Demand Forecasting Using Time Series

Anupriya Thirumurthy

5/18/2019

Import dataset

```
dataPath<-"/Users/anupriyathirumurthy/Documents/AnuBackUp/University/MScA_UoC/Courses/TimeSeries/Projec
dataset <-read.csv(file=paste(dataPath,"Books.csv",sep="/"))
head(dataset)
```

```
##   ID_LABEL1 Date.Ordered ID.. Format Pub.Status Pub.Status.Description
## 1       124      8/24/17    1      C          1           Active Title
## 2       124     11/29/17    1      C          1           Active Title
## 3       124       9/5/18    1      C          1           Active Title
## 4       124      5/21/18    1      C          1           Active Title
## 5       124      6/14/17    1      C          1           Active Title
## 6       124      8/29/18    1      C          1           Active Title
##   SRDP Order.Disc.Class Product.Class Quantity.Ordered Order.Price
## 1   NA            124ND        UCTECH                2          72
## 2   NA            124ND        UCTECH                1          72
## 3   NA            124ND        UCTECH                6          72
## 4   NA            124ND        UCTECH                1          72
## 5   NA            124ND        UCTECH                1          72
## 6   NA            124ND        UCTECH                9          72
##   Disc.Pct Line.Amount
## 1        0         144
## 2      100           0
## 3        0         432
## 4      100           0
## 5      100           0
## 6        0         648
```

```
str(dataset)
```

```
## 'data.frame':    4152 obs. of  13 variables:
##  $ ID_LABEL1             : int  124 124 124 124 124 124 124 124 124 124 ...
##  $ Date.Ordered          : Factor w/ 683 levels "1/10/17","1/10/18",..: 556 133 670 342 383 576 585 
##  $ ID..                  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Format                : Factor w/ 5 levels "C","E","O","P",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Pub.Status            : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Pub.Status.Description: Factor w/ 3 levels "Active Title",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ SRDP                  : logi  NA NA NA NA NA NA ...
##  $ Order.Disc.Class      : Factor w/ 1 level "124ND": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Product.Class         : Factor w/ 1 level "UCTECH": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Quantity.Ordered      : int  2 1 6 1 1 9 -6 1 6 -2 ...
##  $ Order.Price           : num  72 72 72 72 72 72 72 72 72 72 ...
##  $ Disc.Pct              : int  0 100 0 100 100 0 0 0 0 0 ...
##  $ Line.Amount           : Factor w/ 273 levels "-1,080.00","-1,152.00",..: 111 69 196 69 69 232 40 
```

Import relevant packages

```
library(forecast)
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff
```

```
library(tseries)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Step 1: Data clean-up

```
#Rename ISBN column
colnames(dataset)[3] <- "ISBN"

#Separate units ordered vs. returned
dataset$Qty_Ordered <- ifelse(dataset$Quantity.Ordered >0, dataset$Quantity.Ordered, 0)
dataset$Qty_Returned <- ifelse(dataset$Quantity.Ordered <0, dataset$Quantity.Ordered, 0)

#Create timeframe identifier
Date.Ordered.Year <- year(as.Date(dataset$Date.Ordered,"%m/%d/%Y"))
Date.Ordered.Year <- as.numeric(paste("20",Date.Ordered.Year, sep = ""))
Date.Ordered.Year <- as.data.frame(Date.Ordered.Year, "%Y")
```

```
## Warning in as.data.frame.numeric(Date.Ordered.Year, "%Y"): 'row.names' is
## not a character vector of length 4152 -- omitting it. Will be an error!
```

```
#df$Fasta.headers = paste(">",df$Fasta.headers,sep = "")
dataset<- cbind(dataset, Date.Ordered.Year)
```

```r
max <- tapply(dataset$Date.Ordered.Year, dataset$ISBN, max)
min <- tapply(dataset$Date.Ordered.Year, dataset$ISBN, min)
min_max <- cbind(min, max)
min_max <- as.data.frame(min_max)
min_max <- cbind(newColName = rownames(min_max), min_max)
rownames(min_max) <- 1:nrow(min_max)
colnames(min_max)[1] <- "ISBN"
min_max$combined <- paste(min_max$min, "-", min_max$max)

merged_dataset <- merge(x = dataset, y = min_max, by = "ISBN", all.x = TRUE)

#Eliminate unneccessary columns
final_dataset <- merged_dataset[, -c(2,5:9,12:13,17:18)]
head(final_dataset)
```

```
##   ISBN Date.Ordered Format Quantity.Ordered Order.Price Qty_Ordered
## 1    1      8/24/17      C                2          72           2
## 2    1     11/29/17      C                1          72           1
## 3    1       9/5/18      C                6          72           6
## 4    1      5/21/18      C                1          72           1
## 5    1      6/14/17      C                1          72           1
## 6    1      8/29/18      C                9          72           9
##   Qty_Returned Date.Ordered.Year    combined
## 1            0              2017 2016 - 2019
## 2            0              2017 2016 - 2019
## 3            0              2018 2016 - 2019
## 4            0              2018 2016 - 2019
## 5            0              2017 2016 - 2019
## 6            0              2018 2016 - 2019
```

```r
str(final_dataset)
```

```
## 'data.frame':    4152 obs. of  9 variables:
##  $ ISBN             : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Date.Ordered     : Factor w/ 683 levels "1/10/17","1/10/18",..: 556 133 670 342 383 576 585 423 5
##  $ Format           : Factor w/ 5 levels "C","E","O","P",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Quantity.Ordered : int  2 1 6 1 1 9 -6 1 6 -2 ...
##  $ Order.Price      : num  72 72 72 72 72 72 72 72 72 72 ...
##  $ Qty_Ordered      : num  2 1 6 1 1 9 0 1 6 0 ...
##  $ Qty_Returned     : num  0 0 0 0 0 0 -6 0 0 -2 ...
##  $ Date.Ordered.Year: num  2017 2017 2018 2018 2017 ...
##  $ combined         : chr  "2016 - 2019" "2016 - 2019" "2016 - 2019" "2016 - 2019" ...
```

Step 2: Split test/train data

```r
train_data <- final_dataset[final_dataset$Date.Ordered.Year < "2018",]
test_data <- final_dataset[final_dataset$Date.Ordered.Year >= "2018",]
dim(train_data)
```

```
## [1] 3342    9
```

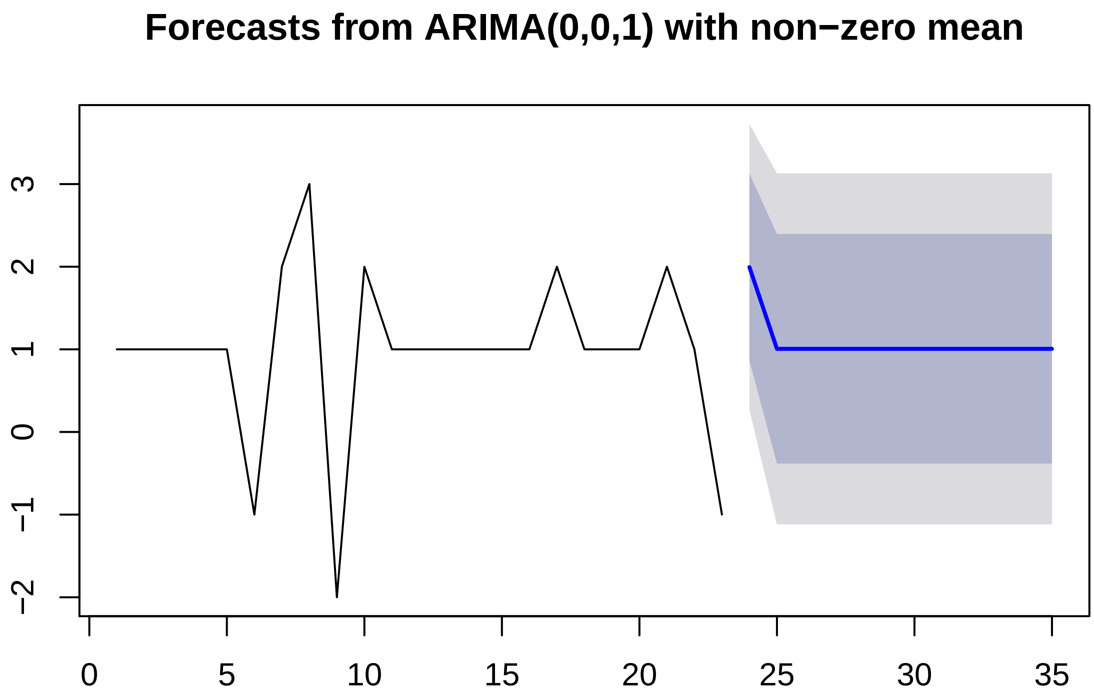
```r
dim(test_data)
```

```
## [1] 810   9
```

Step 3: Auto Arima with just one title

```
ISBN_8 <- final_dataset[final_dataset$ISBN == 8,]
train_data_8 <- ISBN_8[ISBN_8$Date.Ordered.Year < "2018",]
test_data_8 <- ISBN_8[ISBN_8$Date.Ordered.Year >= "2018",]

#Arima model
arima_8 <- auto.arima(ISBN_8$Quantity.Ordered)
forecast_8 <- forecast(arima_8, h = 12)
plot(forecast_8)
```




```
print(forecast_8)
```

```
##    Point Forecast       Lo 80    Hi 80      Lo 95    Hi 95
## 24       1.994563  0.8620119 3.127114  0.2624756 3.726650
## 25       1.006129 -0.3826485 2.394906 -1.1178229 3.130081
## 26       1.006129 -0.3826485 2.394906 -1.1178229 3.130081
## 27       1.006129 -0.3826485 2.394906 -1.1178229 3.130081
## 28       1.006129 -0.3826485 2.394906 -1.1178229 3.130081
## 29       1.006129 -0.3826485 2.394906 -1.1178229 3.130081
## 30       1.006129 -0.3826485 2.394906 -1.1178229 3.130081
## 31       1.006129 -0.3826485 2.394906 -1.1178229 3.130081
## 32       1.006129 -0.3826485 2.394906 -1.1178229 3.130081
## 33       1.006129 -0.3826485 2.394906 -1.1178229 3.130081
## 34       1.006129 -0.3826485 2.394906 -1.1178229 3.130081
## 35       1.006129 -0.3826485 2.394906 -1.1178229 3.130081
```

```
forecast_8 <- as.vector(forecast_8$mean)
pred <- mean(forecast_8)
pred
```

```
## [1] 1.088498
```

```r
test <- mean(test_data_8$Quantity.Ordered)
test
```

```
## [1] 1.4
```

```r
smape <- (sum(abs(test-pred)/(abs(test)+abs(pred))))/length(test)
smape
```

```
## [1] 0.1251765
```

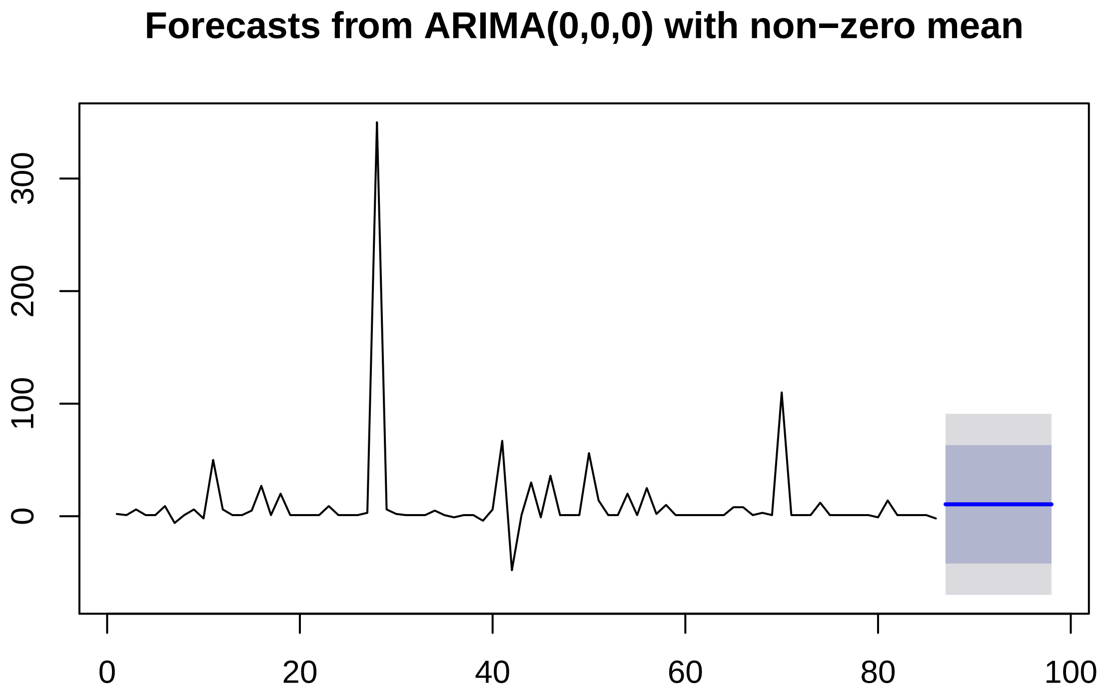
Step 4: Auto-arima with all isbn's

```r
# Auto Arima for all the titles
isbn_list <- unique(sort(final_dataset$ISBN))
isbn_list
```

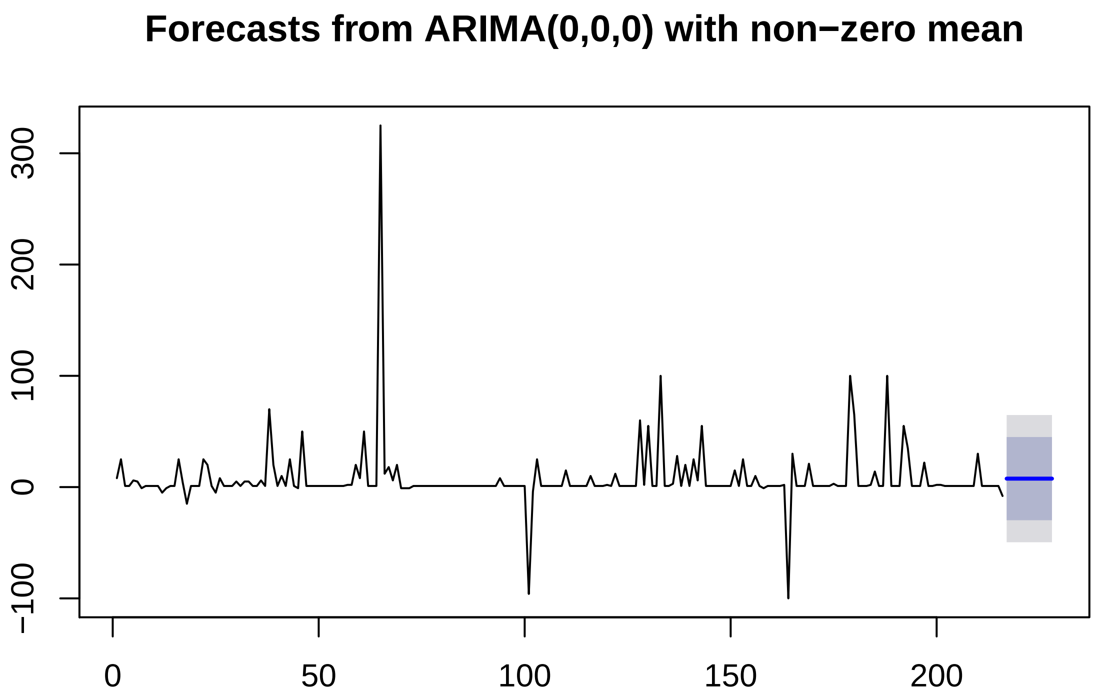
```
##  [1]   1   2   3   4   5   6   7   8   9  10  11  14  15  16  17  18  19
## [18]  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
## [35]  37  38  39  40  41  42  43  49  54  56  57  58  59  60  61  62  63
## [52]  86  90  91  92  93  94  95  96  97 103 109 110 128 129 132 165 166
## [69] 172 195 196 197 239 243 244 324 325 364 542
```

```r
for (i in isbn_list){
  ISBN <- final_dataset[final_dataset$ISBN == i,]
  train_data_all <- ISBN[ISBN$Date.Ordered.Year < "2018",]
  test_data_all <- ISBN[ISBN$Date.Ordered.Year >= "2018",]
  #Arima model
  arima_all <- auto.arima(ISBN$Quantity.Ordered)
  forecast_all <- forecast(arima_all, h = 12)
  cat("Book ID: ", unique(ISBN$ISBN))
  plot(forecast_all)
  #print(forecast_all)
}
```

```
## Book ID:  1
```

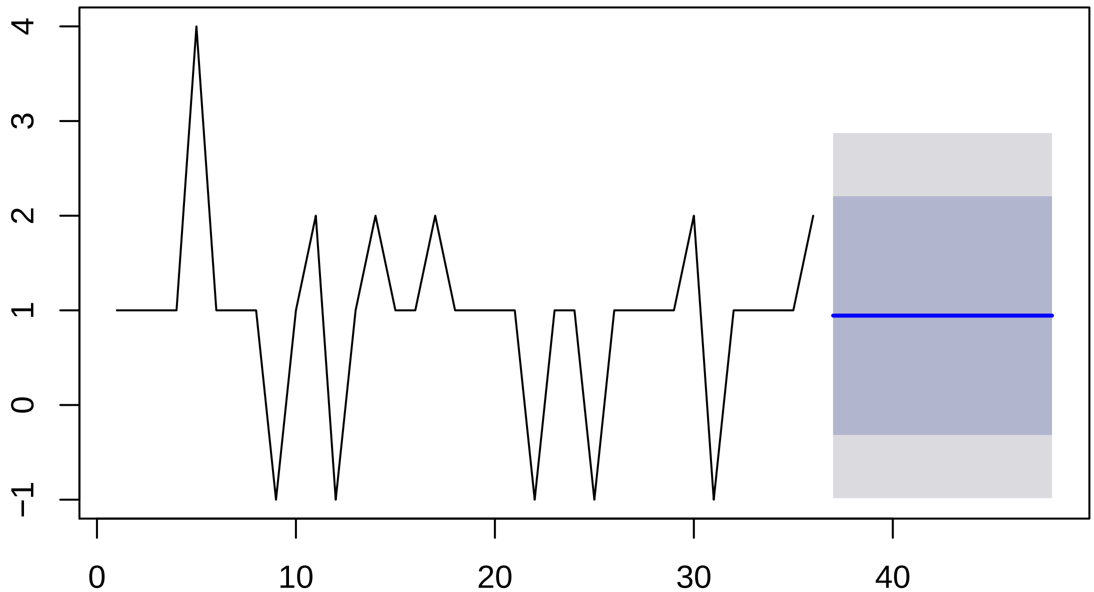


Book ID: 2



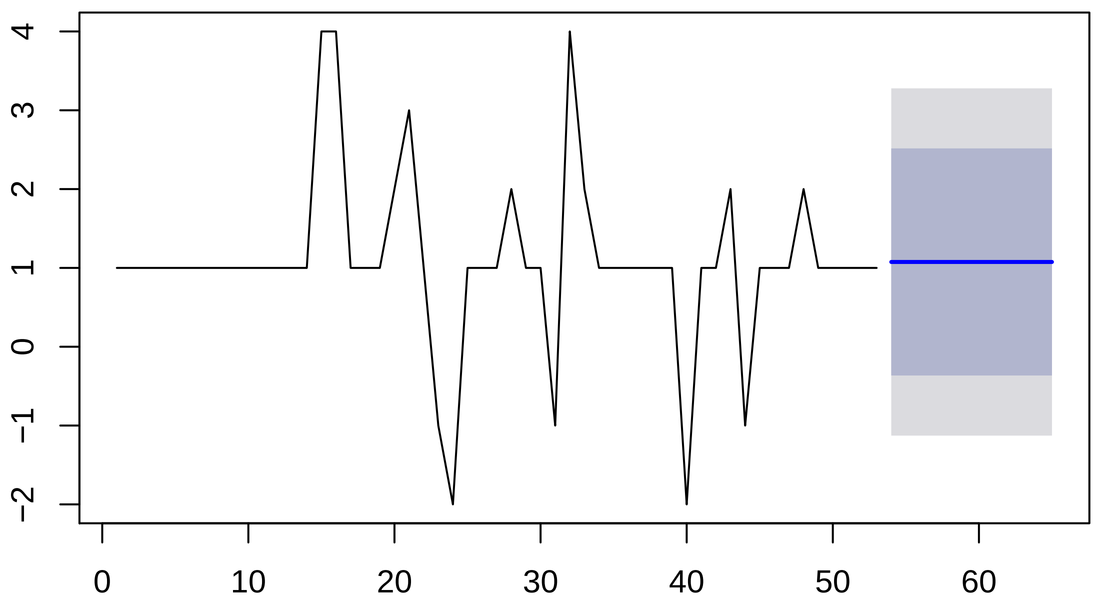
Book ID: 3

Forecasts from ARIMA(0,0,0) with non−zero mean

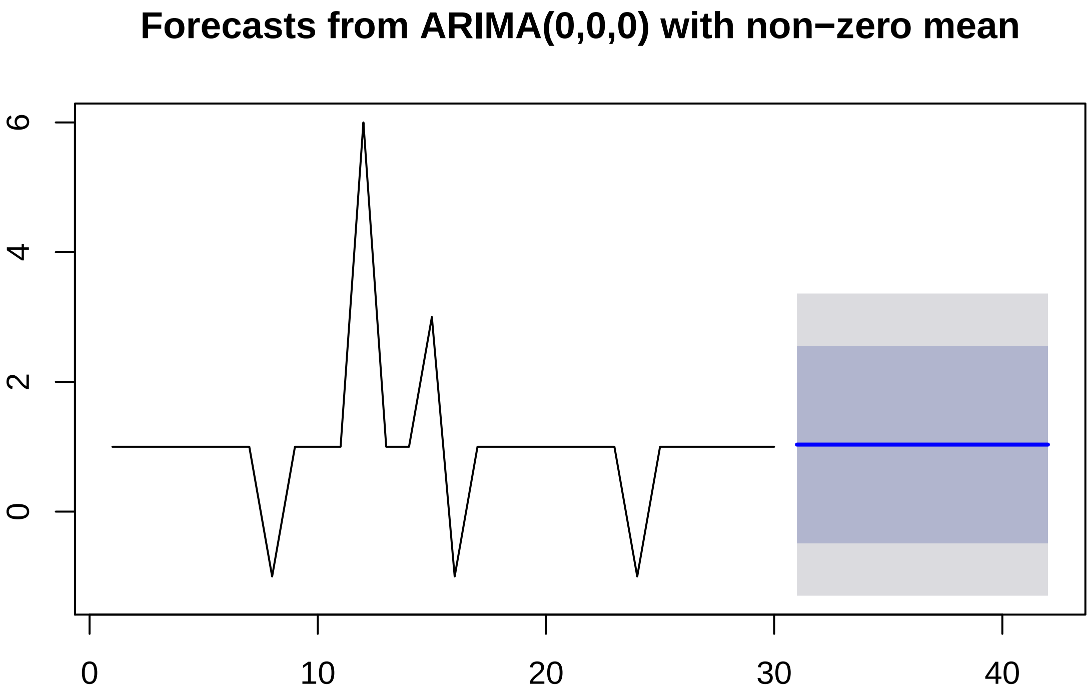



```
## Book ID:  4
```

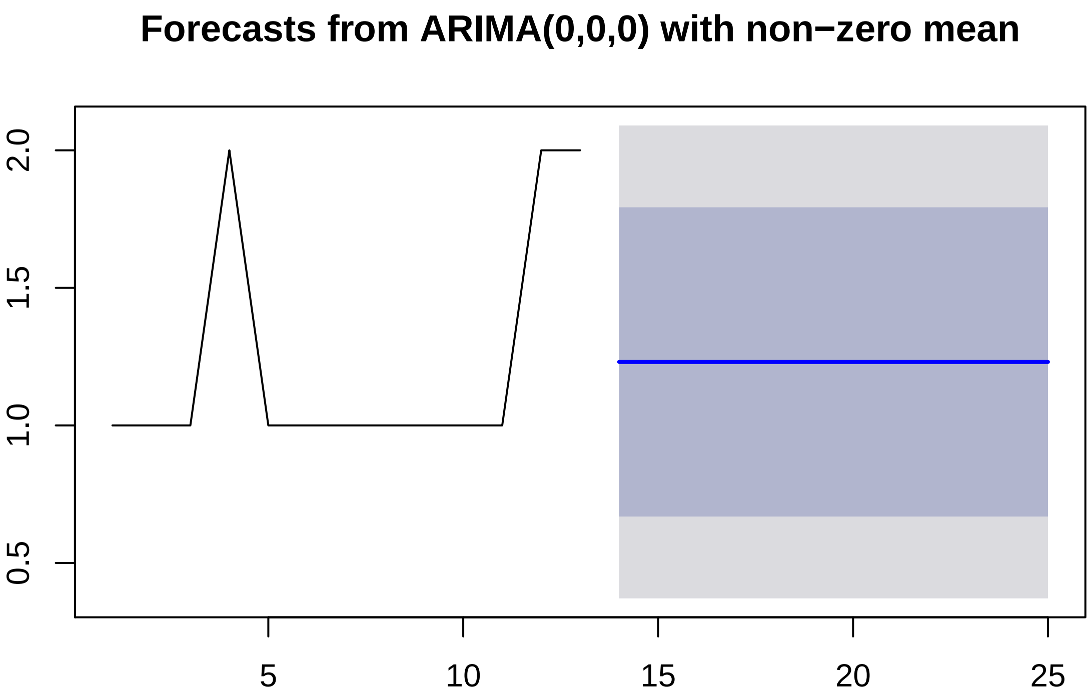
Forecasts from ARIMA(0,0,0) with non−zero mean



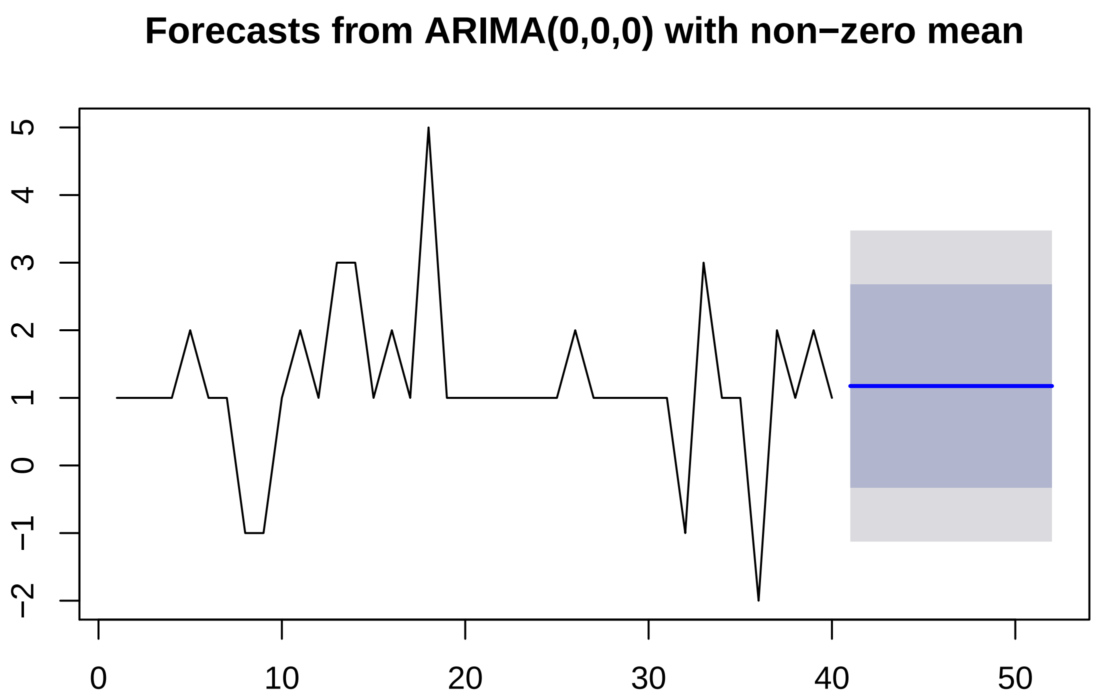

