MATH 545 Assignment 1 - 260677676

Dan Yunheum Seol

2018-09-27

## 0.

We take it for granted that: 1. A sequence of random variables will have their second moments as covariances given that . 2. The linearity of expected value function .

## 1

Let be and let be finite-valued constants. Which of the following processes are stationary? For each stationary process, show that it is stationary and specify the mean and autocovariance functions. For each non-stationary process, show why it is non-stationary.

1. First we check the mean. We are given Since is a linear map, we obtain

Now we will check the autocovariance function . We know

We will start with variances, in other words, the cases where .

Due to independence of ,

When , we would need to scrutinize some cases. (WLOG) for

through re-indexing.

if

if (if would follow similarly)

thus since we have the mean and autocovariance function

not dependent on t, the sequence is stationary We get autocorrelation function

# (b)

We will look at the mean:

The mean does not depend on t. Shall we look at the autocovariance function?

but if

In none of the cases it depends on t, thus it is stationary.

# (c)

The mean does not depend on t:

We can find the sequence is not stationary from its autocovariance function when

So it depends on t, so it is non-stationary. except for the case where for some . if

# (d)

Conjecture: so it should be stationary.

Let us check the mean and covariances, though;

We would have the mean of , which does not depend on t.

For covariances:

It does not depend on t for arbitrary (therefore any) h. Thus the sequence is stationary.

# (e)

The mean would be

It does not depend on t.

Let us check the covariances now: for

We have two cases: If for some , then either or thus the covariance function is zero. Thus it becomes stationary. Otherwise it depends on t. Thus the sequence is not stationary.

# (f)

The mean would be

So it does not depend on t.

It would have the variance

due to independence,

For other cases, if ,

Otherwise,

Thus the sequence is stationary. ##2

Let be iid noise and define:

1. Show that is , but not noise.

Since the mean is zero, the variance will just be the second moments. Remember if then with mean 1 and variance 2. so

(WLOG we assume ; if we reindex) we have already visited the cases where if

A. if h is even, we would have two cases 1. t is even; it follows that t+h is also even.

and is an iid standard normal noise. Due to independence

1. t is odd; it follows that t+h is also odd.
2. and due to independence of

where , thus

B. If h is odd. If t is even, then t+h will be odd, and if t is odd then t+h will be even. In other words, in this case autocovariance function would have the form of where t is even and s is odd ( WLOG ).

if , and are independent. Thus

We would have two subcases: where $ h > 1 $ and . If $ h = 1 $ and $ s < t$ then and are independent, so we get the same result as above.

Otherwise ( and ), we have . So

We use the moment-generating function

for the standard normal distribution. To get the third moment, we differentiate it three times and set t to 0.

This we obtain

Thus

But for all even t and are not independent since . . So it is not an independent sequence.

# b)

Find for n odd and n even and compare the results. ##i) If n is odd. By construction $X\_{n+1} = Z\_{n+1} $, in other words we pick a new random sample from standard normal distribution. Because is an iid noise, is independent with and thus also independent with since all is either a new sample from iid noise or a function of only previous samples. It follows that for all possible cases, thus we would have

## ii) If n is even

Now let be a realization of By construction . Thus once is given

becomes a constant. I.e. we obtain

which is not necessarily zero. This could have been used as an another way to show that is not an iid sequence.

##3. Define be set of all polynomials up to degree k. We use the fact that it is a vector space spanned by

Let

. Show that

. and

We prove by induction on p. Let

which is a polynomial of zero degree.

Assume it holds for with degree p for some . For :

By binomial theorem,

And since

and is a vector space, it follows that

Showing that it is a polynomial of degree at most p. Thus through recursive process we get

## 4.

Let be a stationary process with mean zero and with arbitrary covariance function with .

# a)

if

where is the seasonal component with period 12 (i.e. ) . Show that

is stationary and express its autocovariance function in terms of .

Since , we have

will have mean zero,

suggesting that the covariances become the second moment of the terms in our sequence.

Now we will look at covariances (which is, in fact, the autocovariance functions)

This shows that the covariance does not depend on t, since none of depends on t and therefore their linear combination will also not depend on t. Thus is stationary.

# b)

Suppose where is just as the same as defined above with period 12 (i.e. ).

