Forcasting 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.

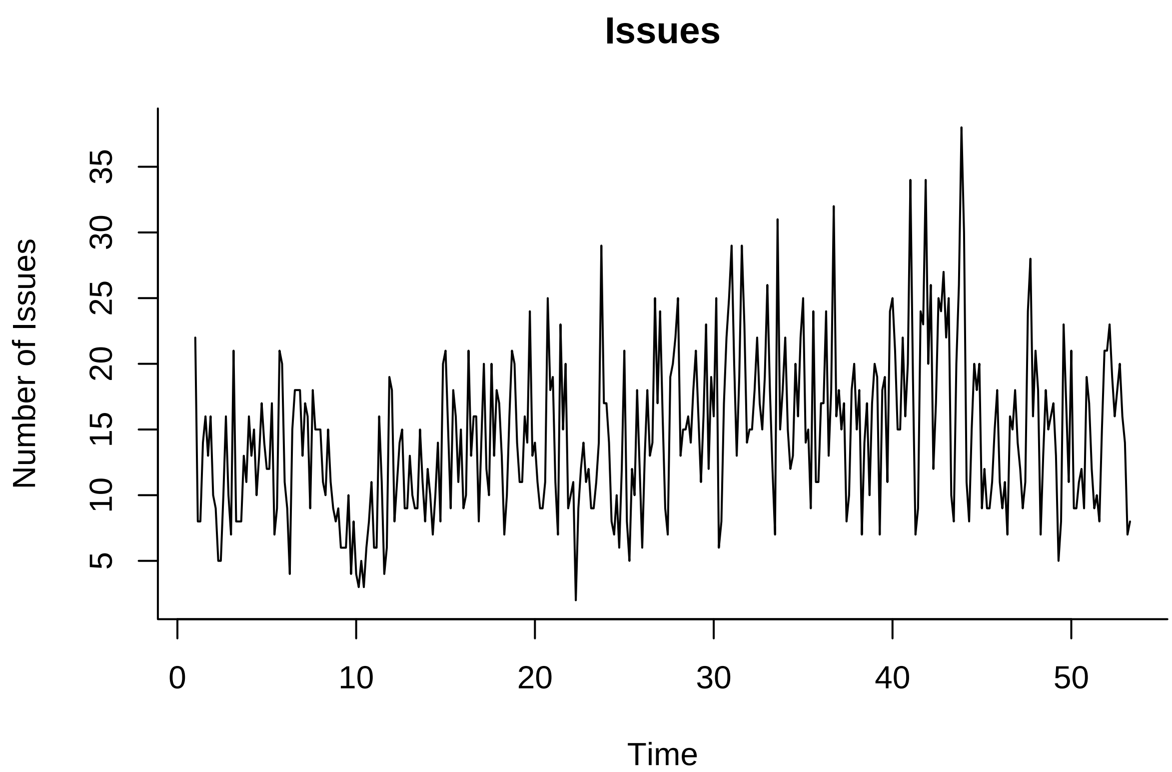
Load the data

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

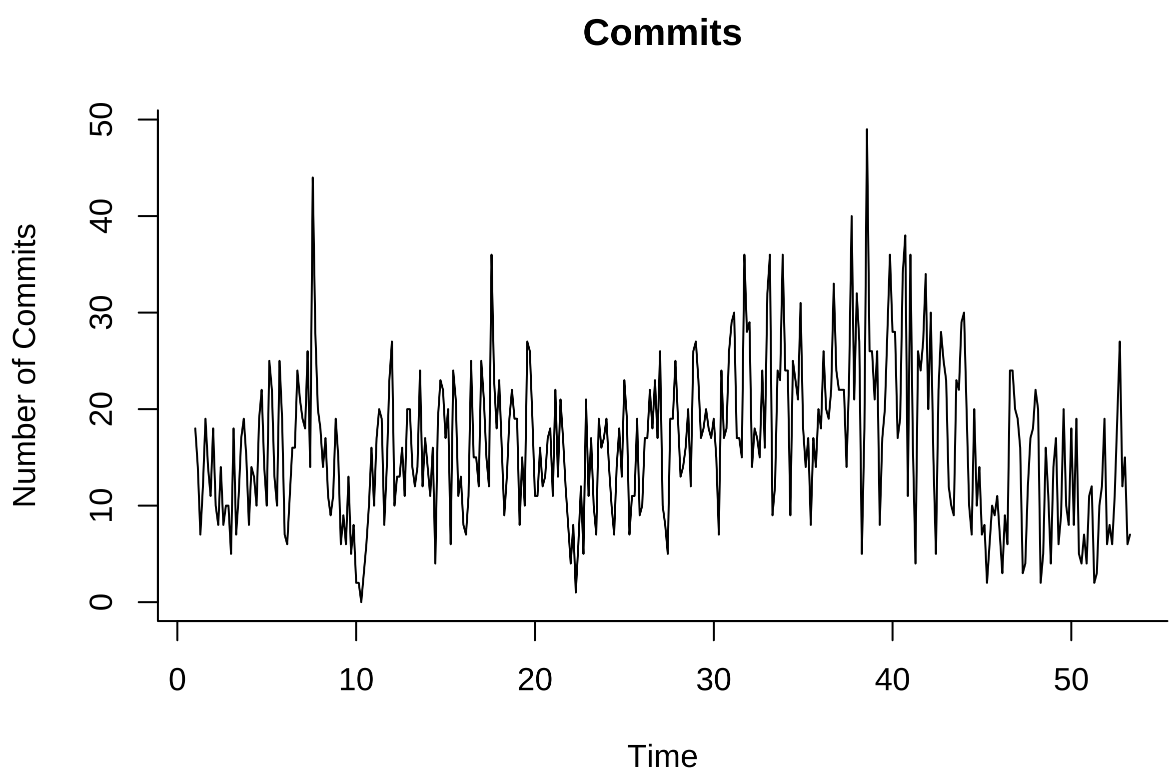
keep the last 12 months

```
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)
```

```r
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])
```