```
## Book ID:  5
```



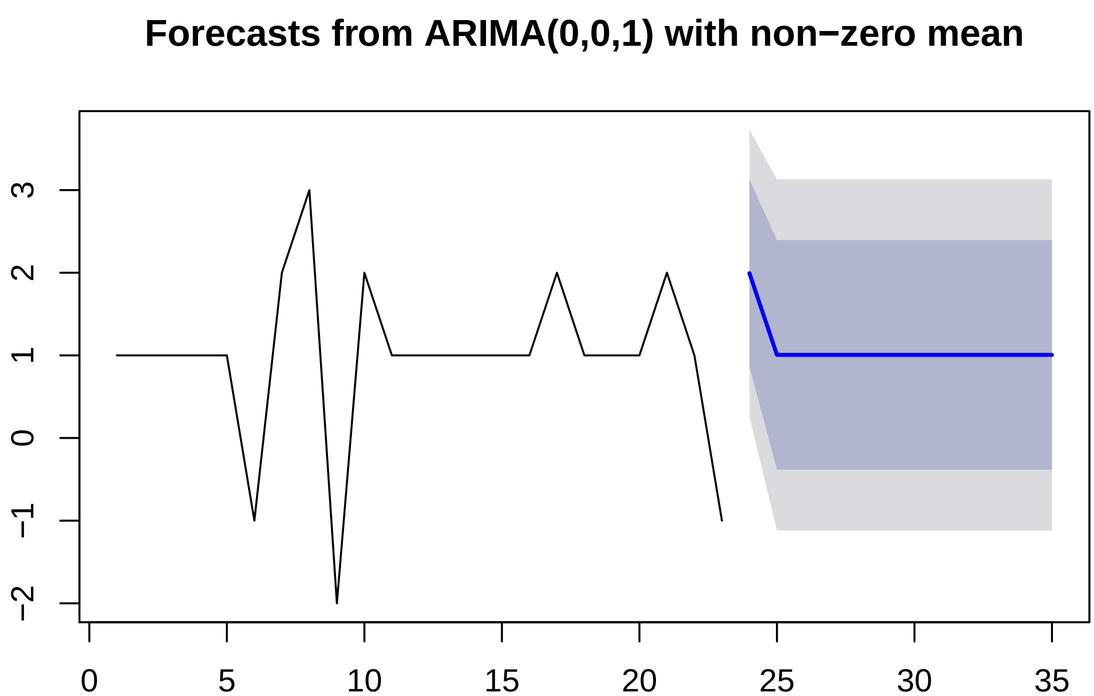
Book ID: 6



Book ID: 7

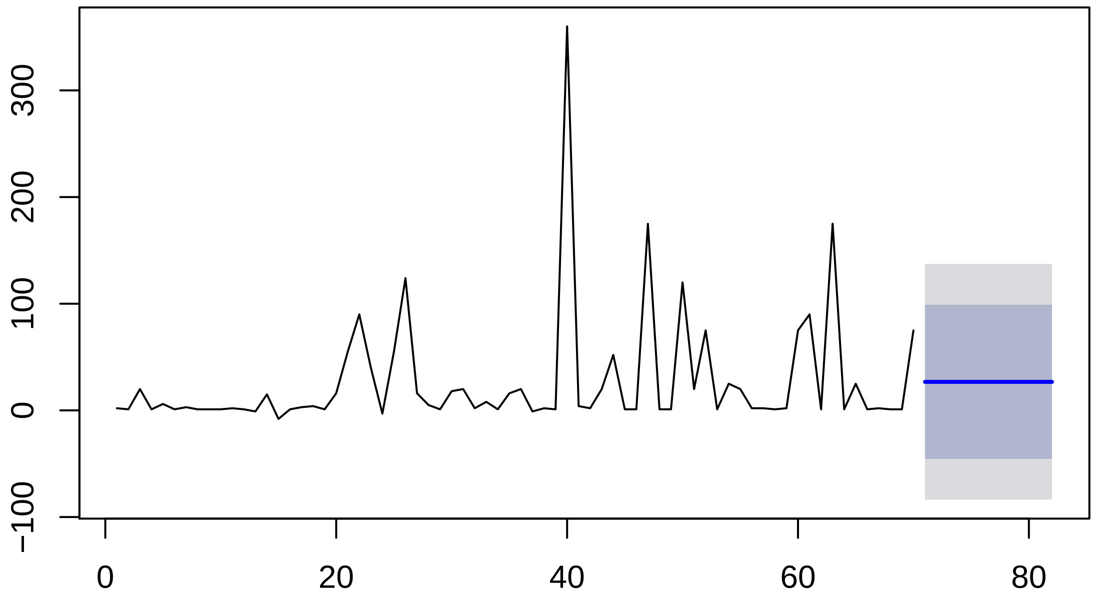



```
## Book ID:  8
```



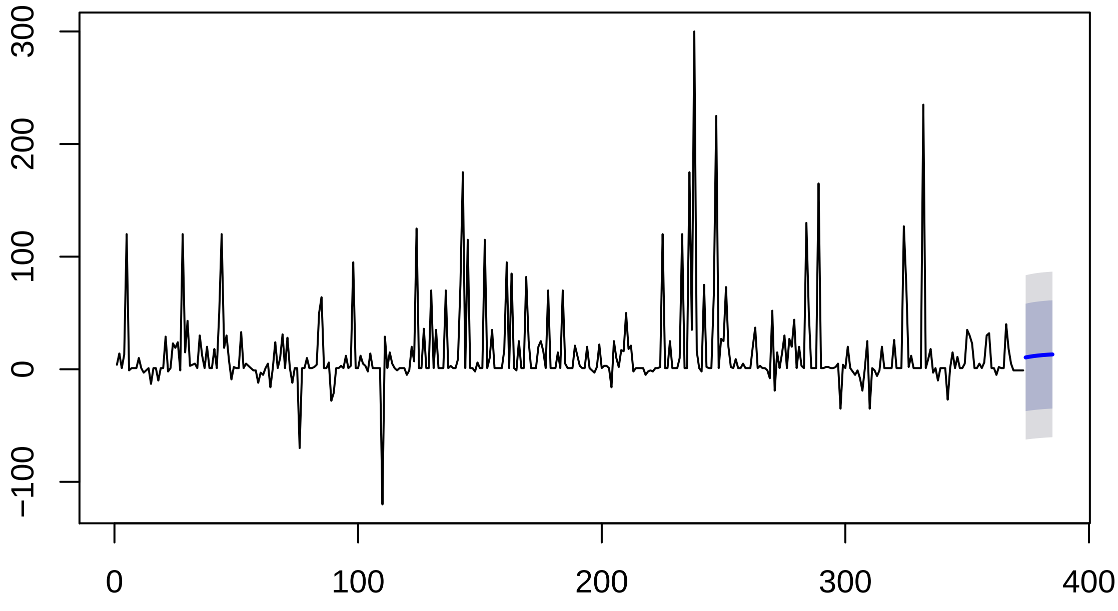

```
## Book ID:  9
```

Forecasts from ARIMA(0,0,0) with non−zero mean

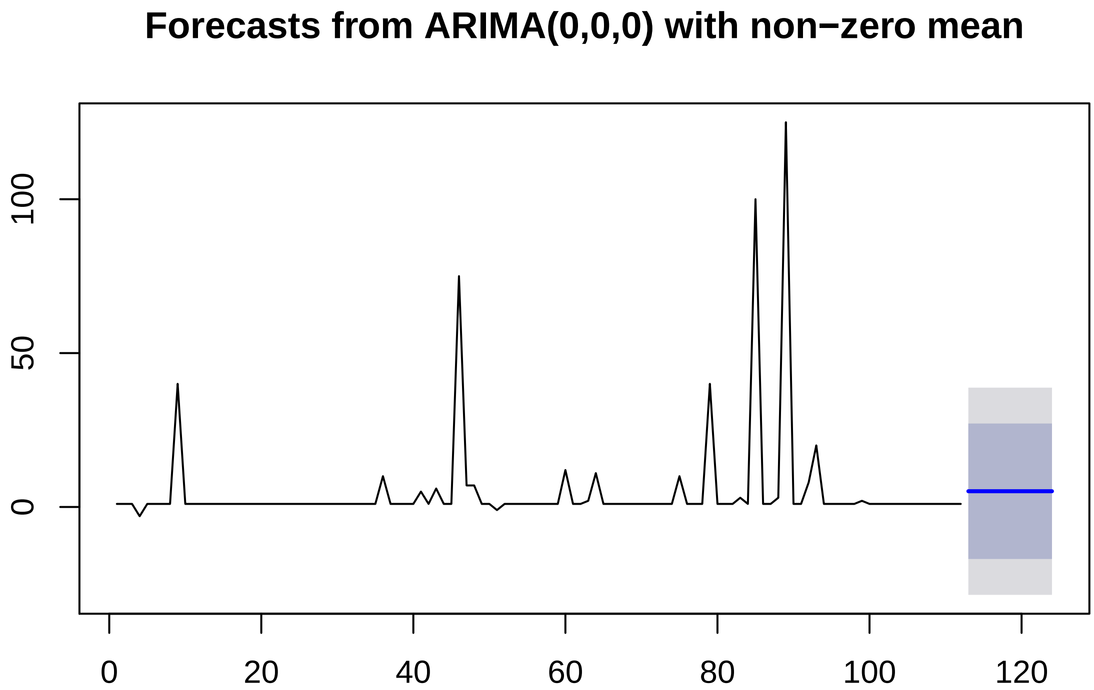



```
## Book ID:  10
```

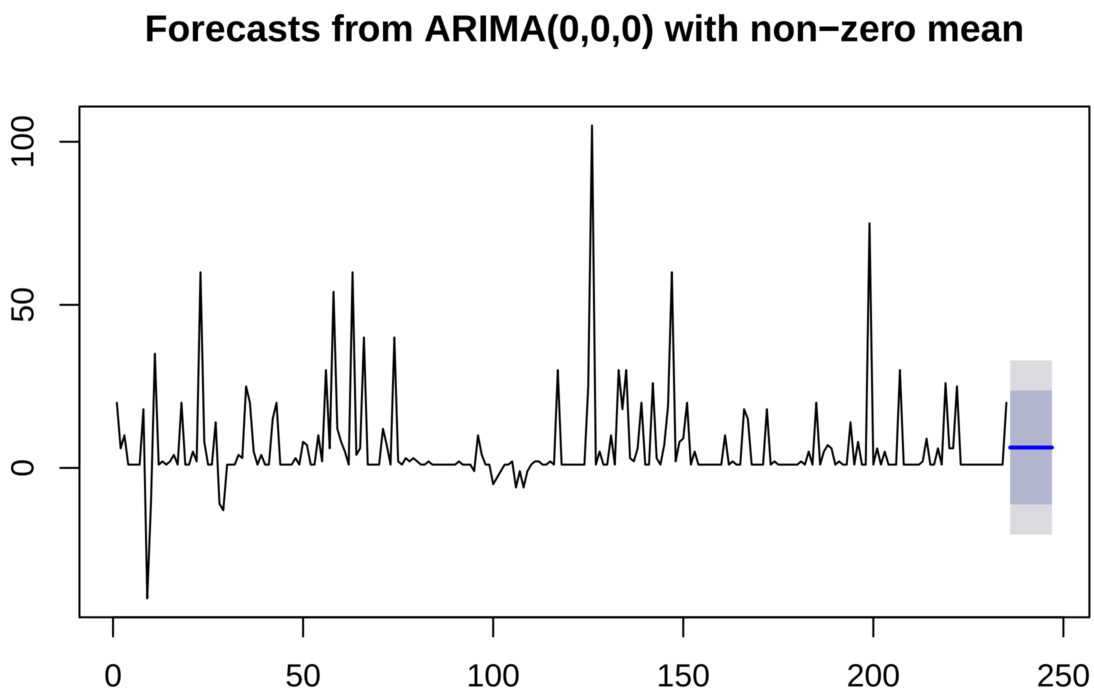
Forecasts from ARIMA(1,0,1) with non−zero mean




```
## Book ID:  11
```

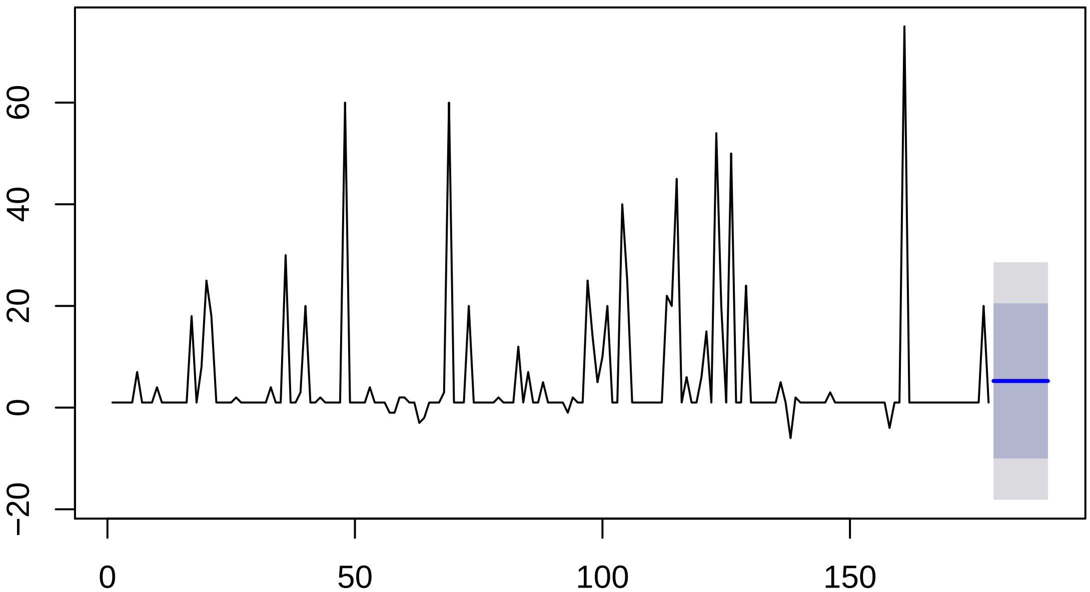



```
## Book ID:  14
```



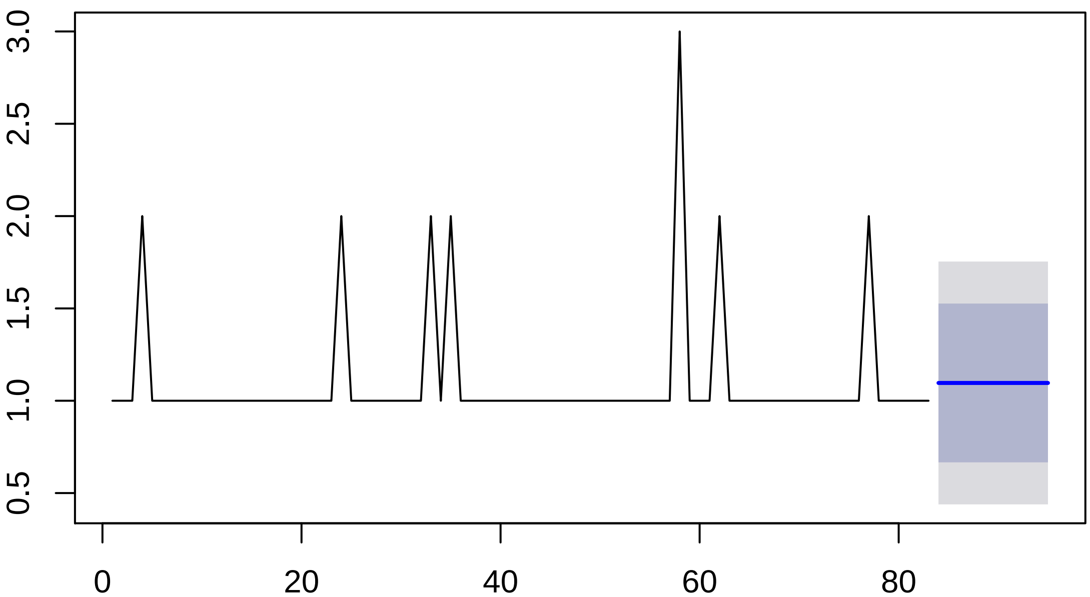

```
## Book ID:  15
```

Forecasts from ARIMA(0,0,0) with non−zero mean

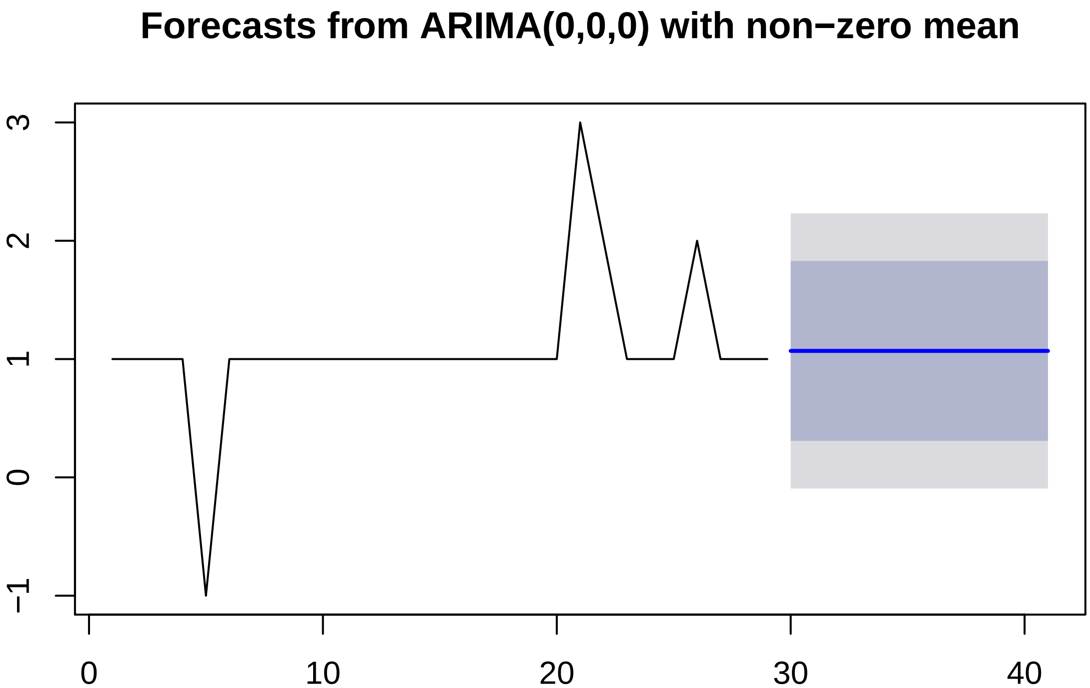



```
## Book ID:  16
```

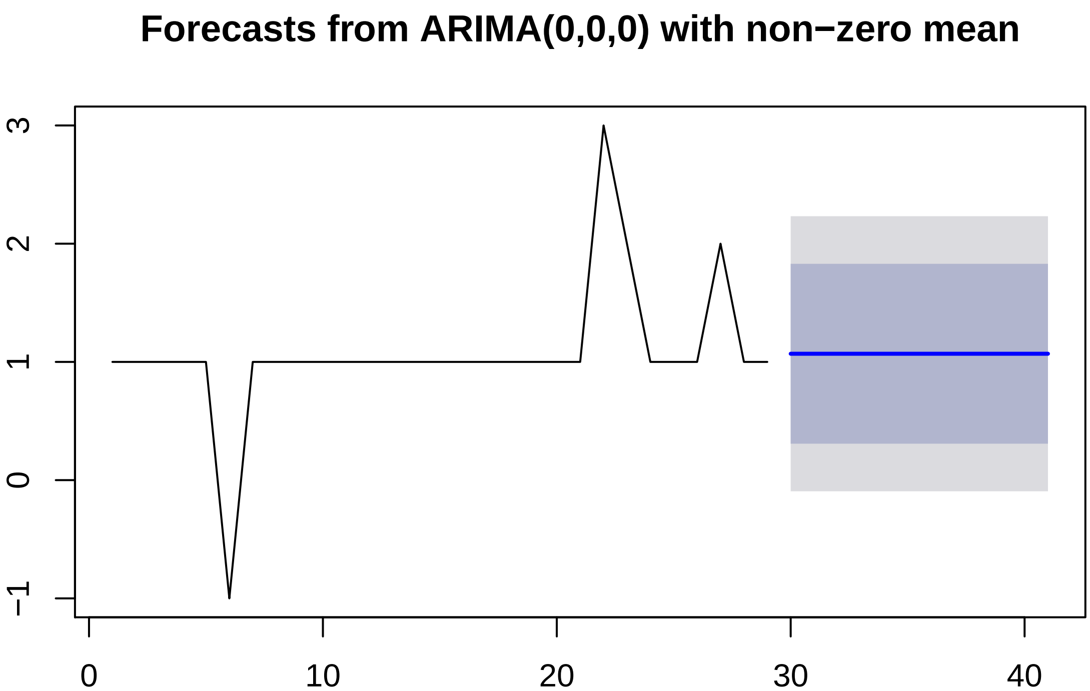
Forecasts from ARIMA(0,0,0) with non−zero mean



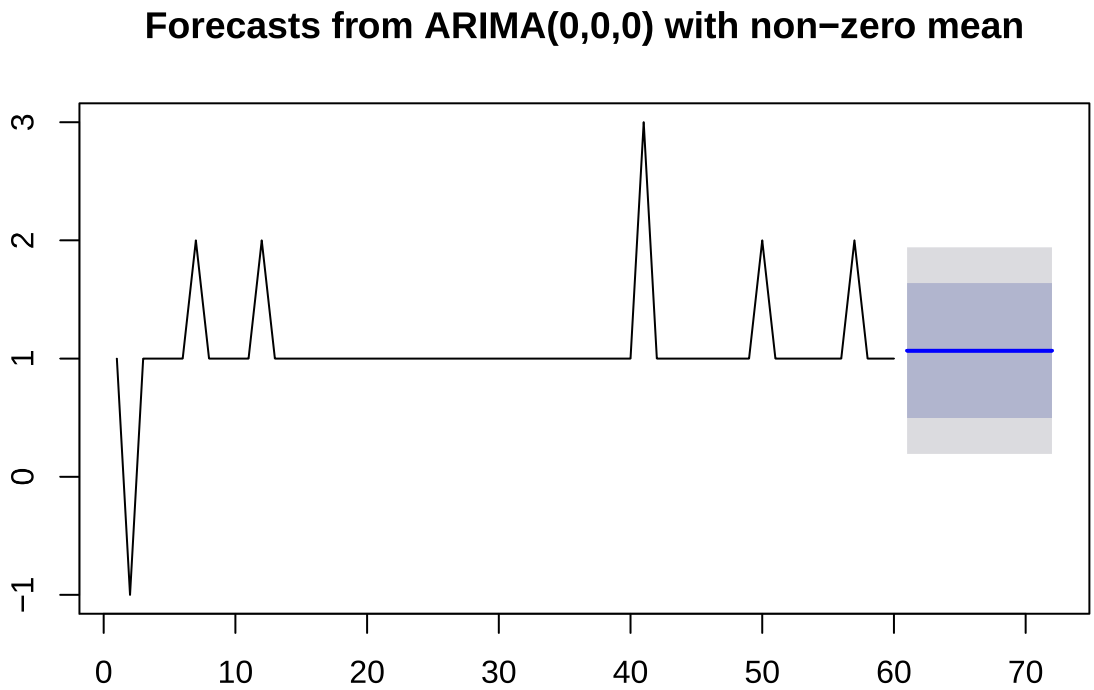