Show that

is stationary and express its autocovariance function in terms of

We recognize the fact that provided that they exist. Thus,

. Now, we can see that has mean zero

And following that we can show its autocovariance function, being the second moment :

Showing that the autocovariance function of is also not dependent on t. Thus is also stationary. ##5. The dataset in the file “AusBeer.csv” (attached to the assignment folder) contains data on monthly Australian beer production from 1956 to 1995, measured in megalitres. You can set it up for analysis with the following code (assuming all packages have been properly installed and the data are in the correct directory):

Analyse the attached data and determine whether you can detect a) a trend with time; b) seasonal variation (based on an appropriate period) If you detect either of these two features, remove them and assess the residuals for stationarity.

We will take an analogous approach to the codes provided on MyCourses with some additional features.

setwd("~/Documents/MATH 204/Working Dir")  
library(tidyverse)

## ── Attaching packages ───────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 2.2.1 ✔ purrr 0.2.4  
## ✔ tibble 1.4.2 ✔ dplyr 0.7.4  
## ✔ tidyr 0.8.0 ✔ stringr 1.3.0  
## ✔ readr 1.1.1 ✔ forcats 0.3.0

## ── Conflicts ──────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(itsmr)   
library(forecast)

##   
## Attaching package: 'forecast'

## The following object is masked from 'package:itsmr':  
##   
## forecast

library(tibbletime)

##   
## Attaching package: 'tibbletime'

## The following object is masked from 'package:stats':  
##   
## filter

library(tsbox)   
library(gridExtra)

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

australian\_beer = read\_csv("AusBeer.csv")

## Parsed with column specification:  
## cols(  
## Date = col\_date(format = ""),  
## Production = col\_double()  
## )

head(australian\_beer)

## # A tibble: 6 x 2  
## Date Production  
## <date> <dbl>  
## 1 1956-01-01 93.2  
## 2 1956-02-01 96.0  
## 3 1956-03-01 95.2  
## 4 1956-04-01 77.1  
## 5 1956-05-01 70.9  
## 6 1956-06-01 64.8

tail(australian\_beer)

## # A tibble: 6 x 2  
## Date Production  
## <date> <dbl>  
## 1 1995-03-01 152.  
## 2 1995-04-01 127.  
## 3 1995-05-01 151.  
## 4 1995-06-01 130.  
## 5 1995-07-01 119.  
## 6 1995-08-01 153.

dim(australian\_beer)

## [1] 476 2

class(australian\_beer)

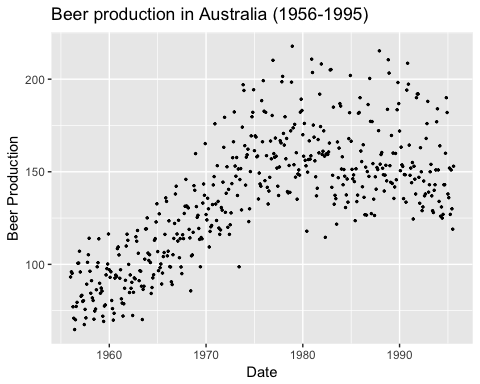
## [1] "tbl\_df" "tbl" "data.frame"

Aus <- ts\_df(australian\_beer)

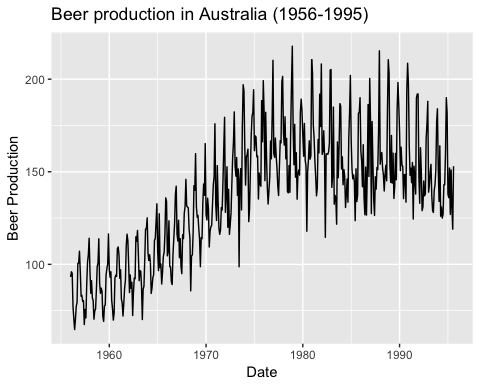
## [time]: 'Date' [value]: 'Production'

Let’s plot our date first:

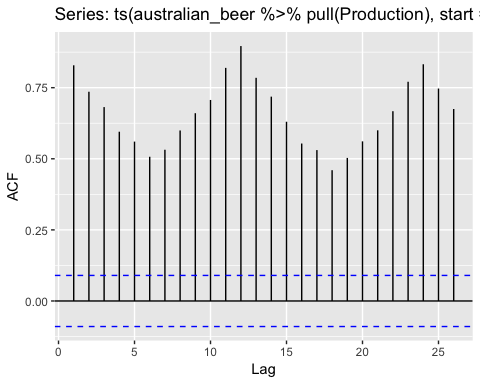
#Plot data points  
ggplot(Aus,aes(x=Date,y=Production)) + geom\_point(size=0.5) + ylab("Beer Production") +  
 ggtitle("Beer production in Australia (1956-1995)") + xlab("Date")