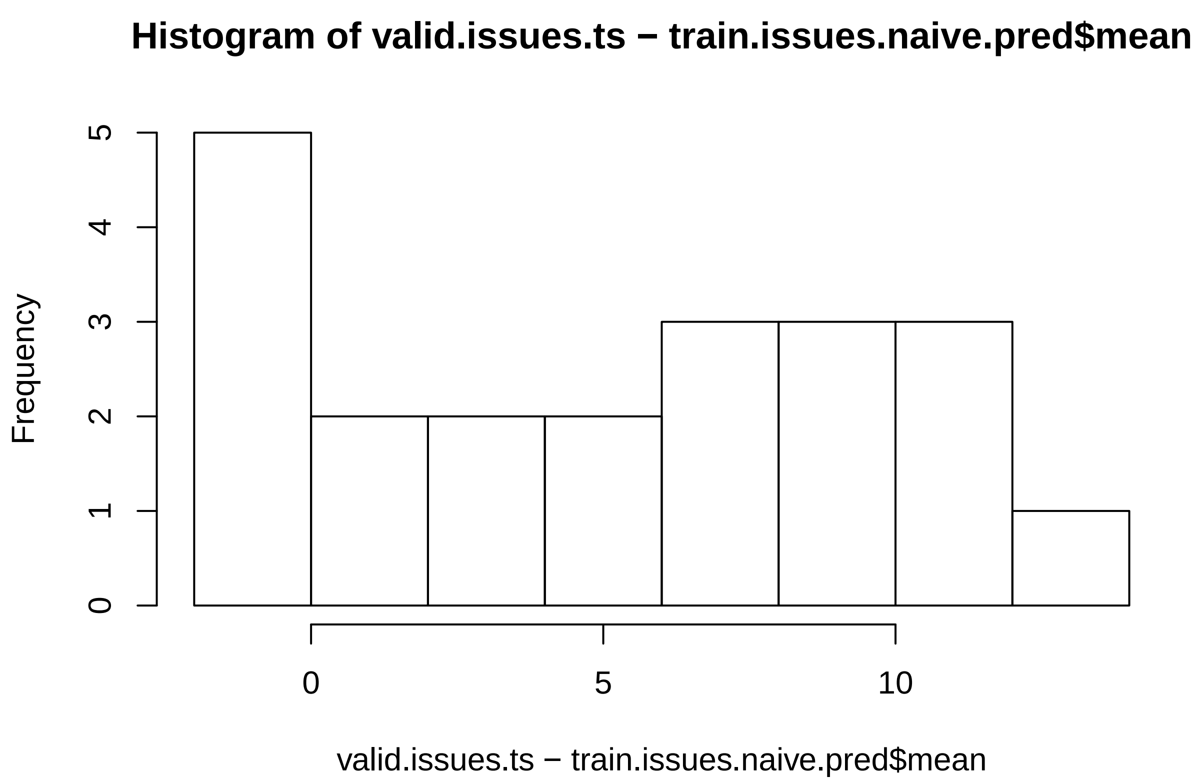
Naive Forecast

Naive

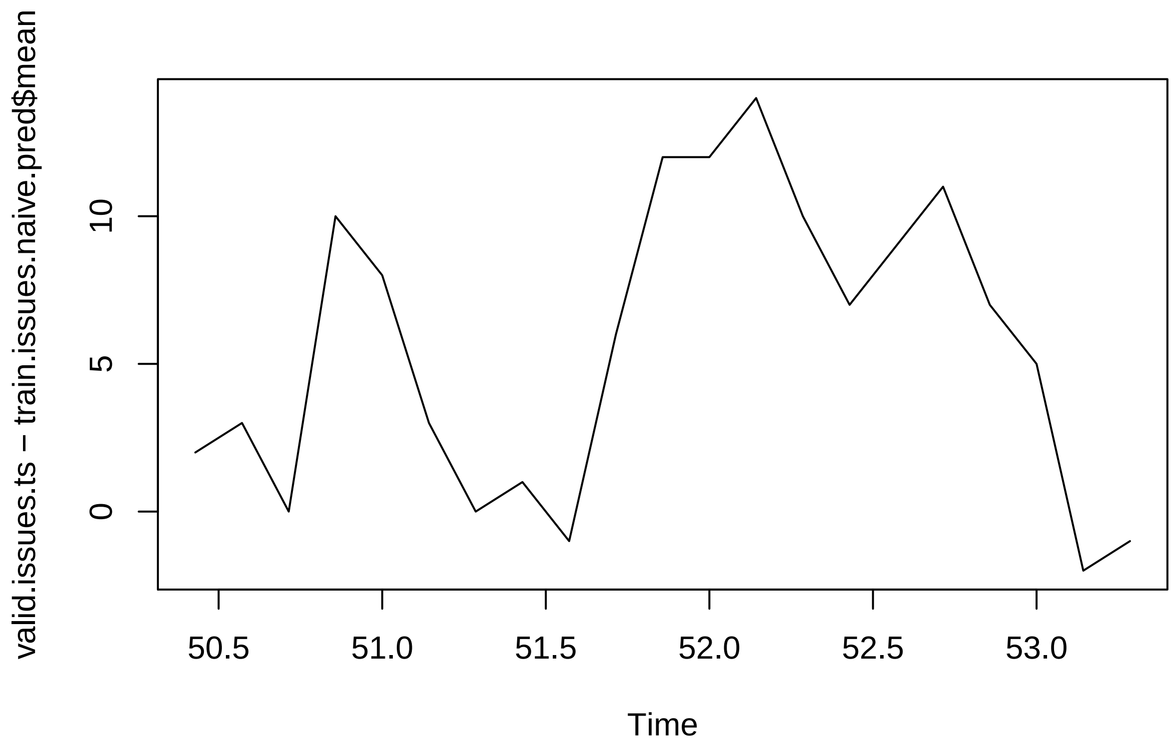
```r
train.issues.naive.pred <- naive(train.issues.ts, h=n.valid)
accuracy(train.issues.naive.pred, valid.issues.ts)
```

```
##                       ME     RMSE      MAE       MPE     MAPE     MASE
## Training set -0.03768116 6.718998 5.260870 -13.08450 42.03105 1.011591
## Test set      5.52380952 7.361418 5.904762  29.53819 34.64023 1.135402
##                    ACF1 Theil's U
## Training set -0.2812264        NA
## Test set      0.5978010  1.293618
```

