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.

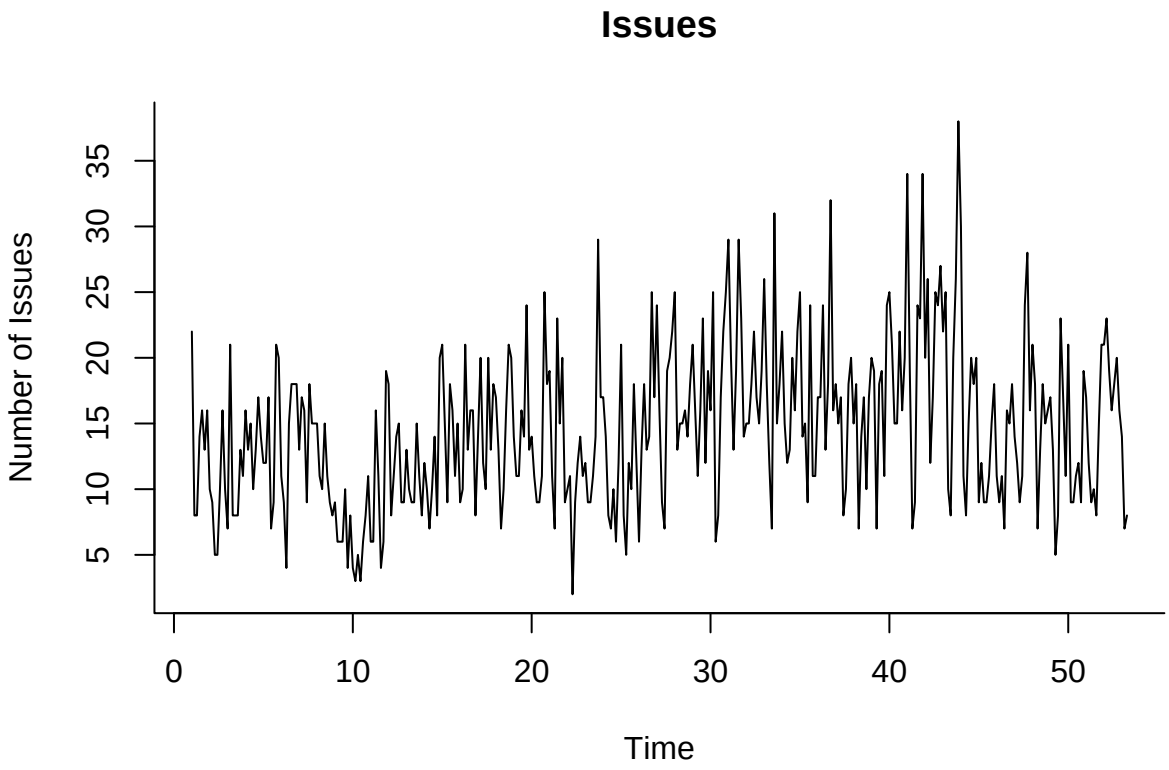
Load the data

```
#install.packages('forecast')
library('forecast')
library(knitr)
#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'))
```

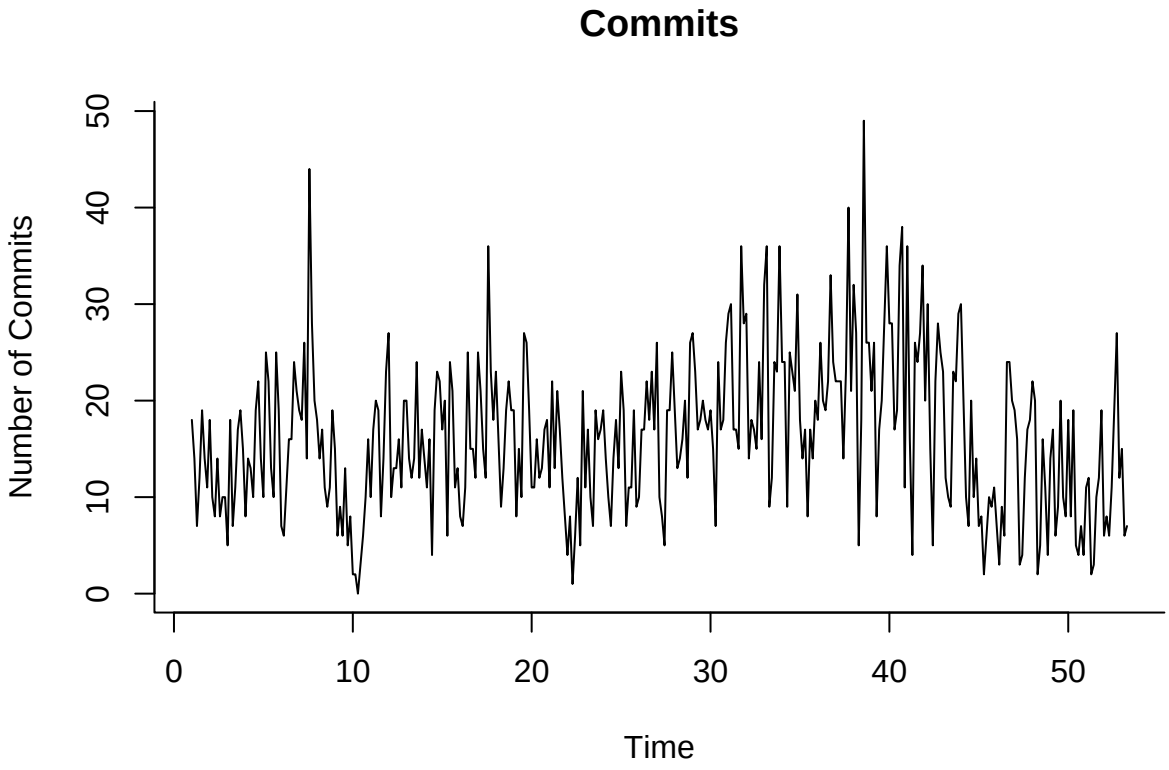
keep the last 12 months

```
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)
```

```
n.valid <- 21


separate.train.test <- function(timeserie, n.valid) {
  time <- time(timeserie)
  n.train <- length(timeserie) - n.valid
  results = list()
  results$train.ts <- window(timeserie, start=time[1], end=time[n.train])
  results$valid.ts <- window(timeserie, start=time[n.train+1], end=time[n.train+n.valid])
  return(results)
}

issues  <- separate.train.test(issues.ts, n.valid)
commits  <- separate.train.test(commits.ts, n.valid)
```

