Demo for Insight Interview: Predict Time Series with ARIMA & HMM

Chaoyi (Mina) Zheng

30 July, 2019

Load Data

This data represents monthly number of major cancer surgery in New York State in a ten year period from 1997 to 2006. There are 120 data pionts in total.

```
total <- read.csv('C:/Users/zcyhi/Downloads/data.csv')
total <- total %>%
  select(month, total)
kable(head(total), row.names=FALSE) %>% kable_styling(bootstrap_options = "striped", ful
l_width = F)
```

month	total
1	1309
2	1057
3	1166
4	1196
5	1206
6	1171

```
kable(tail(total), row.names=FALSE) %>% kable_styling(bootstrap_options = "striped", ful
l_width = F)
```

month	total
115	1338
116	1482
117	1396

month	total
118	1415
119	1450
120	1260

Recode Key Variables

This step prepares variables for subsequent modeling.

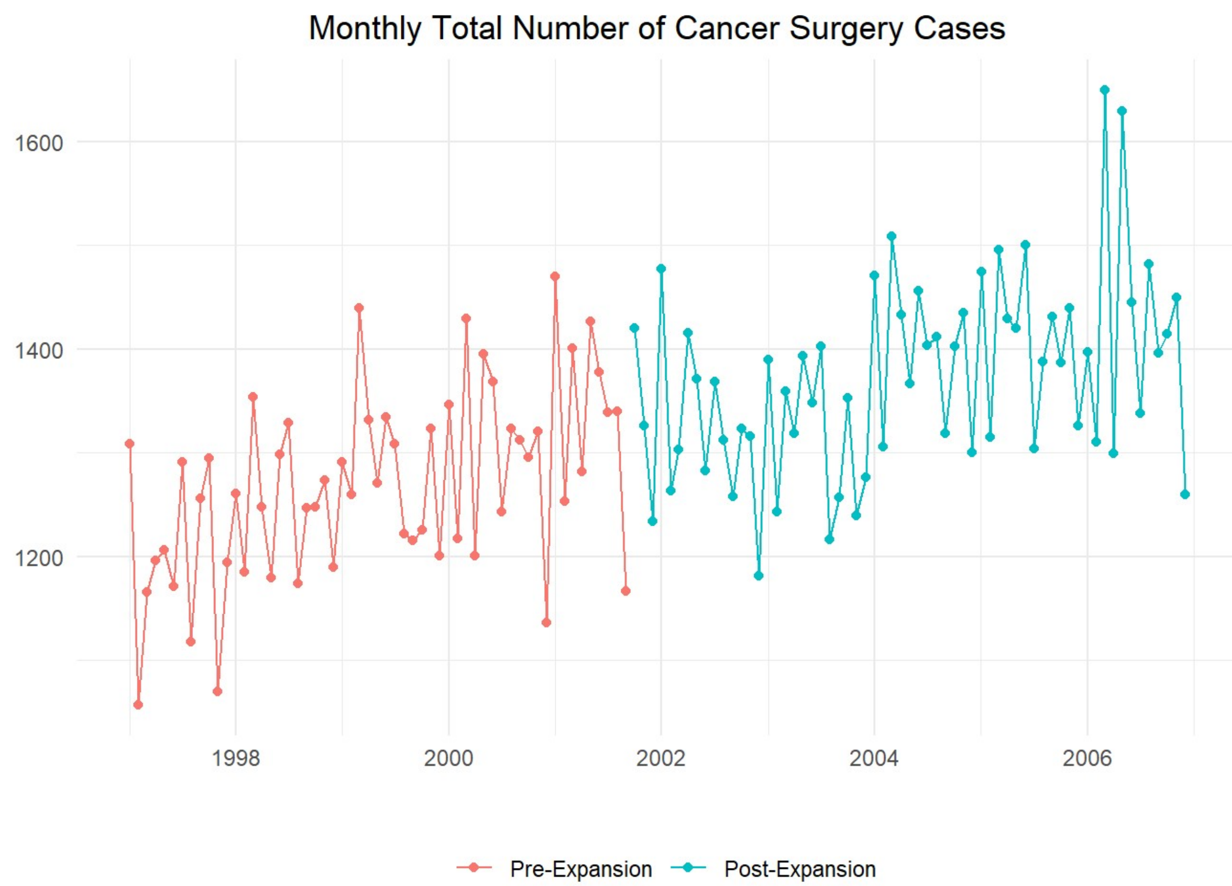
```
total <- slide(data = total, Var = 'total', NewVar = 'lag1', slideBy = -1)
total <- slide(data = total, Var = 'total', NewVar = 'lag2', slideBy = -2)
#kable(head(total), row.names=FALSE)  %>% kable_styling(bootstrap_options = "striped", f
ull_width = F)

total <- total %>%
  mutate(post = as.numeric(month > 57),
         monthpost = if_else(post == 1, month - 57, 0),
         monthdummy = month %% 12,
         date = ymd("1996-12-01") %m+% months(month)) %>%
  arrange(month)

cos <- cos(2 * pi * total$month/12)
sin <- sin(2 * pi * total$month/12)

total <- cbind.data.frame(total, cos, sin)
kable(total[54:63,], row.names=FALSE)  %>% kable_styling(bootstrap_options = "striped",
full_width = F)
```