```
## Book ID:  17
```



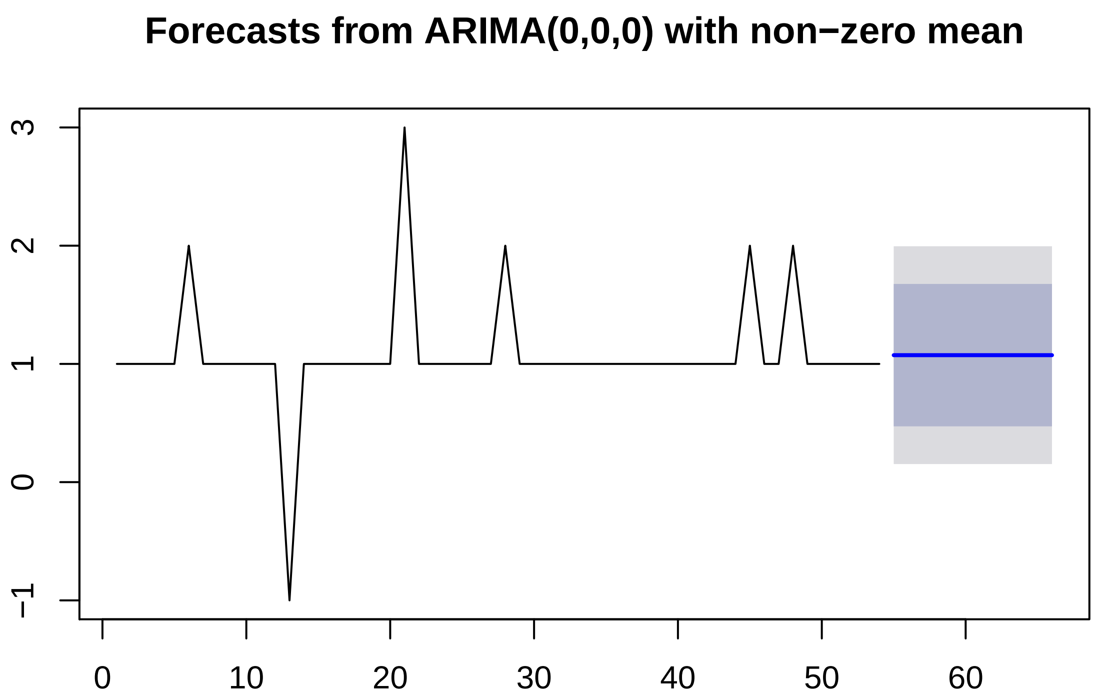

```
## Book ID:  18
```



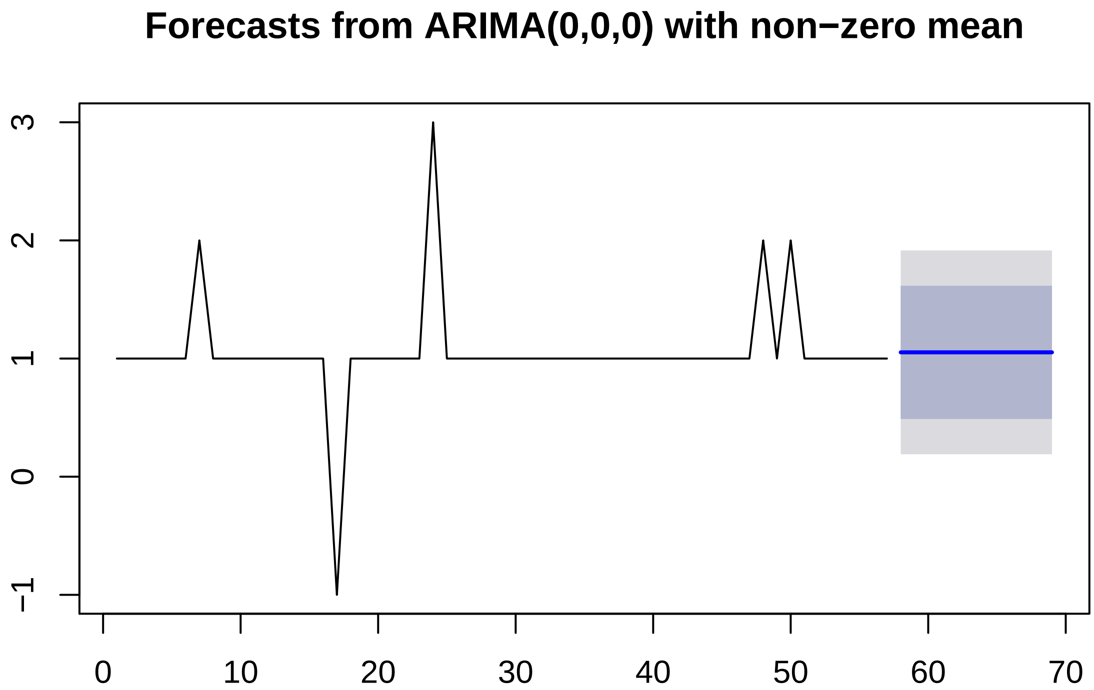

```
## Book ID:  19
```



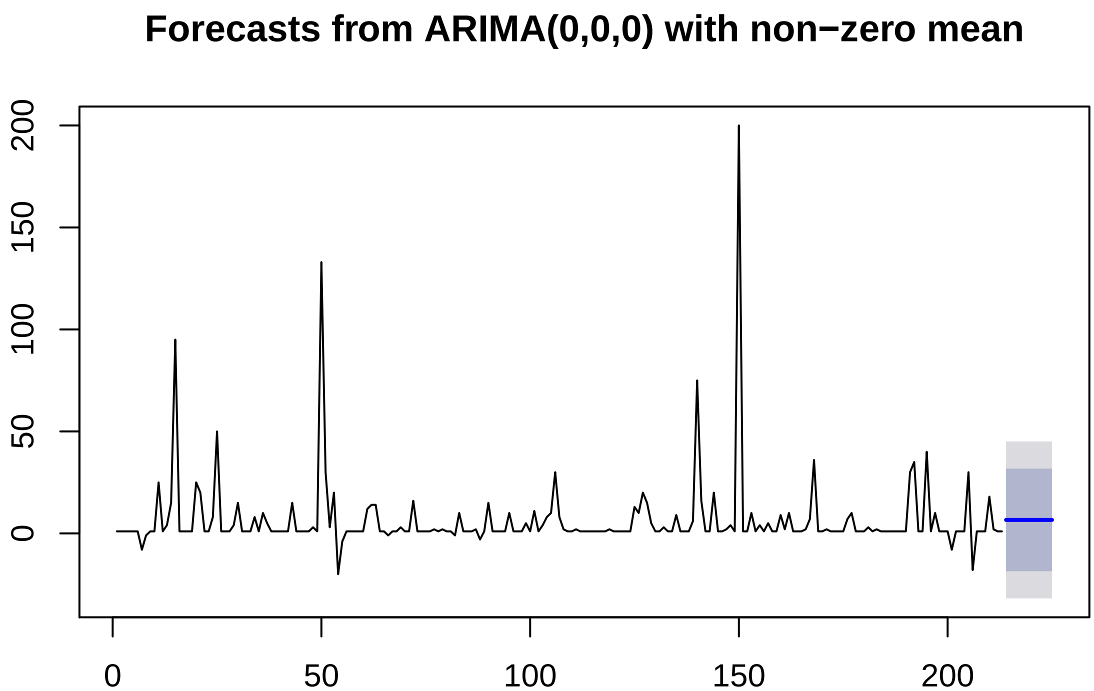
Book ID: 20



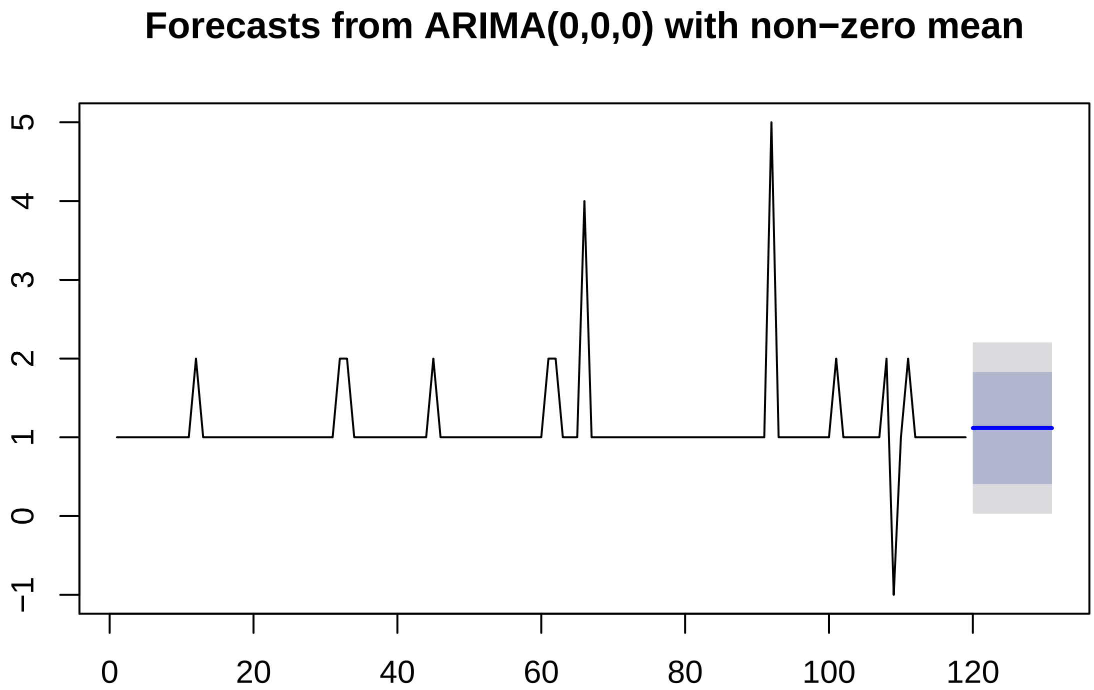
Book ID: 21



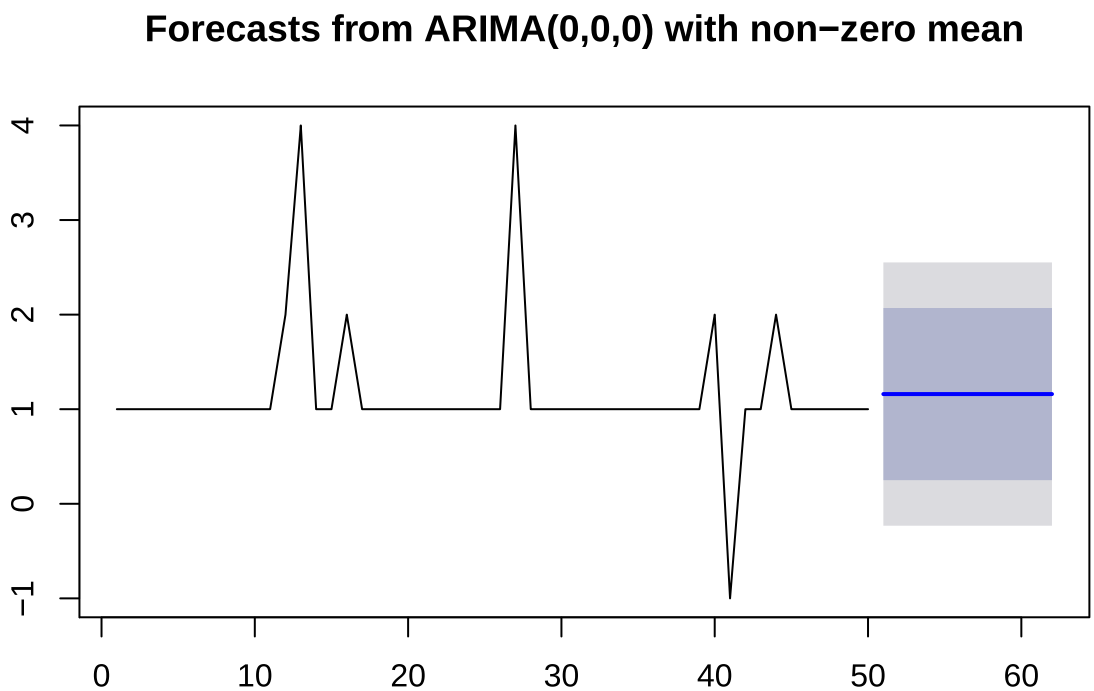

```
## Book ID:  22
```



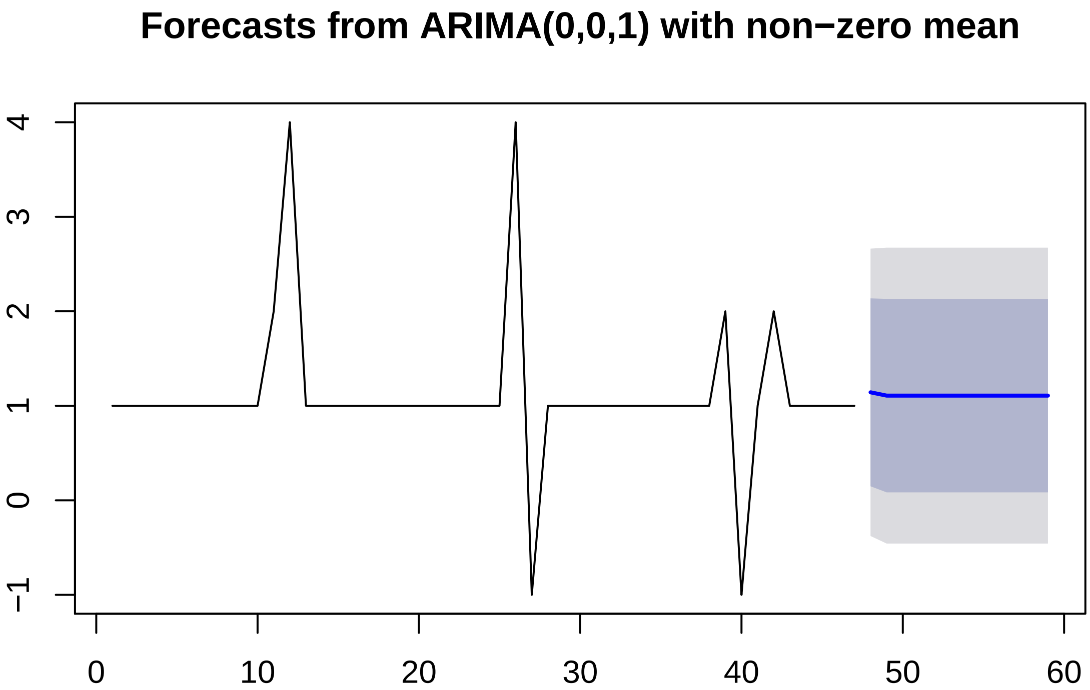

```
## Book ID:  23
```



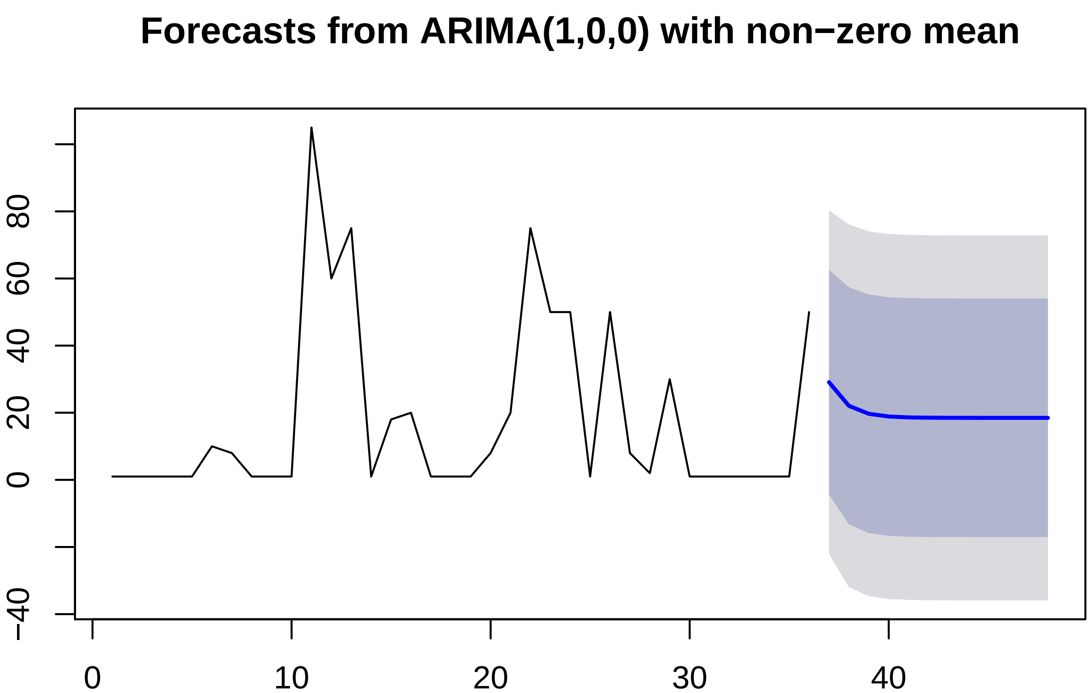

```
## Book ID:  24
```




```
## Book ID:  25
```

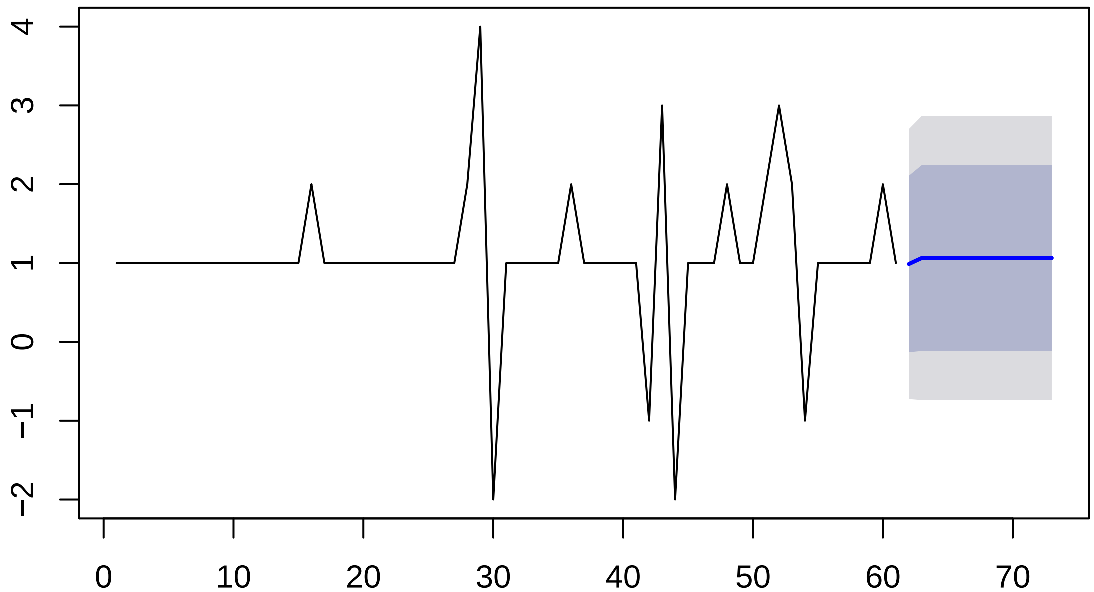


Book ID: 26



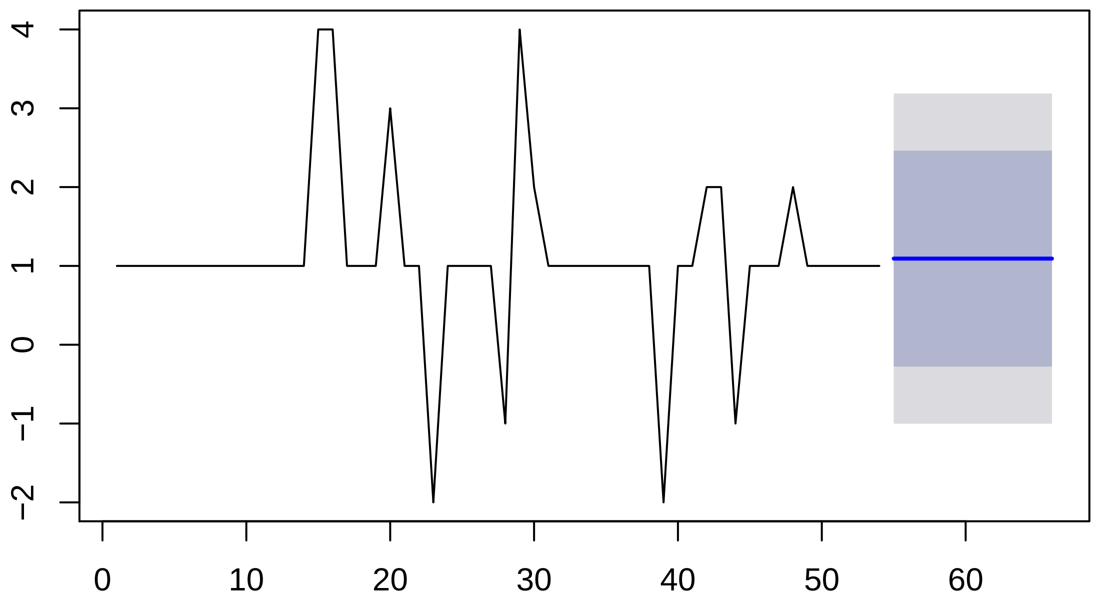
Book ID: 27

Forecasts from ARIMA(0,0,1) with non−zero mean



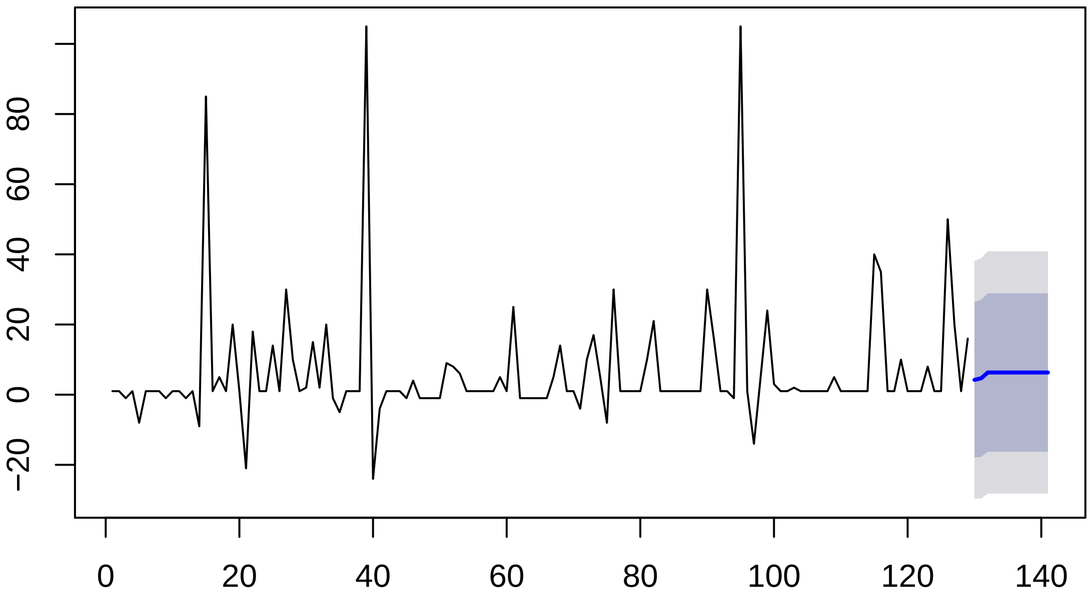

```
## Book ID:  28
```

Forecasts from ARIMA(0,0,0) with non−zero mean



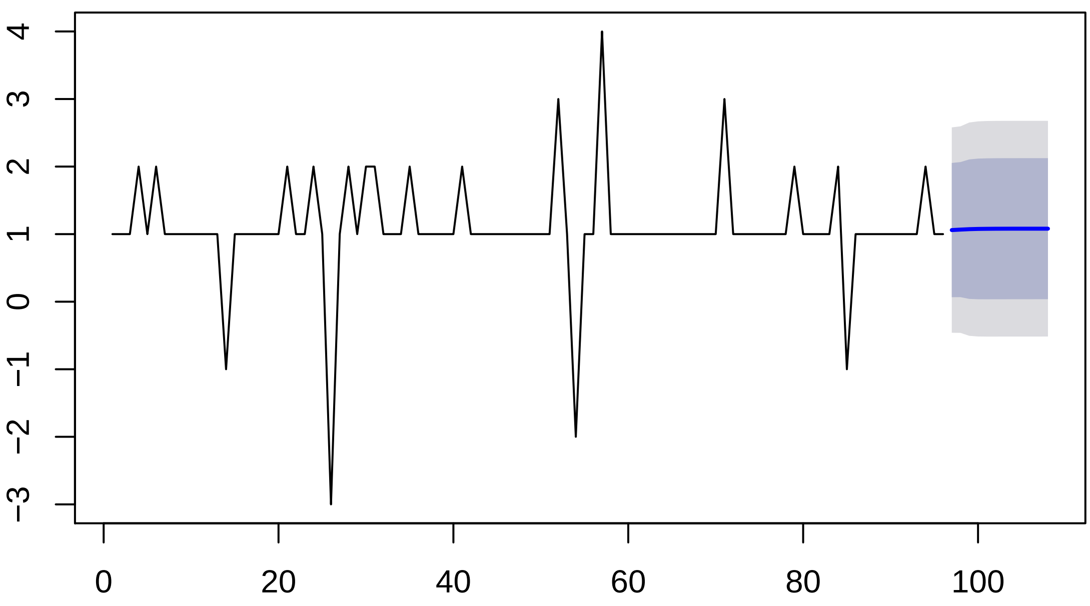