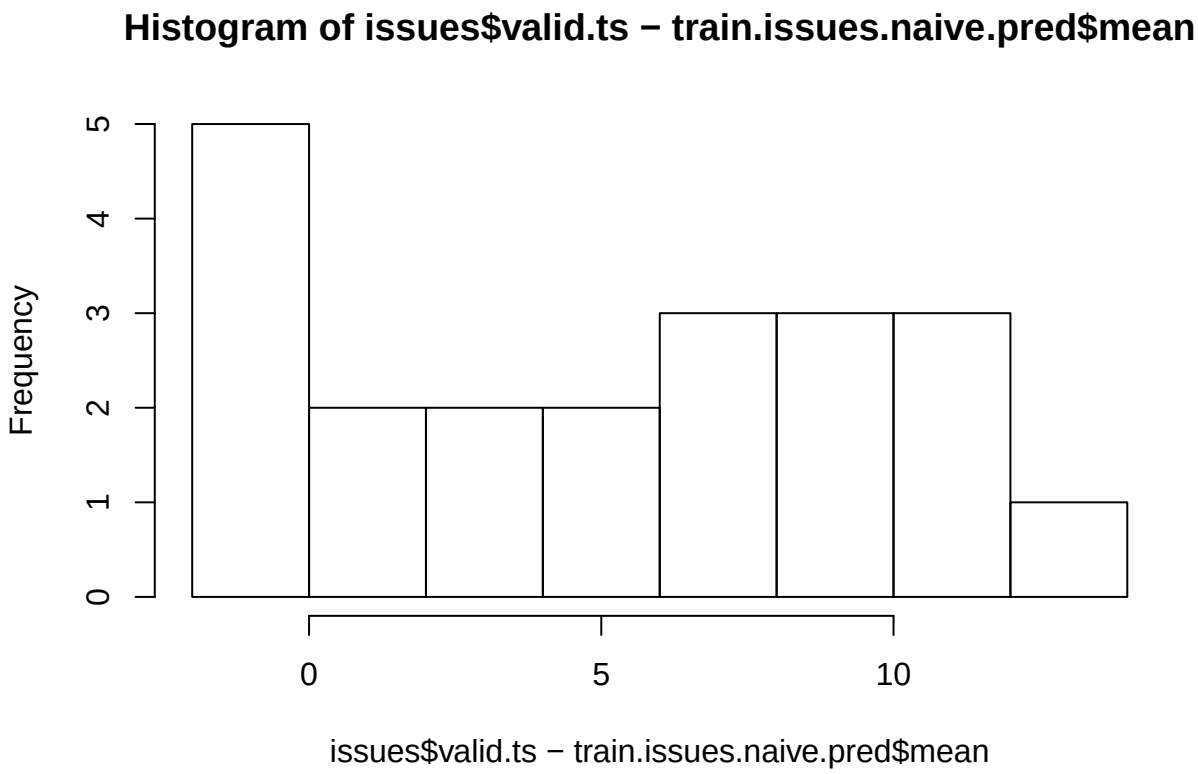
Naive Forecast

Naive

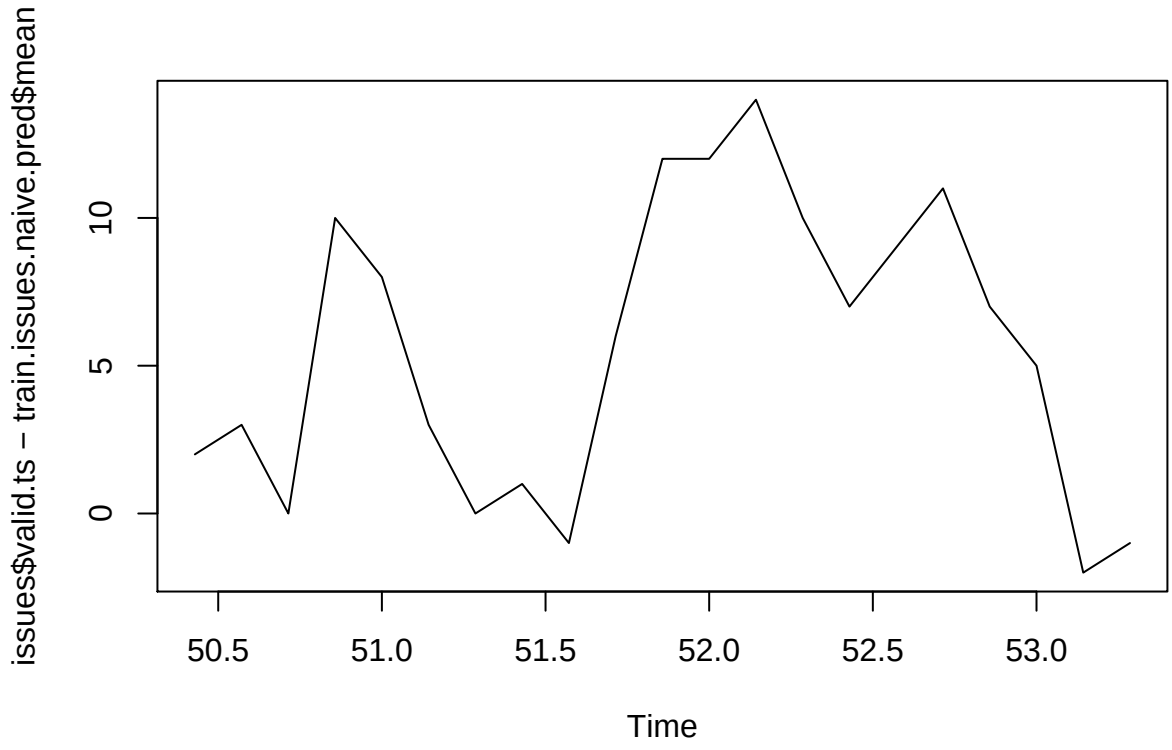
```
train.issues.naive.pred <- naive(issues$train.ts, h=n.valid)
kable(accuracy(train.issues.naive.pred, issues$valid.ts))
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	-0.0376812	6.718998	5.260870	-13.08450	42.03105	1.011591	-0.2812264	NA
Test set	5.5238095	7.361418	5.904762	29.53819	34.64023	1.135402	0.5978010	1.293618

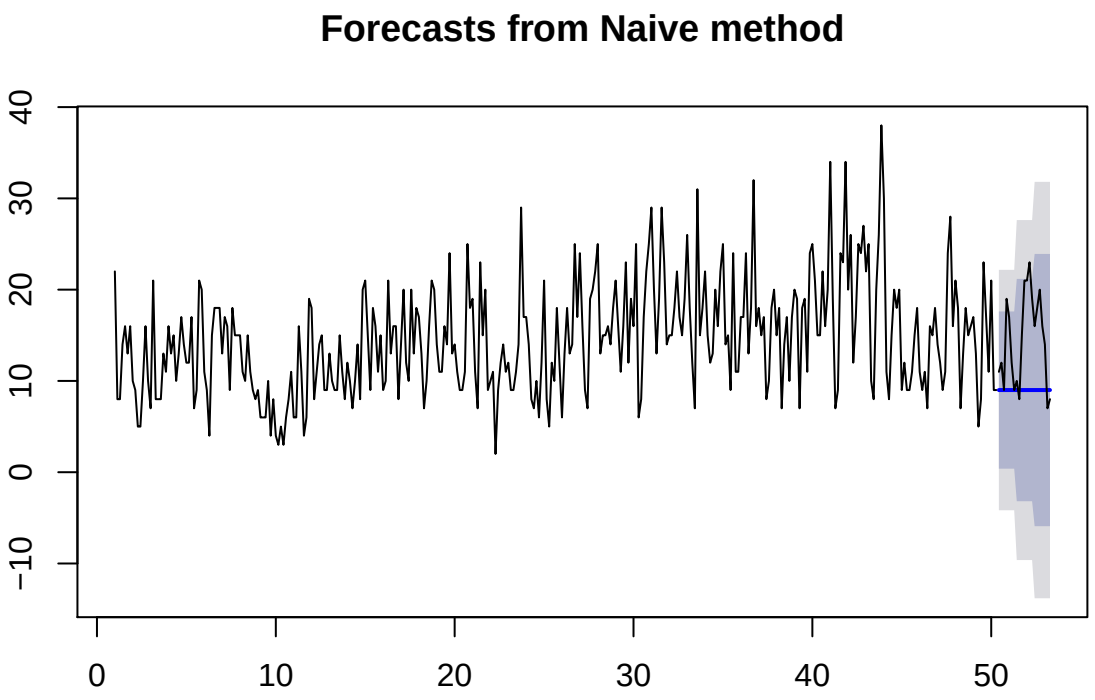
```
hist(issues$valid.ts - train.issues.naive.pred$mean)
```




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




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

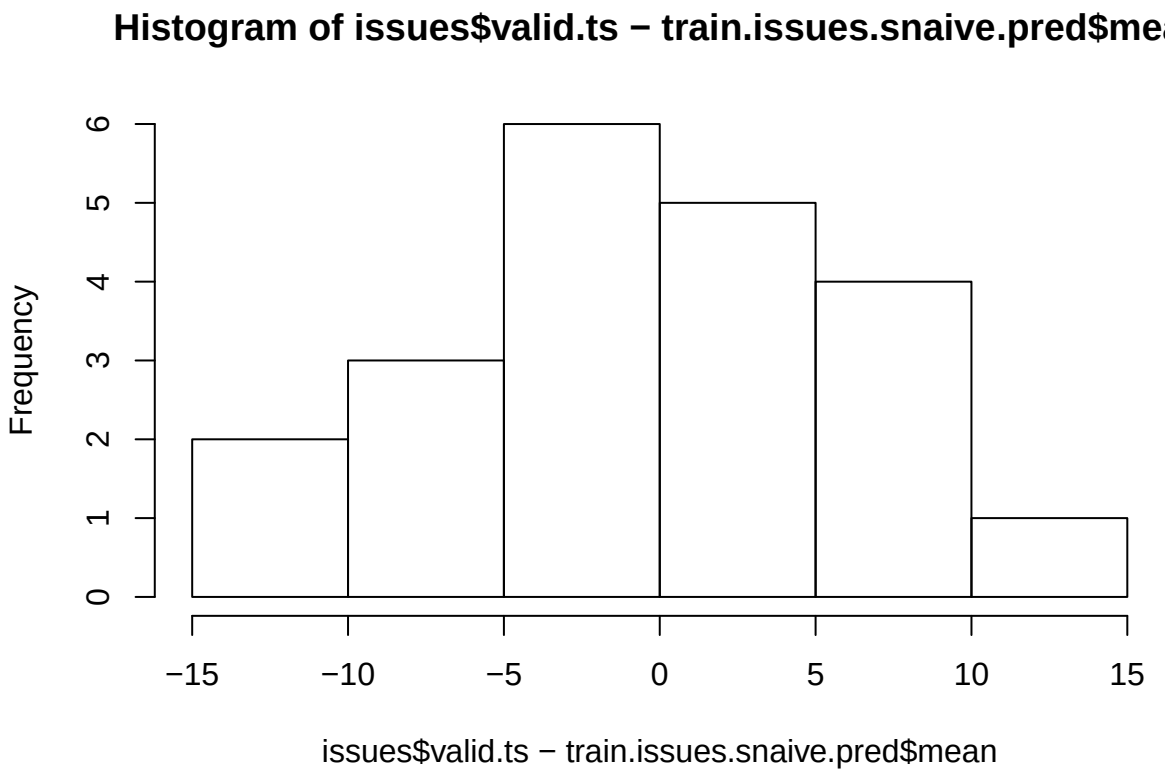


Seasonal Naive

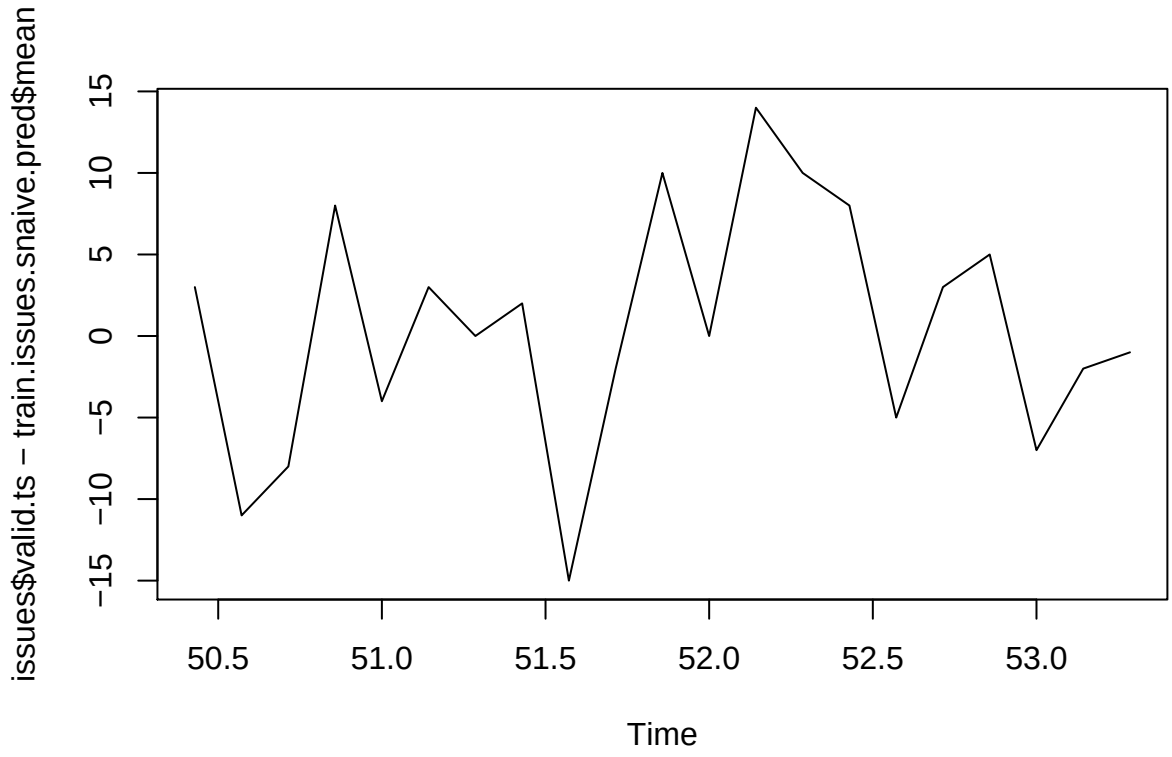
```
train.issues.snaive.pred <- snaive(issues$train.ts, h=n.valid)
kable(accuracy(train.issues.snaive.pred, issues$valid.ts))
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	0.0029499	6.552038	5.200590	-12.839076	42.21488	1.000000	0.1720590	NA
Test set	0.5238095	7.201190	5.761905	-7.239015	42.64360	1.107933	0.0766326	1.489315