month	total	lag1	lag2	post	monthpost	monthdummy	date	cos	sin
54	1378	1427	1282	0	0	6	2001-06-01	-1.0000000	0.0000000
55	1339	1378	1427	0	0	7	2001-07-01	-0.8660254	-0.5000000
56	1340	1339	1378	0	0	8	2001-08-01	-0.5000000	-0.8660254
57	1167	1340	1339	0	0	9	2001-09-01	0.0000000	-1.0000000
58	1420	1167	1340	1	1	10	2001-10-01	0.5000000	-0.8660254

month	total	lag1	lag2	post	monthpost	monthdummy	date	cos	sin
59	1326	1420	1167	1	2	11	2001-11-01	0.8660254	-0.5000000
60	1234	1326	1420	1	3	0	2001-12-01	1.0000000	0.0000000
61	1477	1234	1326	1	4	1	2002-01-01	0.8660254	0.5000000
62	1263	1477	1234	1	5	2	2002-02-01	0.5000000	0.8660254
63	1303	1263	1477	1	6	3	2002-03-01	0.0000000	1.0000000

Plot Data

```
ggplot(total, aes(x = date, y = total, colour = as.factor(post))) +
  geom_point() +
  geom_path() +
  labs(y = '', x = '') +
  scale_colour_discrete(name = "", labels = c("Pre-Expansion", "Post-Expansion")) +
  theme_minimal() +
  theme(legend.position = 'bottom', plot.title = element_text(hjust = 0.5)) +
  ggtitle("Monthly Total Number of Cancer Surgery Cases")
```



Segmented Linear Regression (OLS)

The standard approach is to regress the outcome measure (total) on time (month), indicator for intervention (post), and post-intervention time (postmonth). As results from this section shows, the residuals from this model are still autocorrelated, indicating this model is not adequate.

We build models using the first 9 years of data, saving the last 1 year of data for testing. Prediction accuracy is measured by sum of squared errors (SSE).

