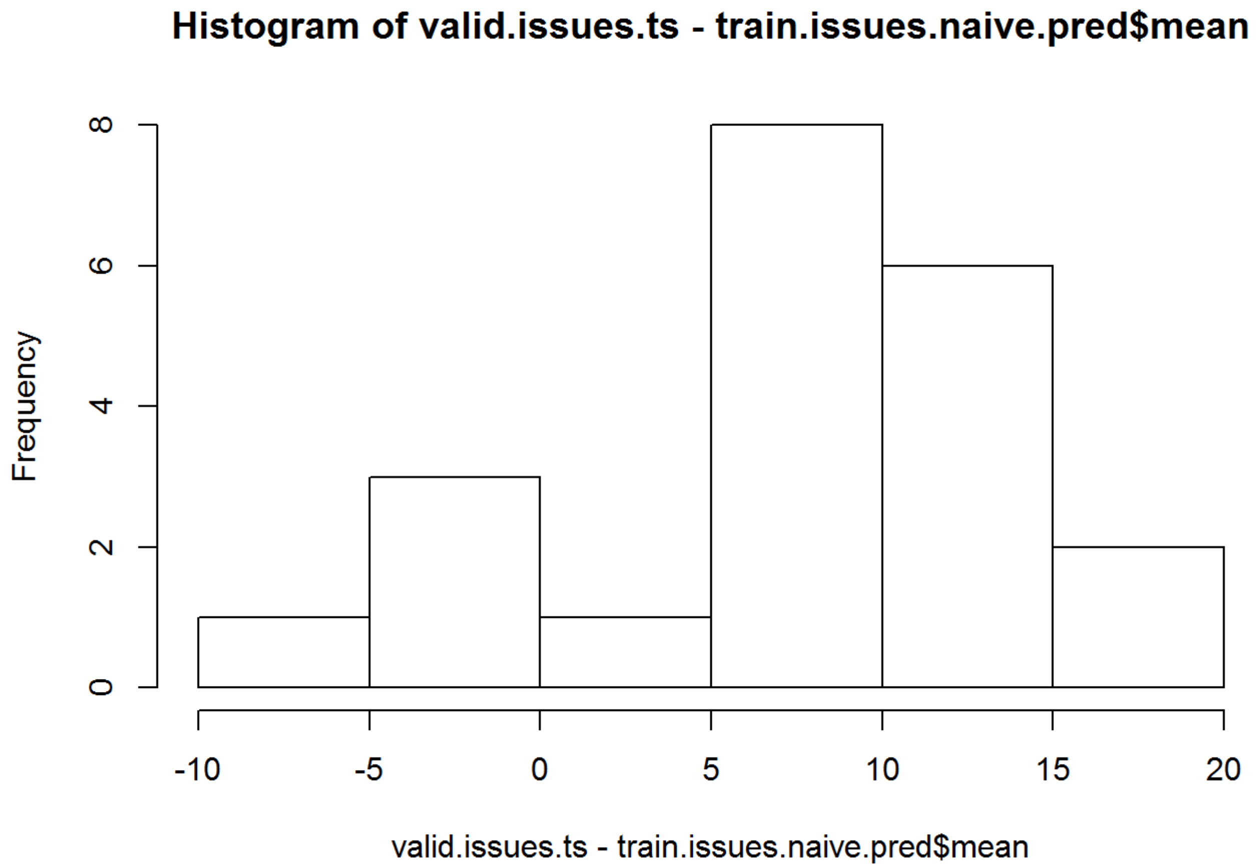
Naive Forecast

Naive

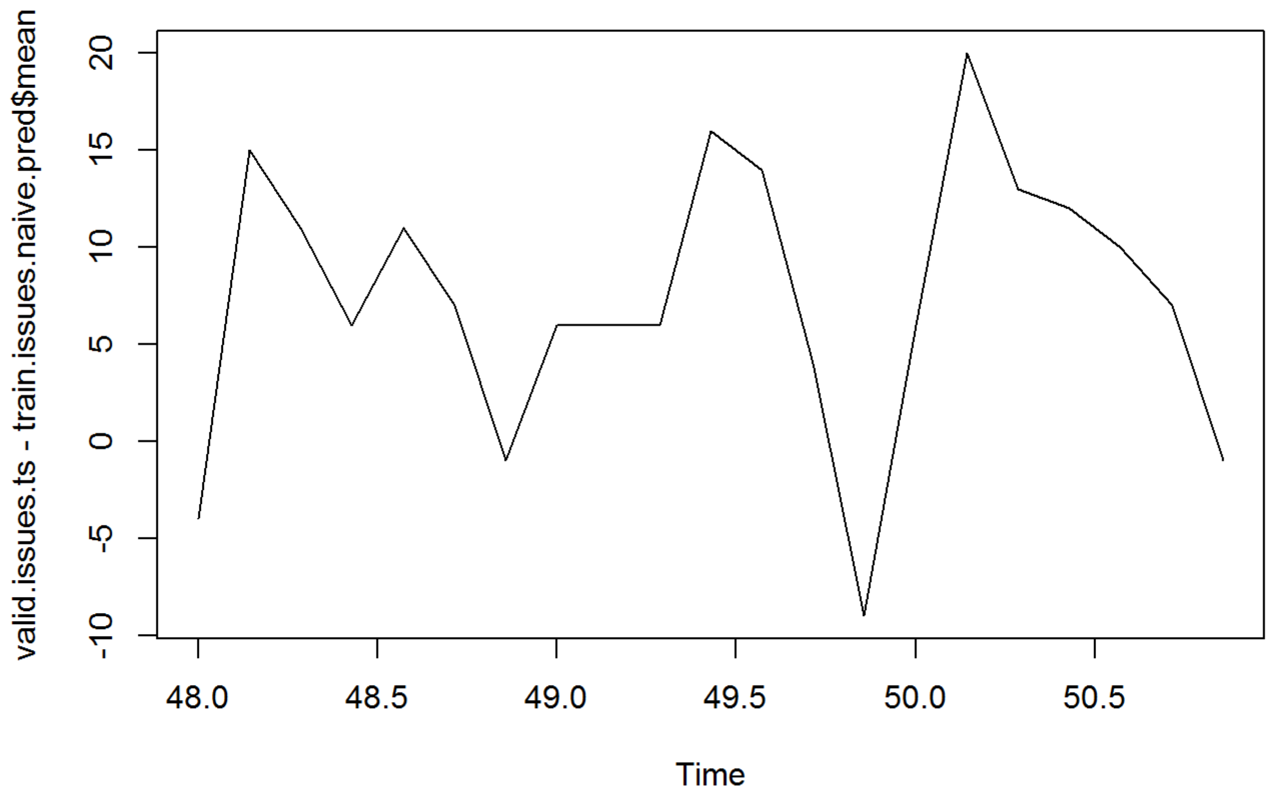
```
train.issues.naive.pred <- naive(train.issues.ts, h=n.valid)
kable(accuracy(train.issues.naive.pred, valid.issues.ts))
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	0.027439	8.695212	5.564024	-19.15979	49.74022	0.9565488	-0.2721137	NA
Test set	7.380952	10.059348	8.809524	19.98380	47.23247	1.5145044	0.1453375	0.6586884

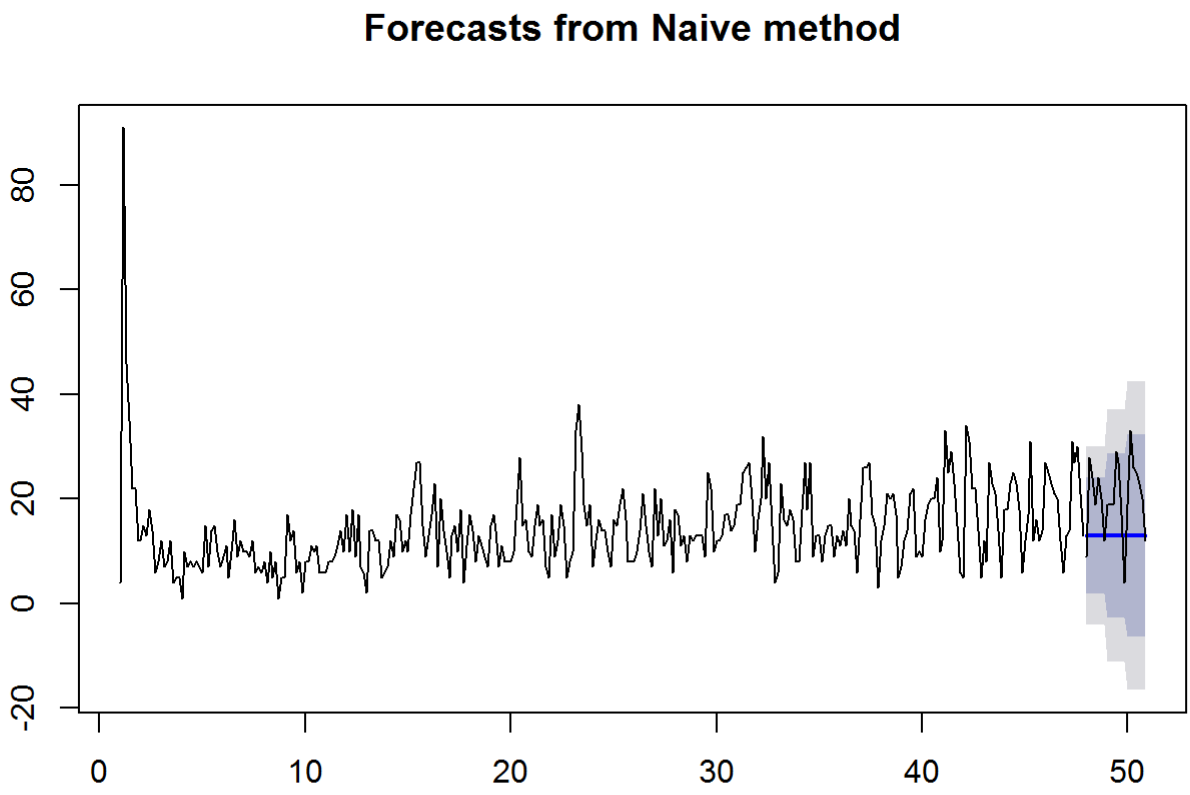
```
hist(valid.issues.ts - train.issues.naive.pred$mean)
```