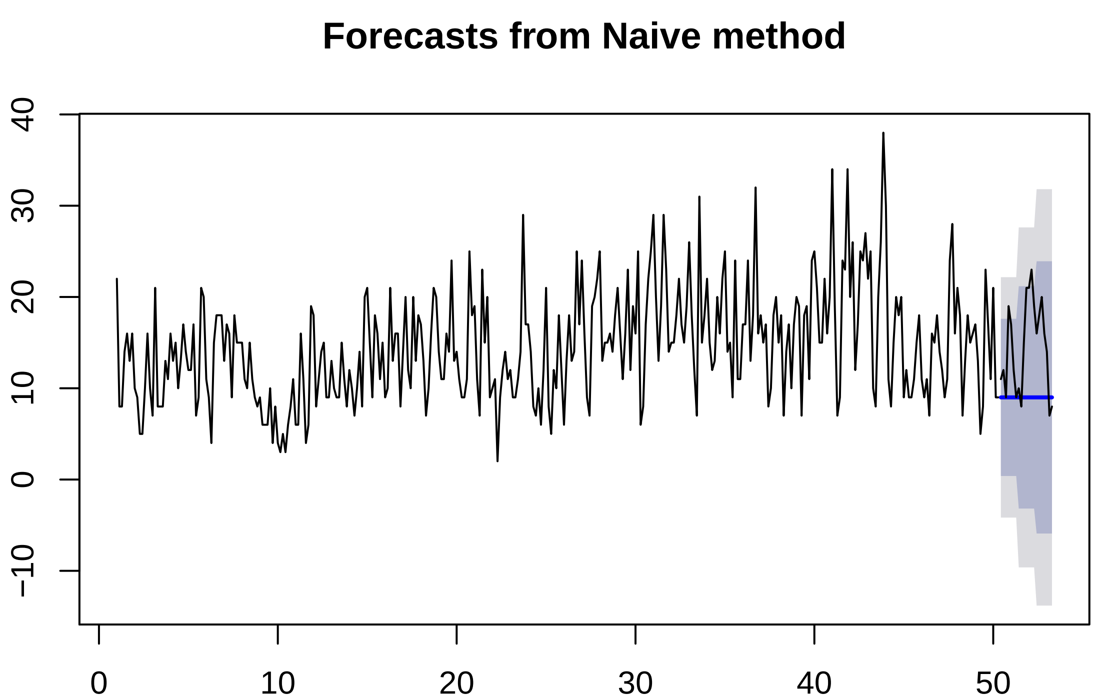
```r
hist(valid.issues.ts - train.issues.naive.pred$mean)
```