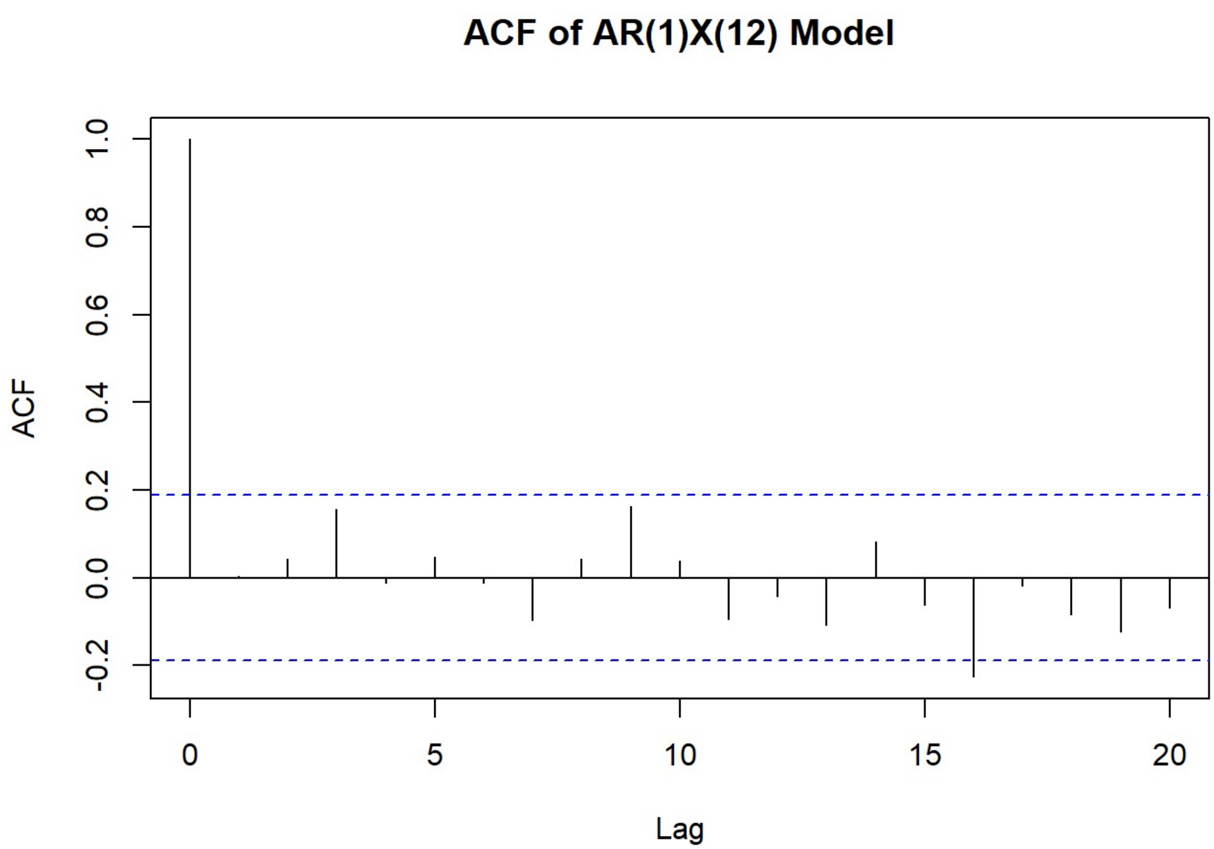
ARIMA Model with 1st Order and Seasonal Autoregression

```
ar <- arima(total[1:108,2],
            order = c(1,0,0),
            seasonal = list(order = c(1,0,0), period = 12),
            xreg = total[1:108,c(1,5,6)])
kable(tidy(coeftest(ar))) %>% kable_styling(bootstrap_options = "striped", full_width =
F)
```

term	estimate	std.error	statistic	p.value

term	estimate	std.error	statistic	p.value
ar1	-0.1990809	0.0995724	-1.9993575	0.0455697
sar1	0.4882089	0.0899229	5.4291950	0.0000001
intercept	1185.9924203	19.2646882	61.5630220	0.0000000
month	2.7088977	0.5231431	5.1781200	0.0000002
post	-23.3213241	20.7340594	-1.1247833	0.2606809
monthpost	-0.8489078	0.8682474	-0.9777258	0.3282100

```
ar.pred <- predict(ar, 12, newxreg = total[109:120, c(1,5,6)])$pred
ar.sse <- sum((ar.pred-total[109:120, 2])^2)

(acf(residuals(ar), main = 'ACF of AR(1)X(12) Model'))
```



```
##
## Autocorrelations of series 'residuals(ar)', by lag
##
##      0      1      2      3      4      5      6      7      8      9
##  1.000  0.004  0.043  0.158 -0.012  0.048 -0.010 -0.096  0.043  0.163
##     10     11     12     13     14     15     16     17     18     19
##  0.038 -0.095 -0.042 -0.107  0.082 -0.061 -0.226 -0.018 -0.083 -0.123
##     20
## -0.068
```

```
(ar.qtest <- c(
  Box.test(residuals(ar), type="Ljung-Box", lag=6)$p.value,
  Box.test(residuals(ar), type="Ljung-Box", lag=12)$p.value,
  Box.test(residuals(ar), type="Ljung-Box", lag=18)$p.value,
  Box.test(residuals(ar), type="Ljung-Box", lag=24)$p.value
))
```

```
## [1] 0.7685199 0.6777519 0.3578611 0.2017350
```