#Plot a curve  
ggplot(Aus,aes(x=Date,y=Production)) + geom\_line() + ylab("Beer Production") +  
 ggtitle("Beer production in Australia (1956-1995)") + xlab("Date")



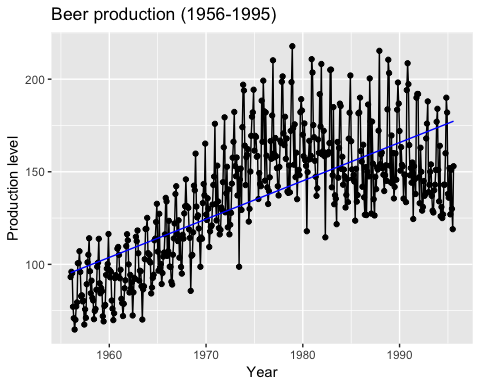
ggAcf(ts(australian\_beer %>% pull(Production),start=1,end=476))

 And we shall fit a linear trend

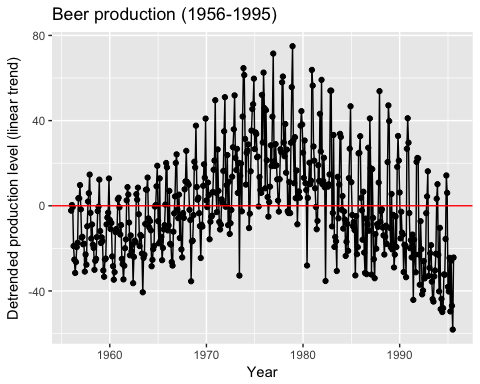
Aus\_tbl <- as\_tbl\_time(Aus,index=Date)  
Aus\_tbl <- Aus\_tbl %>% mutate(Aus\_lin=trend(Production,p=1)) ## Linear trend  
  
  
  
Aus\_tbl = Aus\_tbl %>% mutate(lin\_residuals = Production - Aus\_lin)

We will fit the linear trend for the production level first, and then will check whether it was a good fit using residuals for both the model assuming a linear trend and another assuming a quadratic trend.

ggplot(Aus\_tbl,aes(x=Date,y=Production))+ geom\_line() + geom\_point()+ geom\_line(aes(y=Aus\_lin),color="blue")+ylab("Production level") +  
 ggtitle("Beer production (1956-1995)") + xlab("Year")

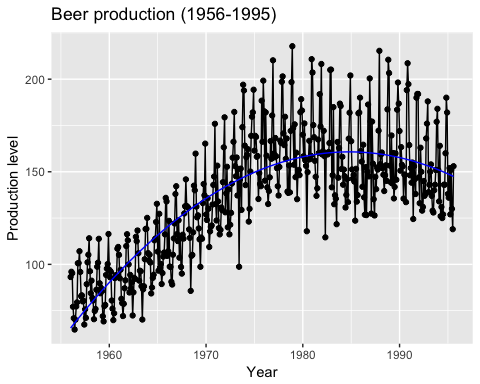


ggplot(Aus\_tbl,aes(x=Date,y=lin\_residuals))+   
 geom\_line() +   
 geom\_point()+   
 geom\_hline(yintercept=0,color="red") +  
 ylab("Detrended production level (linear trend)") +  
 ggtitle("Beer production (1956-1995)") + xlab("Year")

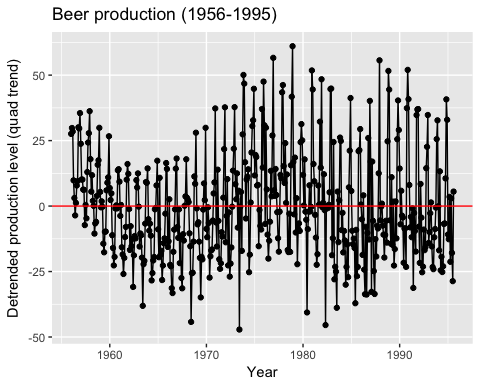


The visual inspection suggests there exists a positive trend that can be roughly explained by a linear trend (but later it shows that without assuming seasonality the quadratric trend seems to be a better one).