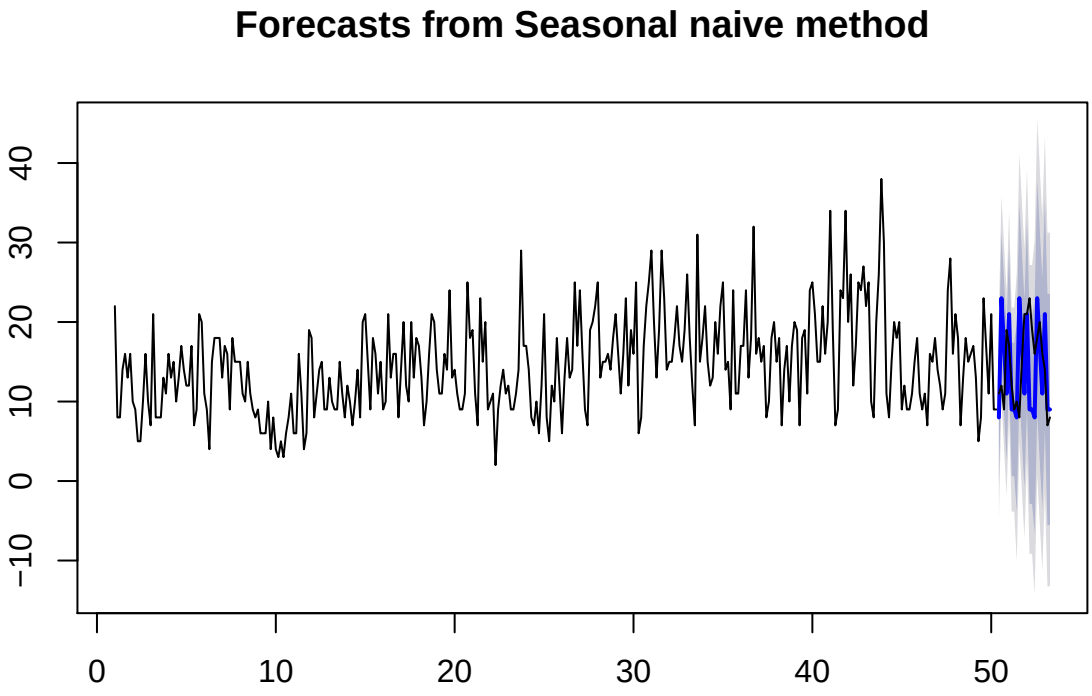
```
hist(issues$valid.ts - train.issues.snaive.pred$mean)
```