HMM with 1st-Order Autoregression and Fourier Seasonal Terms

This section fits a hidden Markov model (HMM) with 3 states for the latent Markov chain and normally distributed source distributions. This model also accounts for first-order autocorrelation and seasonal pattern.

```
set.seed(29)

# FIT MODEL

mod.lag <- depmix(
  total ~ month + monthpost + post + lag1 + cos + sin,
  data=total[3:108,], ns=3, family=gaussian())

(fm.lag <- fit(mod.lag,
               verbose=F))
```

```
## converged at iteration 74 with logLik: -539.8393
```

```
## Convergence info: Log likelihood converged to within tol. (relative change)
## 'log Lik.' -539.8393 (df=32)
## AIC:  1143.679
## BIC:  1228.909
```

```
# OBTAIN PARAMETER ESTIMATES

## TRANSITION MATRIX
kable(trans <- t(matrix(getpars(fm.lag)[4:12], c(3,3))))  %>% kable_styling(bootstrap_op
tions = "striped", full_width = F)
```

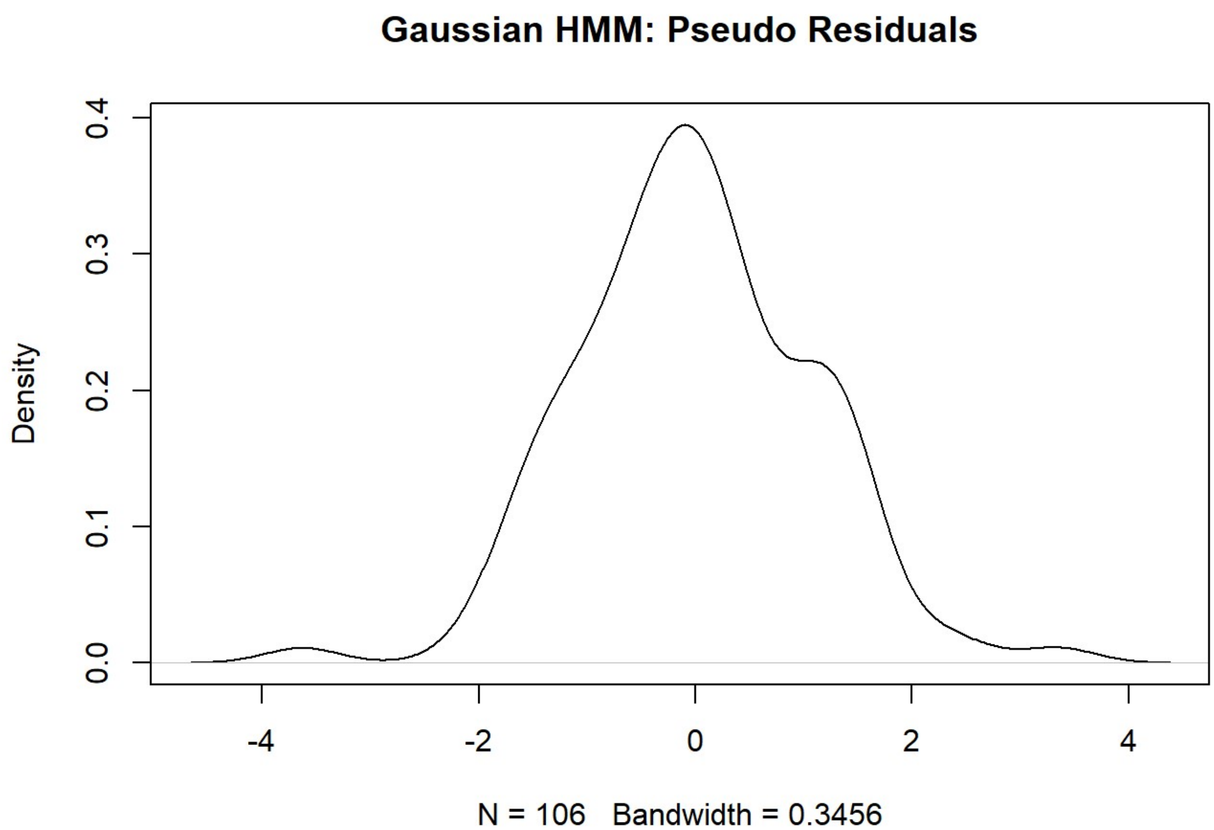
0.2364537	0.6764410	0.0871053
0.6263355	0.0000004	0.3736641
0.7170669	0.1913419	0.0915912

```
## OTHER PARAMETERS
kable(pars <- cbind(getpars(fm.lag)[13:19],getpars(fm.lag)[21:27],getpars(fm.lag)[29:3
5]))  %>% kable_styling(bootstrap_options = "striped", full_width = F)
```