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




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

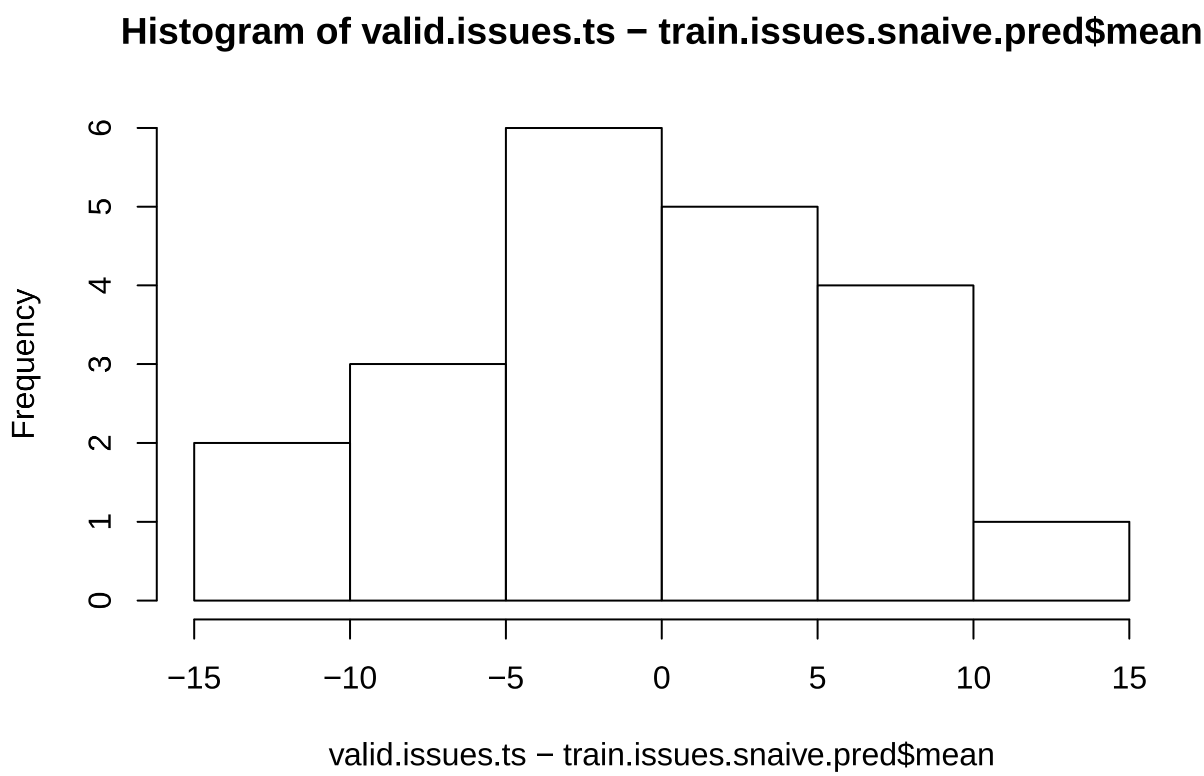


Seasonal Naive

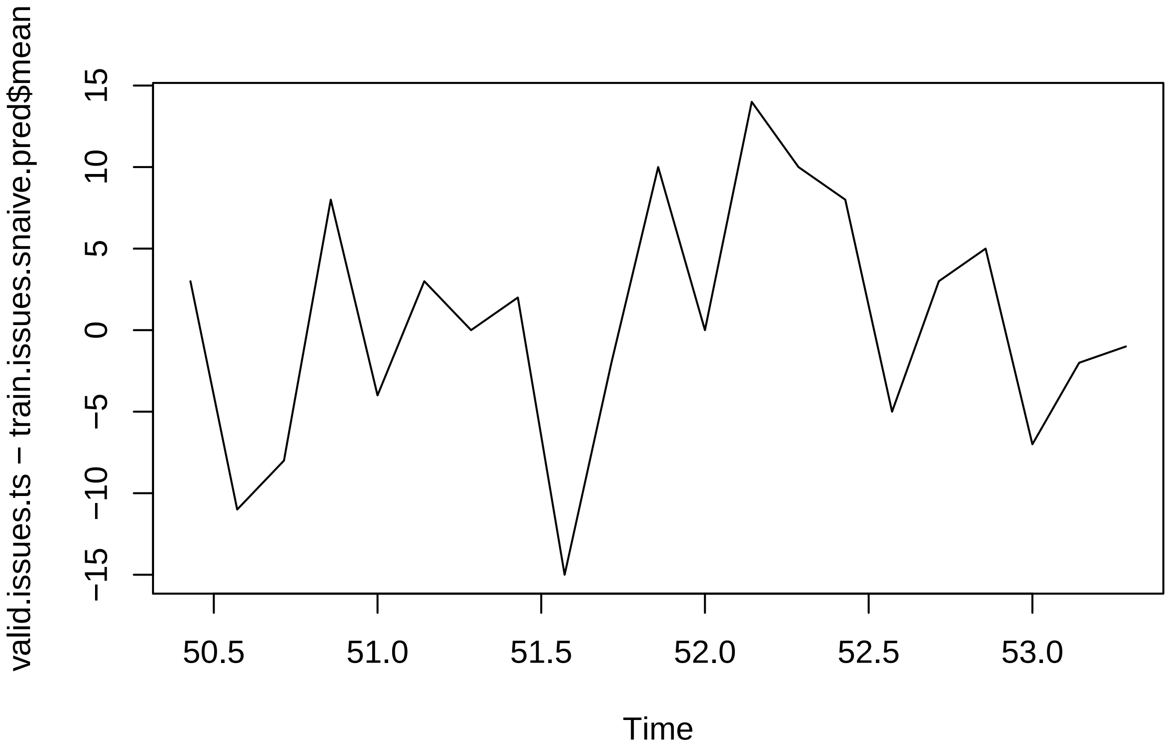
```
train.issues.snaive.pred <- snaive(train.issues.ts, h=n.valid)
accuracy(train.issues.snaive.pred, valid.issues.ts)
```

```
##                       ME     RMSE      MAE        MPE     MAPE     MASE
## Training set 0.002949853 6.552038 5.200590 -12.839076 42.21488 1.000000
## Test set     0.523809524 7.201190 5.761905  -7.239015 42.64360 1.107933
##                    ACF1 Theil's U
## Training set 0.17205895        NA
## Test set     0.07663259  1.489314
```

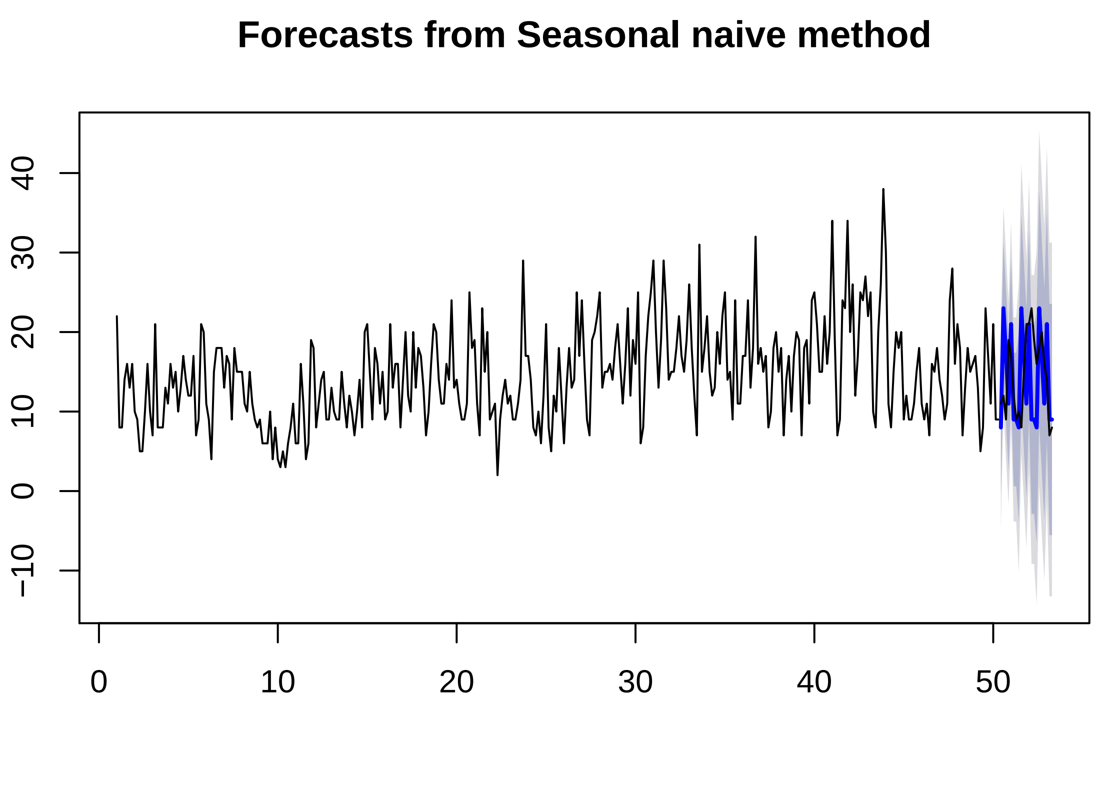
```
hist(valid.issues.ts - train.issues.snaive.pred$mean)
```