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




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



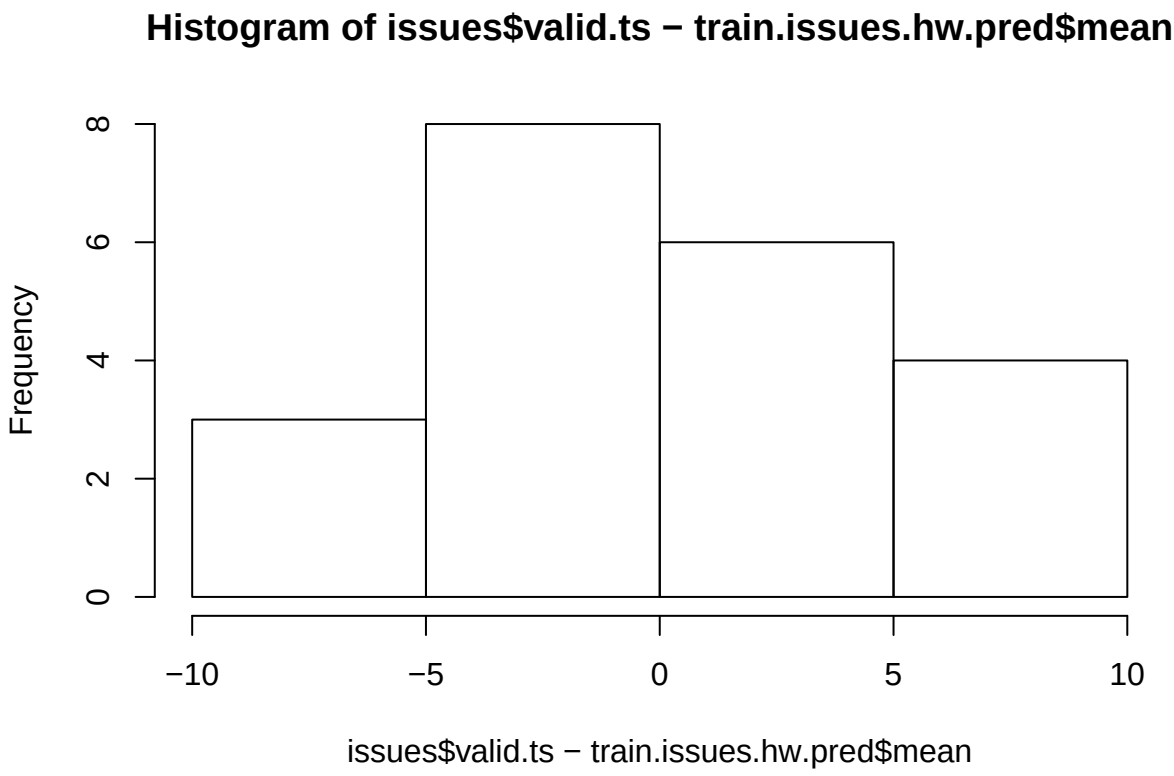
Smoothing

Holt Winter

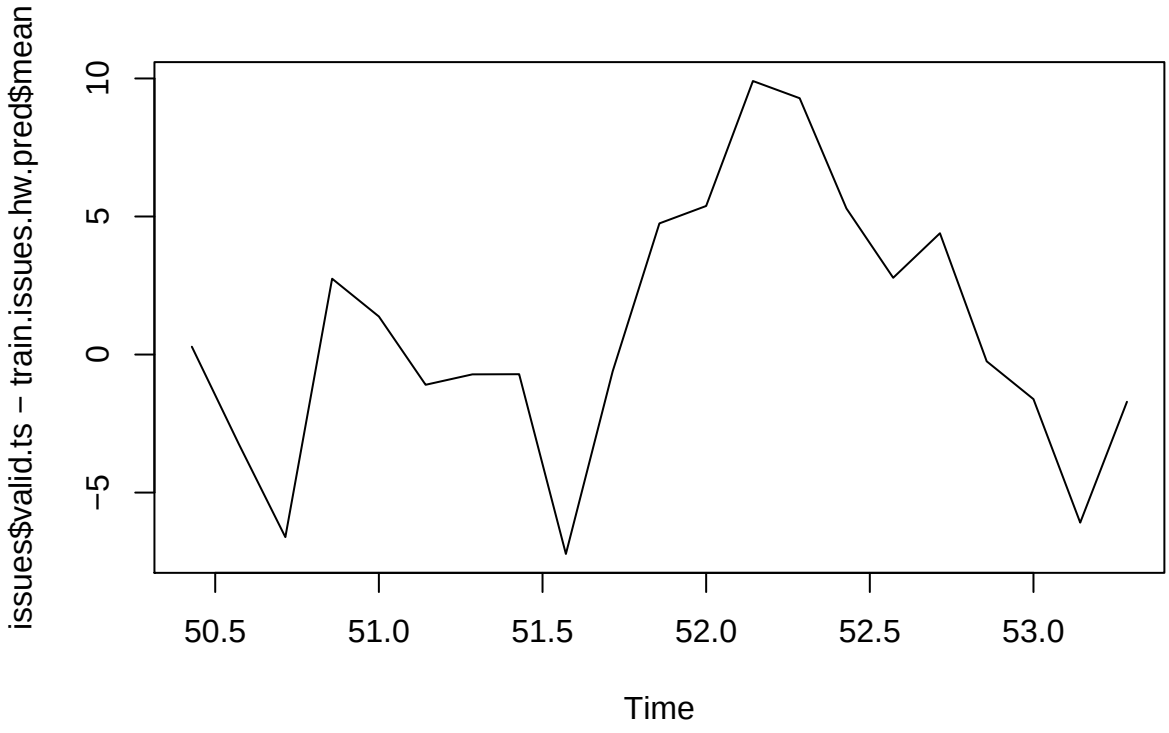
```
train.issues.hw.pred <- hw(issues$train.ts, hw = "ZAA", h = n.valid)
kable(accuracy(train.issues.hw.pred, issues$valid.ts))
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	-0.0265046	4.977548	3.872611	-13.061762	32.46036	0.7446484	0.077384	NA
Test set	0.7779834	4.639578	3.621891	-4.980661	27.43986	0.6964384	0.602180	0.837638

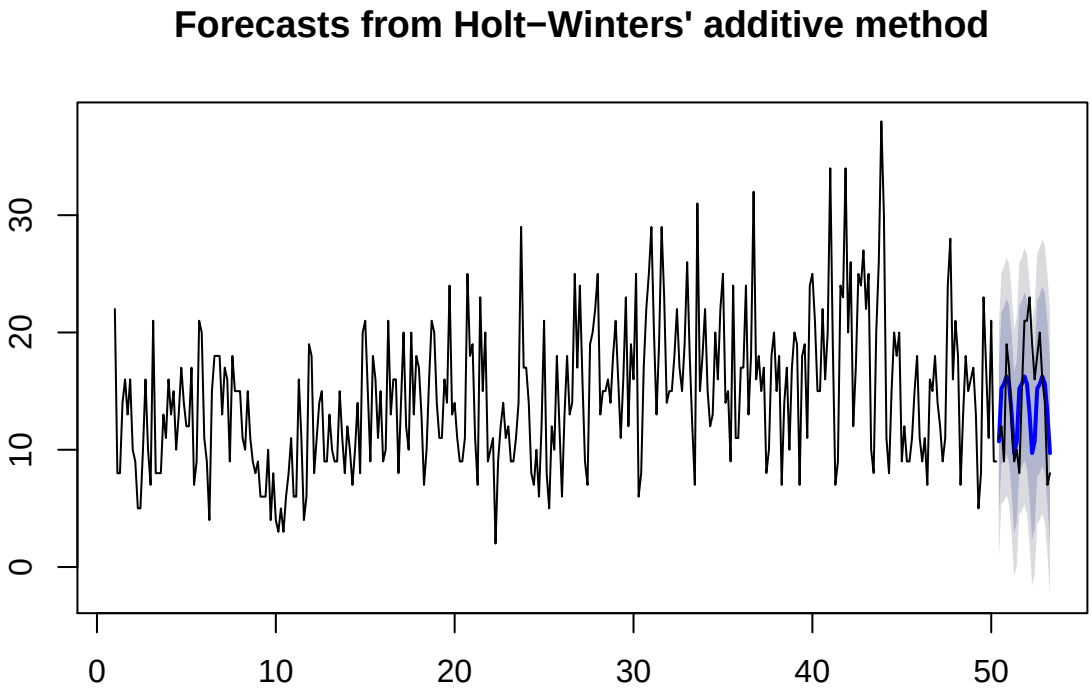
```
hist(issues$valid.ts - train.issues.hw.pred$mean)
```




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

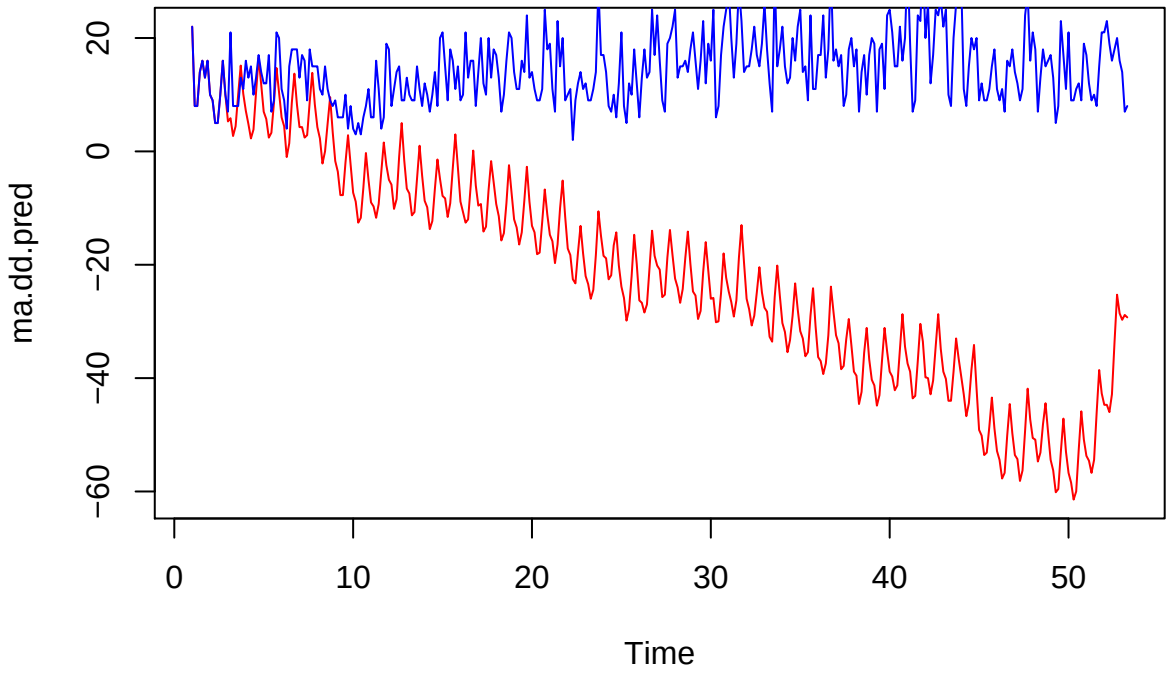



```
plot(train.issues.hw.pred)