```
plot(valid.issues.ts - train.issues.naive.pred$mean)
```



```
plot(train.issues.naive.pred)
lines(valid.issues.ts)
```

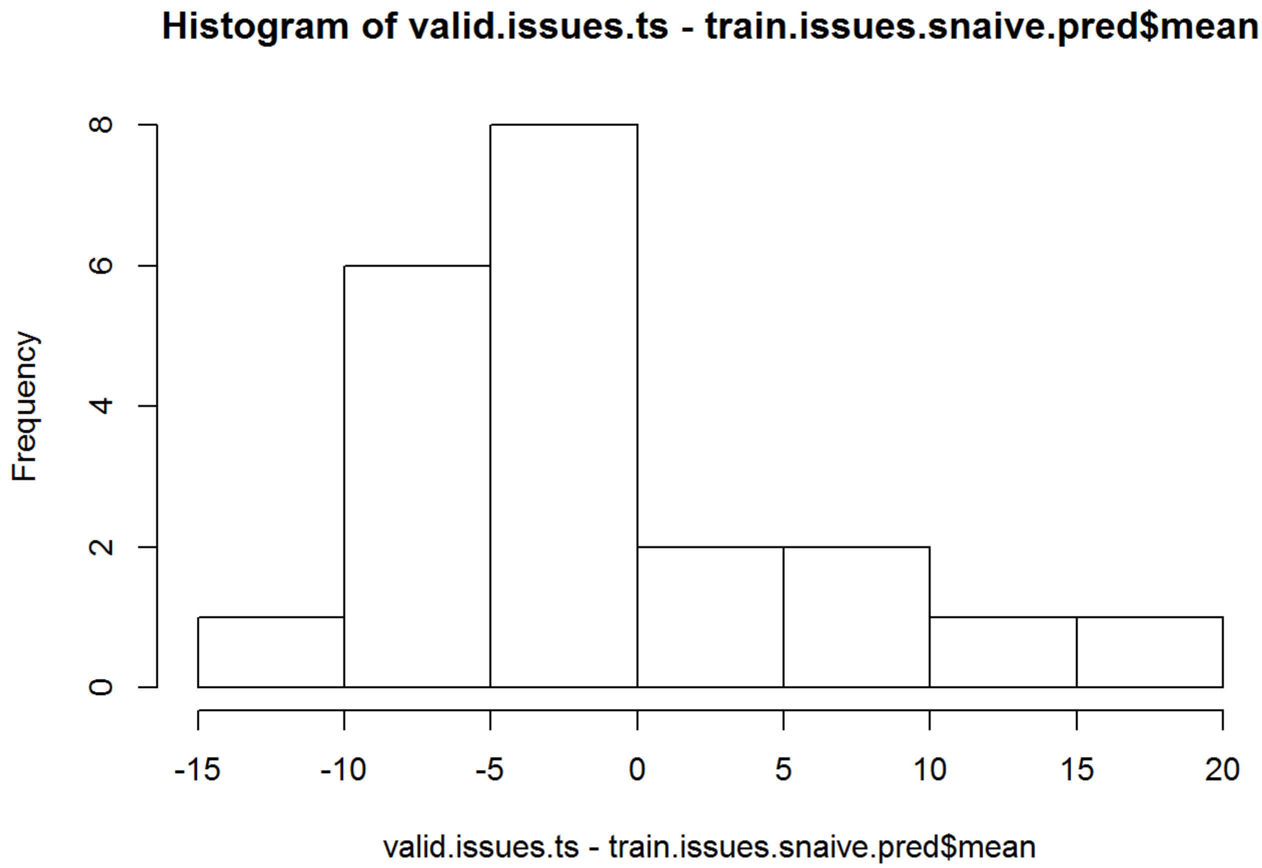


Seasonal Naive

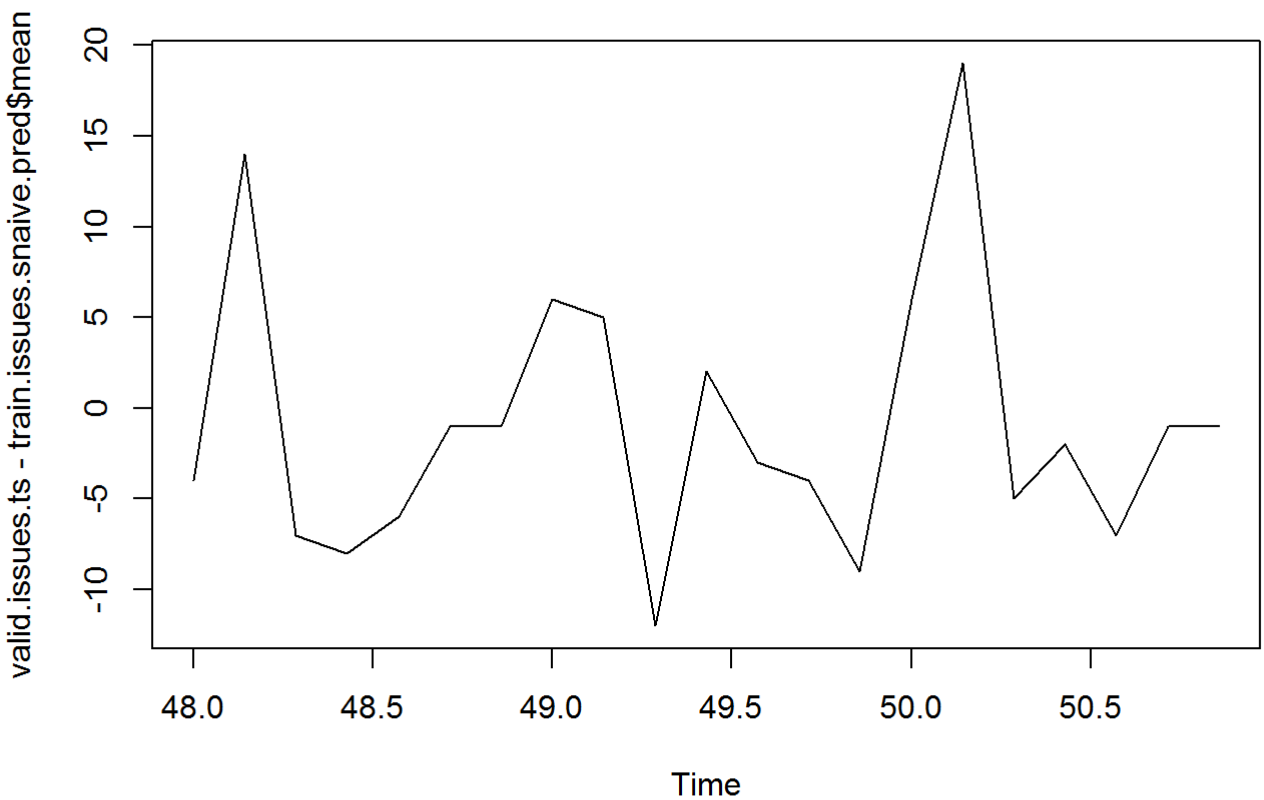
```
train.issues.snaive.pred <- snaive(train.issues.ts, h=n.valid)
kable(accuracy(train.issues.snaive.pred, valid.issues.ts))
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	-0.2577640	8.519798	5.816770	-22.20475	54.82299	1.000000	0.3057894	NA
Test set	-0.9047619	7.416199	5.857143	-16.37624	35.79967	1.006941	-0.0681004	0.5821032

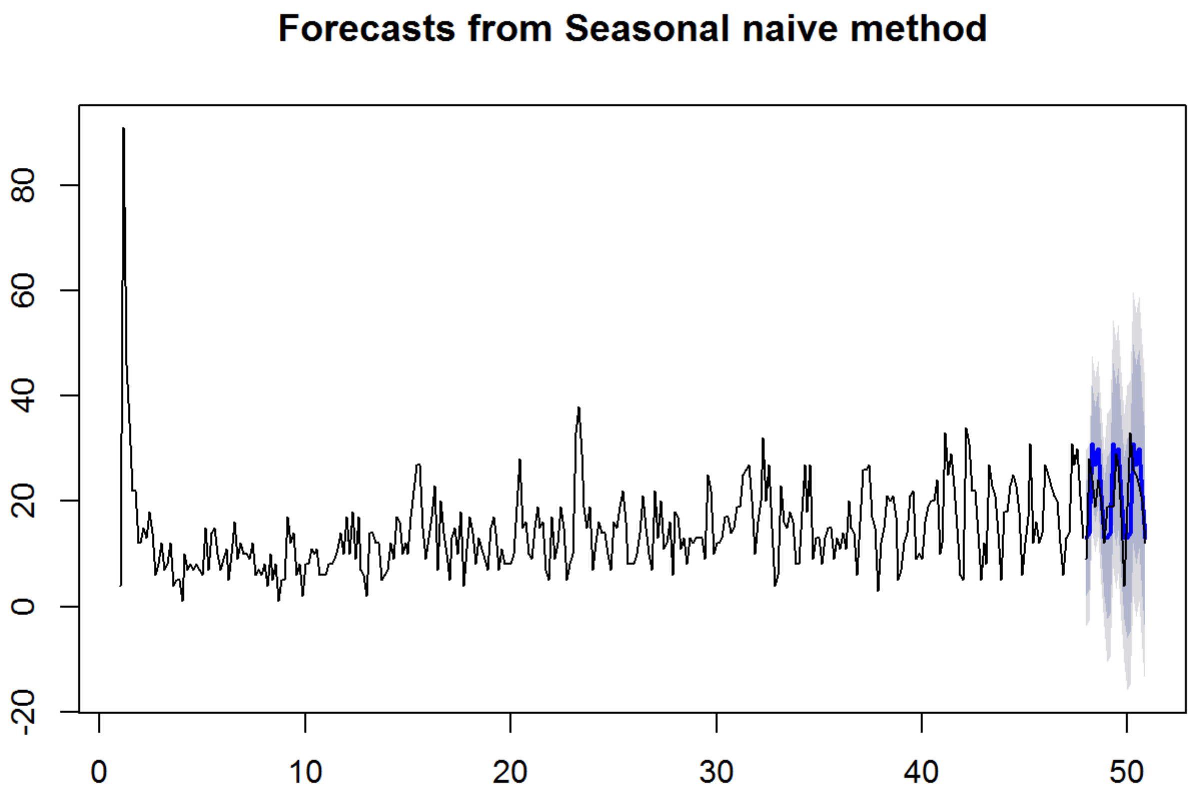
```
hist(valid.issues.ts - train.issues.snaive.pred$mean)
```




```
plot(valid.issues.ts - train.issues.snaive.pred$mean)
```



```
plot(train.issues.snaive.pred)
lines(valid.issues.ts)
```