```
## Book ID:  29
```

Forecasts from ARIMA(0,0,2) with non−zero mean

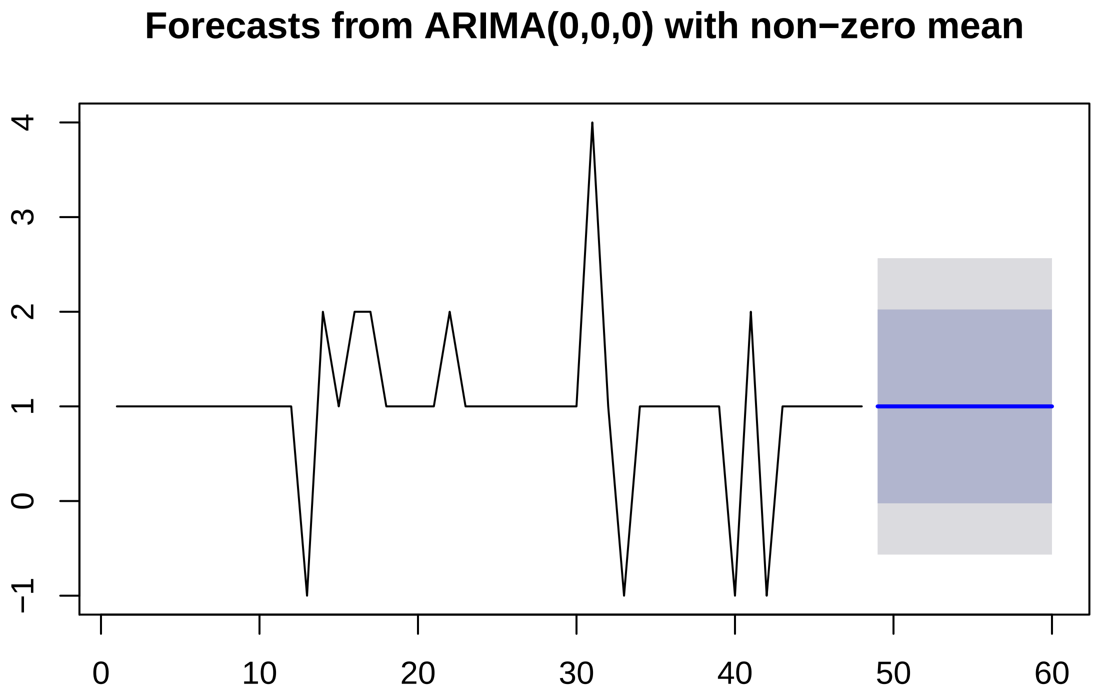



```
## Book ID:  30
```

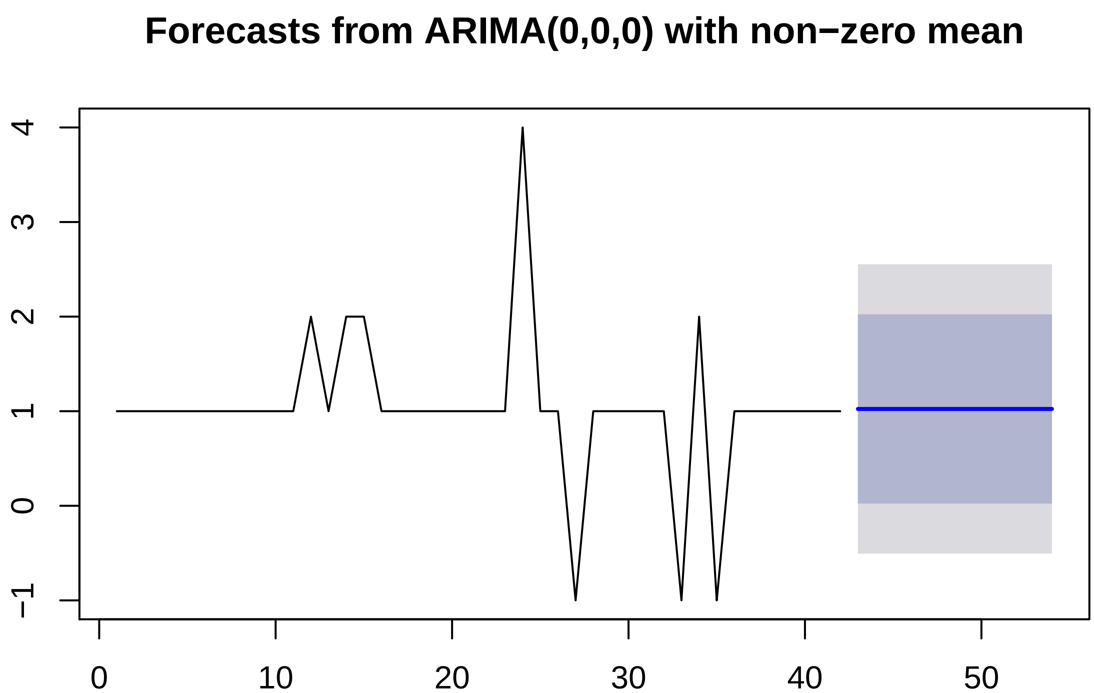
Forecasts from ARIMA(1,0,2) with non−zero mean



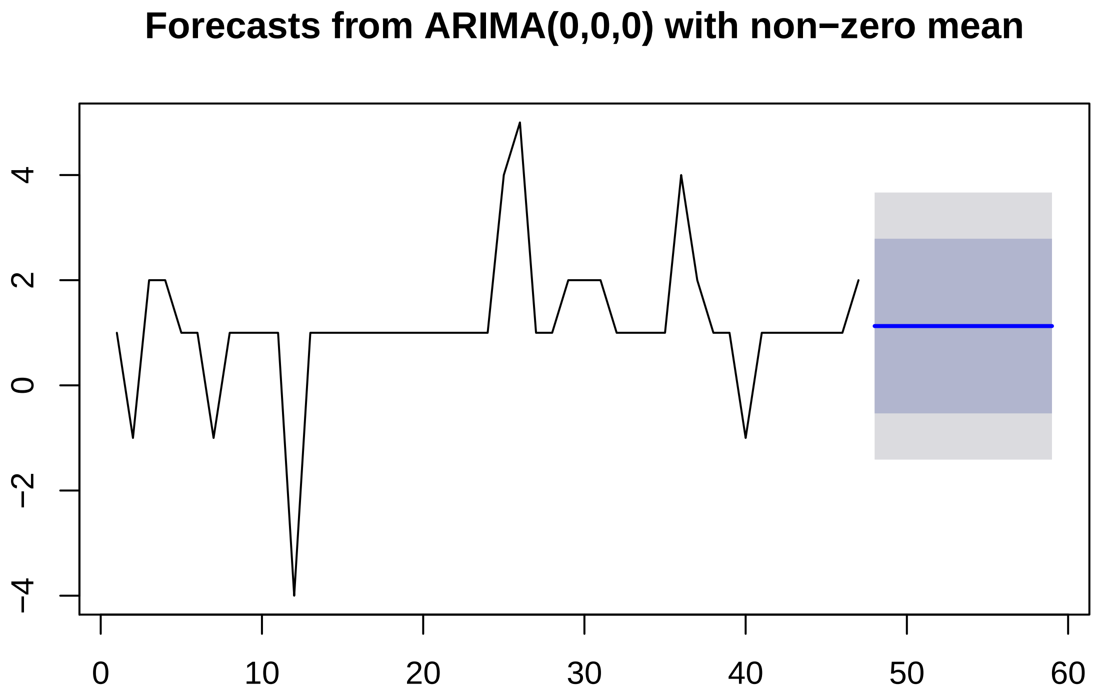

```
## Book ID:  31
```



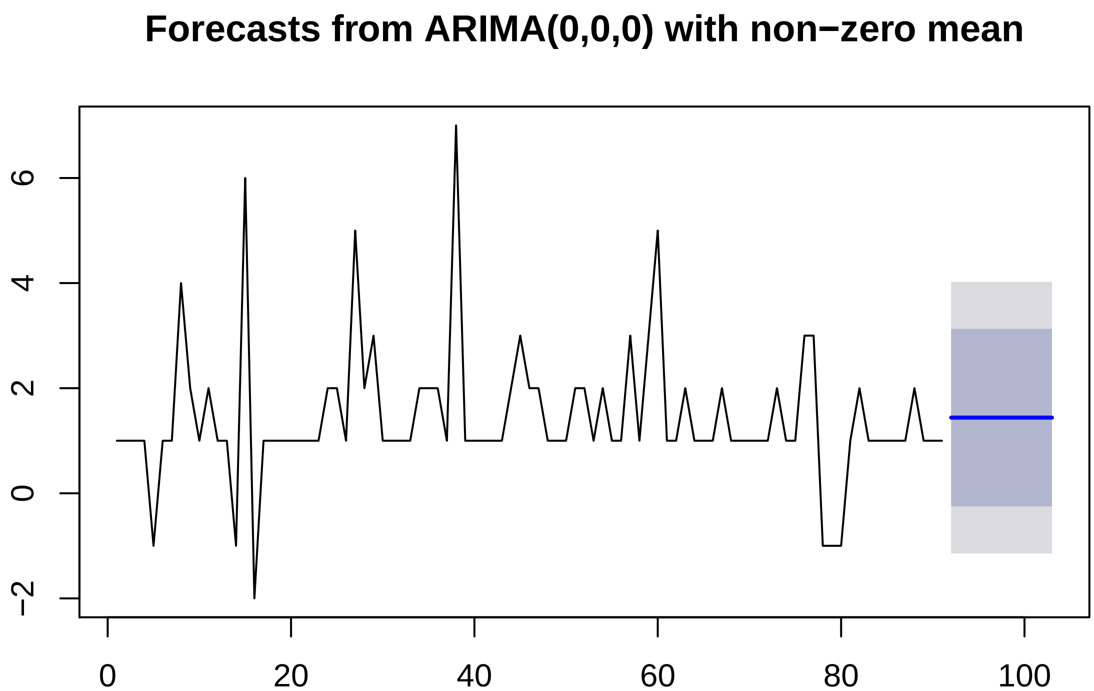

```
## Book ID:  32
```



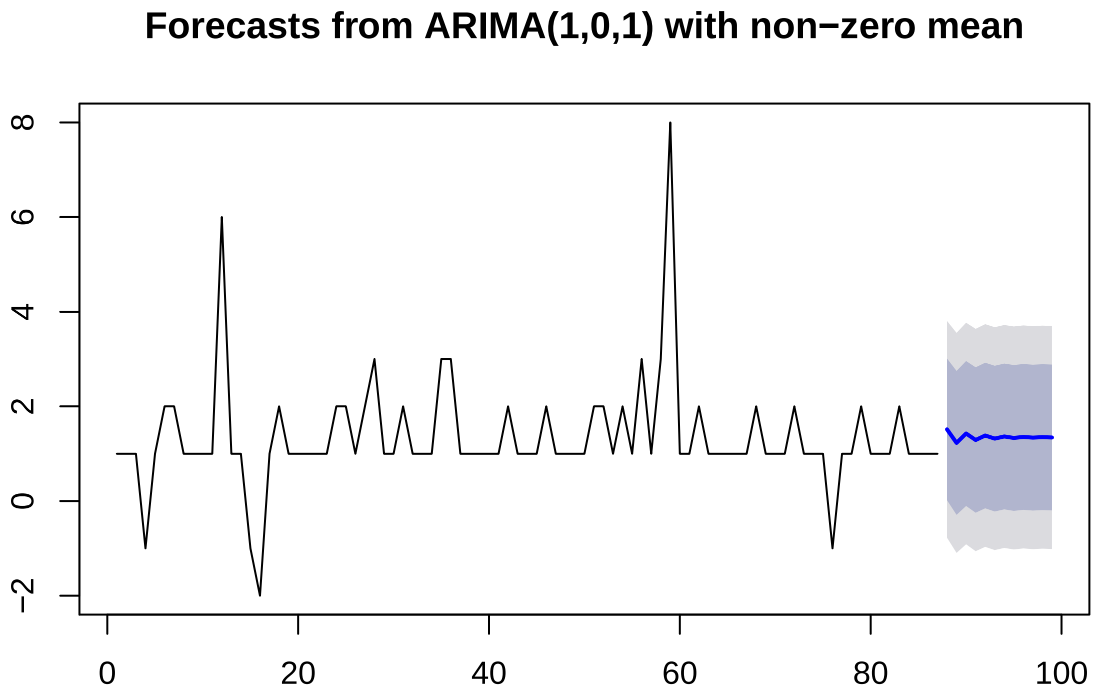

```
## Book ID:  33
```



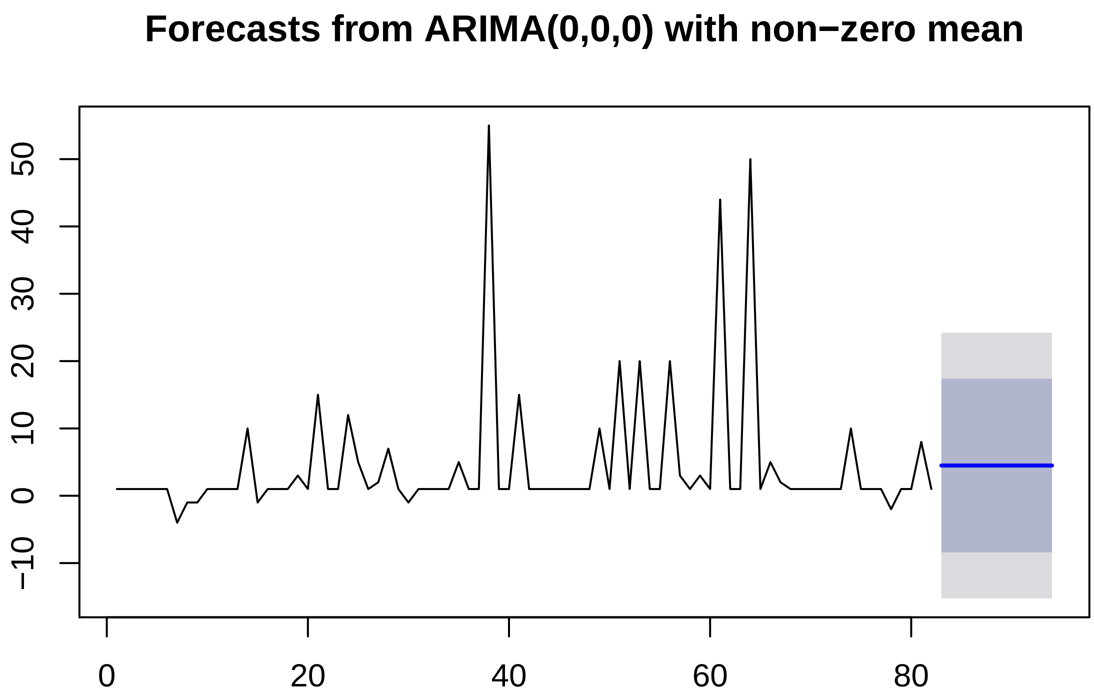

```
## Book ID:  34
```




```
## Book ID:  35
```

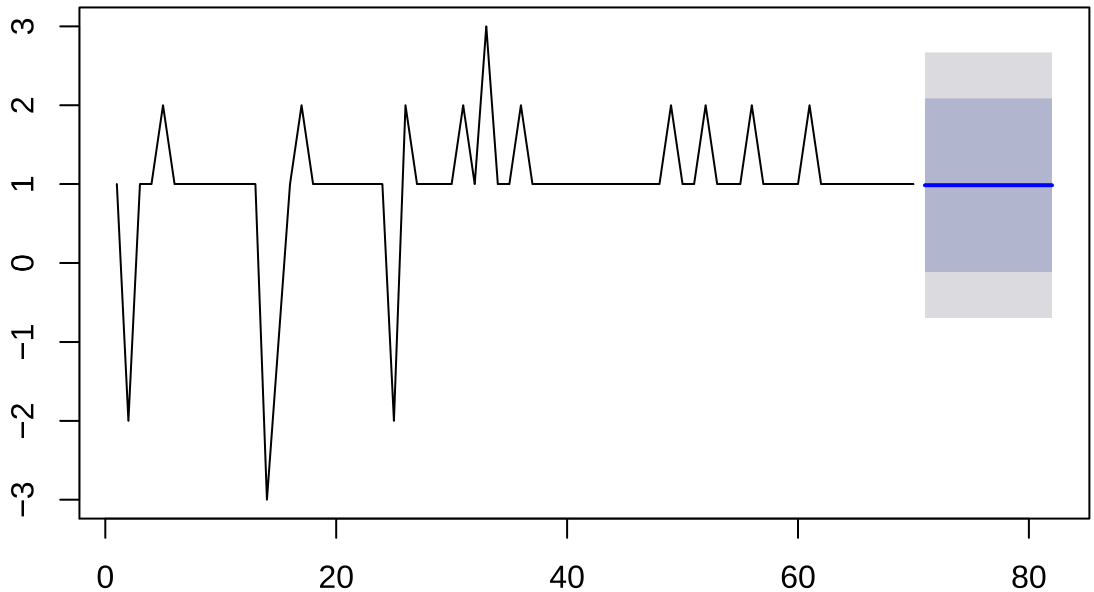


Book ID: 36



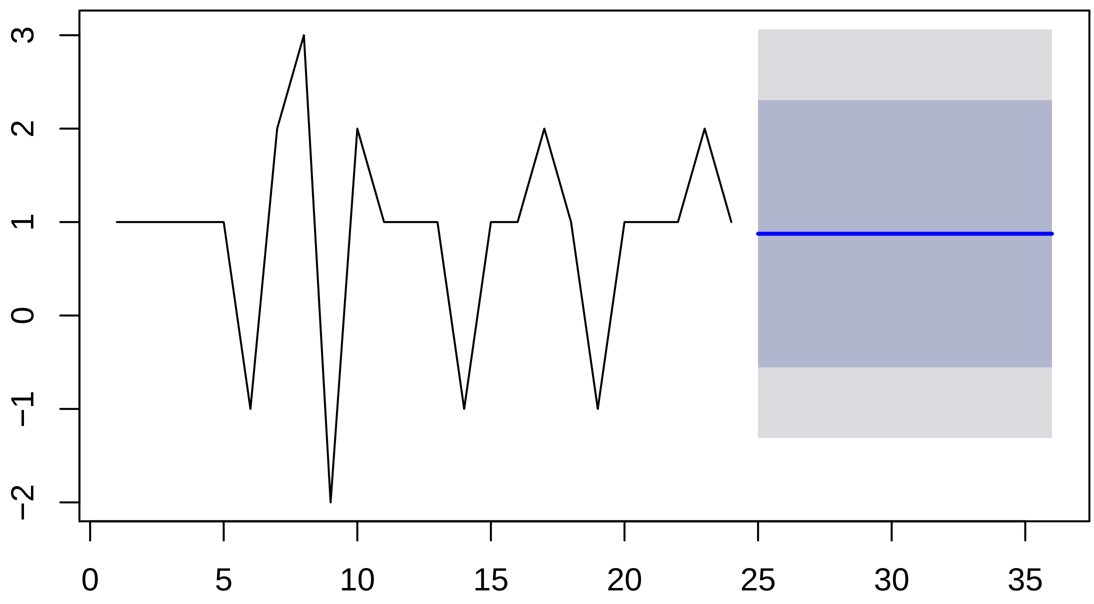
Book ID: 37

Forecasts from ARIMA(0,0,0) with non−zero mean



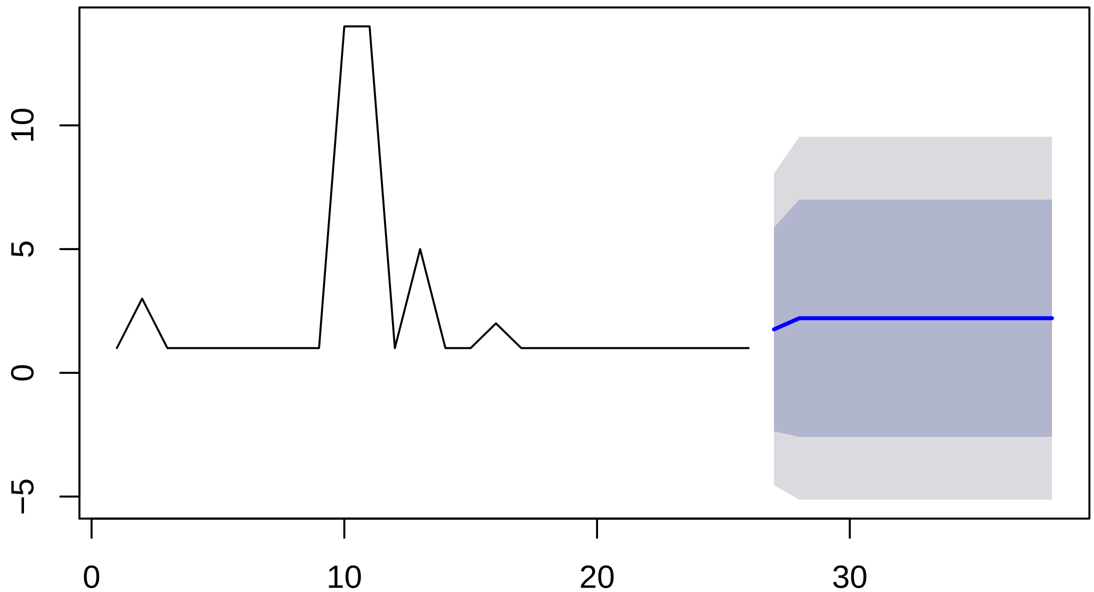

```
## Book ID:  38
```

Forecasts from ARIMA(0,0,0) with non−zero mean



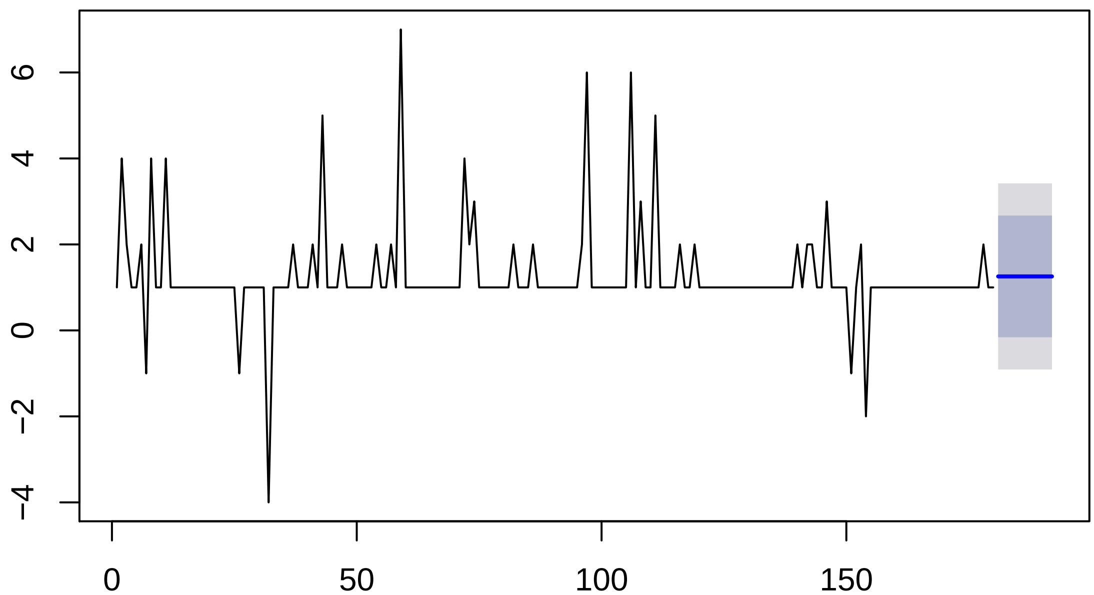

```
## Book ID:  39
```

Forecasts from ARIMA(0,0,1) with non−zero mean



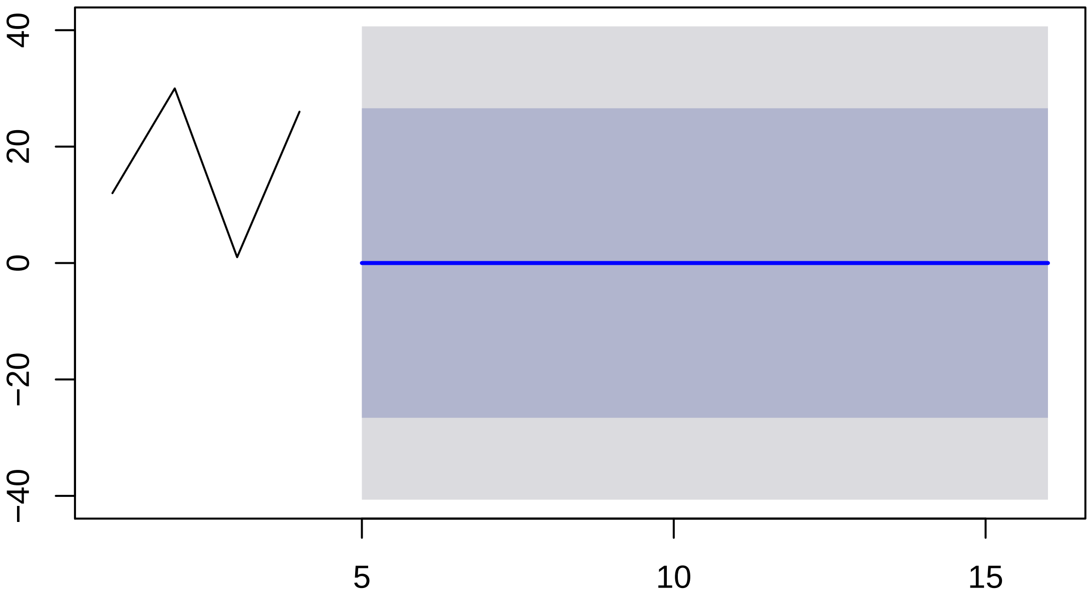