lines(issues$valid.ts)
```



Double differencing

```
train.issues.d1 <- diff(issues$train.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), fr

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=issues$train.ts[1])

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



Regression

Linear additive regression

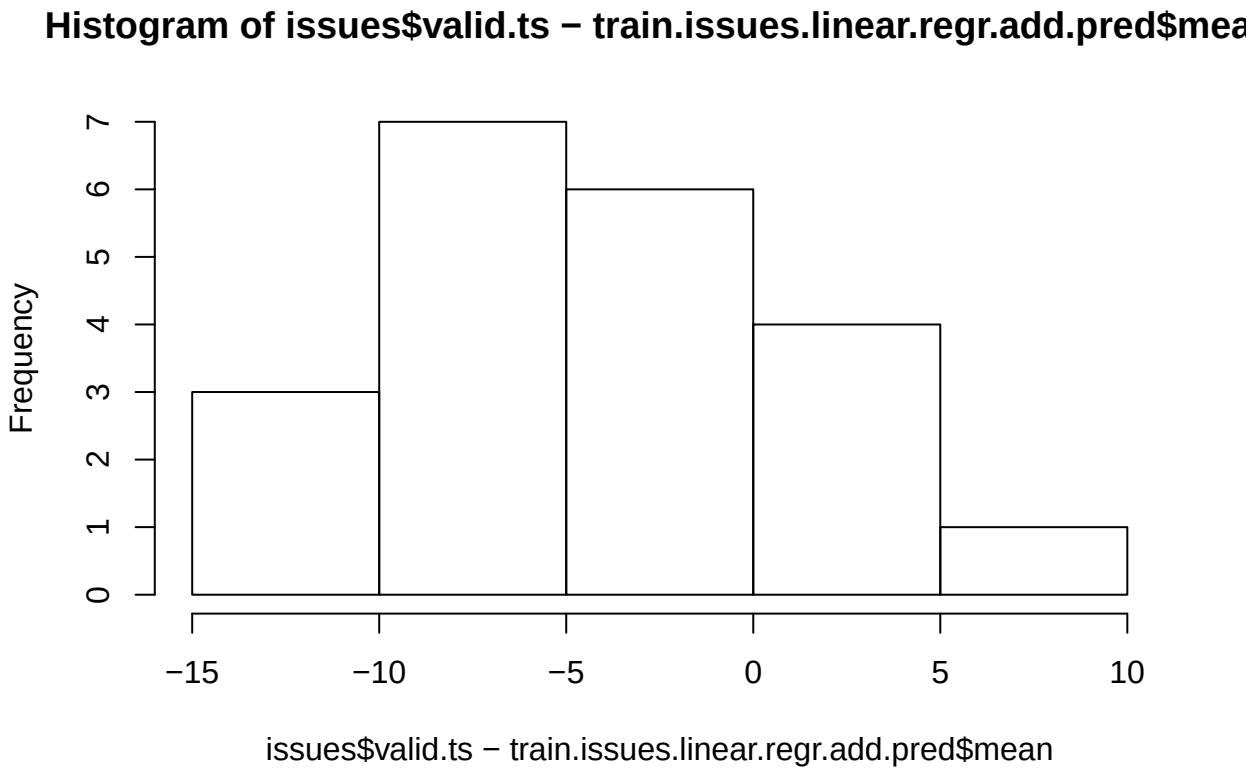
```
train.issues.linear.regr.add.m <- tslm(issues$train.ts ~ trend + season)
train.issues.linear.regr.add.m
```

```
##
## Call:
## tslm(formula = issues$train.ts ~ trend + season)
##
## Coefficients:
## (Intercept)        trend      season2      season3      season4
##    13.32711      0.02002     -3.10002     -6.36003     -5.17815
##     season5      season6      season7
##    -0.50429     -0.36105     -0.09535
```