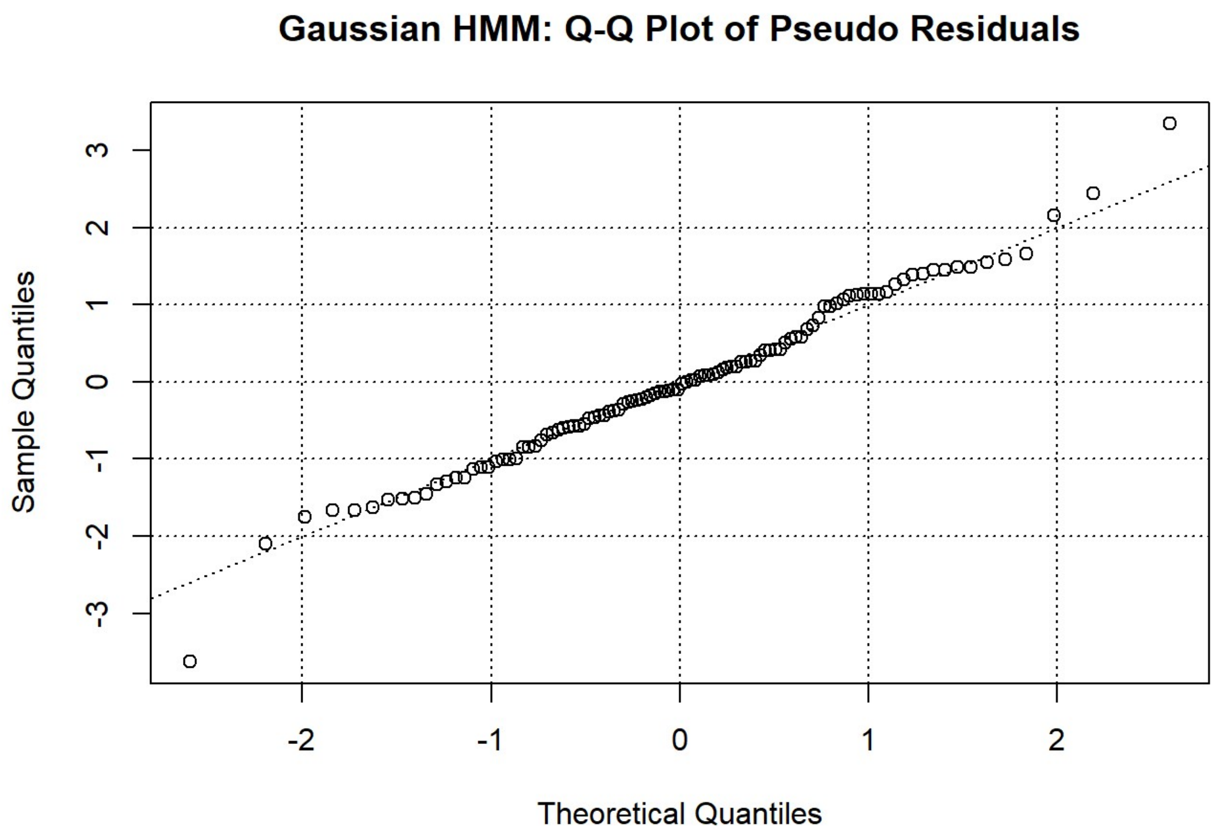
(Intercept)	2313.0692815	1292.4902549	974.4524602
month	6.1474732	-1.4065804	4.1815895
monthpost	-0.9921967	3.6738858	-1.1263824
post	-119.9130717	40.0750409	-55.4206261
lag1	-0.9702878	0.0200643	0.1241805
cos	-16.3172782	-88.1384443	14.2784040
sin	29.3766873	29.0132211	48.0836843

```r
## INITIAL AND ENDING DISTRIBUTIONS
init<-t(matrix(getpars(fm.lag)[1:3], c(3,1)))
end.distr <- as.matrix(posterior(fm.lag)[106,2:4])

## PREDICTED VALUES FOR YEARS 1997 TO 2005
base <- cbind.data.frame(
  rep(1,106),
  select(total[3:108,], month, monthpost, post, lag1, cos, sin))
base <- as.matrix(base) %*% pars

pred<-NULL
for (i in 1 : 106){
  pred<-c(pred, as.matrix(attributes(fm.lag)$posterior[i,2:4]) %*% base[i,])
}

## SSE FOR FORECAST VALUES FOR YEAR 2006
forecast.base <- cbind.data.frame(
  rep(1,12),
  select(total[109:120,], month, monthpost, post, lag1, cos, sin))

forecast.distr <- matrix(0, nrow = 12, ncol = 3)
for (i in (1:12)){
  forecast.distr[i,] <- end.distr %*% (trans %^% i)
}

forecast.m <- as.matrix(forecast.base) %*% as.matrix(pars)
hmm.forecast <- NULL
for (i in (1:12)){
  hmm.forecast <- c(hmm.forecast, forecast.m[i,] %*% t(t(forecast.distr[i,])))
}

hmm.sse <- sum((hmm.forecast-total$total[109:120])^2)

## EXAMINE DISTRIBUTION OF PSEUDO-RESIDUALS

design <- cbind.data.frame(
  rep(1,106),
  select(total[3:108,], month, monthpost, post, lag1, cos, sin))

HidMarkov <- mmglm1(
  total[3:108, 2],
  Pi = trans,
  delta = c(1,0,0),
  glmfamily = gaussian(link = "identity"),
  beta = pars,
  Xdesign = as.matrix(design),
  sigma = c(getpars(fm.lag)[c(20,28,36)]))

res <- residuals(HidMarkov)
```

```
plot(density(res), main="Gaussian HMM: Pseudo Residuals")
box()
```