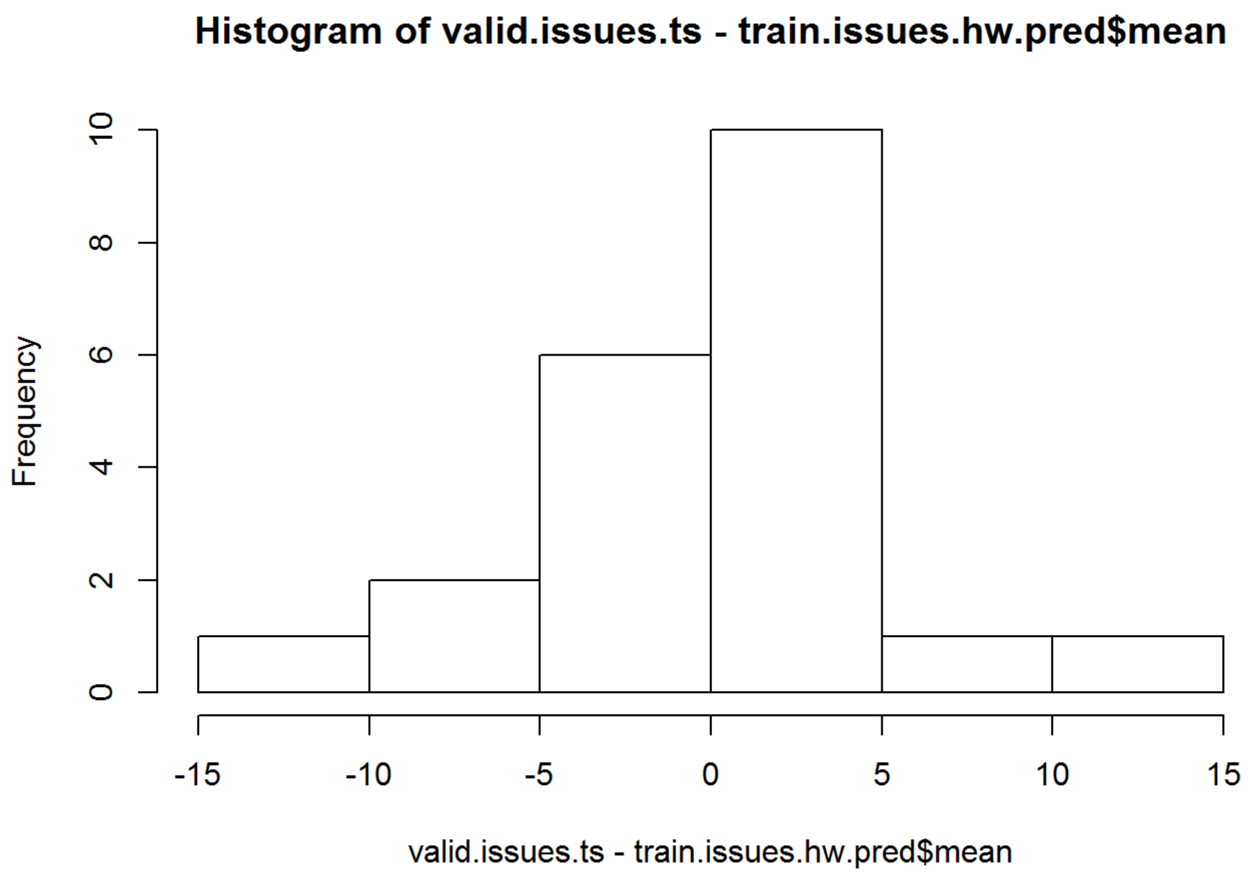
Smoothing

Holt Winter

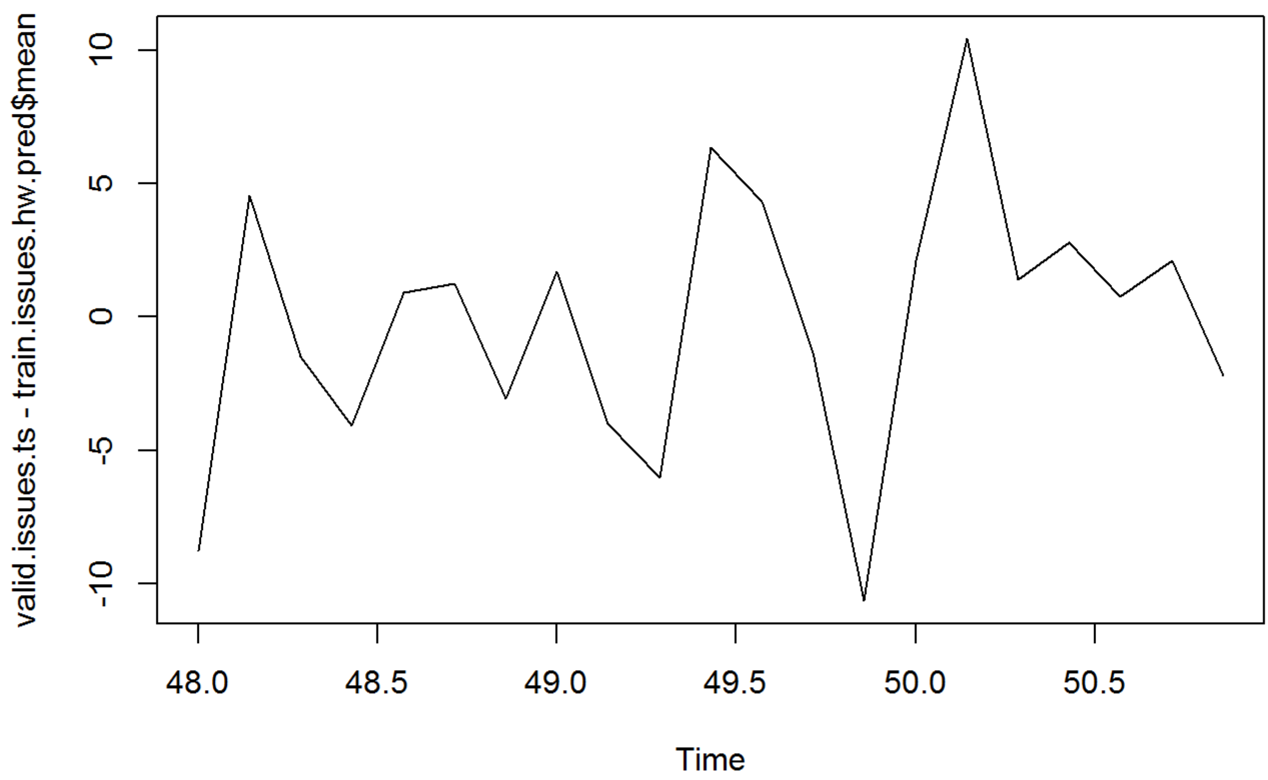
```
train.issues.hw.pred <- hw(train.issues.ts, hw = "ZAA", h = n.valid)
kable(accuracy(train.issues.hw.pred, valid.issues.ts))
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	0.0490254	6.728816	4.608885	-17.34177	43.16867	0.7923444	0.0172489	NA
Test set	-0.1403043	4.831118	3.822583	-16.63936	30.55990	0.6571659	-0.0384511	0.2786967

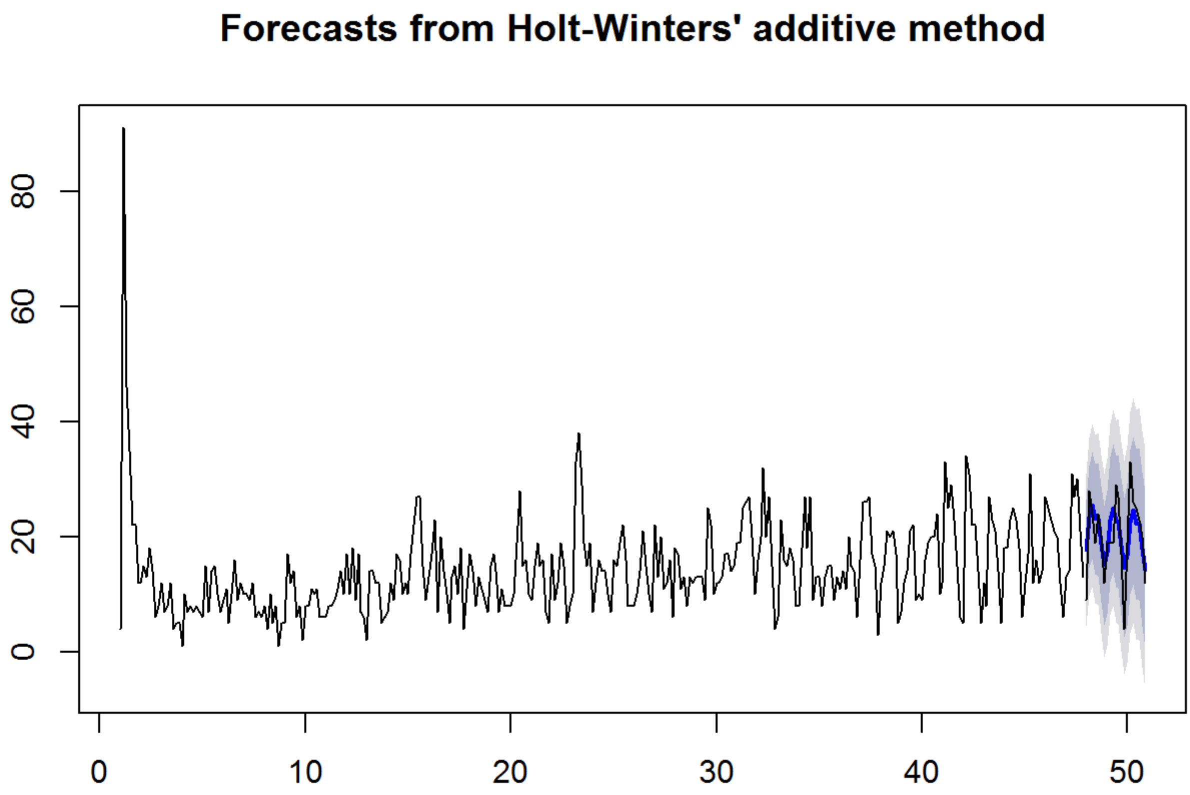
```
hist(valid.issues.ts - train.issues.hw.pred$mean)
```