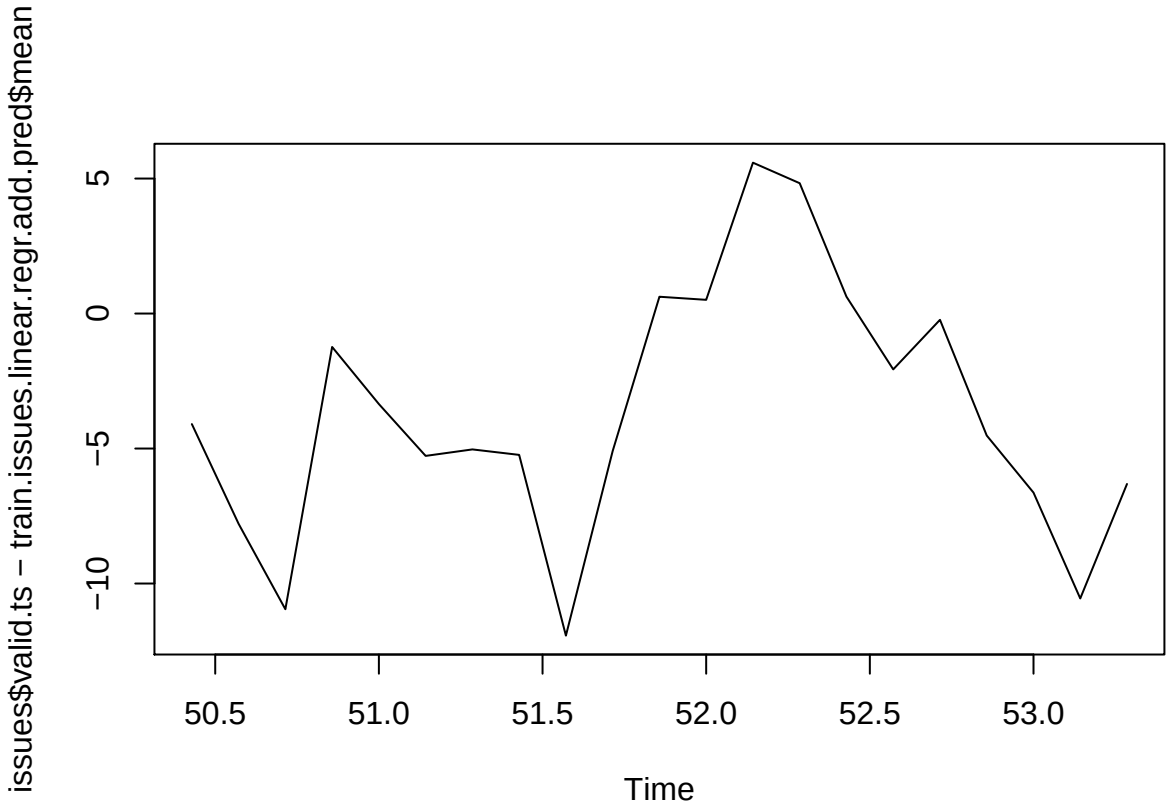
```
train.issues.linear.regr.add.pred <- forecast(train.issues.linear.regr.add.m , h=n.valid)

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

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	0.000000	5.191735	4.097533	-16.39242	36.04025	0.7878976	0.2745091	NA
Test set	-3.720864	5.913790	4.879717	-40.20440	45.82142	0.9383006	0.5850591	1.394207

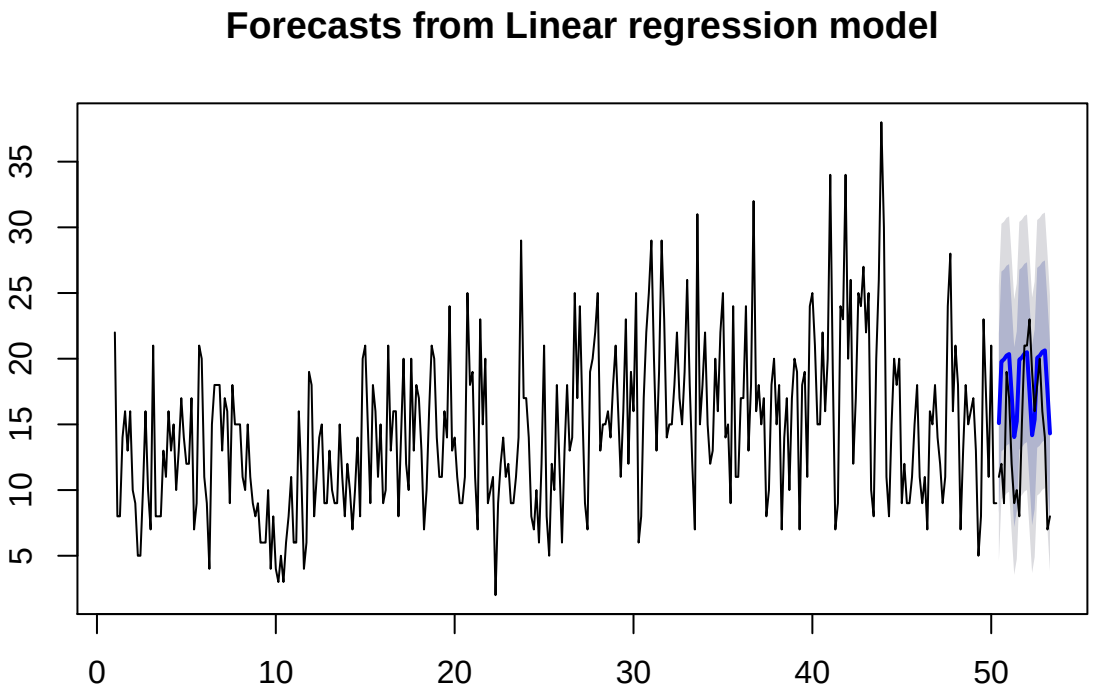
```
hist(issues$valid.ts - train.issues.linear.regr.add.pred$mean)
```




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




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



linear multiplicative regression

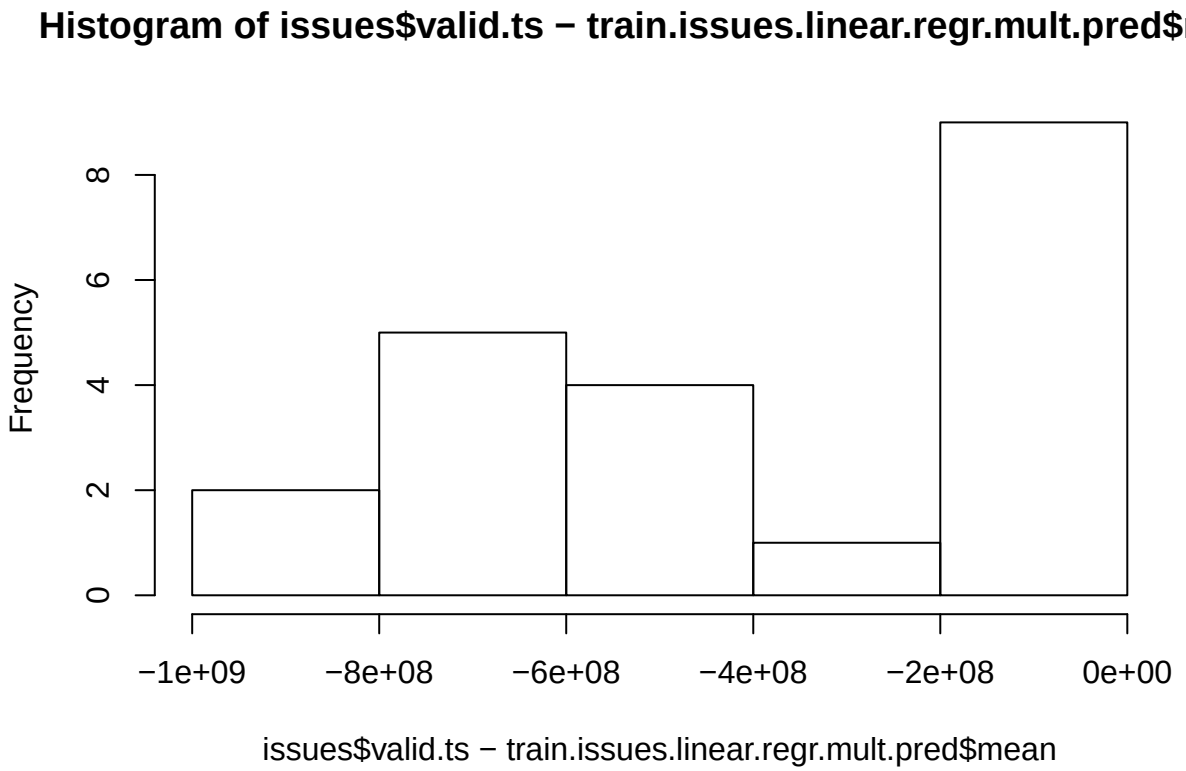
```
train.issues.linear.regr.mult.m <- tslm(issues$train.ts ~ trend + season, lambda = 0)
train.issues.linear.regr.mult.m
```