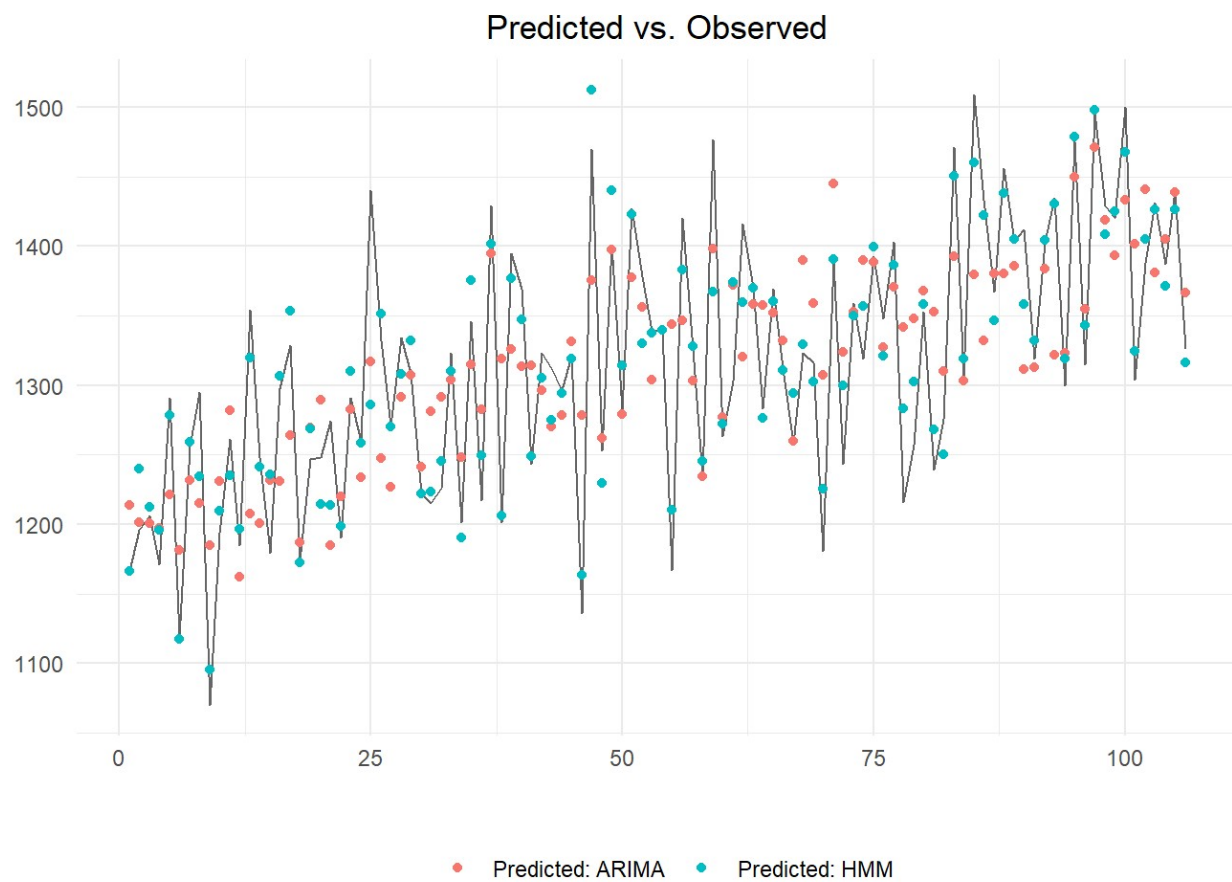



```
qqnorm(res, main="Gaussian HMM: Q-Q Plot of Pseudo Residuals")
abline(a=0, b=1, lty=3)
abline(h=seq(-2, 2, 1), lty=3)
abline(v=seq(-2, 2, 1), lty=3)
```