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




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



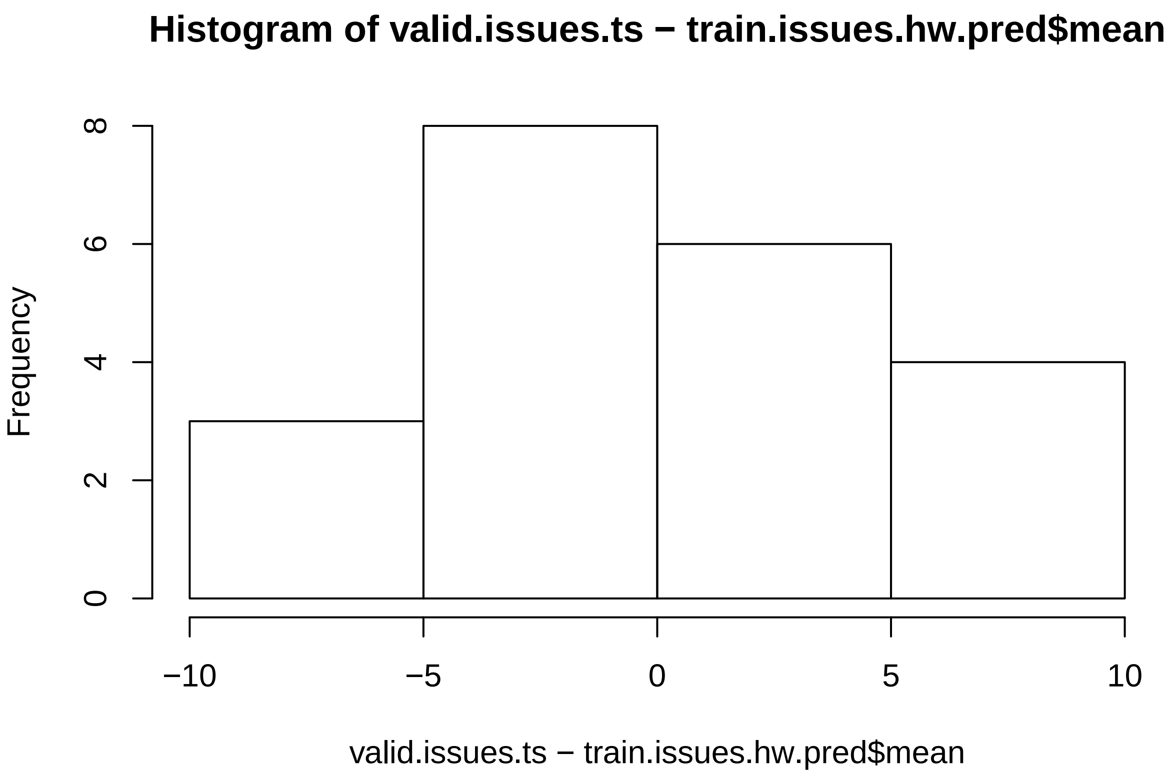
Smoothing

Holt Winter

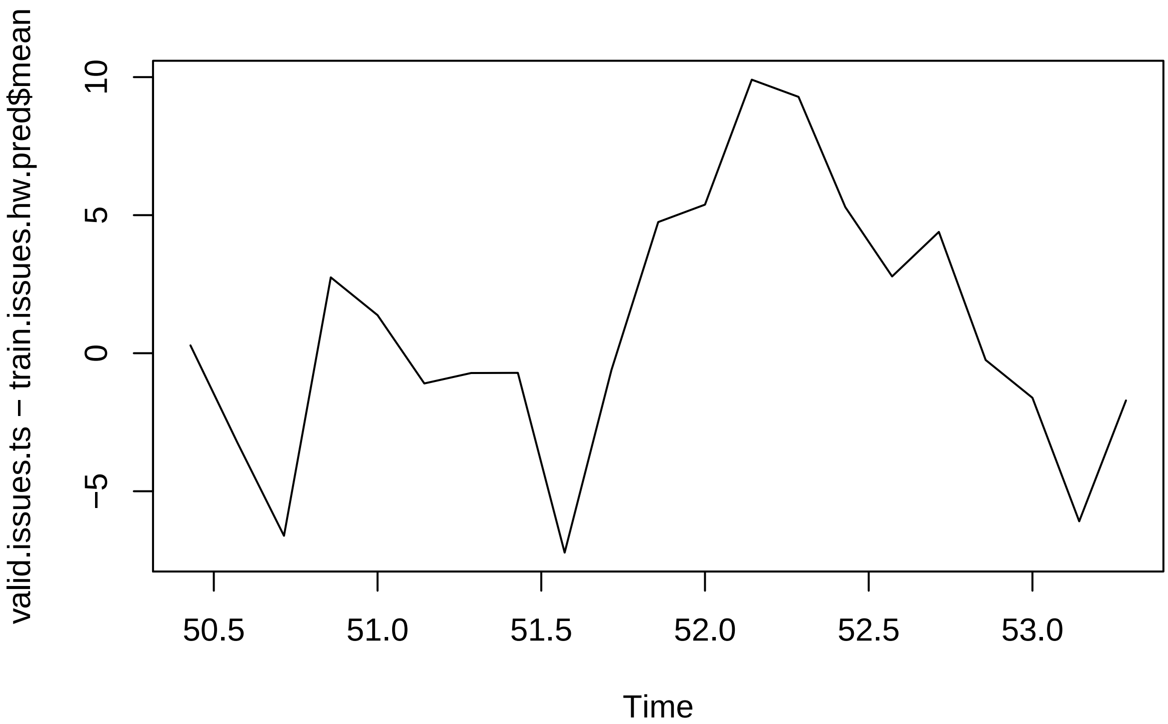
```r
train.issues.hw.pred <- hw(train.issues.ts, hw = "ZAA", h = n.valid)
accuracy(train.issues.hw.pred, valid.issues.ts)
```

```
##                       ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.02650463 4.977548 3.872611 -13.061762 32.46036 0.7446484
## Test set      0.77798345 4.639578 3.621891  -4.980661 27.43986 0.6964384
##                    ACF1 Theil's U
## Training set 0.07738404        NA
## Test set     0.60217996  0.837638
```