```
##
## Call:
## tslm(formula = issues$train.ts ~ trend + season, lambda = 0)
##
## Coefficients:
## (Intercept)        trend      season2      season3      season4
##    13.32711      0.02002     -3.10002     -6.36003     -5.17815
##     season5      season6      season7
##    -0.50429     -0.36105     -0.09535
```

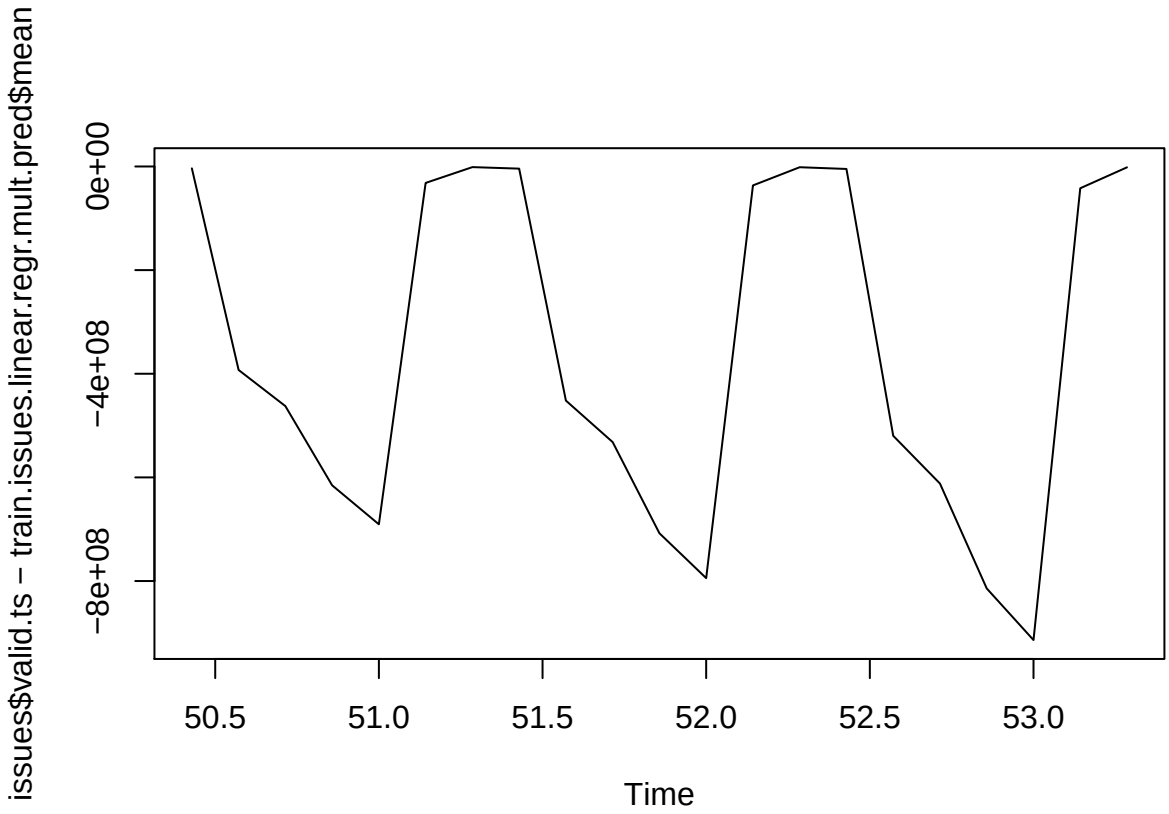
```
train.issues.linear.regr.mult.pred <- forecast(train.issues.linear.regr.mult.m , h=n.valid)

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

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	9.582908e+13	1.713166e+15	9.582912e+13	-8.716828e+04	8.724629e+04	0.4899045	-0.0028054
Test set	-3.635191e+08	4.874506e+08	3.635191e+08	-2.418645e+09	2.418645e+09	0.0000019	0.3705441