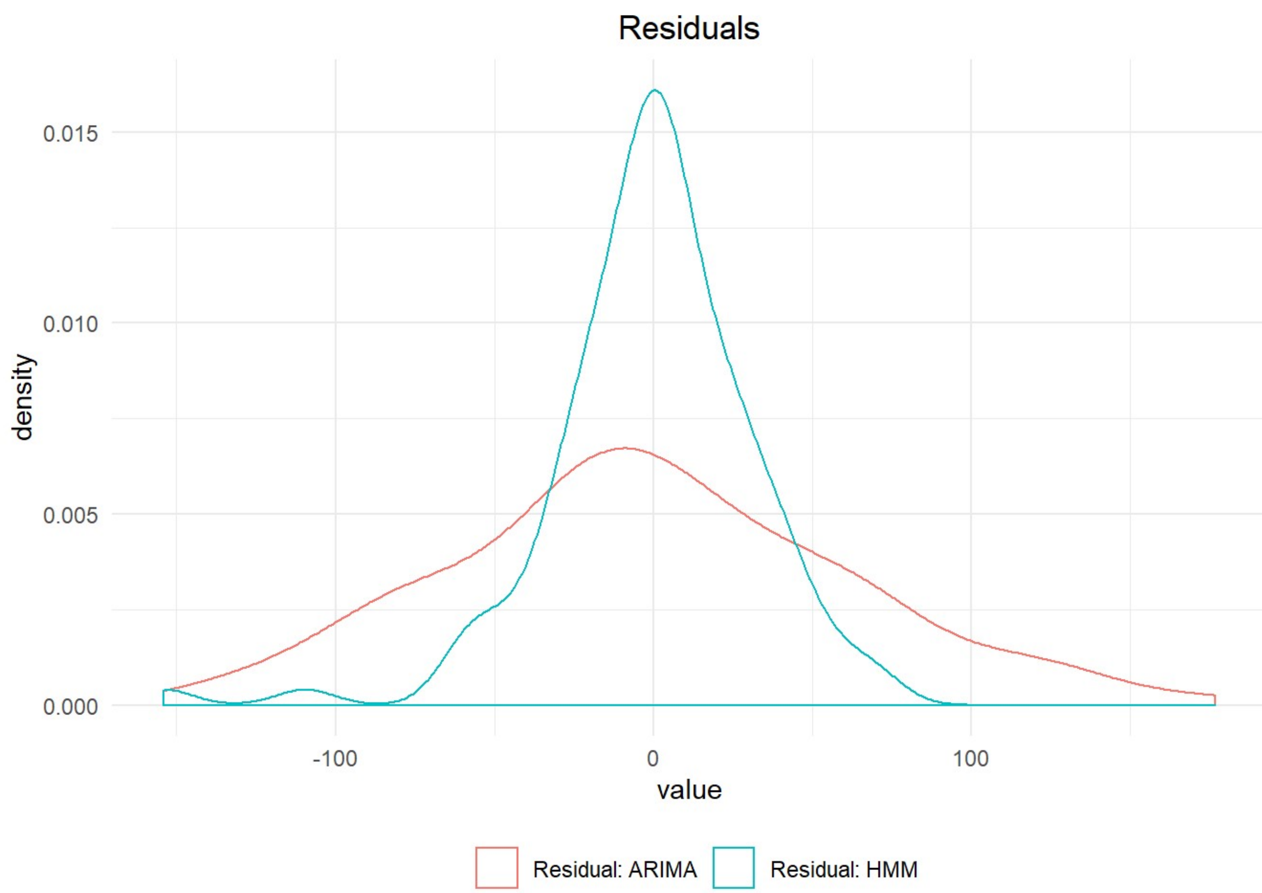

Gaussian HMM: Q-Q Plot of Pseudo Residuals
Sample Quantiles
Theoretical Quantiles
-3
-2
-1
0
1
2
3
-2
-1
0
1
2

Summary: Compare Predictions and Forecast between ARIMA and HMM

```
pred.table = cbind.data.frame(
  1:106,
  total[3:108, 2],
  total[3:108, 2] - residuals(ar)[3:108],
  pred
)
colnames(pred.table) <- c('Time', 'Observed', 'Predicted: ARIMA', 'Predicted: HMM')
pred.table <- pred.table %>%
  mutate(`Residual: ARIMA` = `Predicted: ARIMA` - Observed,
         `Residual: HMM` = `Predicted: HMM` - Observed)

pred.values <- melt(pred.table[, 1:4], id.vars = c('Time', 'Observed'))
res.values <- melt(pred.table[, c(1,2,5,6)], id.vars = c('Time', 'Observed'))
(ggplot(data=pred.values) +
  geom_line(aes(x = Time, y = Observed), alpha = 0.6) +
  geom_point(aes(x = Time, y = value, colour = variable))) +
  theme_minimal() +
  theme(legend.position = 'bottom', plot.title = element_text(hjust = 0.5)) +
  labs(y = '', x = '') +
  scale_colour_discrete(name = "") +
  ggtitle("Predicted vs. Observed")
```




```
(ggplot(res.values,
        aes(x = value, colour = variable)) +
  geom_density(alpha = 0.5) +
  theme_minimal() +
  theme(legend.position = 'bottom', plot.title = element_text(hjust = 0.5)) +
  scale_colour_discrete(name = "") +
  ggtitle("Residuals"))
```




```
summary.table = cbind.data.frame(
  c('OLS', 'ARIMA', 'HMM'),
  c(lm.sse, ar.sse, hmm.sse),
  c(AIC(lm), AIC(ar), NA))
colnames(summary.table) = c('Model', 'Forecast SSE', 'AIC')
kable(summary.table) %>% kable_styling(bootstrap_options = "striped", full_width = F)
```

Model	Forecast SSE	AIC
OLS	167141.5	1252.629
ARIMA	125742.6	1220.973
HMM	107104.0	NA