```
plot(valid.issues.ts - train.issues.hw.pred$mean)
```



```
plot(train.issues.hw.pred)
lines(valid.issues.ts)
```

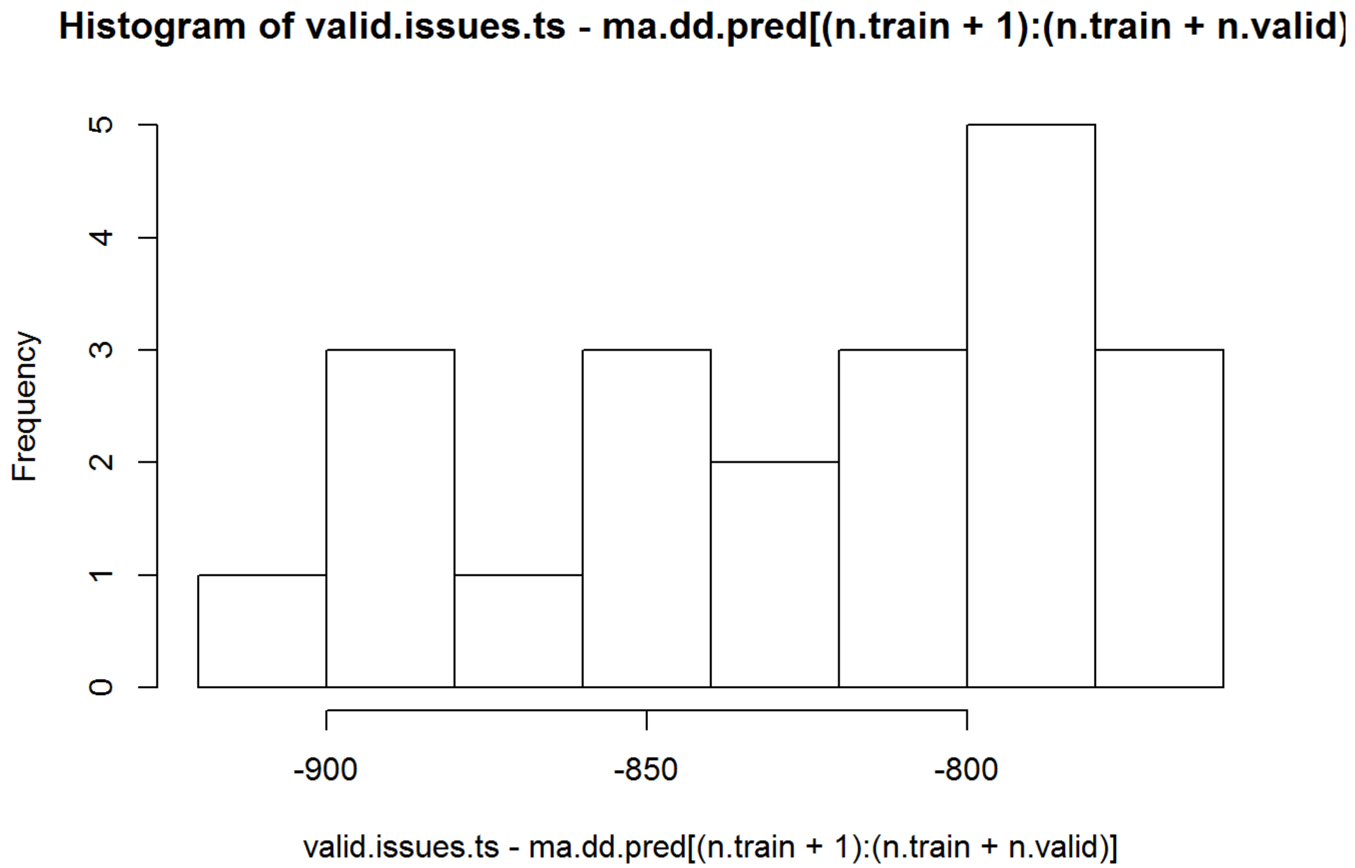


Double differencing

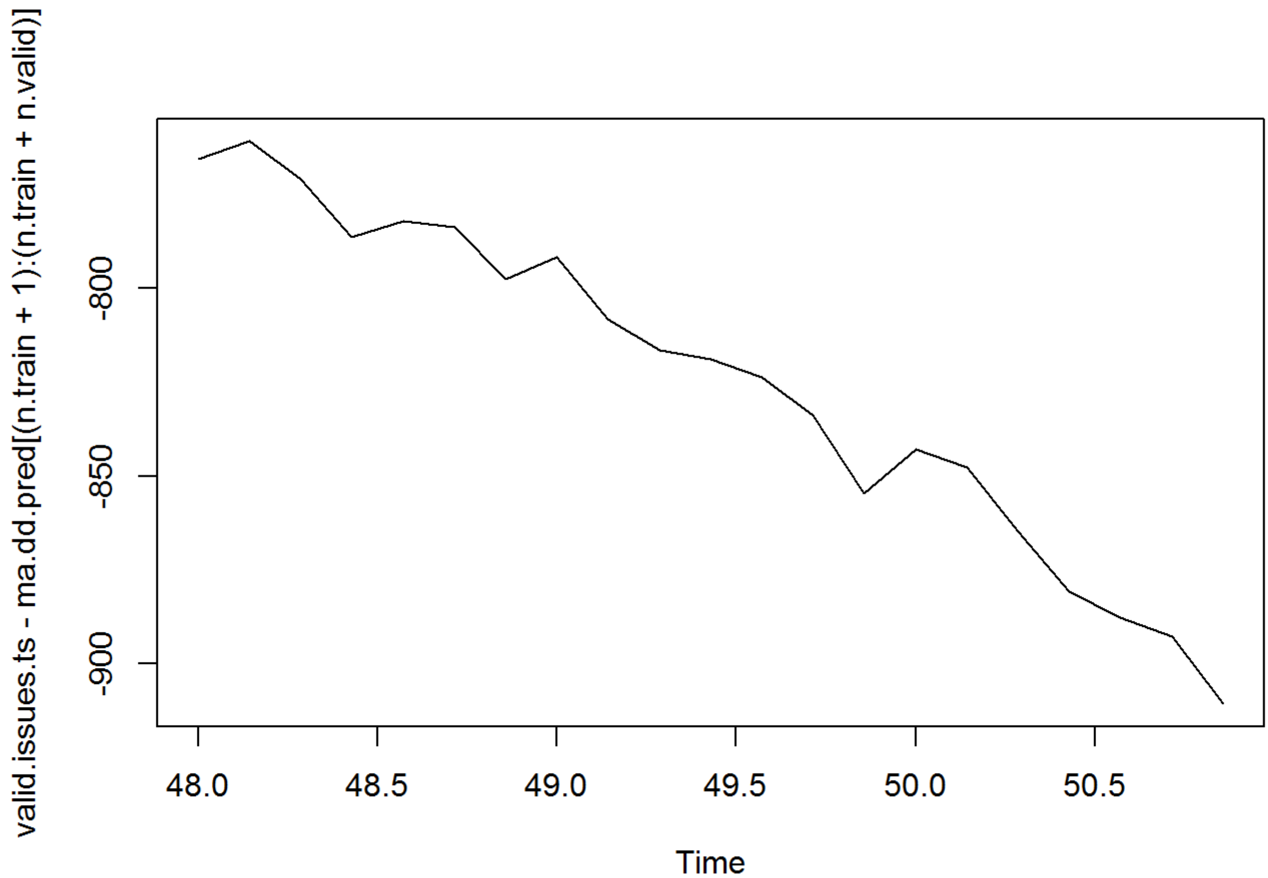
```
train.issues.d1 <- diff(train.issues.ts, lag = 1)
train.issues.d1.d7 <- diff(train.issues.d1, lag = 7)

ma.trailing <- rollmean(train.issues.d1.d7, k = 7, align = "right")
last.ma <- tail(ma.trailing, 1)
ma.trailing.pred <- ts(c(train.issues.d1.d7[1:6], ma.trailing, rep(last.ma, n.valid)),
start=c(2,2), frequency = 7)

ma.dd.pred.d1 <- diffinv(ma.trailing.pred, lag = 7, xi=train.issues.d1[1:7])
ma.dd.pred <- diffinv(ma.dd.pred.d1, lag = 1, xi=train.issues.ts[1])

kable(accuracy(ma.dd.pred[(n.train+1):(n.train+n.valid)], valid.issues.ts))
```

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	-824.8435	825.9994	824.8435	-5096.455	5096.455	0.8318993	61.9973

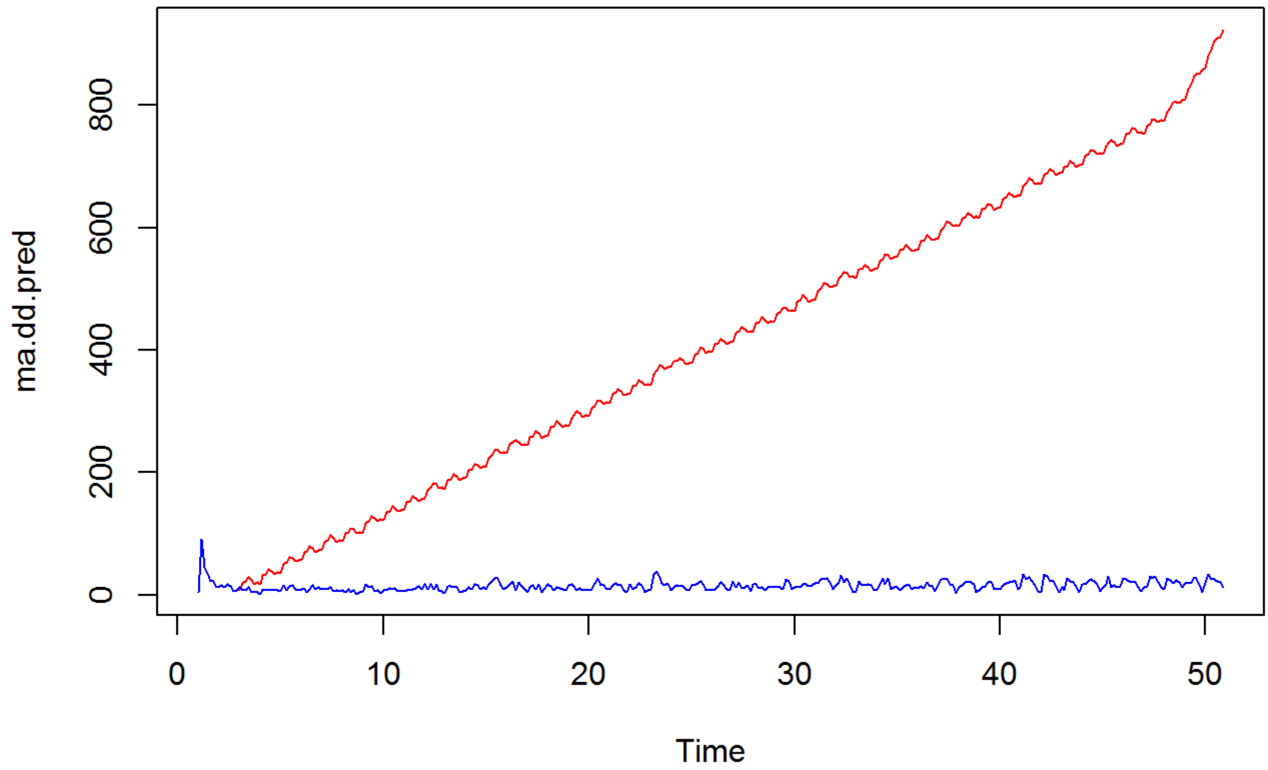
```
hist(valid.issues.ts - ma.dd.pred[(n.train+1):(n.train+n.valid)])
```




```
plot(valid.issues.ts - ma.dd.pred[(n.train+1):(n.train+n.valid)])
```



```
plot(ma.dd.pred,col='red')
lines(issues.ts,col='blue')
```



Regression

Linear additive regression

```
train.issues.linear.regr.add.m <- tslm(train.issues.ts ~ trend + season)
train.issues.linear.regr.add.m
```

```
##
## Call:
## tslm(formula = train.issues.ts ~ trend + season)
##
## Coefficients:
## (Intercept)        trend      season2      season3      season4
##     7.34596      0.02334      5.63623      6.86821      5.99380
##     season5      season6      season7
##     5.54492      0.62796     -2.90602
```

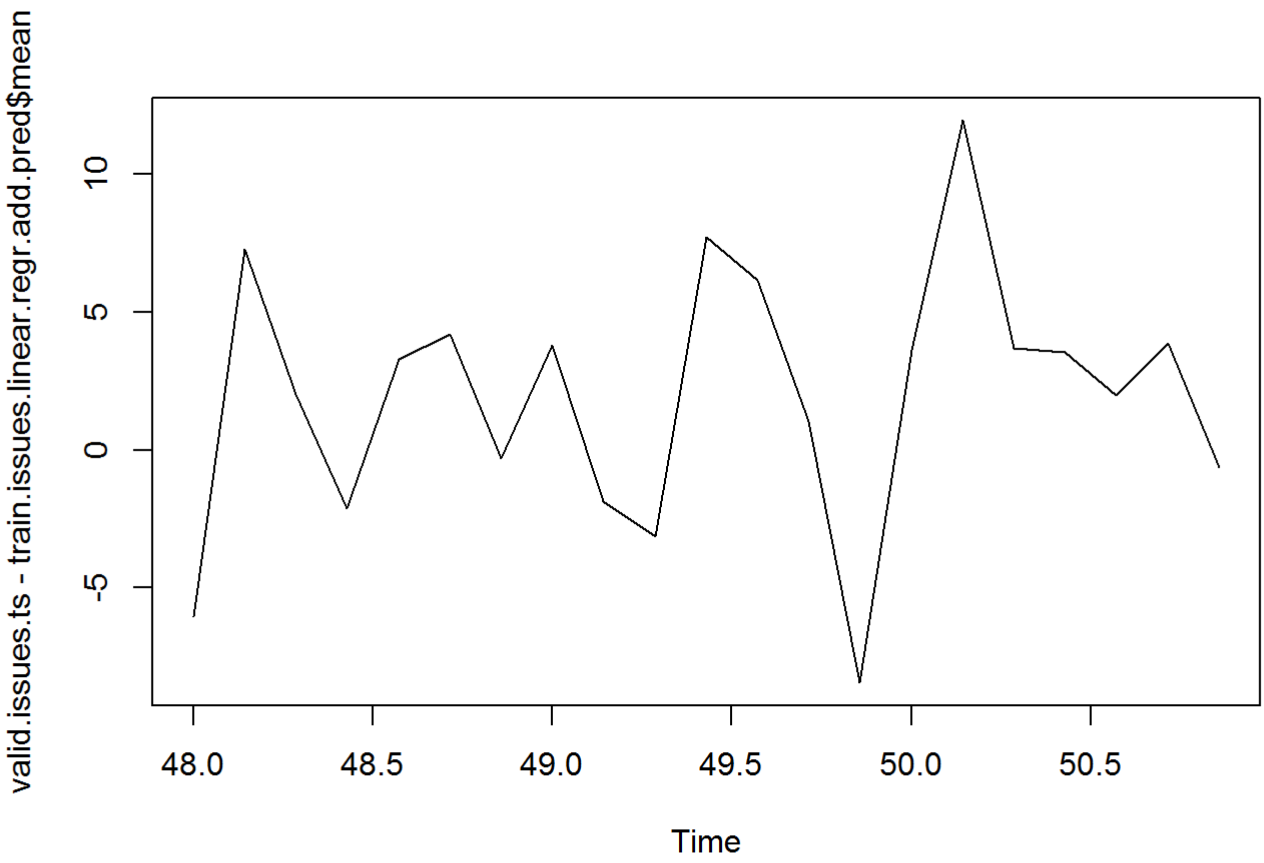
```
train.issues.linear.regr.add.pred <- forecast(train.issues.linear.regr.add.m , h=n.valid)