```
## Book ID:  40
```

Forecasts from ARIMA(0,0,0) with non−zero mean



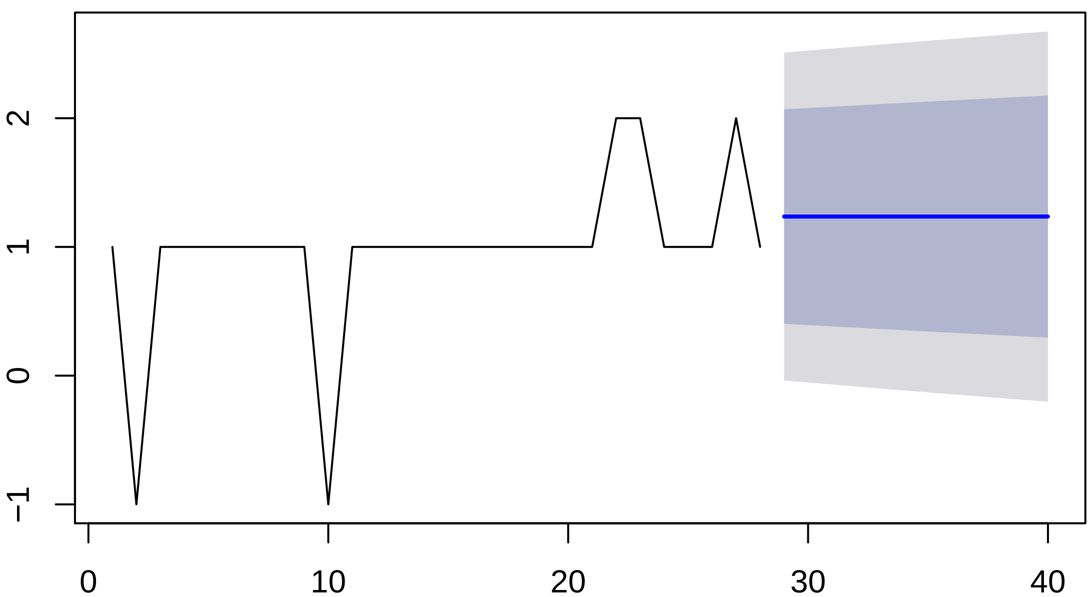

```
## Book ID:  41
```

Forecasts from ARIMA(0,0,0) with zero mean

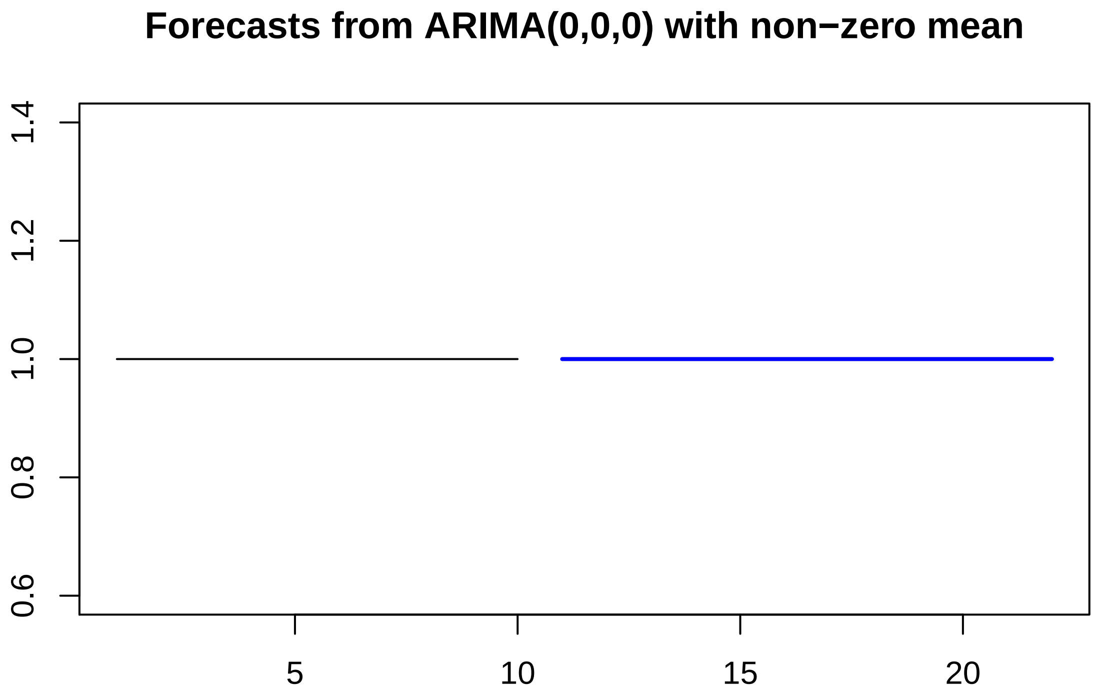


Book ID: 42

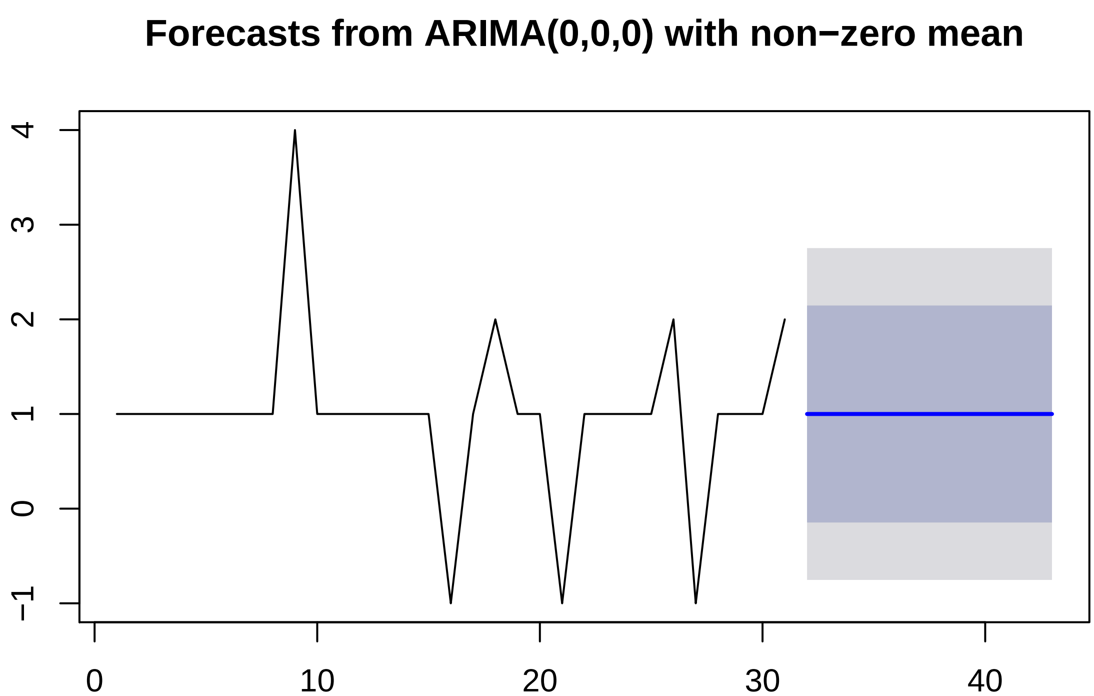
Forecasts from ARIMA(0,1,1)



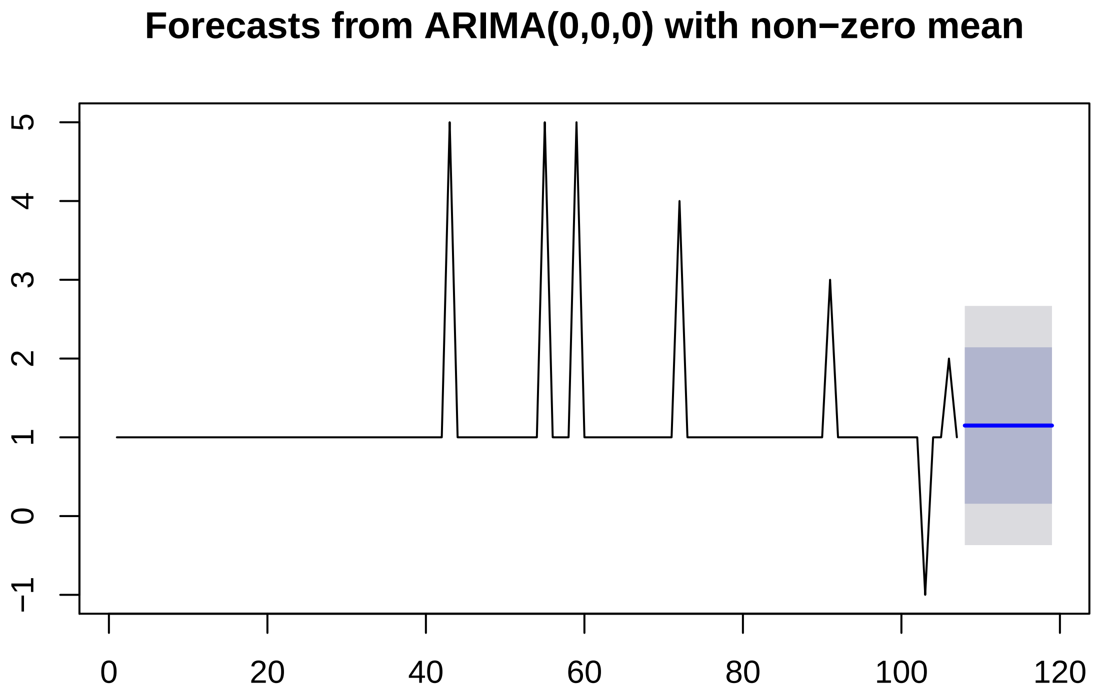
Book ID: 43



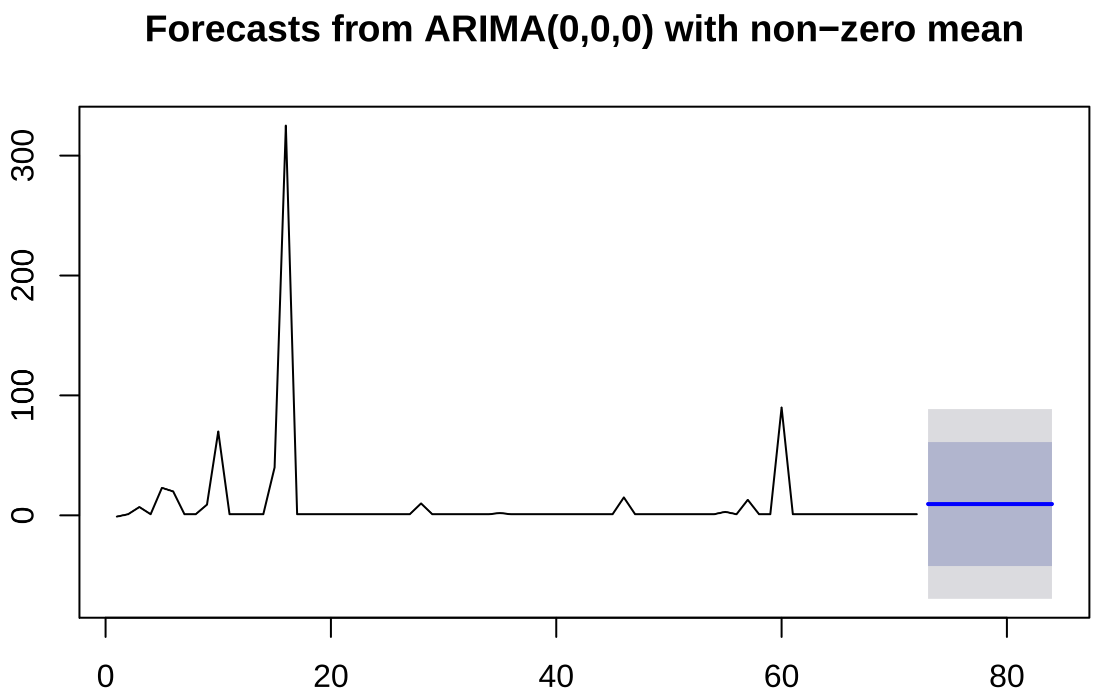

```
## Book ID:  49
```




```
## Book ID:  54
```

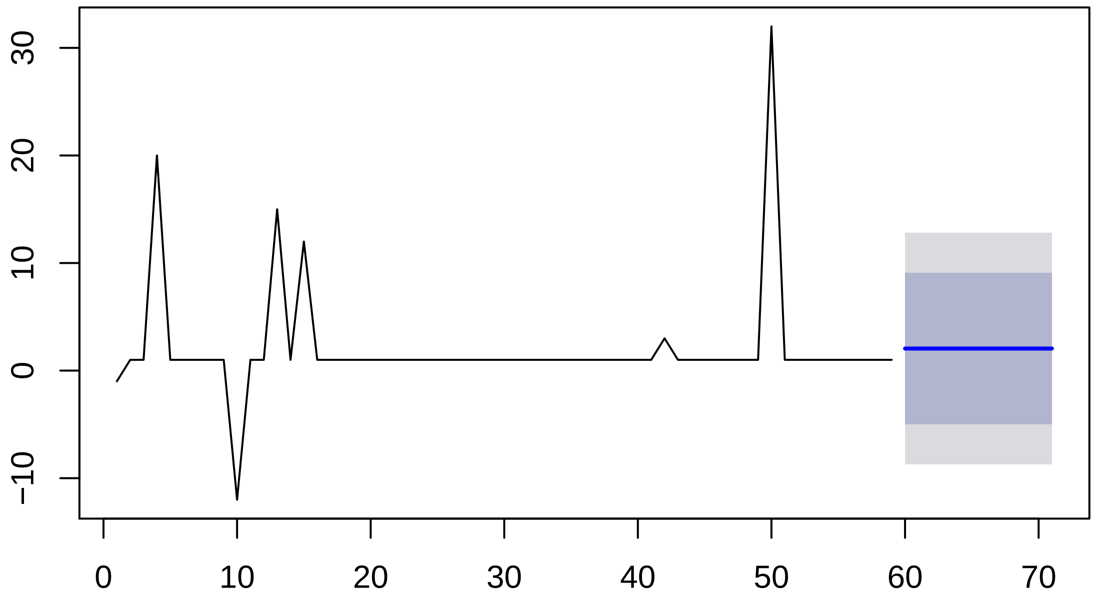



```
## Book ID:  56
```



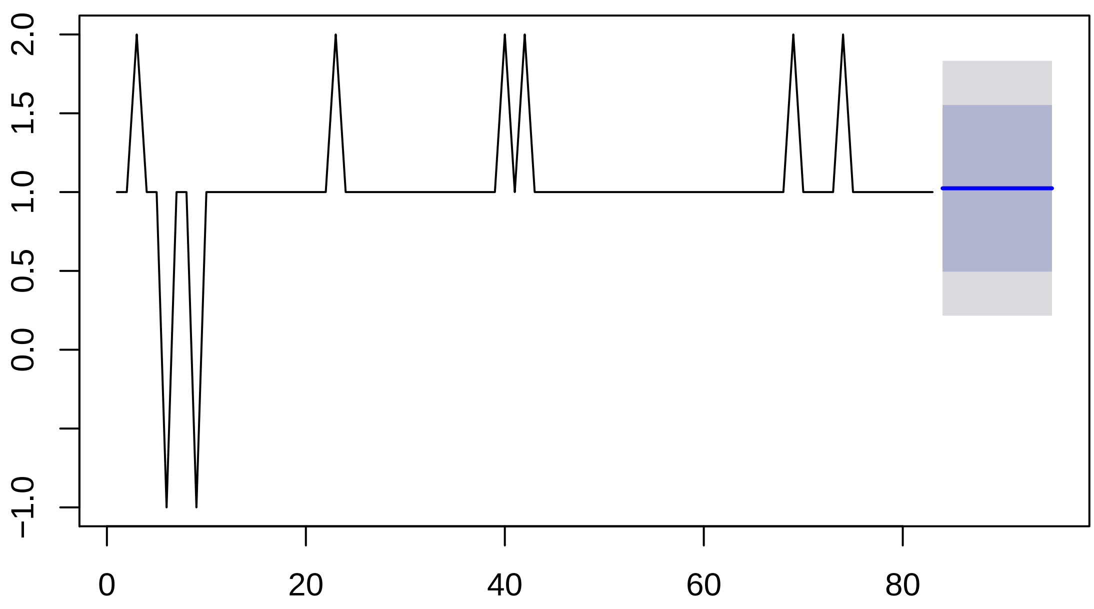

```
## Book ID:  57
```

Forecasts from ARIMA(0,0,0) with non−zero mean

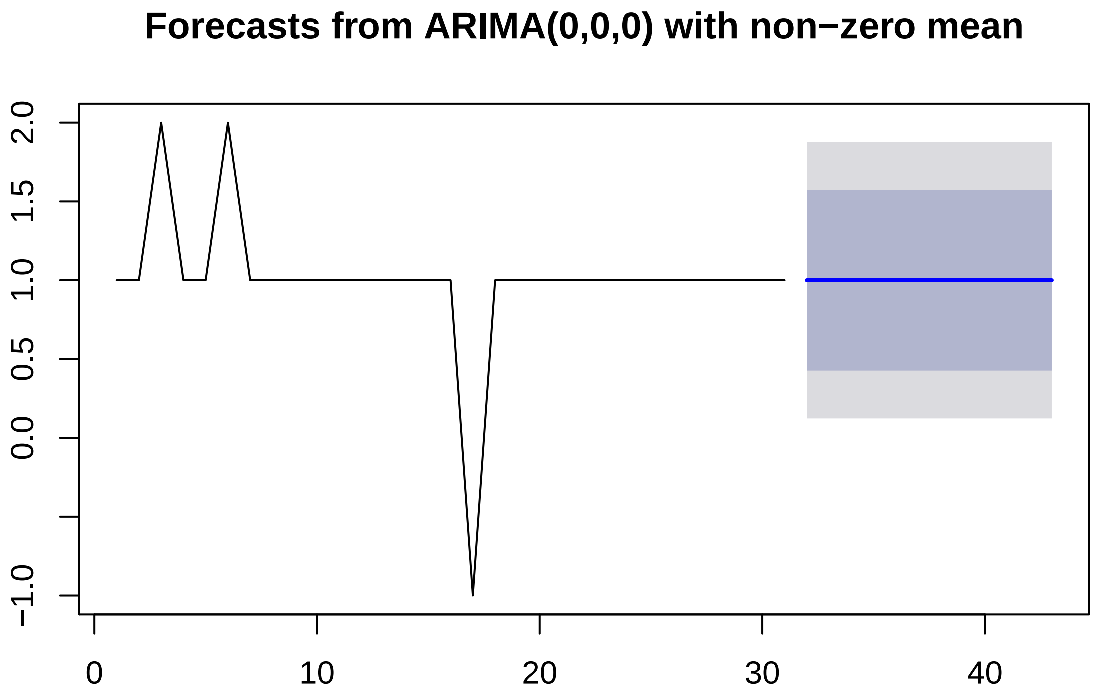



```
## Book ID:  58
```

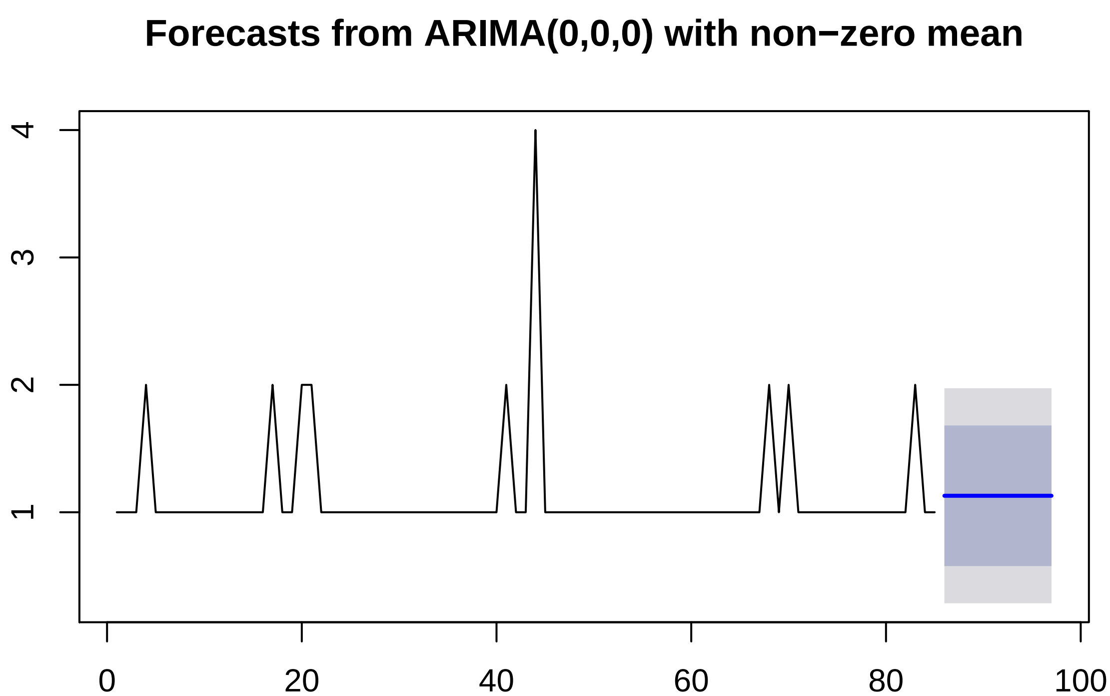
Forecasts from ARIMA(0,0,0) with non−zero mean



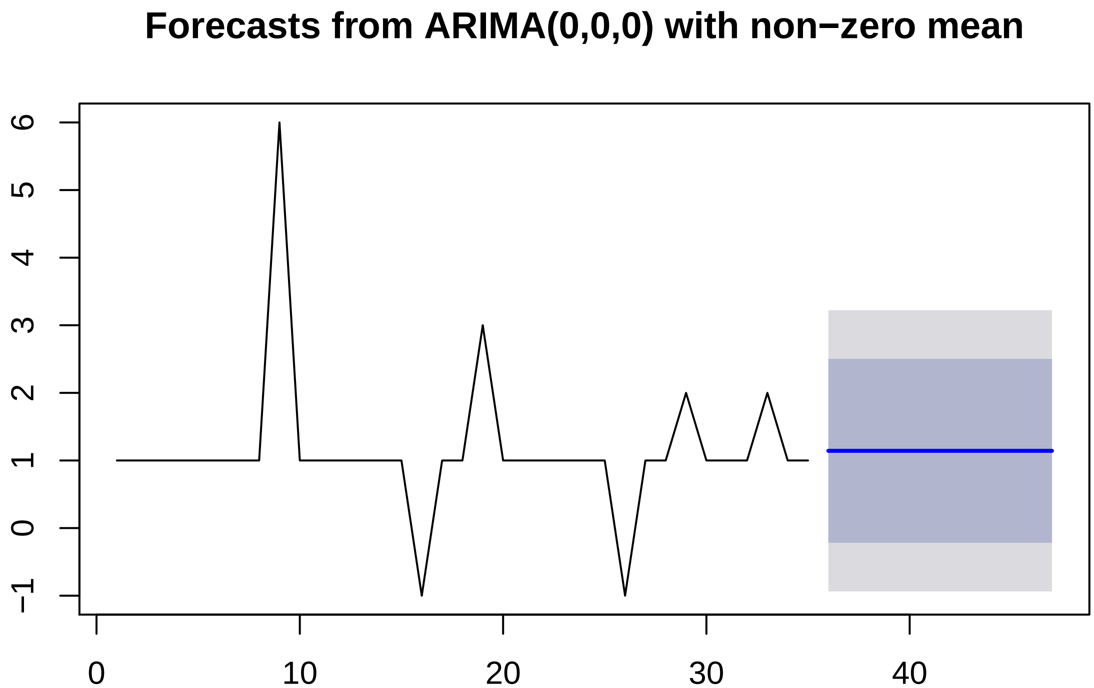