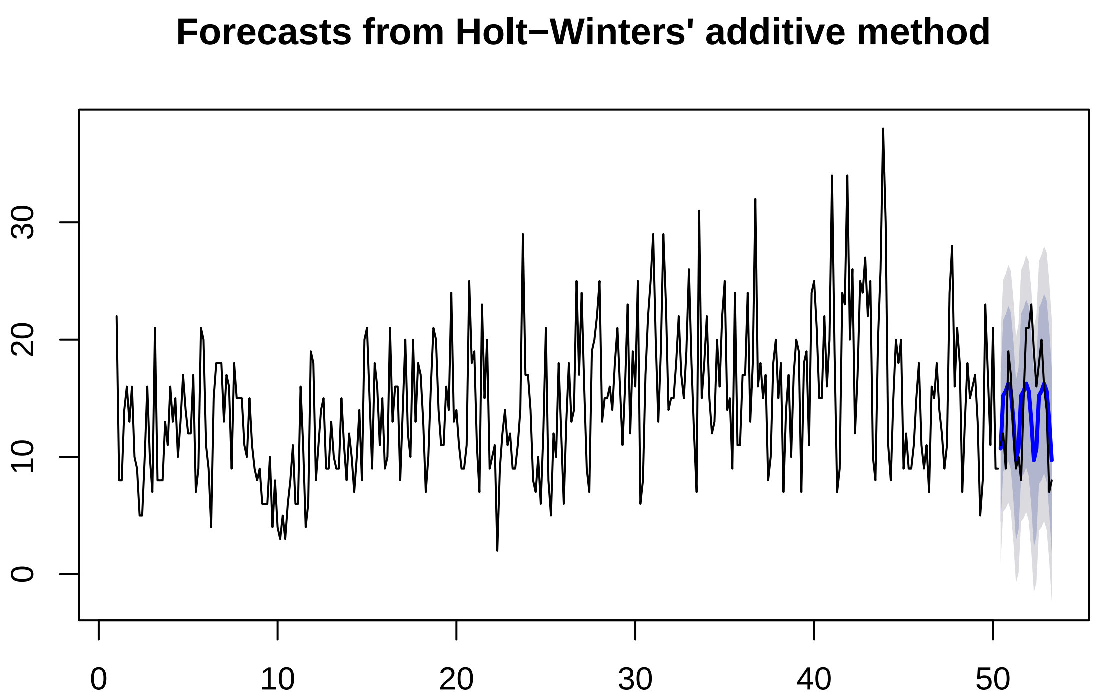
```r
hist(valid.issues.ts - train.issues.hw.pred$mean)
```