kable(accuracy(train.issues.linear.regr.add.pred, valid.issues.ts))
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U

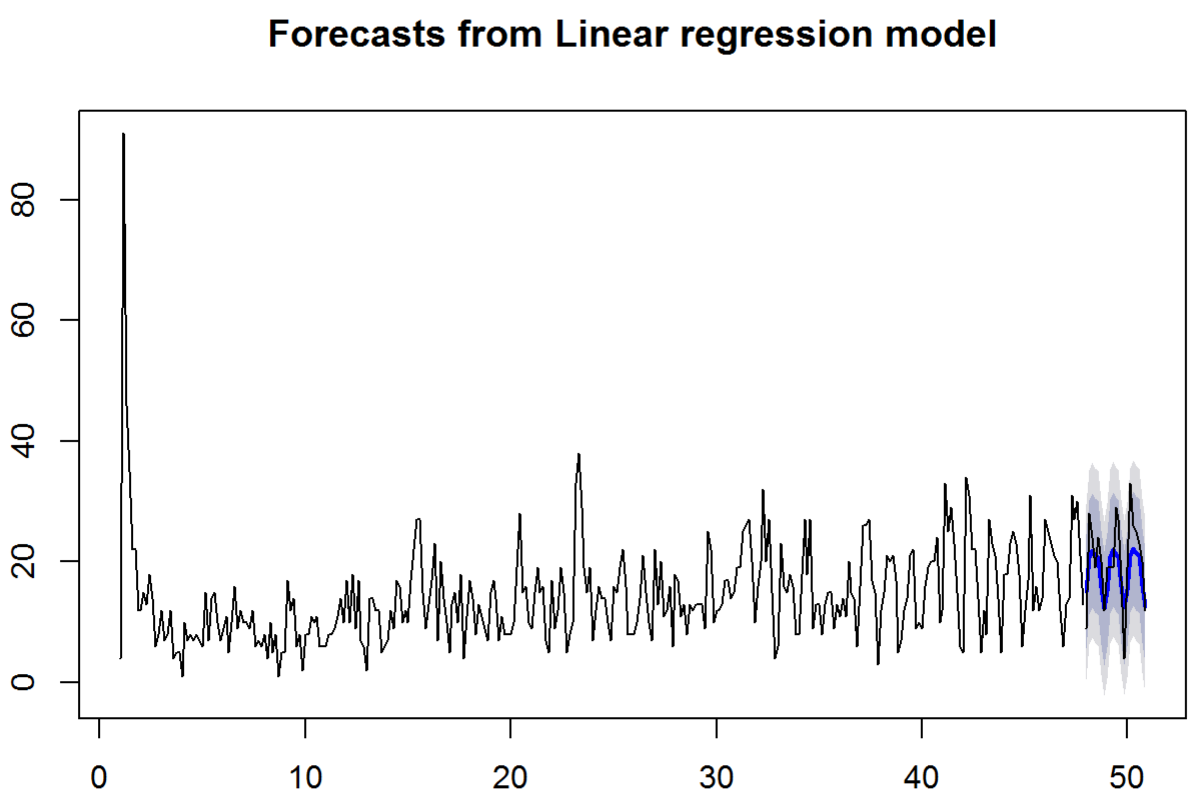
Training set	0.000000	7.153206	4.596331	-24.24642	44.30184	0.7901861	0.3743566	NA
Test set	1.988789	5.008067	4.131973	-3.17105	27.61700	0.7103553	-0.0602594	0.3528974

```
hist(valid.issues.ts - train.issues.linear.regr.add.pred$mean)
```


```
plot(valid.issues.ts - train.issues.linear.regr.add.pred$mean)
```




```
plot(train.issues.linear.regr.add.pred)
lines(valid.issues.ts)
```



linear multiplicative regression

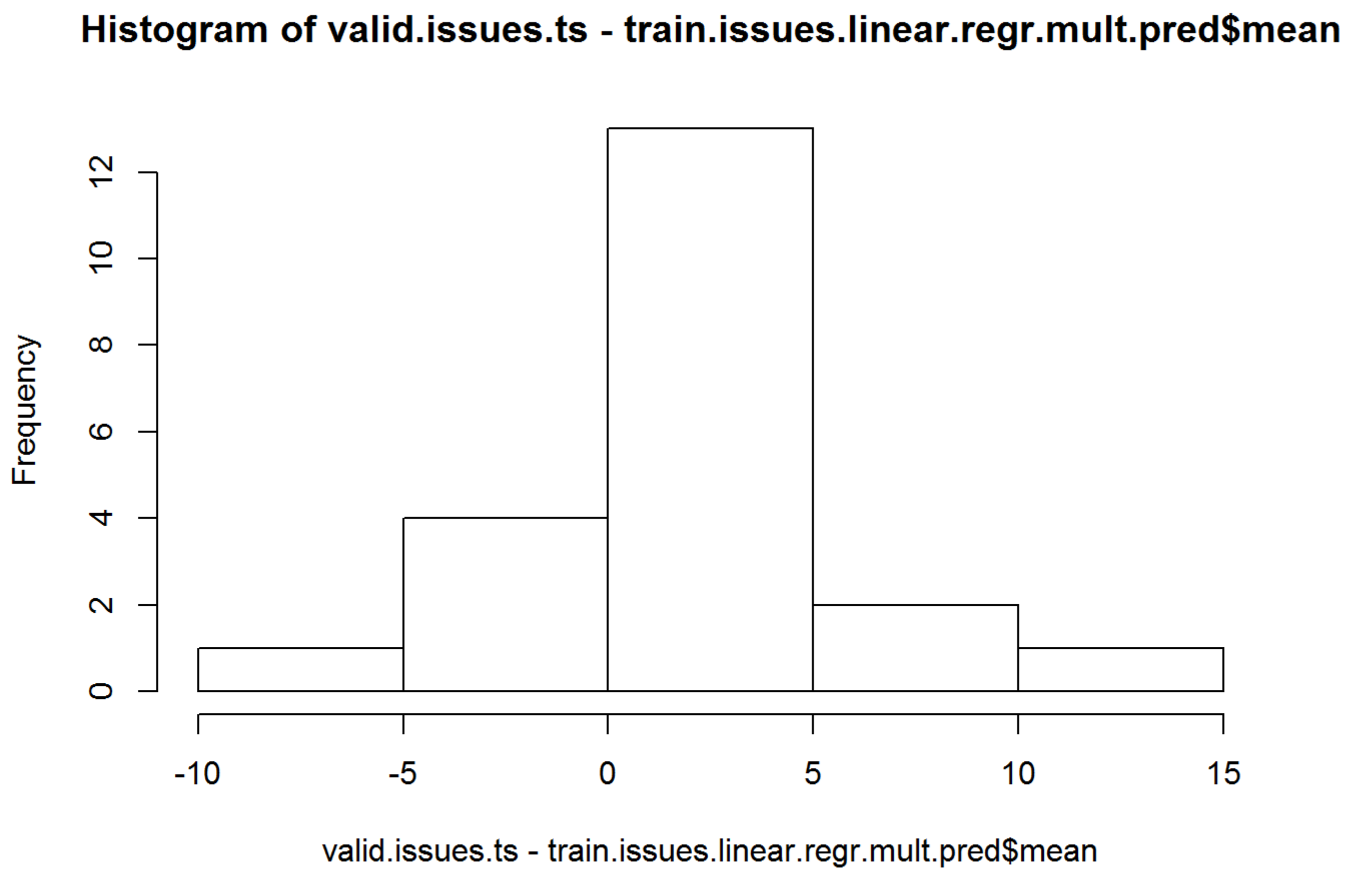
```
train.issues.linear.regr.mult.m <- tslm(train.issues.ts ~ trend + season, lambda = 0)
train.issues.linear.regr.mult.m
```

```
##
## Call:
## tslm(formula = train.issues.ts ~ trend + season, lambda = 0)
##
## Coefficients:
## (Intercept)        trend      season2      season3      season4
##    1.925750     0.002127     0.400373     0.499783     0.476474
##     season5      season6      season7
##    0.458874     0.070092    -0.246390
```