```
## Book ID:  59
```



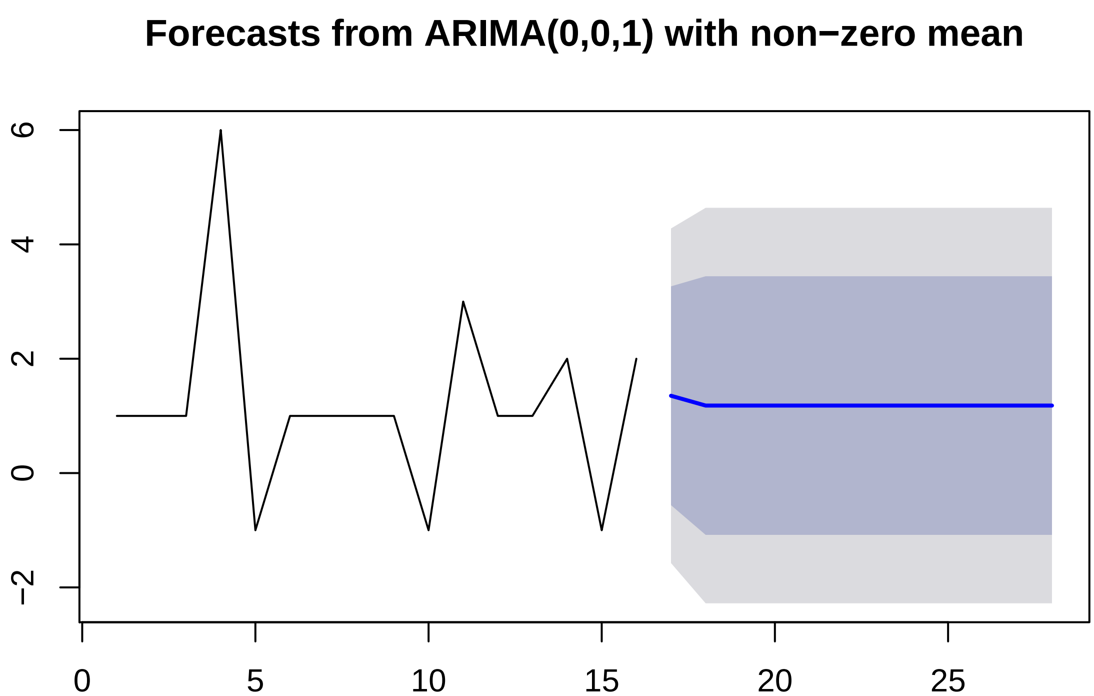

```
## Book ID:  60
```



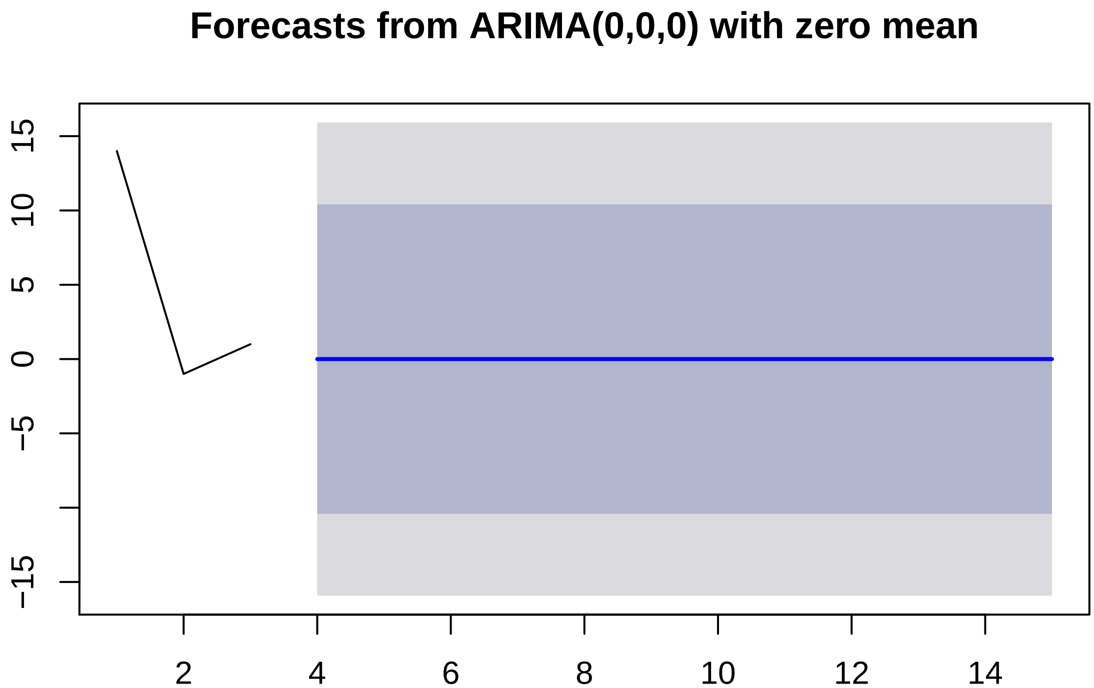

```
## Book ID:  61
```



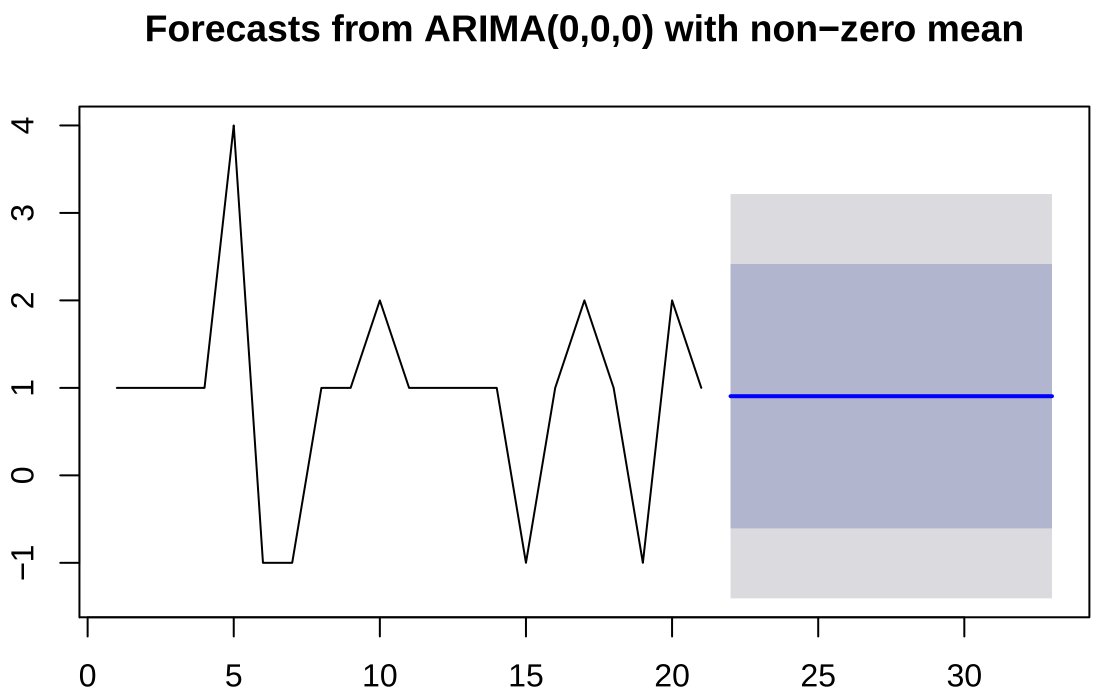

```
## Book ID:  62
```



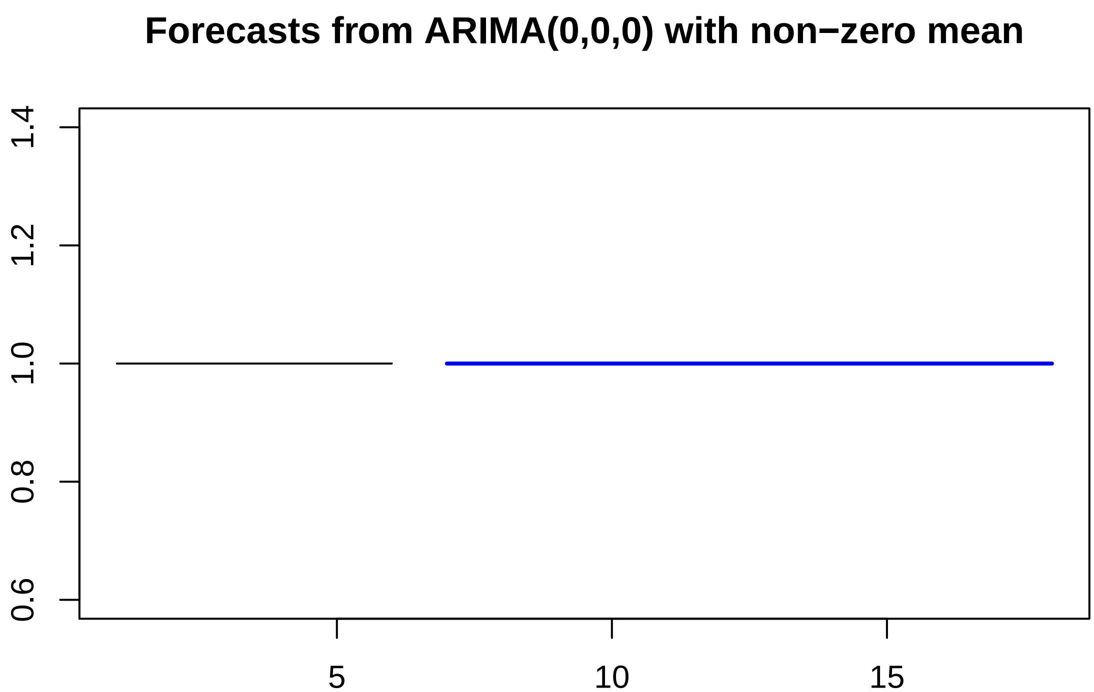

```
## Book ID:  63
```



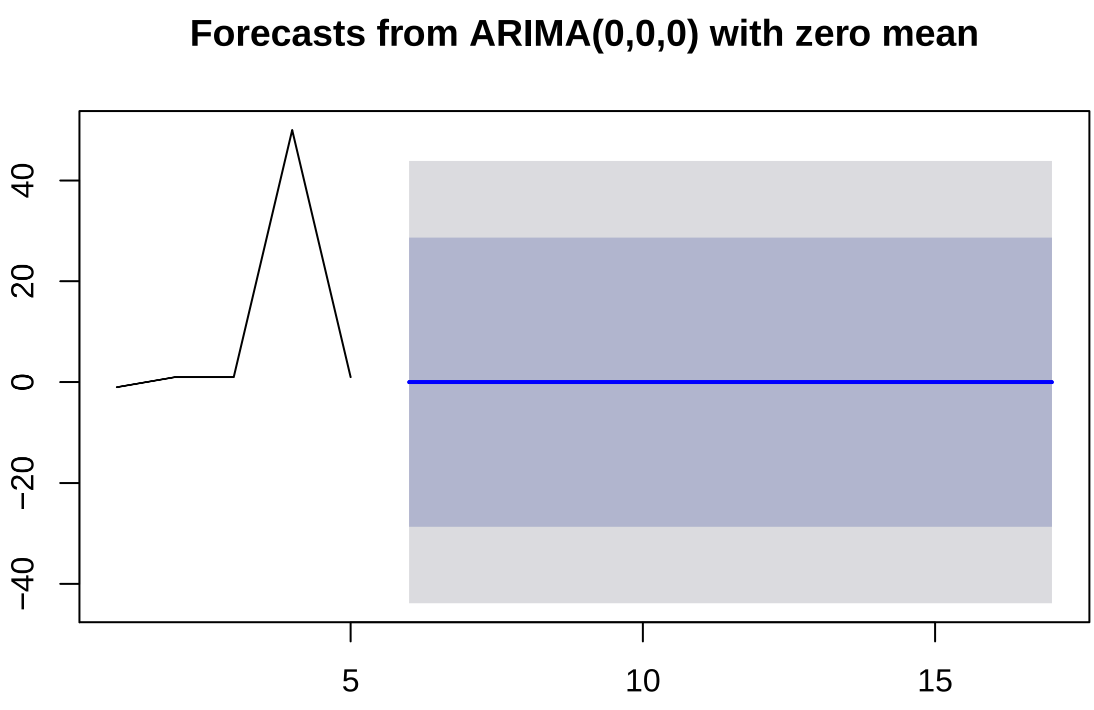

```
## Book ID:  86
```



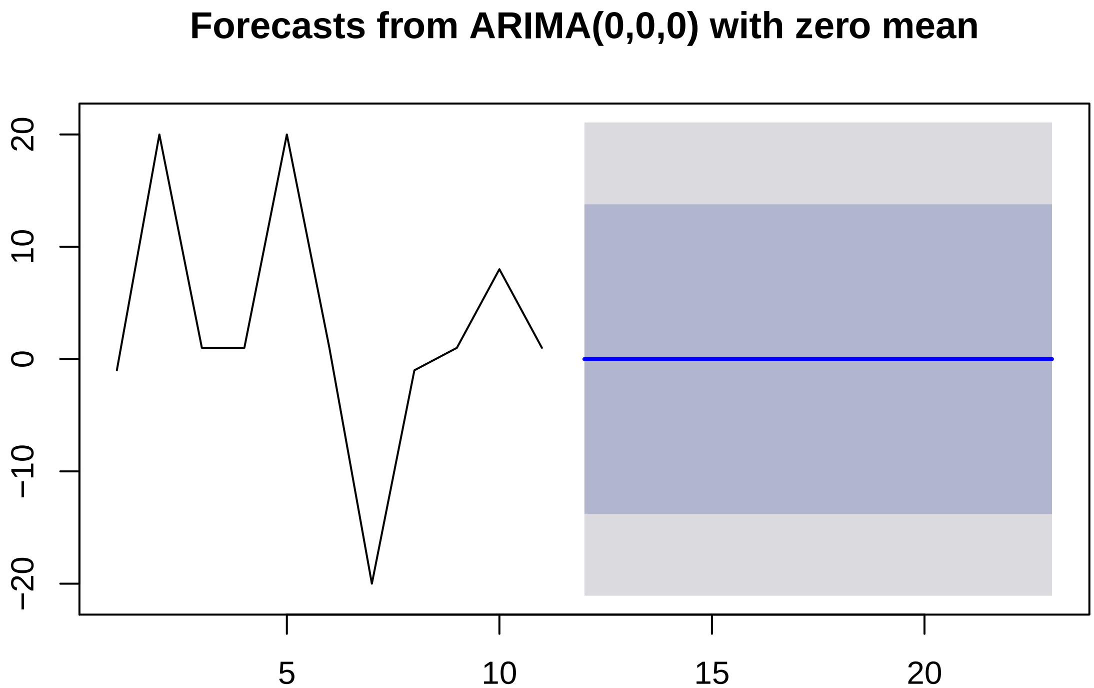

```
## Book ID:  90
```



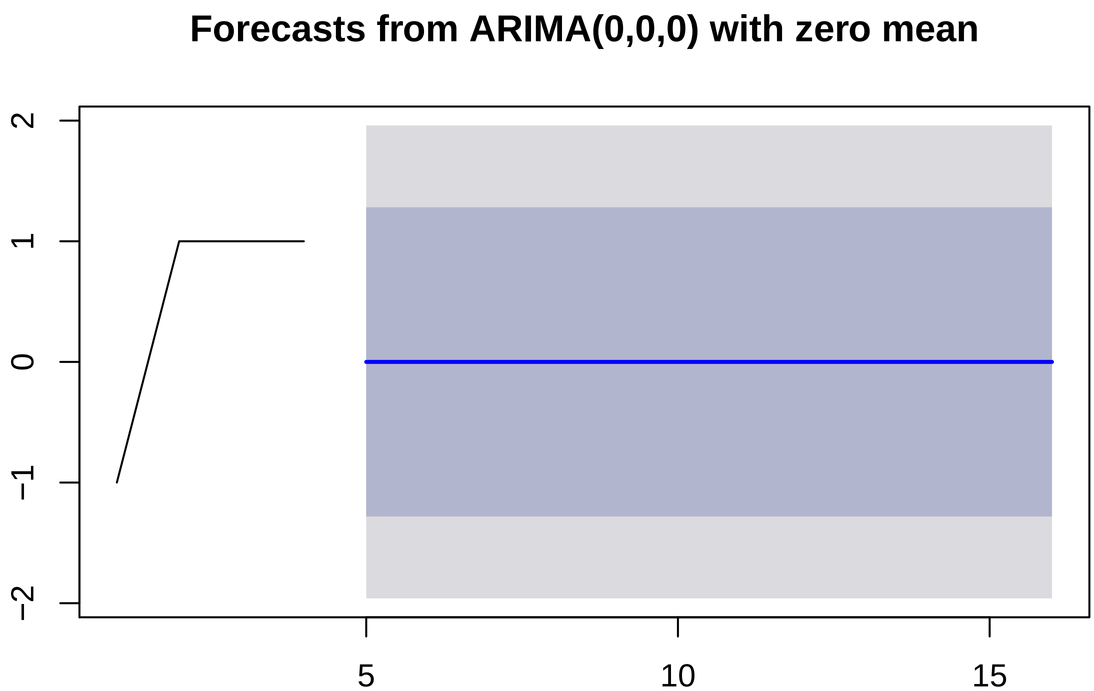

```
## Book ID:  91
```



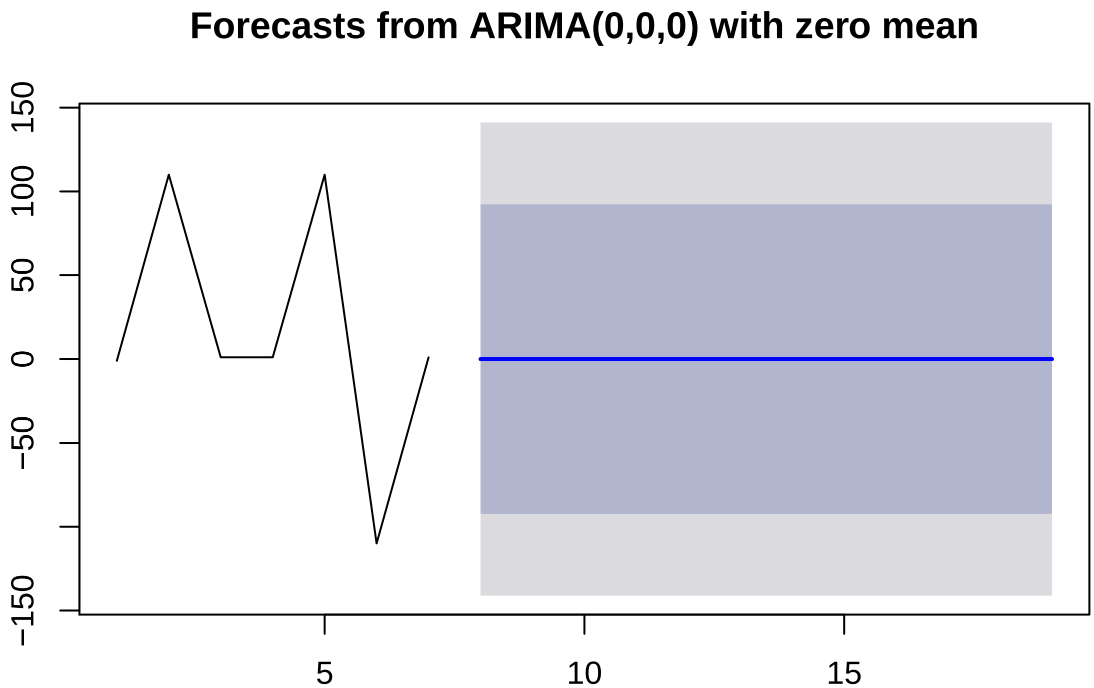

```
## Book ID:  92
```



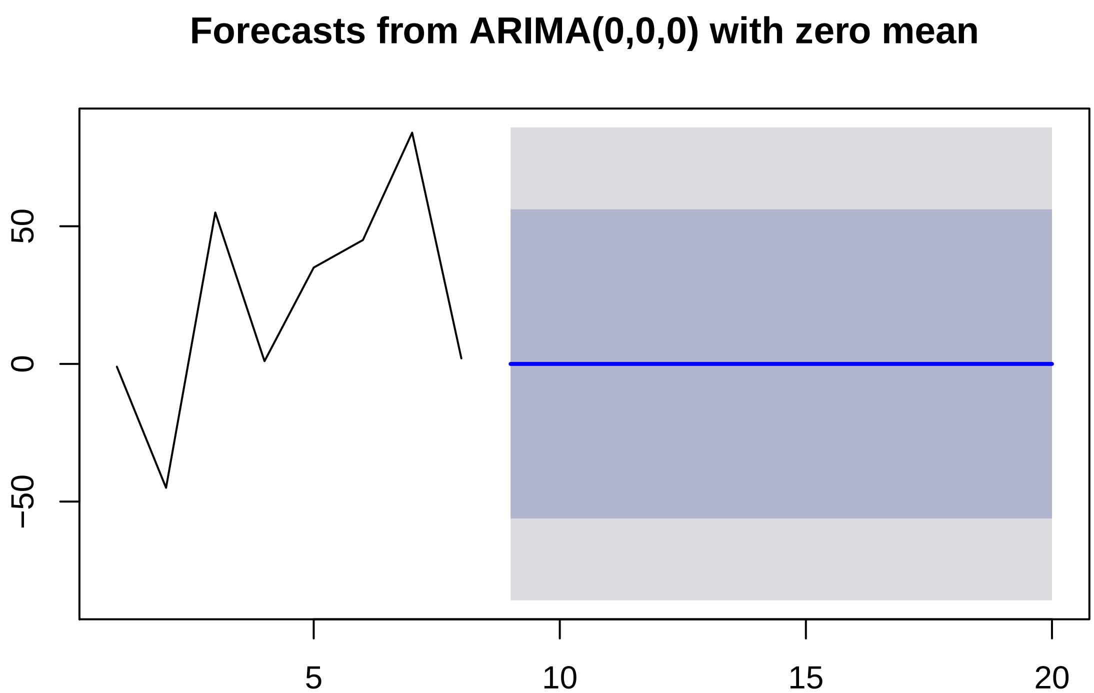

```
## Book ID:  93
```




```
## Book ID:  94
```

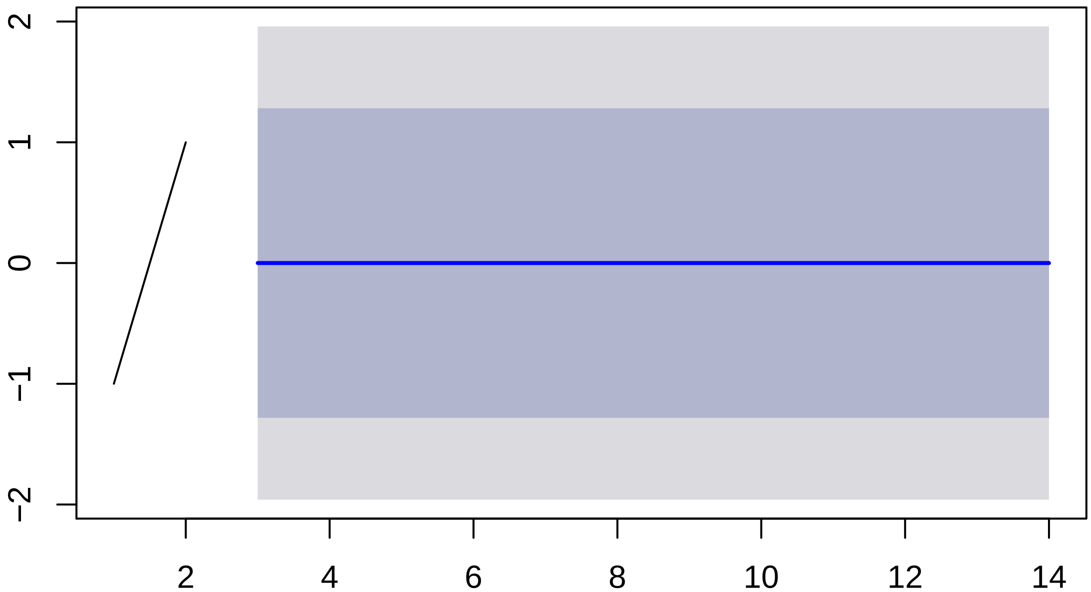



```
## Book ID:  95
```



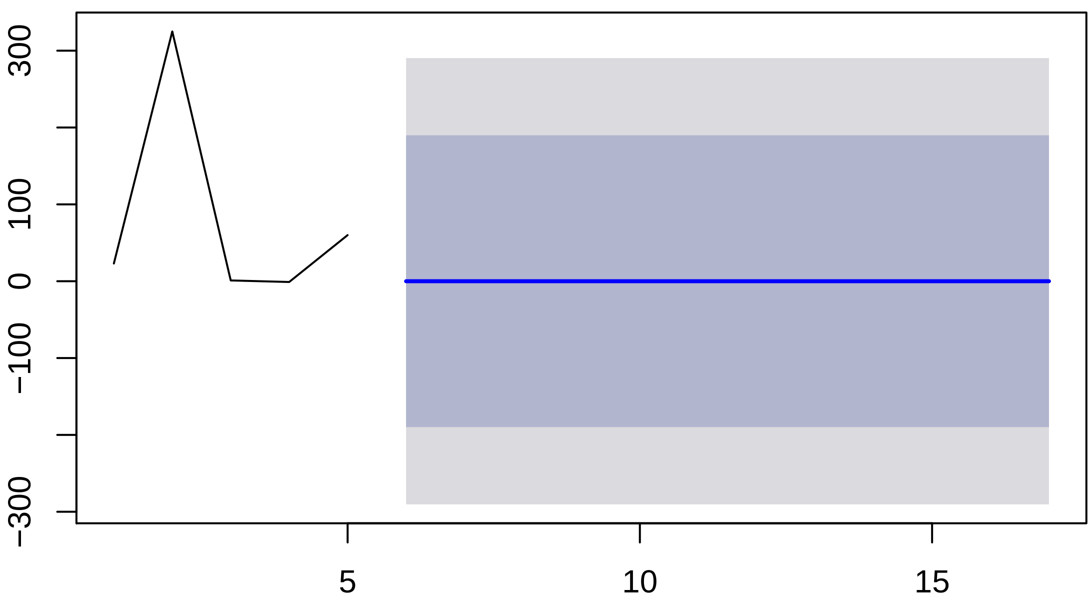

```
## Book ID:  96
```

Forecasts from ARIMA(0,0,0) with zero mean



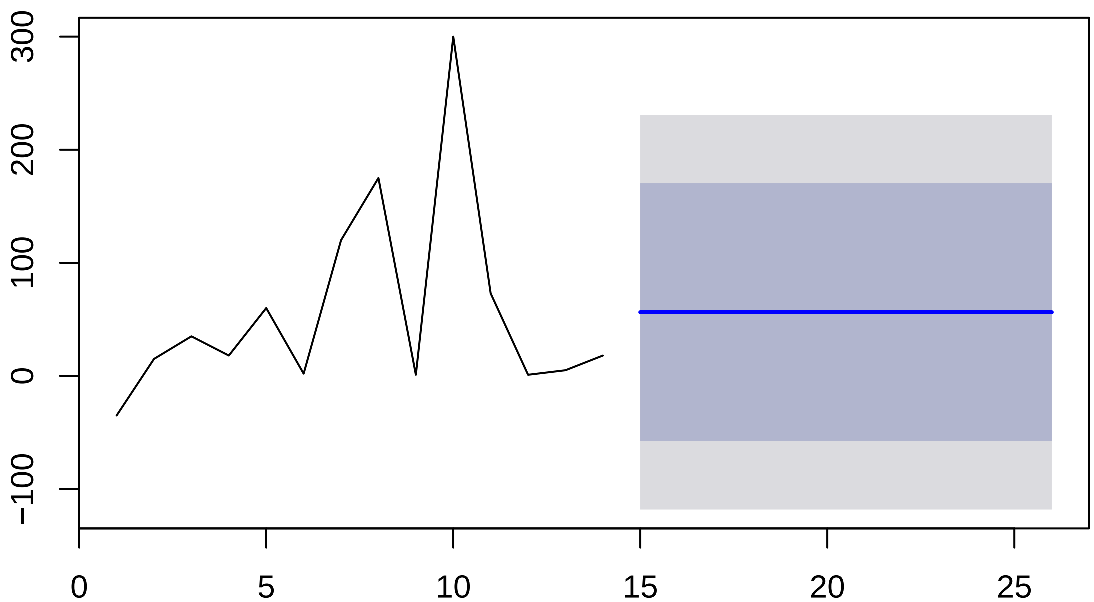