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




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

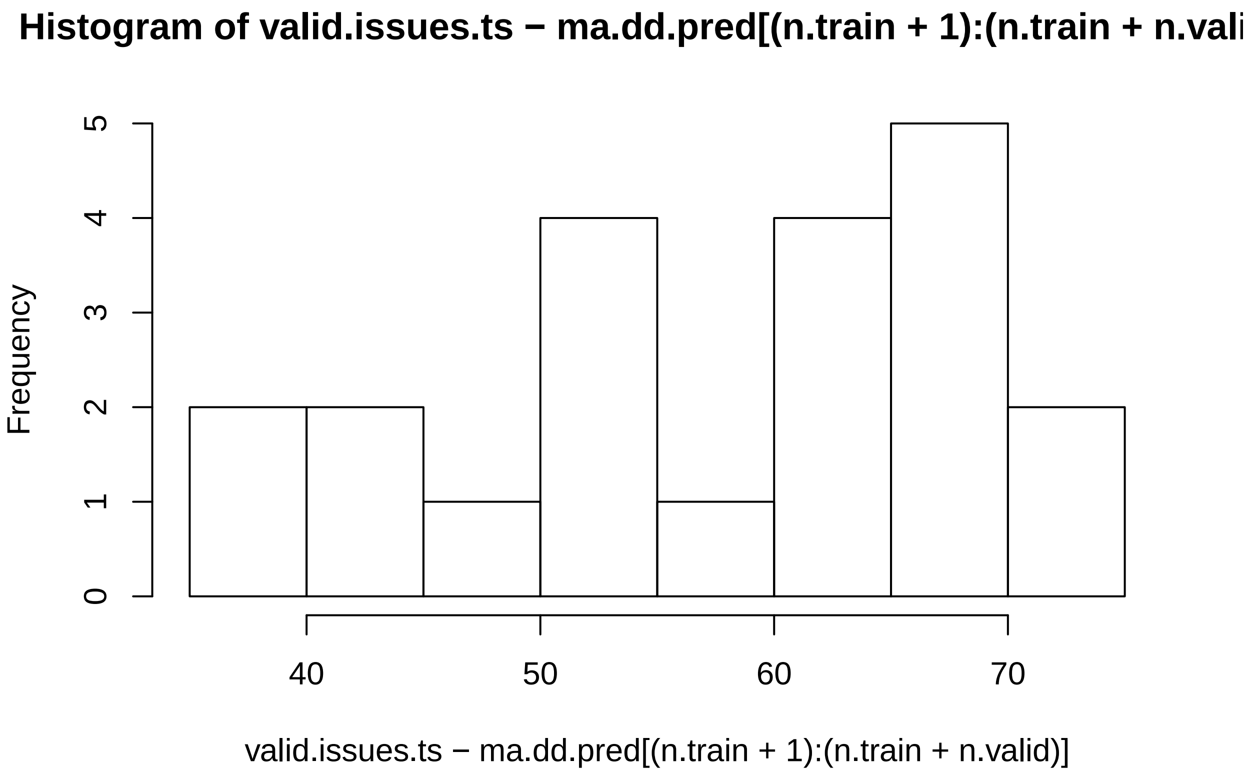


Double differencing

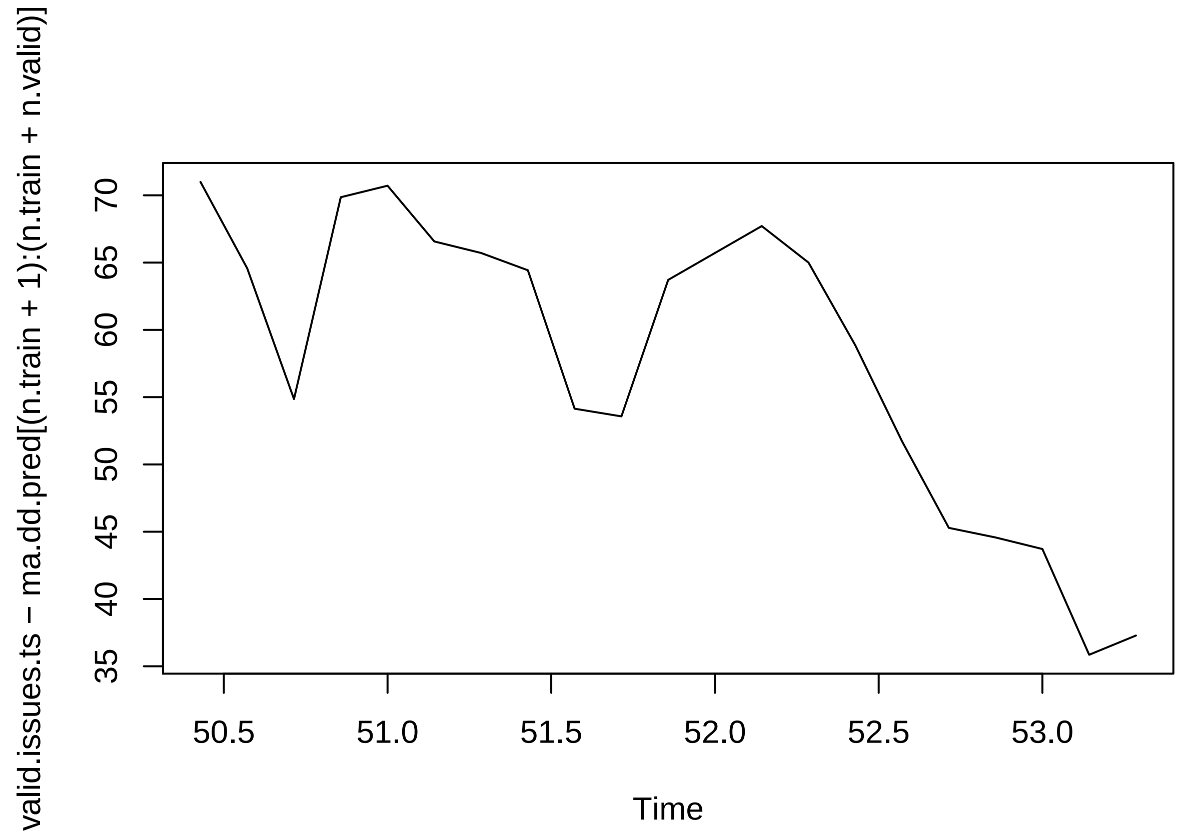
```
train.issues.d1 <- diff(train.issues.ts, lag = 1)
train.issues.d1.d7 <- diff(train.issues.d1, lag = 7)

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), fre

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])

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 57.85034 58.8597 57.85034 440.1622 440.1622 0.7177956   12.1168
```

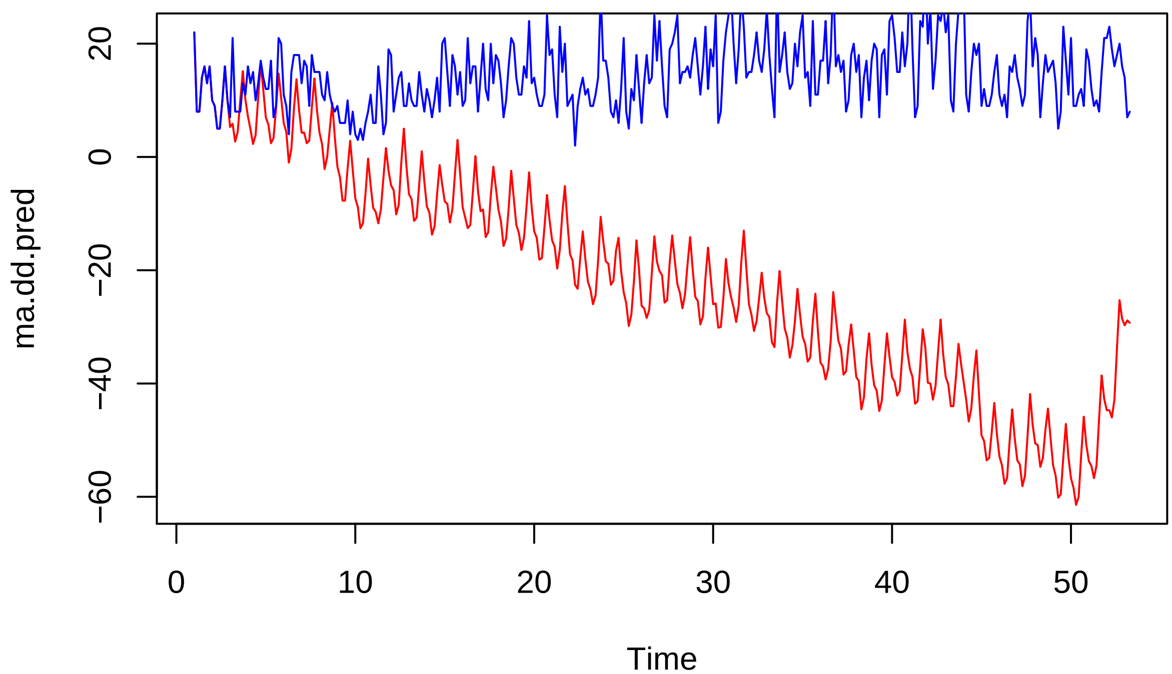
```
hist(valid.issues.ts - ma.dd.pred[(n.train+1):(n.train+n.valid)])
```