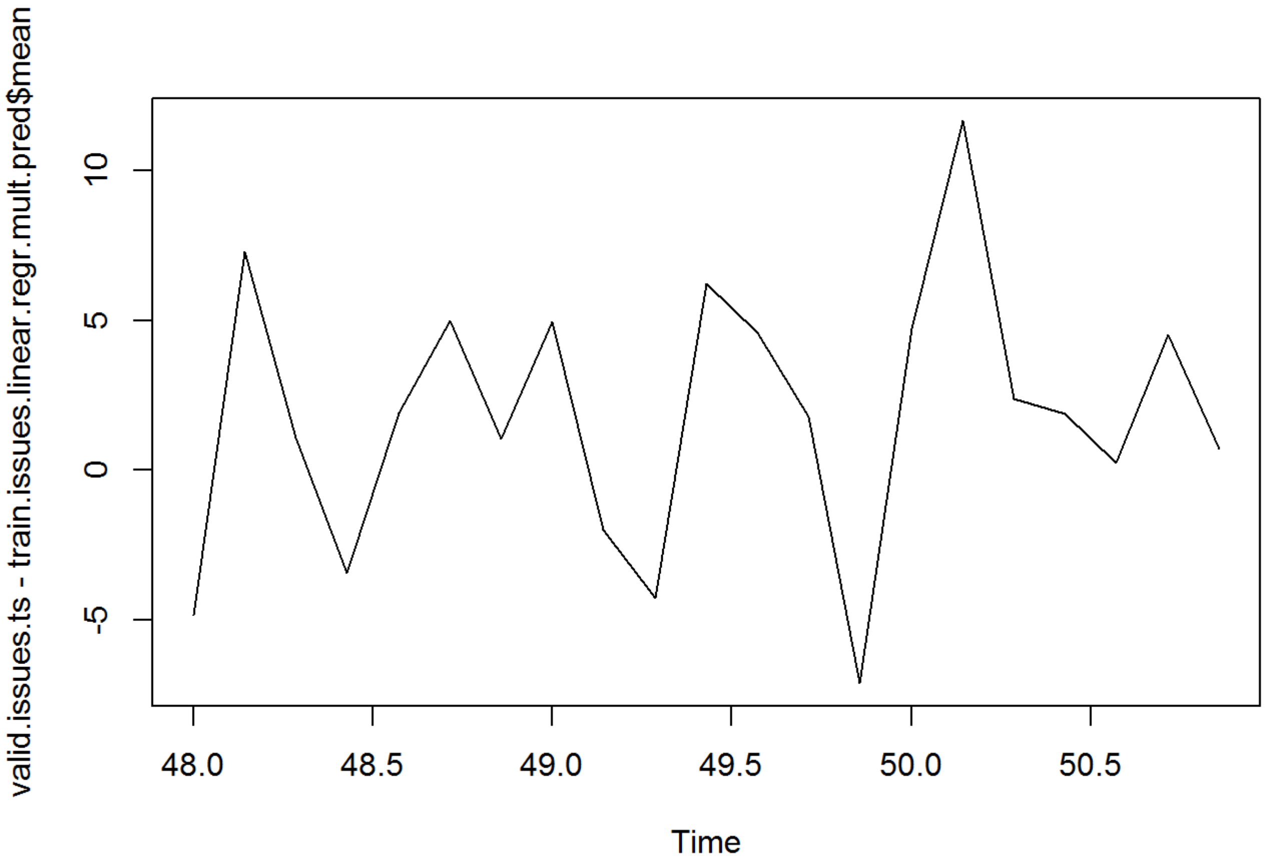
```
train.issues.linear.regr.mult.pred <- forecast(train.issues.linear.regr.mult.m , h=n.valid)

kable(accuracy(train.issues.linear.regr.mult.pred, valid.issues.ts))
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	1.258735	7.219268	4.349847	-12.087206	37.99997	0.7478114	0.3673273	NA
Test set	1.827121	4.724967	3.890974	-1.386916	25.56117	0.6689235	-0.1101274	0.3853172

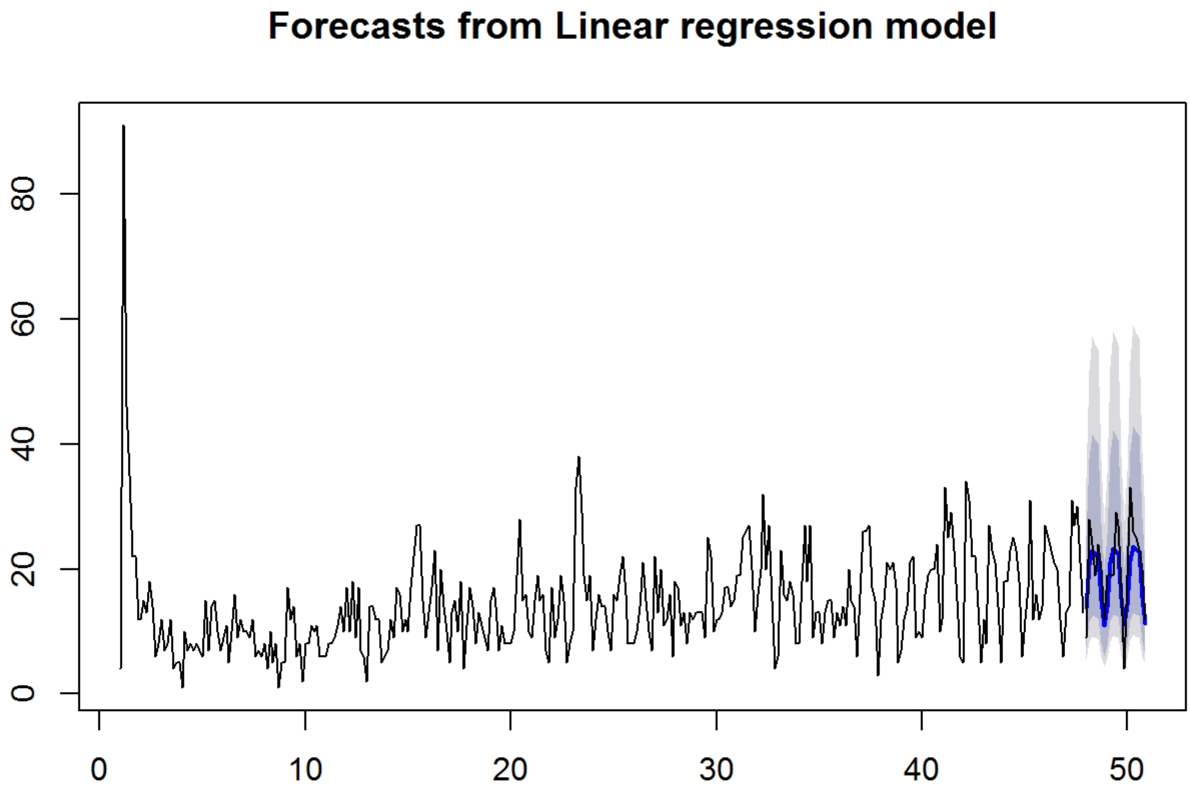
```
hist(valid.issues.ts - train.issues.linear.regr.mult.pred$mean)
```



```
plot(valid.issues.ts - train.issues.linear.regr.mult.pred$mean)
```

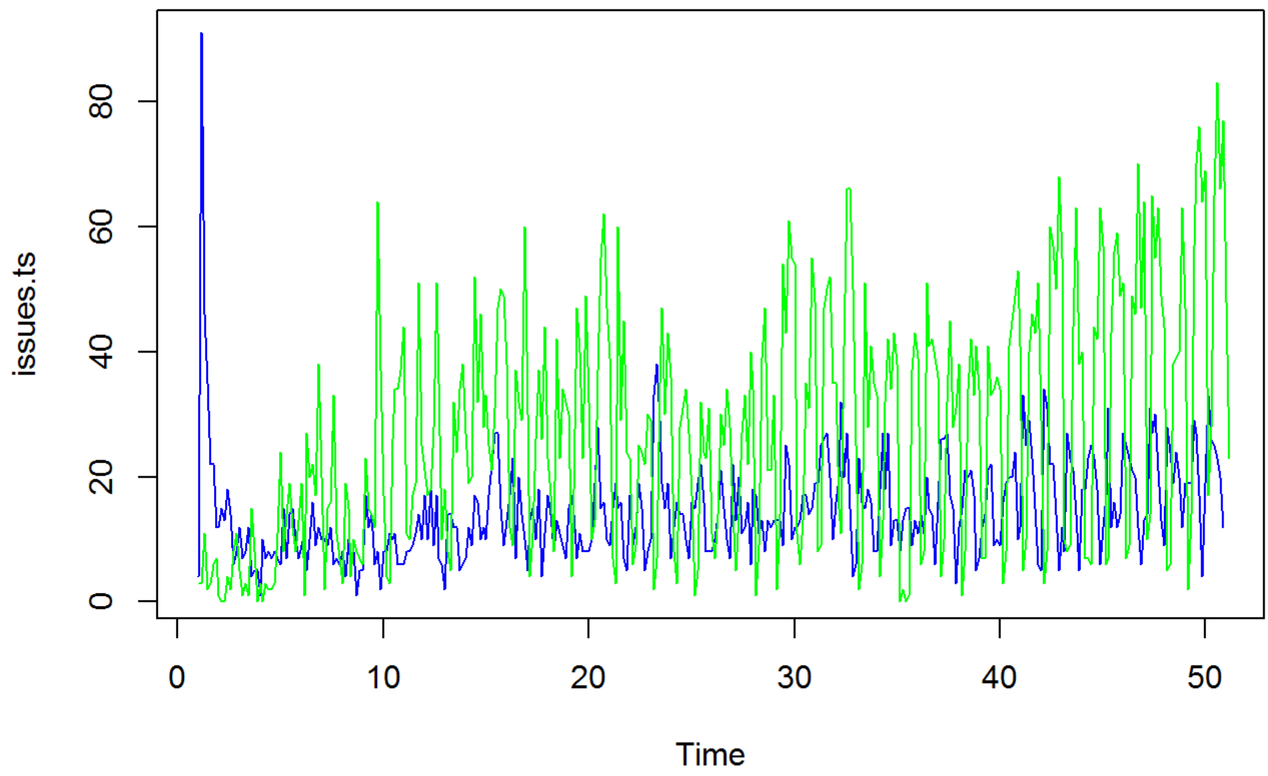