```
## Book ID:  97
```

Forecasts from ARIMA(0,0,0) with zero mean



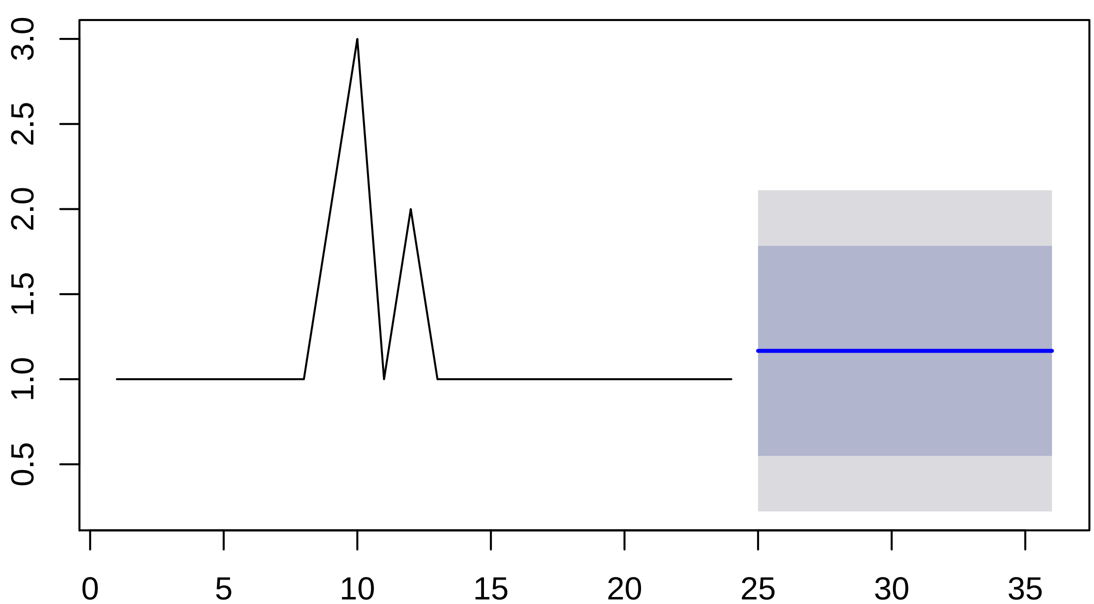

```
## Book ID:  103
```

Forecasts from ARIMA(0,0,0) with non−zero mean



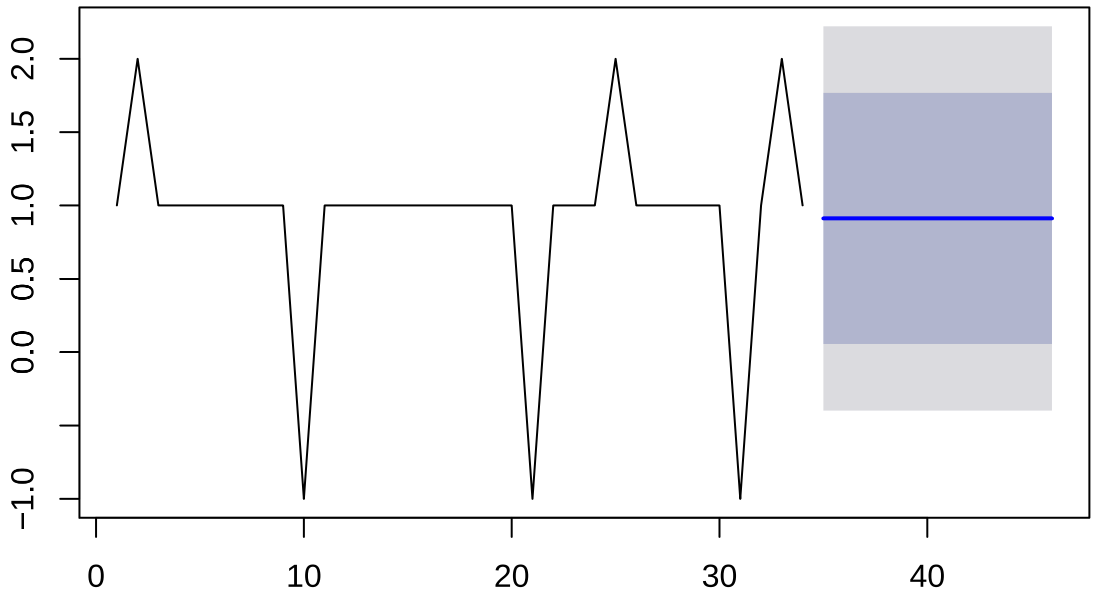

```
## Book ID:  109
```

Forecasts from ARIMA(0,0,0) with non−zero mean



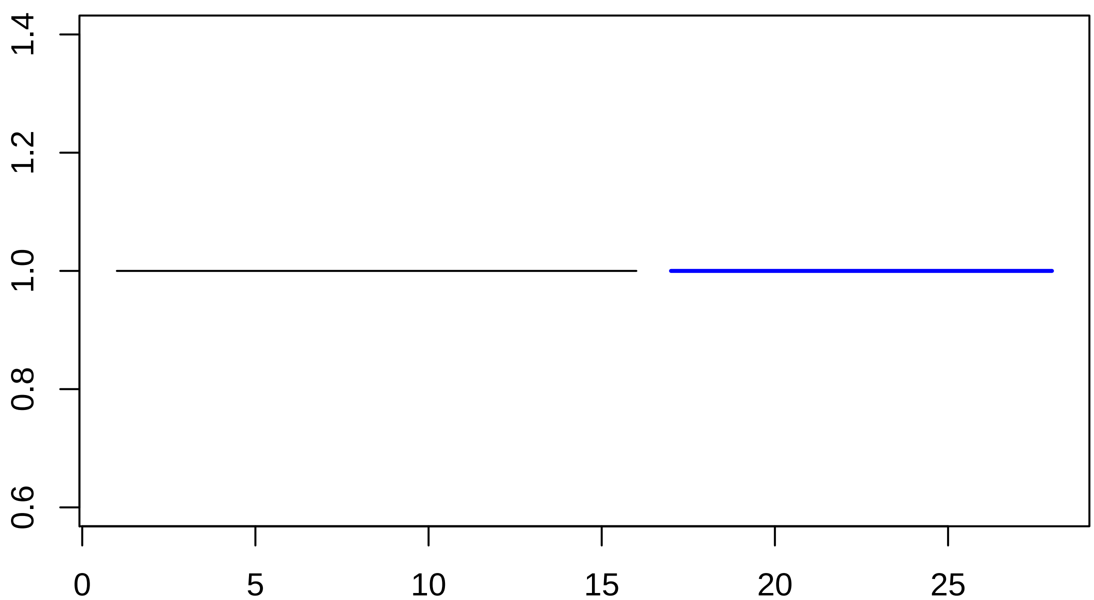

```
## Book ID:  110
```

Forecasts from ARIMA(0,0,0) with non−zero mean



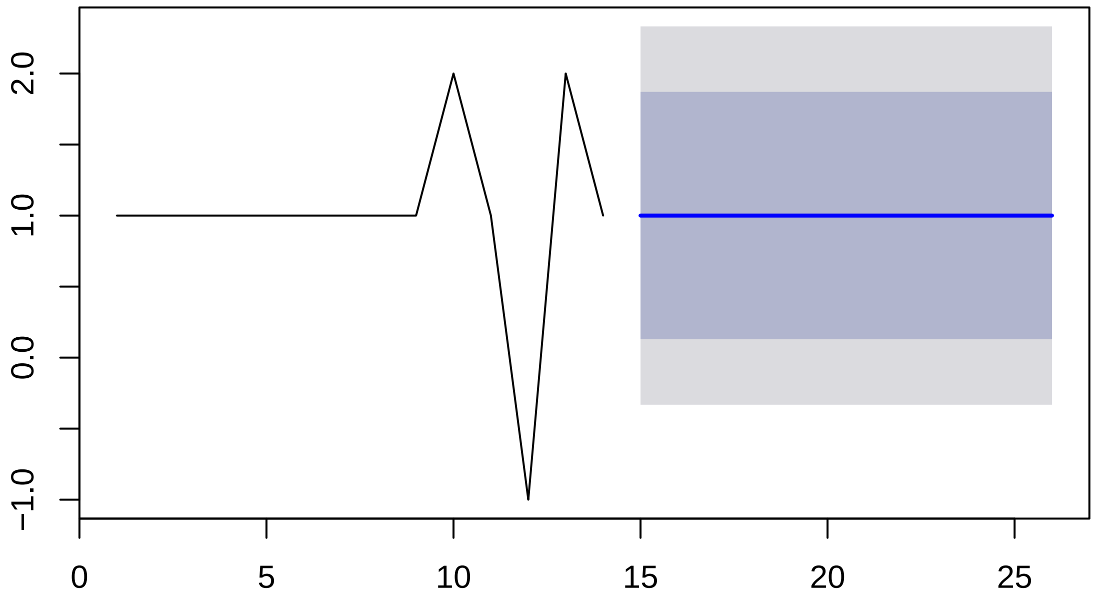

```
## Book ID:  128
```

Forecasts from ARIMA(0,0,0) with non−zero mean



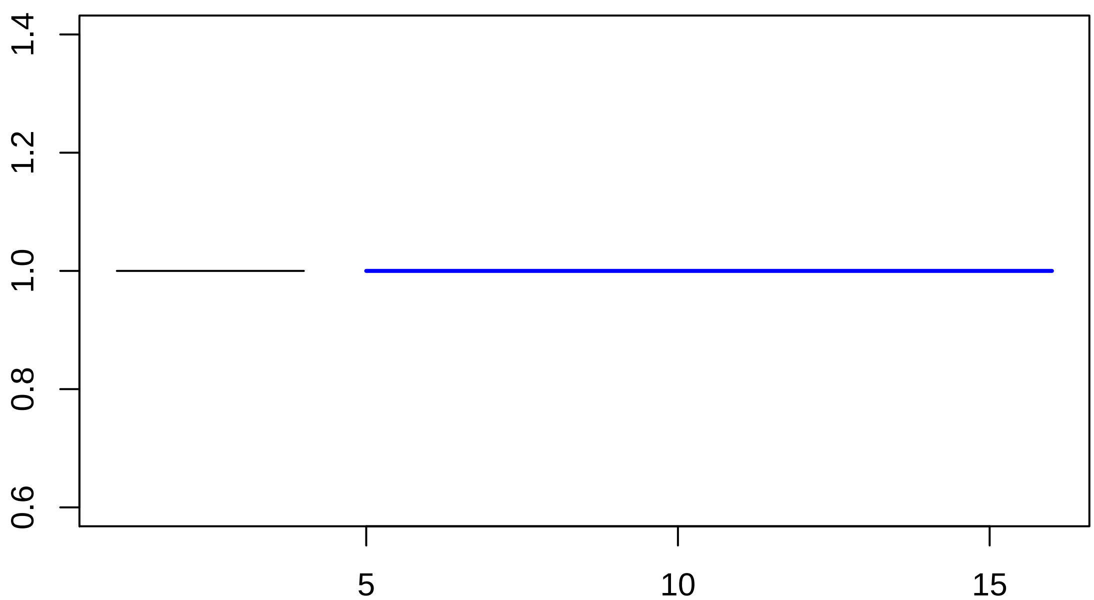

```
## Book ID:  129
```

Forecasts from ARIMA(0,0,0) with non−zero mean

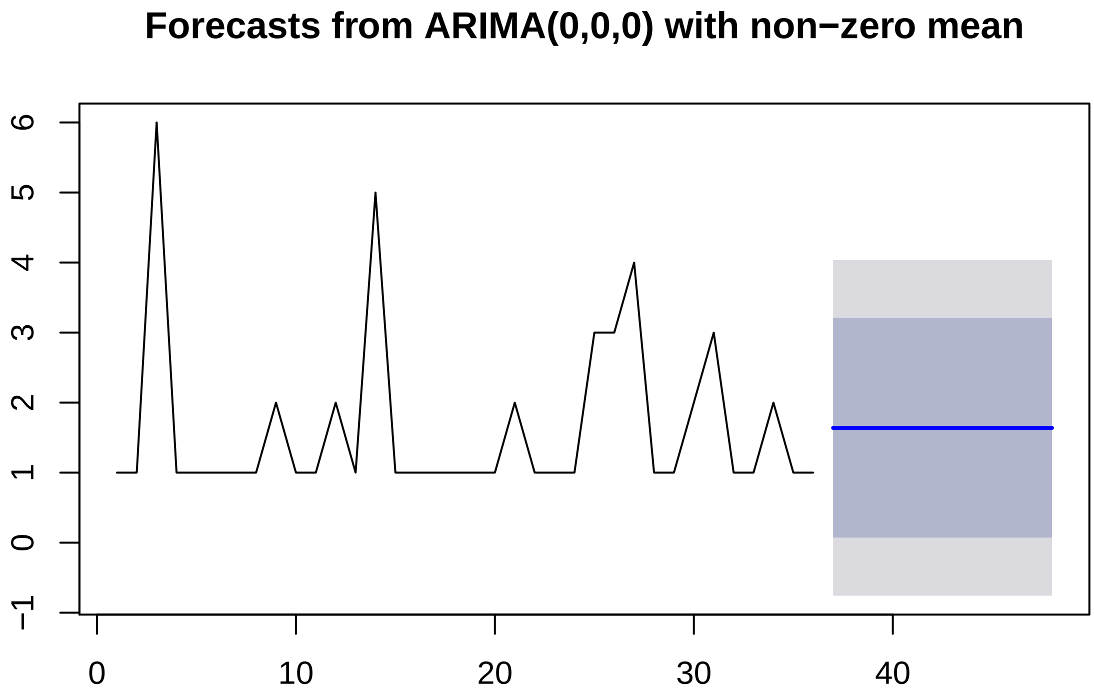



```
## Book ID:  132
```

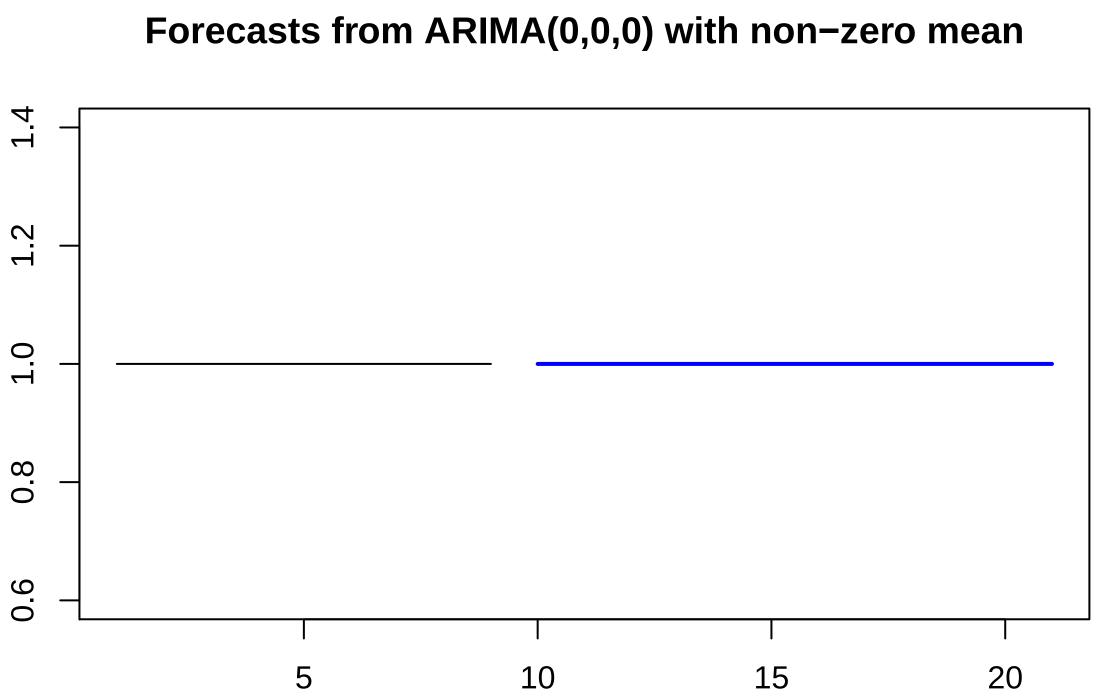
Forecasts from ARIMA(0,0,0) with non−zero mean



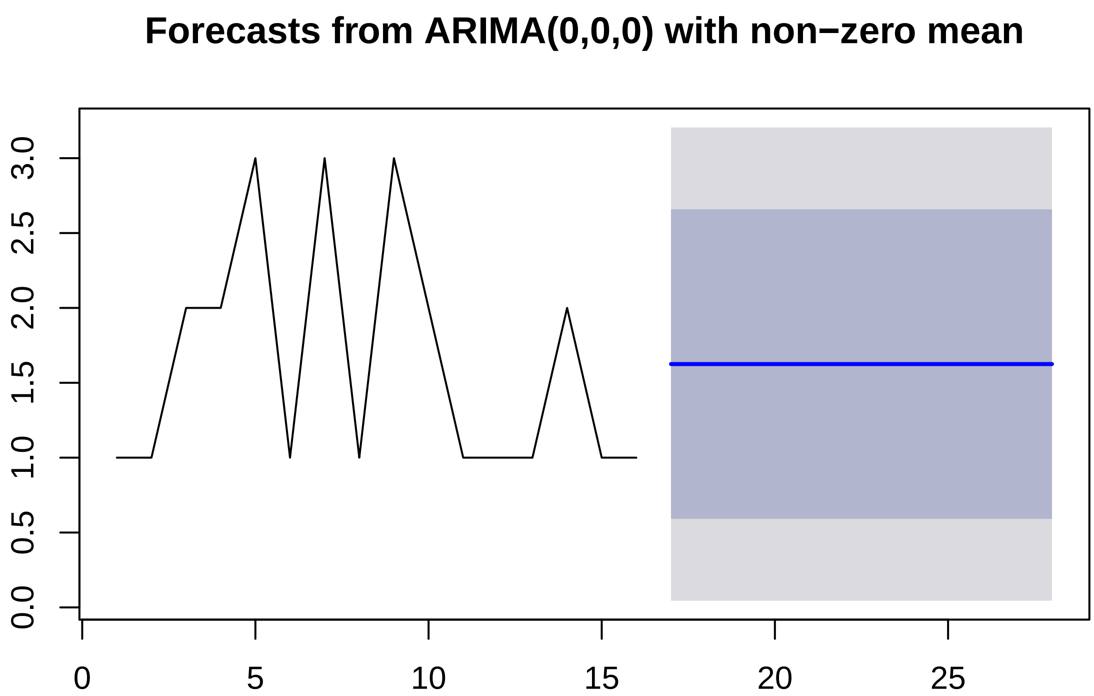

```
## Book ID:  165
```



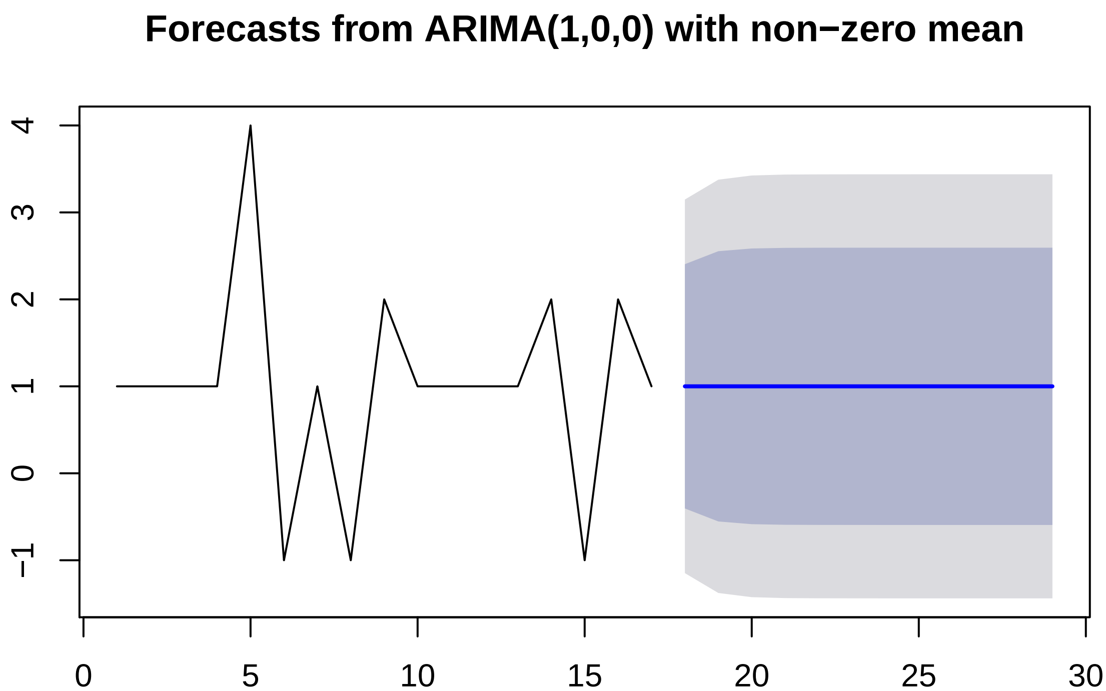
Book ID: 166



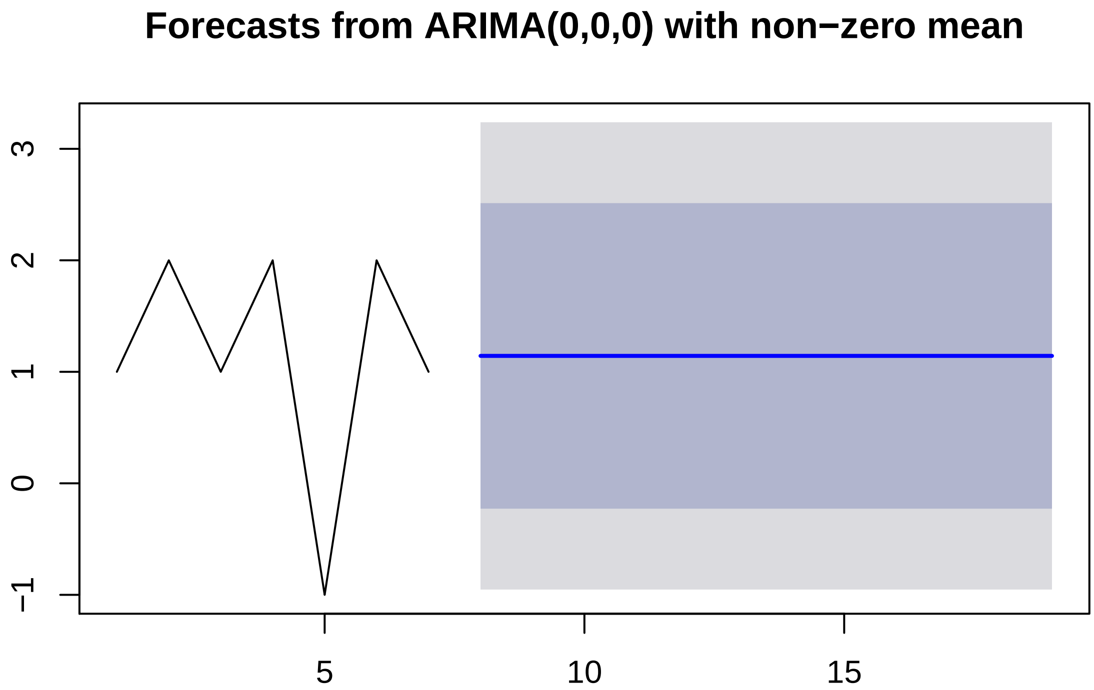
Book ID: 172



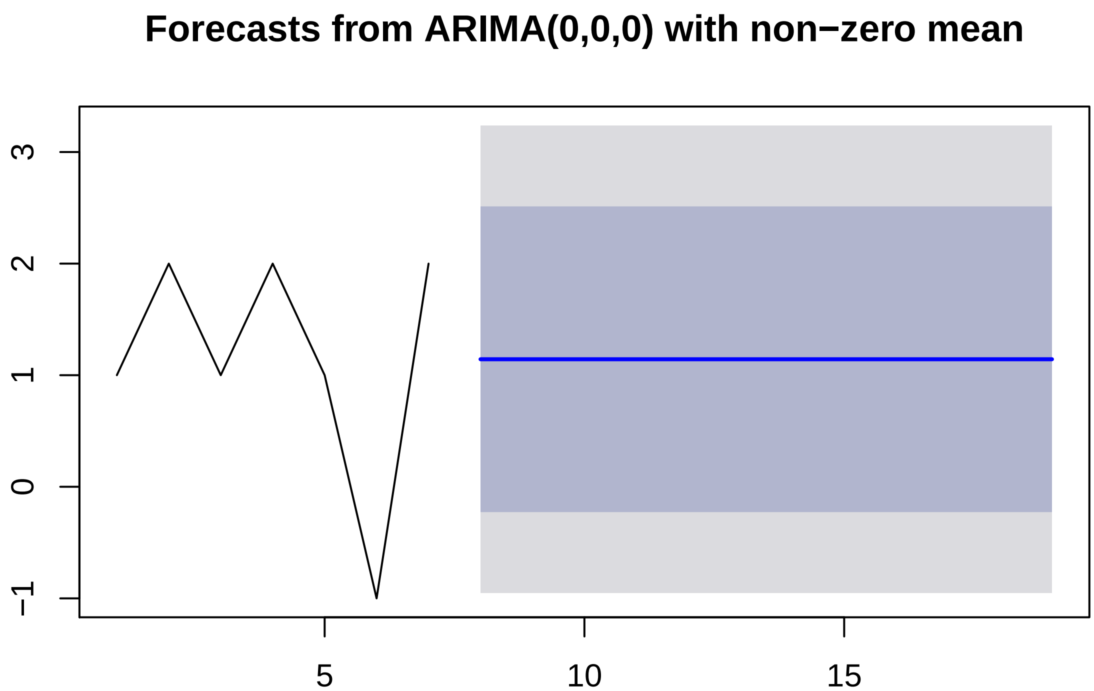