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




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



Regression

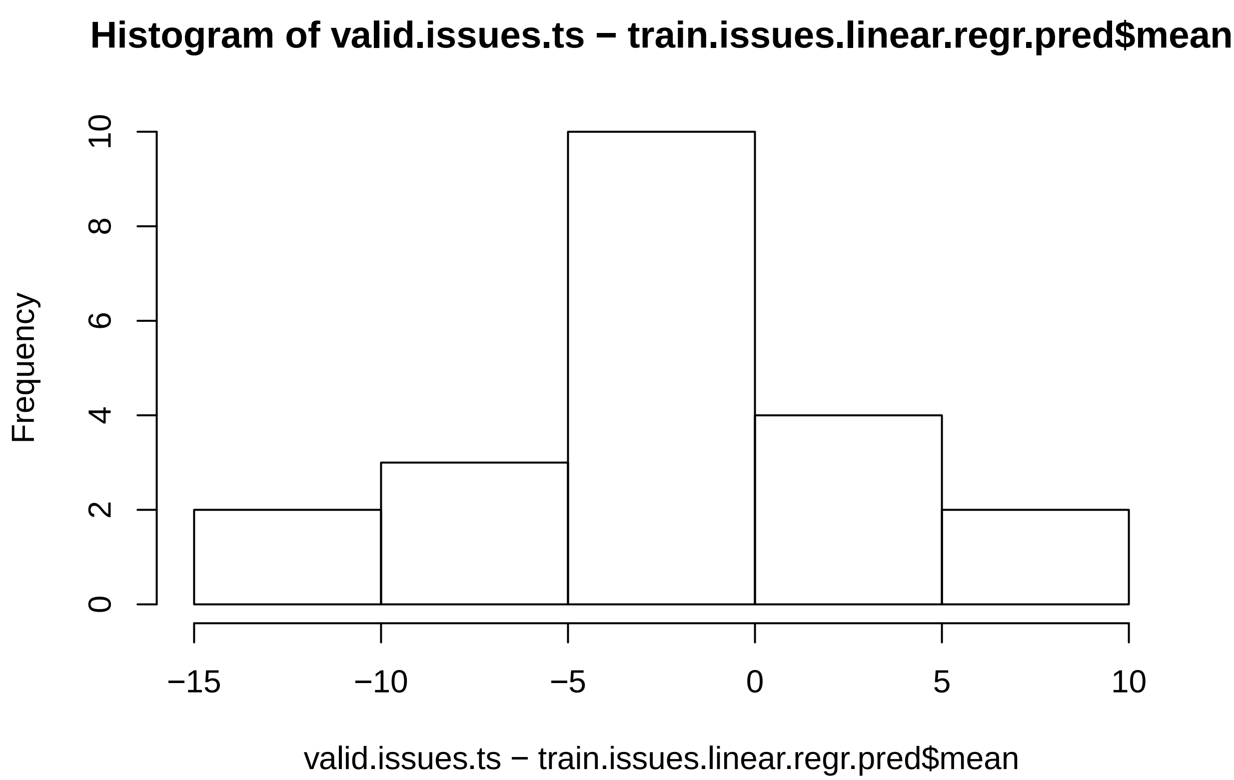
```r
train.issues.linear.regr.m <- tslm(train.issues.ts ~ trend + season, lambda = 0)
train.issues.linear.regr.m
```

```
##
## Call:
## tslm(formula = train.issues.ts ~ trend + season, lambda = 0)
##
## Coefficients:
## (Intercept)        trend      season2      season3      season4
##    2.483884     0.001444    -0.191684    -0.488595    -0.354584
##     season5      season6      season7
##   -0.011783    -0.018838     0.016470
```

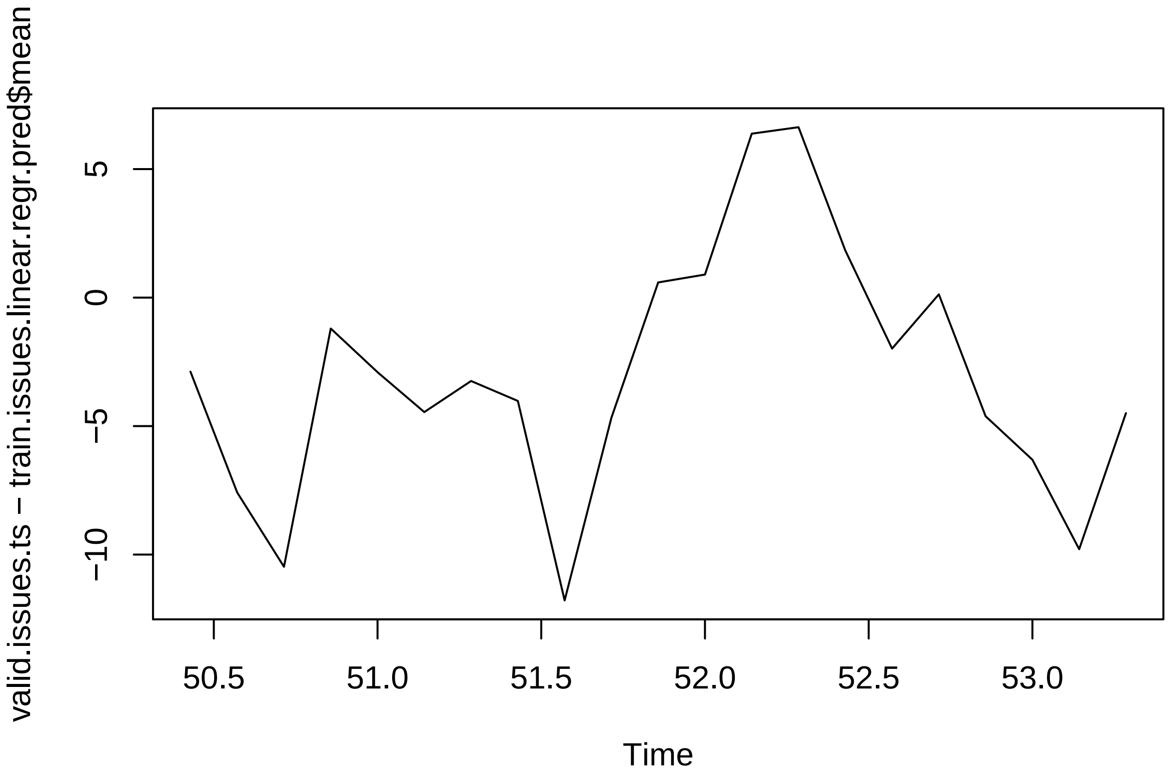
```r
train.issues.linear.regr.pred <- forecast(train.issues.linear.regr.m , h=n.valid)

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

```
##                       ME     RMSE      MAE        MPE     MAPE      MASE
## Training set   0.9509424 5.238848 4.086120  -8.514036 33.40185 0.7857032
## Test set      -3.0456515 5.618531 4.613154 -34.294688 42.08708 0.8870443
##                    ACF1 Theil's U
## Training set 0.2766360        NA
## Test set     0.5703958  1.276547
```

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



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




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

Forecasts from Linear regression model