```
plot(train.issues.linear.regr.mult.pred)
lines(valid.issues.ts)
```

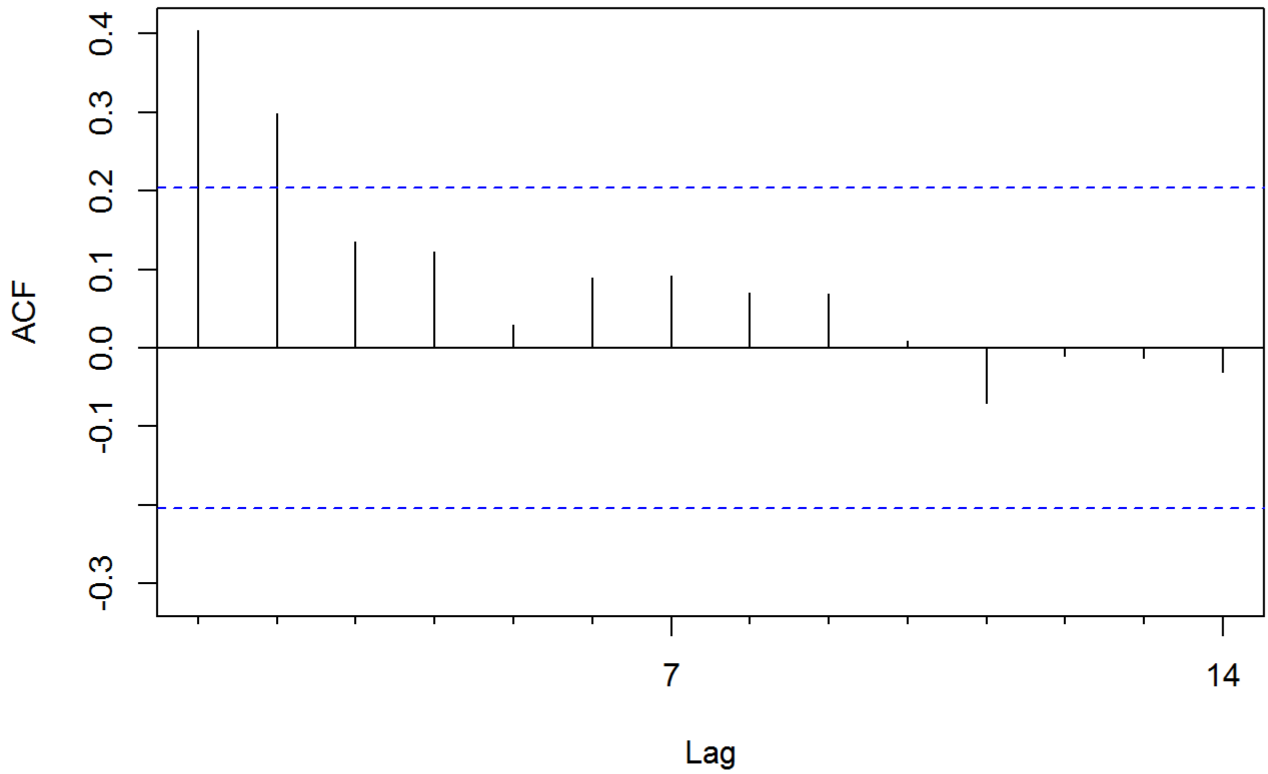


external regression

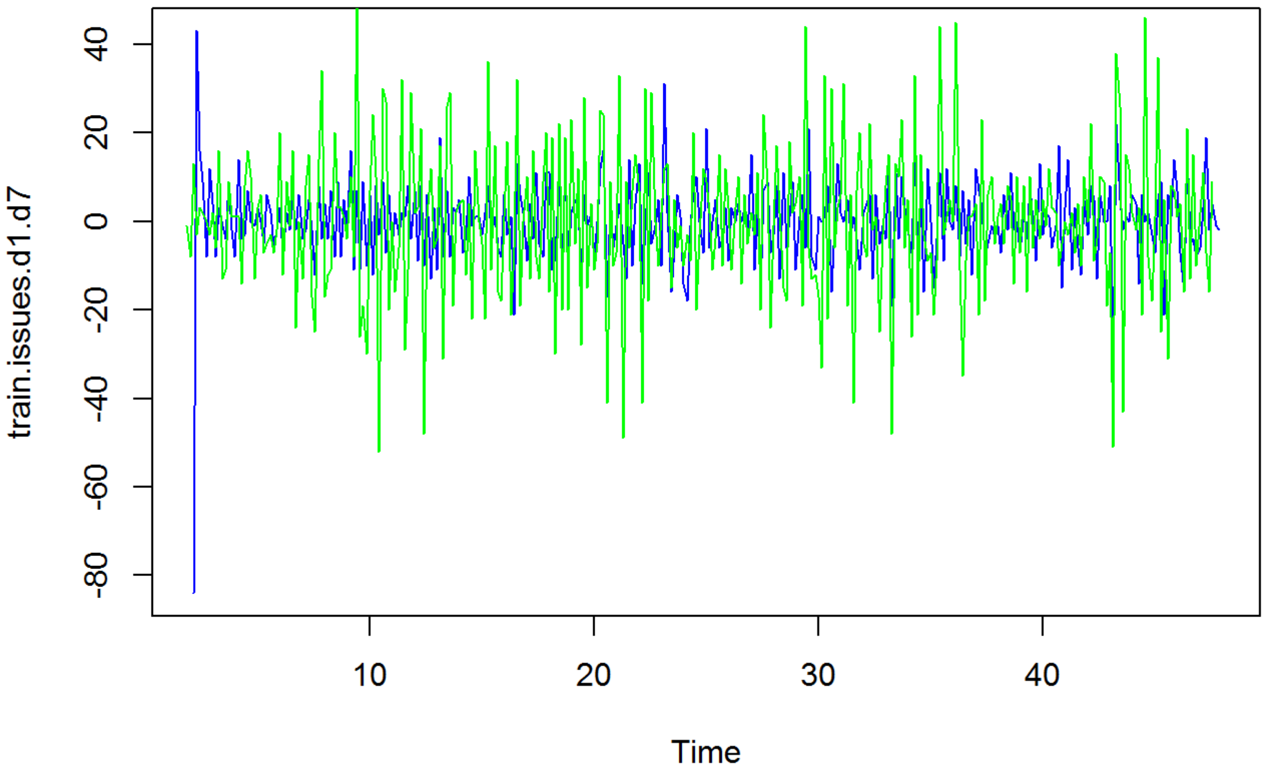
```
plot(issues.ts, col='blue')
lines(commits.ts, col='green')
```




```
issues.14.ts <- window(issues.ts, start = 1, end = 14)
Acf(issues.14.ts, lag.max = 14, main = "")
```

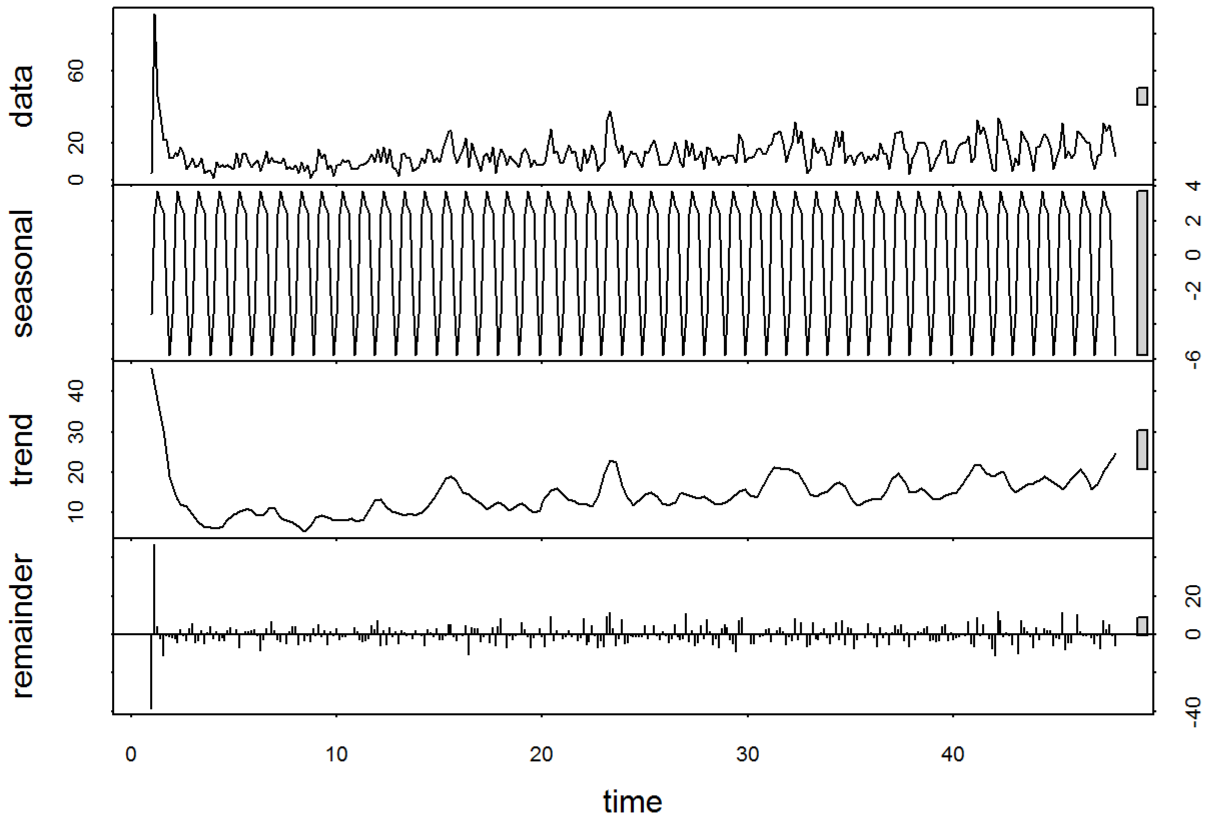


```
train.commits.d1 <- diff(train.commits.ts, lag = 1)
train.commits.d1.d7 <- diff(train.commits.d1, lag = 7)

plot(train.issues.d1.d7, col='blue')
lines(lag(train.commits.d1.d7,2), col='green')
```



external regression using comb.file, stl

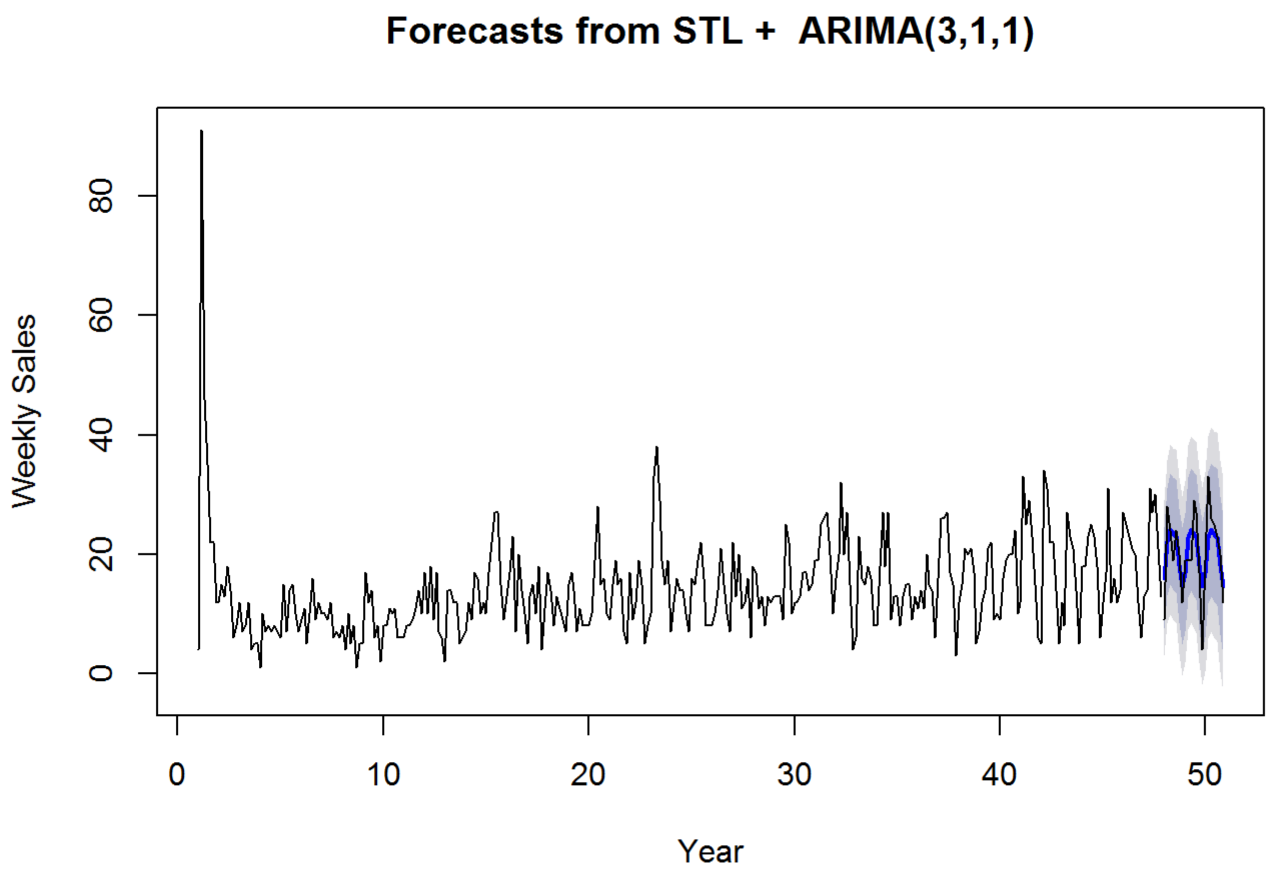
```
comb.issues.commits <- read.csv("issues/tensorflow_combined.csv")
yTrainexternal.ts <- ts(comb.issues.commits$number_of_issues[1:n.train], freq = 7, start = 1)
stl.trainexternal <- stl(yTrainexternal.ts, s.window = "periodic")
plot(stl.trainexternal)
```