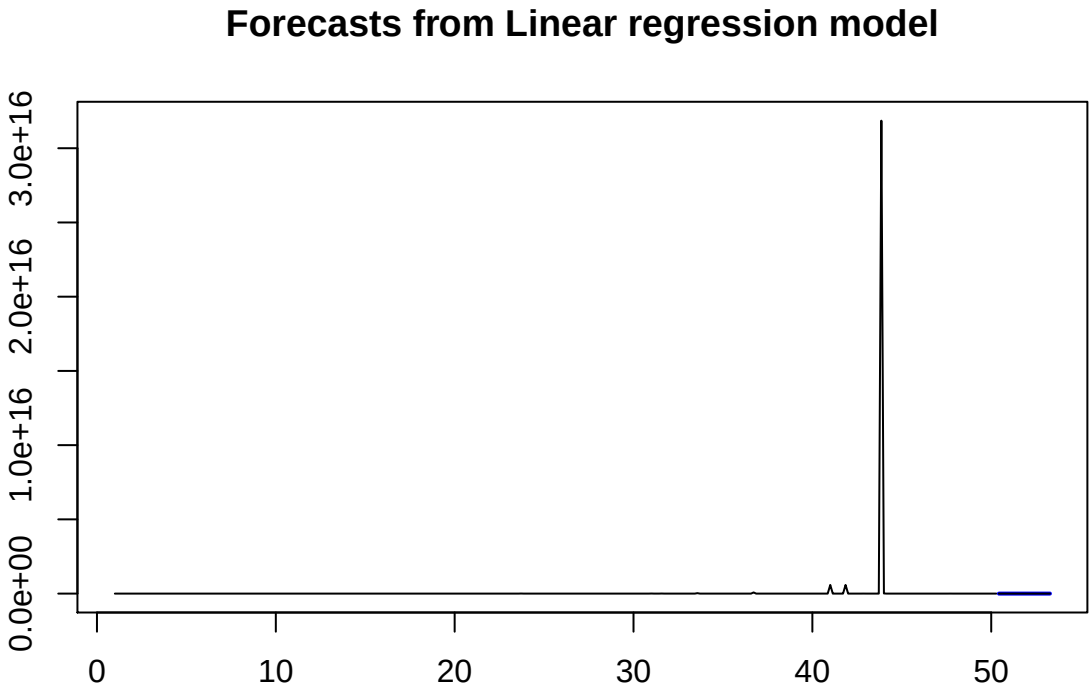
```
hist(issues$valid.ts - train.issues.linear.regr.mult.pred$mean)
```



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

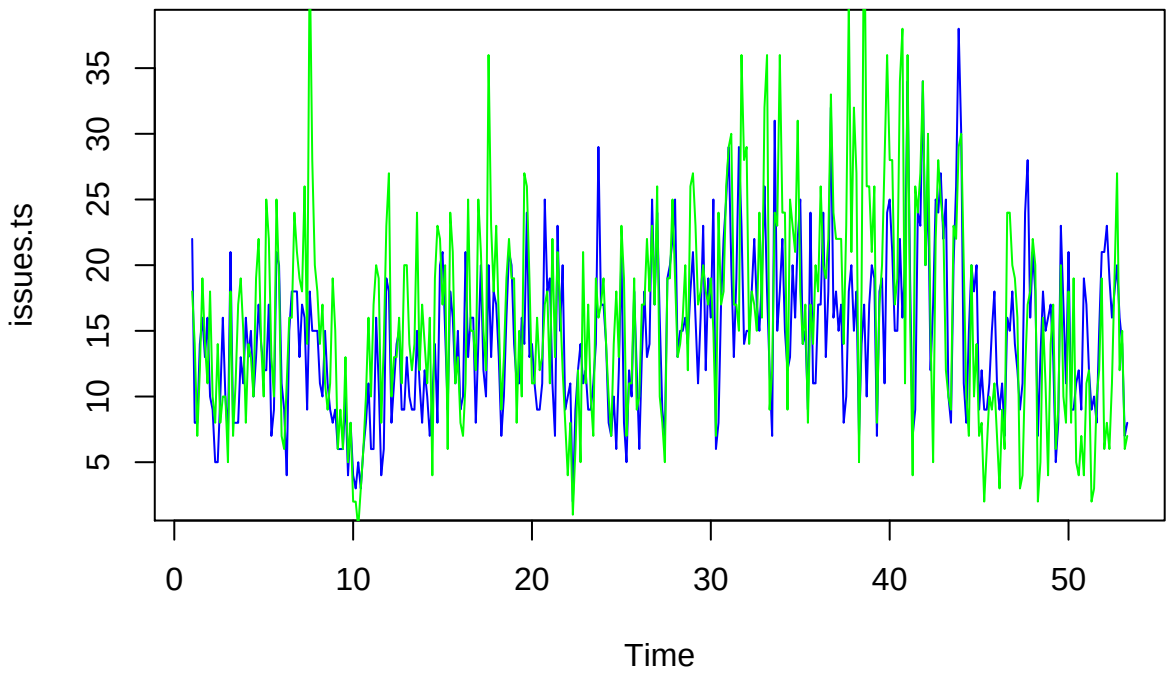



```
plot(train.issues.linear.regr.mult.pred)
lines(issues$valid.ts)
```

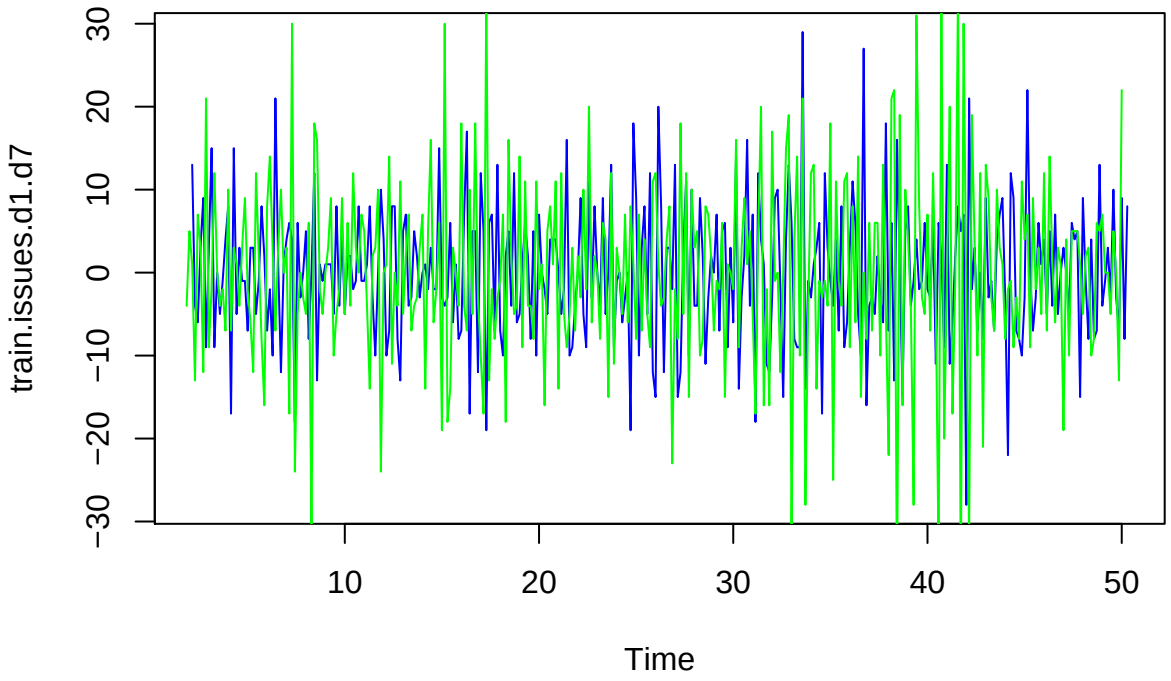


external regression

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




```
train.commits.d1 <- diff(commits$train.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')
```




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

```
## Series: issues$train.ts
## ARIMA(1,0,0)(1,0,0)[7] with non-zero mean
##
## Coefficients:
##          ar1    sar1  intercept  commits$train.ts
##       0.2050  0.2187     7.9198            0.4004
## s.e.  0.0548  0.0565     0.7248            0.0366
##
## sigma^2 estimated as 21.16:  log likelihood=-1017.15
## AIC=2044.3   AICc=2044.48   BIC=2063.53
```