```
## Book ID:  195
```




```
## Book ID:  196
```

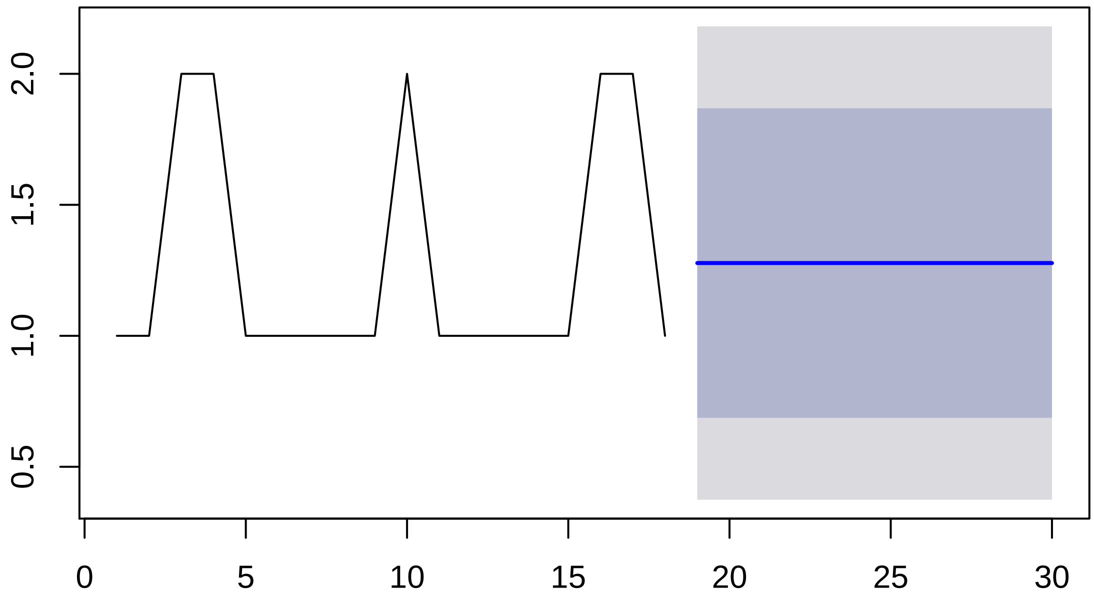



```
## Book ID:  197
```



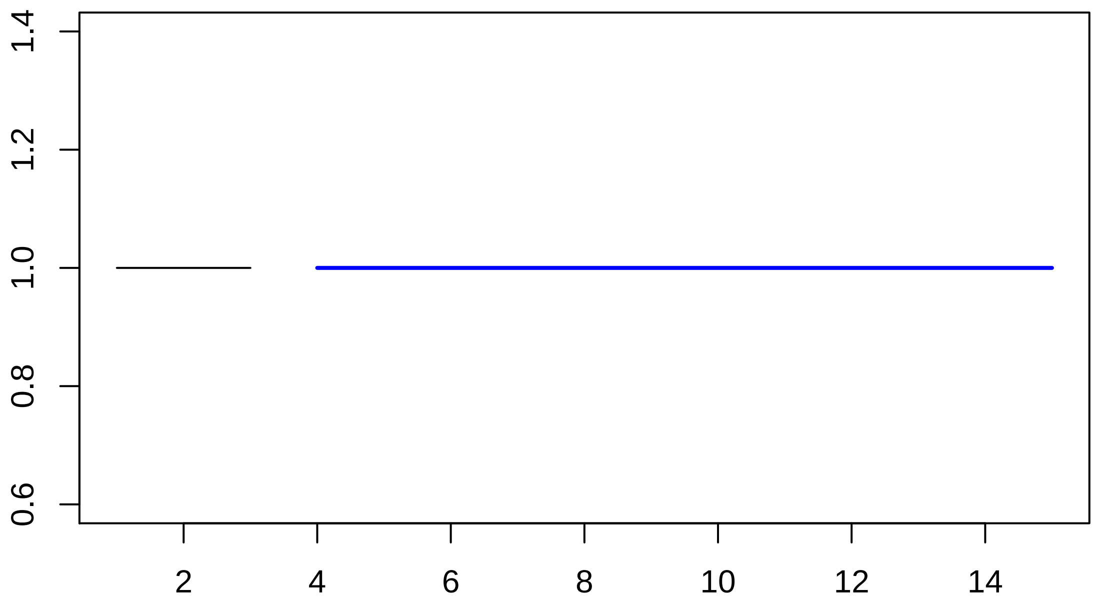

```
## Book ID:  239
```

Forecasts from ARIMA(0,0,0) with non−zero mean

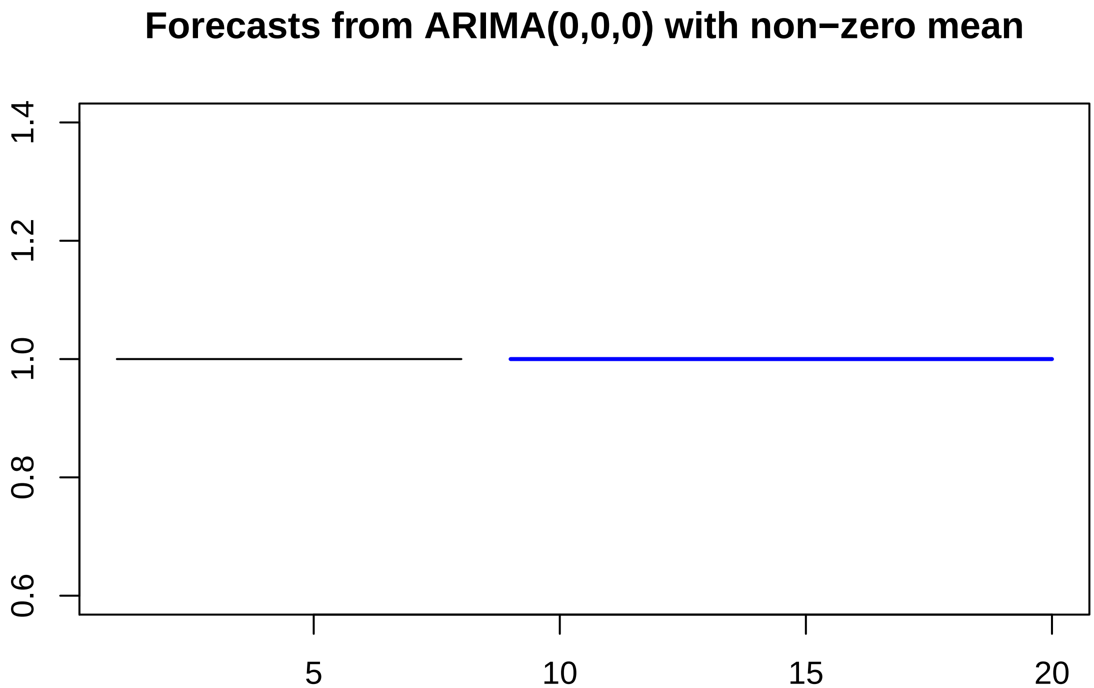



```
## Book ID:  243
```

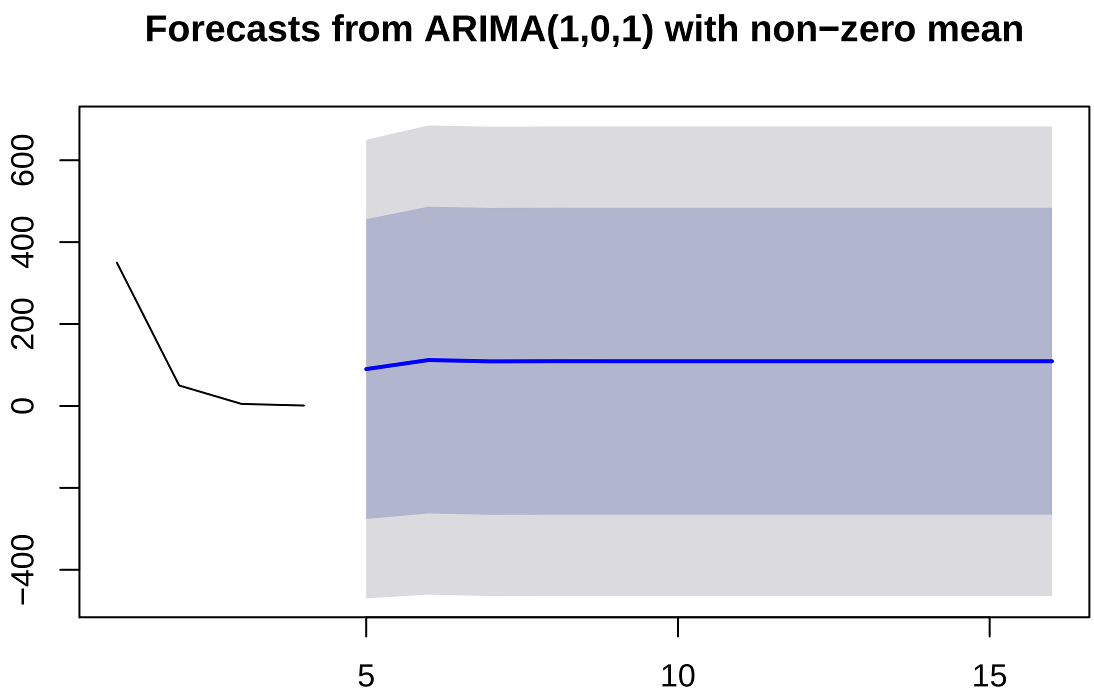
Forecasts from ARIMA(0,0,0) with non−zero mean




```
## Book ID:  244
```

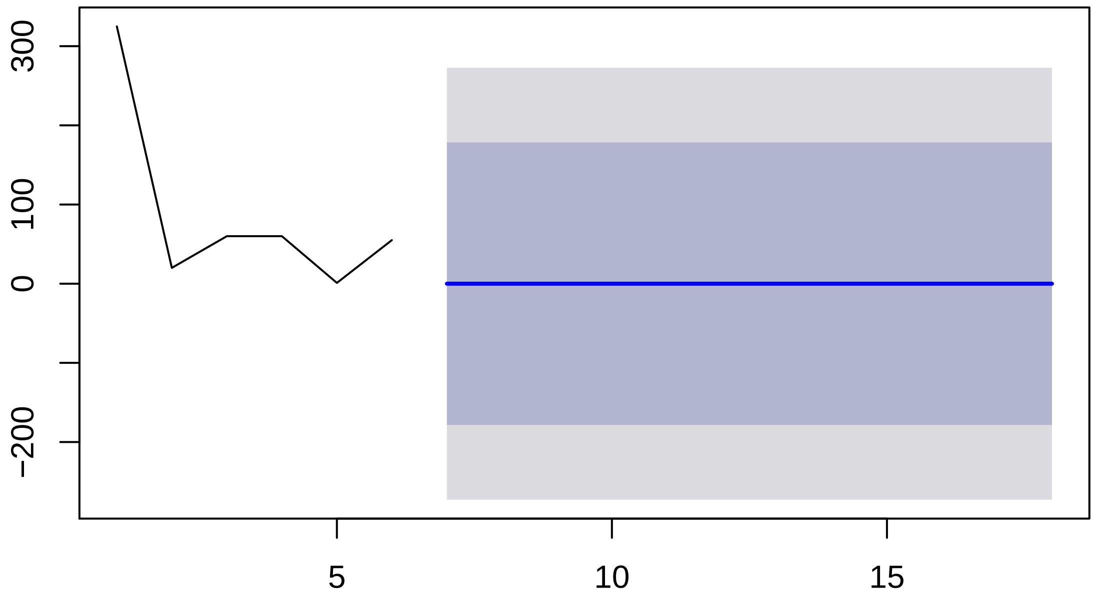



```
## Book ID:  324
```



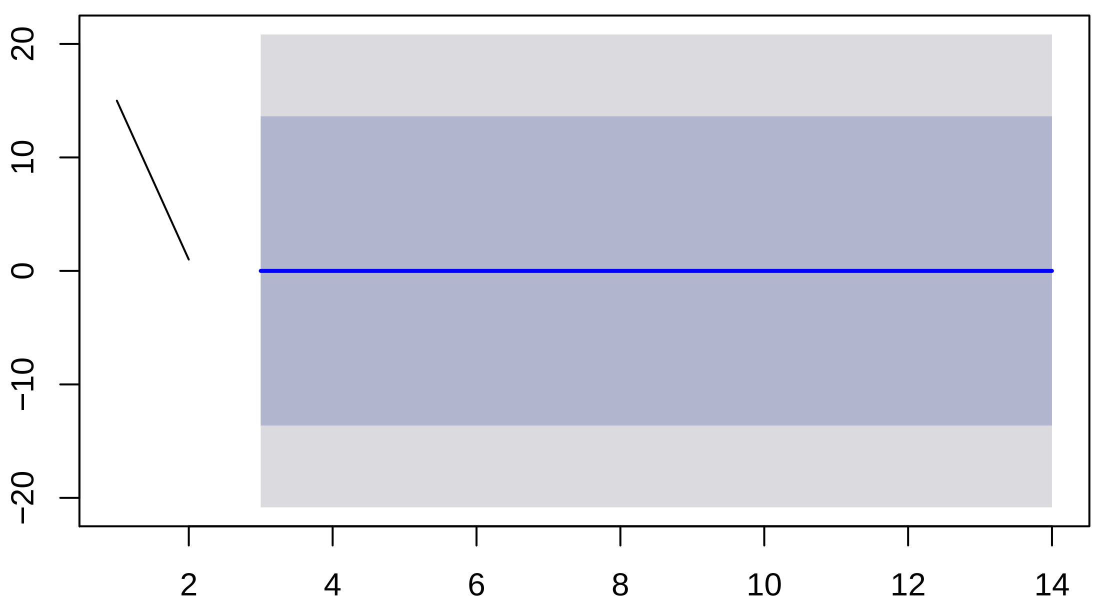

```
## Book ID:  325
```

Forecasts from ARIMA(0,0,0) with zero mean

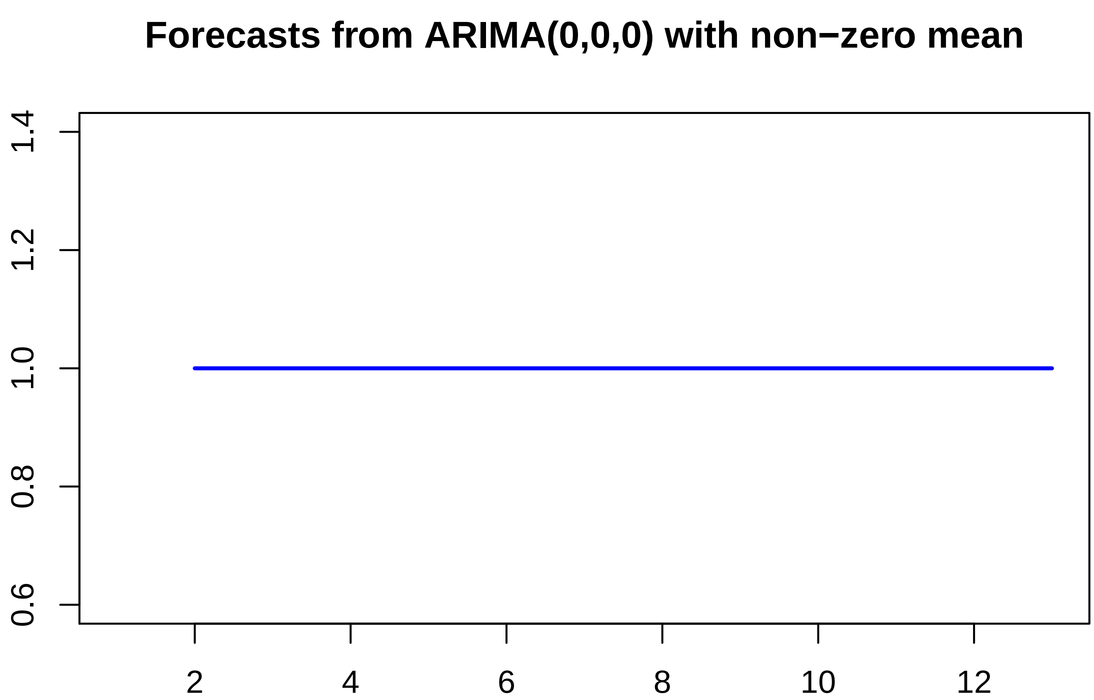



```
## Book ID:  364
```

Forecasts from ARIMA(0,0,0) with zero mean




```
## Book ID:  542
```




```
# Print SMAPE Score for all the book titles
for (i in isbn_list){
  ISBN <- final_dataset[final_dataset$ISBN == i,]
  cat(i)
  train_data_all <- ISBN[ISBN$Date.Ordered.Year < "2018",]
  test_data_all <- ISBN[ISBN$Date.Ordered.Year >= "2018",]
  #Arima model
  arima_all <- auto.arima(ISBN$Quantity.Ordered)
  forecast_all <- forecast(arima_all, h = 12)
  pred <- mean(as.vector(forecast_all$mean))
  test <- mean(test_data_all$Quantity.Ordered)
  smape <- (sum(abs(test-pred)/(abs(test)+abs(pred))))/length(test)
  smape
}
```

```
## 12345678910111415161718192021222324252627282930313233343536373839404142434954565758596061626386909191
```

Step 5: Determine Dependent Vs Independent Variables in Regression Modelling for one title

Price as independent variable

```
Quantity.model <- lm(as.formula(Quantity.Ordered~Order.Price), data = train_data_8)
Quantity.model
```

```
##
## Call:
## lm(formula = as.formula(Quantity.Ordered ~ Order.Price), data = train_data_8)
##
## Coefficients:
## (Intercept)  Order.Price
##     3.92857     -0.03571
```

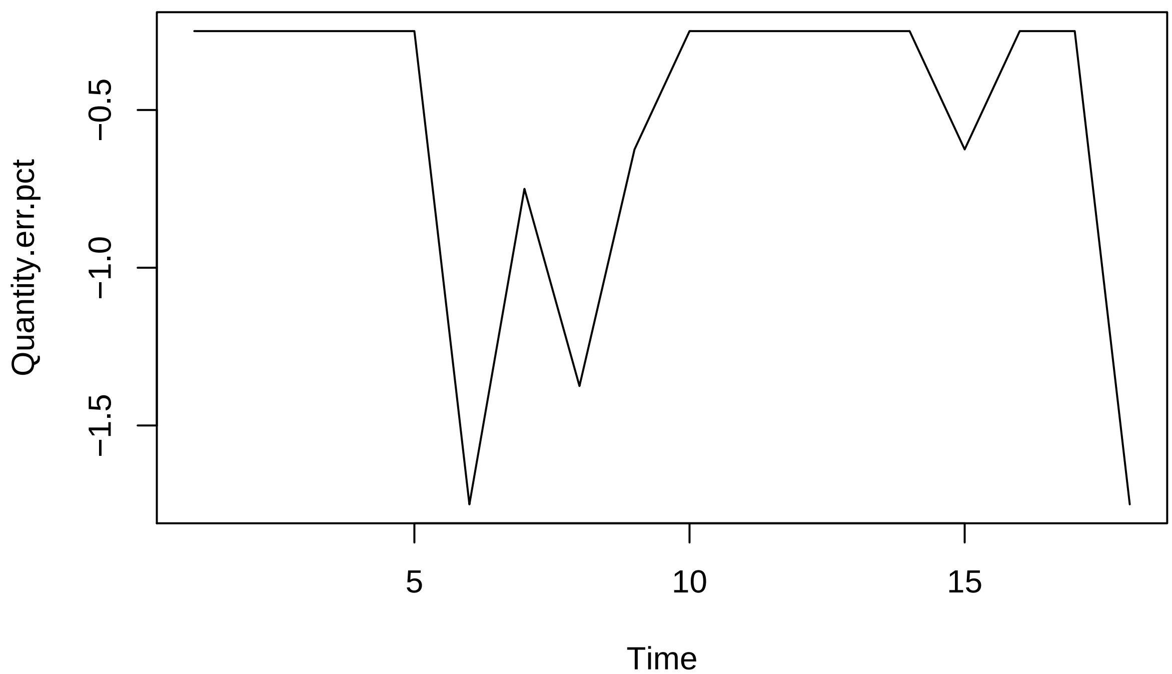
```r
options(warn=-1)
Quantity.model.predict <- predict.lm(Quantity.model, newdata = test_data_8)
options(warn=1)
Quantity.model.predict
```

```
##  481  487  492  493  495
## 0.75 0.75 0.75 0.75 0.75
```

```r
Quantity.actual <- train_data_8[,'Quantity.Ordered']
Quantity.err.pct <- (Quantity.model.predict-Quantity.actual)/Quantity.actual
```

```
## Warning in Quantity.model.predict - Quantity.actual: longer object length
## is not a multiple of shorter object length
```

```r
plot.ts(Quantity.err.pct)
```




```r
# SSE
Quantity.lm.SSE <- sum(sapply((Quantity.model.predict-Quantity.actual), function(z) z^2))
```

```
## Warning in Quantity.model.predict - Quantity.actual: longer object length
## is not a multiple of shorter object length
```

```r
Quantity.lm.SSE
```

```
## [1] 22.625
```

```r
# SSE as a percentage of mean of actual values
Quantity.lm.SSE/mean(unlist(Quantity.actual))
```

```
## [1] 27.15
```

Quantity as independent variable

```r
Price.model <- lm(as.formula(Order.Price~Quantity.Ordered), data = train_data_8)
Price.model
```

```
##
```

```
## Call:
## lm(formula = as.formula(Order.Price ~ Quantity.Ordered), data = train_data_8)
##
## Coefficients:
##      (Intercept)  Quantity.Ordered
##          86.9259           -0.3111
```

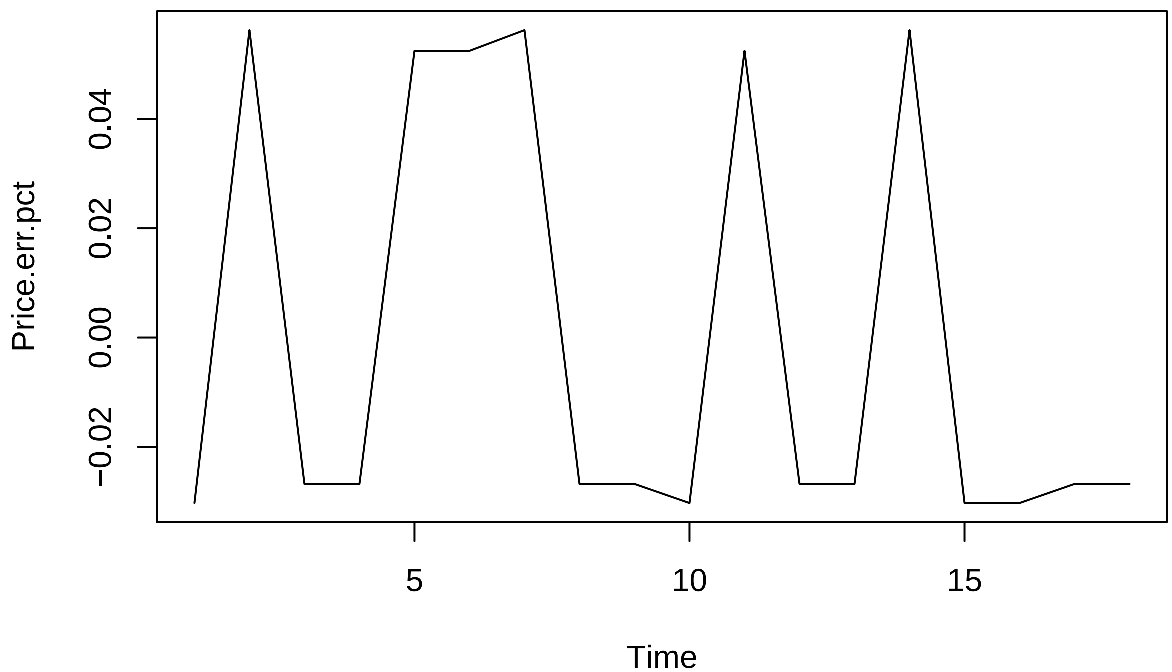
```r
options(warn=-1)
Price.model.predict <- predict.lm(Price.model, newdata = test_data_8)
options(warn=1)
Price.model.predict
```

```
##      481      487      492      493      495
## 86.30370 86.61481 86.61481 86.61481 86.30370
```

```r
Price.actual <- train_data_8[,'Order.Price']
Price.err.pct <- (Price.model.predict-Price.actual)/Price.actual
```

```
## Warning in Price.model.predict - Price.actual: longer object length is not
## a multiple of shorter object length
```

```r
plot.ts(Price.err.pct)
```




```r
# SSE
Price.lm.SSE <- sum(sapply((Price.model.predict-Price.actual), function(z) z^2))
```

```
## Warning in Price.model.predict - Price.actual: longer object length is not
## a multiple of shorter object length
```

```r
Price.lm.SSE
```

```
## [1] 194.0481
```

```r
# SSE as a percentage of mean of actual values
Price.lm.SSE/mean(unlist(Price.actual))
```

```
## [1] 2.239016
```

When comparing the 2 models using the sum of squared error as a percenatge of the mean of the actual values, the SSE when Price as the dependent variable is less when compared to Quantity being the dependent variable. However, in terms of business interests, we would like to forecast the quantity and hence I believe Quantity should be the dependent variable and Price the independent variable in case of Regression Time Series Problems.