```
xTrainIScommit <- data.frame(IsCommit = comb.issues.commits$IS_commit[1:n.train])
stlm.reg.fit <- stlm(yTrainexternal.ts, s.window = "periodic", xreg = xTrainIScommit, method
= "arima")

stlm.reg.fit$model
```

```
## Series: x
## ARIMA(3,1,1)
##
## Coefficients:
##          ar1     ar2      ar3      ma1  IsCommit
##       0.1197  0.1045  -0.1520  -0.8141    3.5996
## s.e.  0.1252  0.1002   0.0916   0.1092    2.9148
##
## sigma^2 estimated as 44.12:  log likelihood=-1084.45
## AIC=2180.89   AICc=2181.16   BIC=2203.65
```

```
xValidIScommit <- data.frame(IsCommit = comb.issues.commits$IS_commit[(n.train+1): (n.train +
 n.valid)])
stlm.reg.pred <- forecast(stlm.reg.fit, xreg = xValidIScommit, h = n.valid)
plot(stlm.reg.pred, xlab = "Year", ylab = "Weekly Sales")
lines(valid.issues.ts)
```




```
kable(accuracy(stlm.reg.pred, valid.issues.ts))
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	-0.1269172	6.581360	4.419095	-18.05605	41.05984	0.7597163	0.0917343	NA
Test set	0.0268061	4.625172	3.616953	-15.32520	29.14017	0.6218147	-0.0567364	0.2854848

ACF of raw shows lag-1 correl, but no seasonality

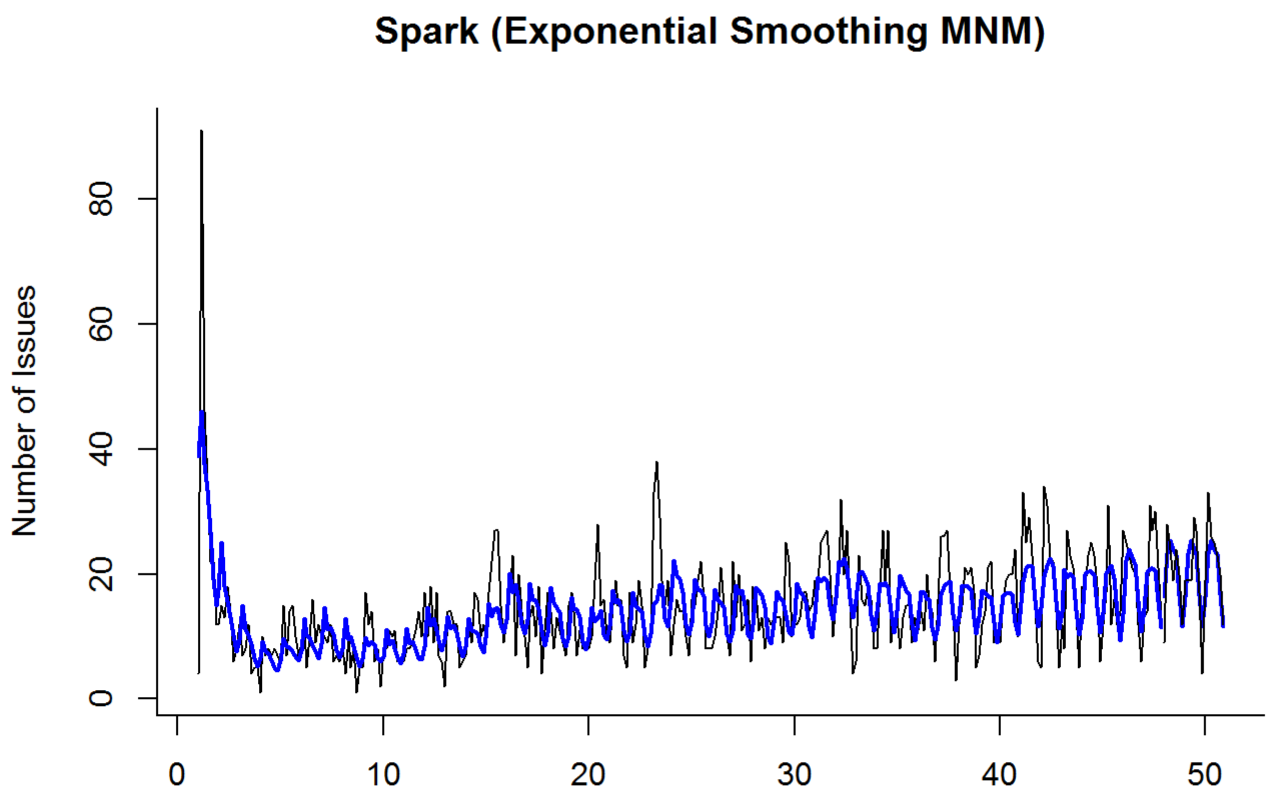
```
train.issues.arima.ext.m <- Arima(train.issues.ts, order=c(1,0,0), seasonal=c(1,0,0), xreg=tr
ain.commits.ts )
train.issues.arima.ext.m
```

```
## Series: train.issues.ts
## ARIMA(1,0,0)(1,0,0)[7] with non-zero mean
##
## Coefficients:
##          ar1    sar1  intercept  train.commits.ts
##       0.3873  0.3312    15.0726           -0.0202
## s.e.  0.0533  0.0709     1.2217            0.0301
##
## sigma^2 estimated as 51.4:  log likelihood=-1113.37
## AIC=2236.75   AICc=2236.93   BIC=2255.73
```

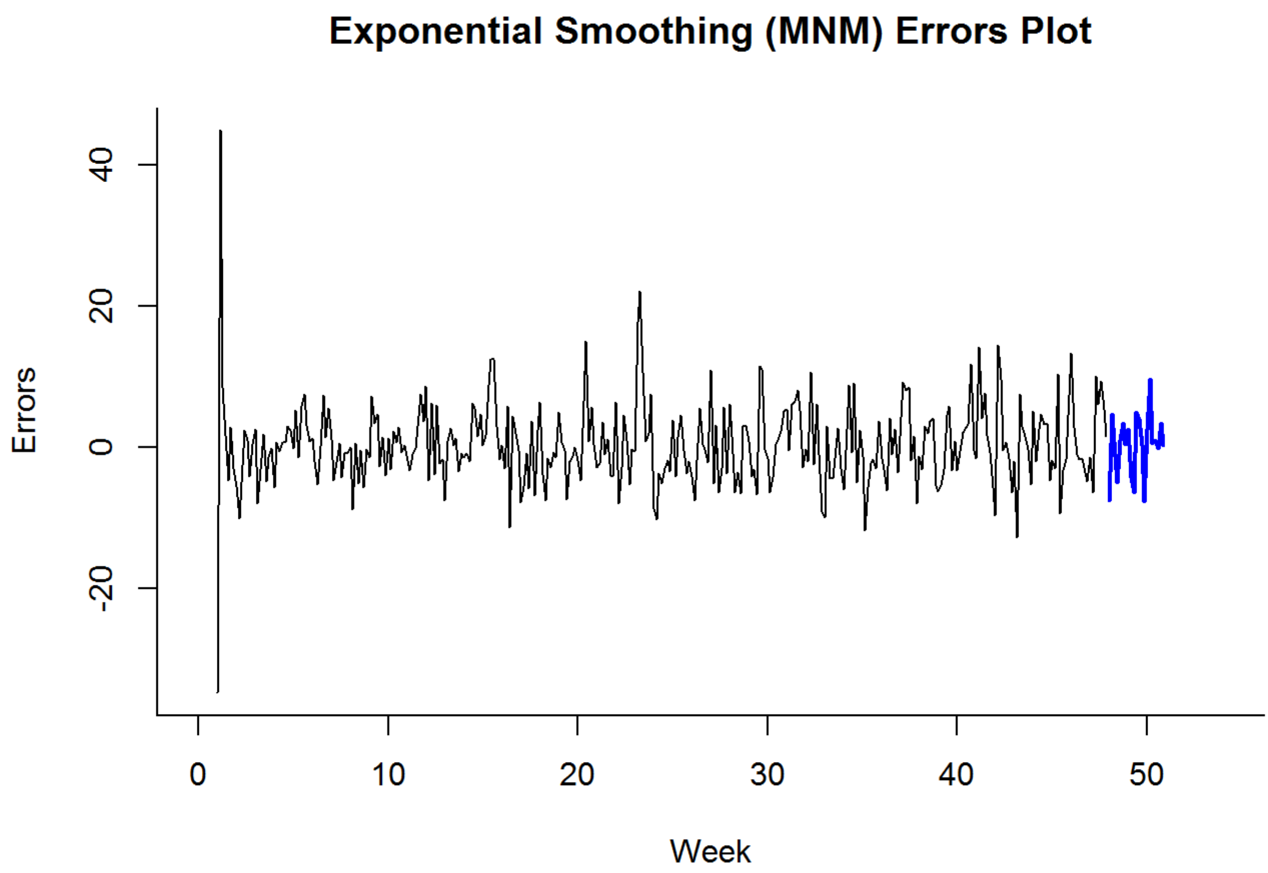
```
ets = ets(train.issues.ts, model = 'ZZZ', restrict = FALSE, allow.multiplicative.trend =
TRUE)
summary(ets)
```

```
## ETS(M,Ad,M)
##
## Call:
##  ets(y = train.issues.ts, model = "ZZZ", restrict = FALSE, allow.multiplicative.trend = TR
UE)
##
##   Smoothing parameters:
##     alpha = 0.1054
##     beta  = 1e-04
##     gamma = 0.034
##     phi   = 0.8466
##
##   Initial states:
##     l = 48.4921
##     b = -6.9831
##     s=0.6735 0.7979 0.9863 1.1128 1.1453 1.3726
##            0.9115
##
##   sigma:  0.398
##
##      AIC     AICc      BIC
## 3015.650 3016.806 3064.999
##
## Training set error measures:
##                      ME     RMSE     MAE       MPE     MAPE      MASE
## Training set 0.3665056 6.102004 4.25467 -18.85287 41.05495 0.7314489
##                     ACF1
## Training set 0.05822546
```

```
ets.pred = forecast(ets, h = n.valid, level = 0)

plot(ets.pred, main = 'Spark (Exponential Smoothing MNM)', bty = 'l', ylab = 'Number of Issue
s')
lines(ets.pred$fitted, lwd = 2, col = 'blue')
lines(valid.issues.ts)
```




```
plot(train.issues.ts - ets.pred$fitted, main = 'Exponential Smoothing (MNM) Errors Plot',
     bty = 'l', xlab = 'Week', ylab = 'Errors', xlim = c(0, 54))
lines(valid.issues.ts - ets.pred$mean, lwd = 2, col = 'blue')
```




```
kable(accuracy(ets.pred, valid.issues.ts))
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	0.3665056	6.102004	4.254670	-18.852866	41.05495	0.7314489	0.0582255	NA
Test set	0.3211297	4.335239	3.377791	-9.755863	24.69860	0.5806987	-0.0597373	0.2696809