#quadratic trend?  
Aus\_tbl <- Aus\_tbl %>% mutate(Aus\_quad=trend(Production,p=2))  
Aus\_tbl <- Aus\_tbl %>% mutate(quad\_residuals = Production - Aus\_quad)  
ggplot(Aus\_tbl,aes(x=Date,y=Production))+ geom\_line() + geom\_point()+ geom\_line(aes(y=Aus\_quad),color="blue")+ylab("Production level") +  
 ggtitle("Beer production (1956-1995)") + xlab("Year")



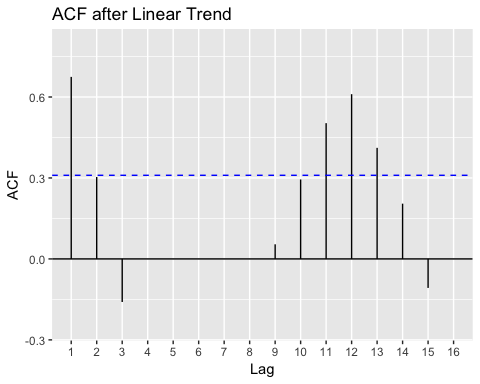
ggplot(Aus\_tbl,aes(x=Date,y=quad\_residuals))+   
 geom\_line() +   
 geom\_point()+   
 geom\_hline(yintercept=0,color="red") +  
 ylab("Detrended production level (quad trend)") +  
 ggtitle("Beer production (1956-1995)") + xlab("Year")



p\_lin = ggAcf(ts(Aus\_tbl %>% pull(lin\_residuals),start=1956,end=1995)) + ggtitle("ACF after Linear Trend") + ylim(c(-0.25,0.8))  
p\_lin

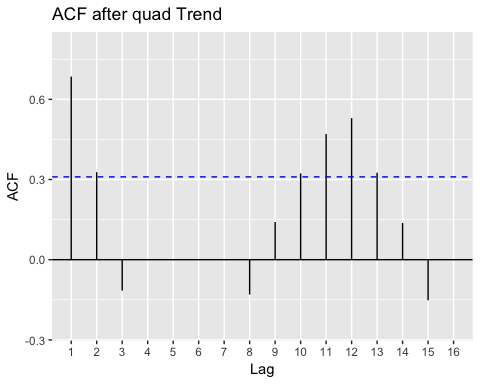
## Warning: Removed 6 rows containing missing values (geom\_segment).

## Warning: Removed 1 rows containing missing values (geom\_hline).

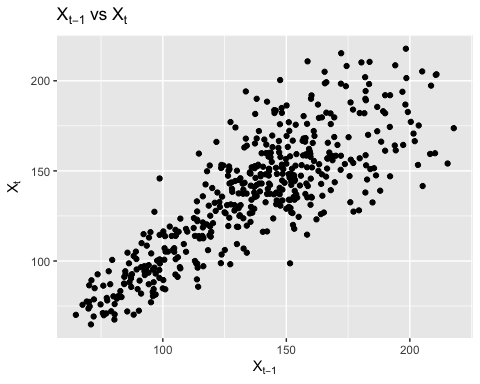


p\_quad = ggAcf(ts(Aus\_tbl %>% pull(quad\_residuals),start=1956,end=1995)) + ggtitle("ACF after quad Trend") + ylim(c(-0.25,0.8))  
p\_quad

## Warning: Removed 5 rows containing missing values (geom\_segment).  
  
## Warning: Removed 1 rows containing missing values (geom\_hline).



ggplot(Aus\_tbl %>% mutate(production\_lag1 = lag(Production,n=1)) %>%   
 slice(2:n()), aes(x=production\_lag1,y=Production)) +   
 geom\_point() + xlab(expression(X[t-1])) + ylab(expression(X[t])) +   
 ggtitle(expression(X[t-1]~"vs"~X[t]))

 Setting ARIMA models:

Aus\_AR1 = arima(ts\_ts(Aus\_tbl %>% select(Date,Production)),order=c(1,0,0))

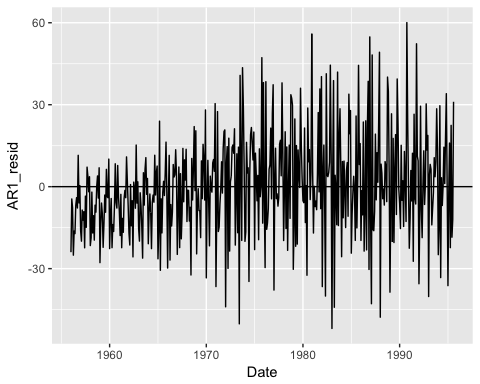
## [time]: 'Date'

print(Aus\_AR1)

##   
## Call:  
## arima(x = ts\_ts(Aus\_tbl %>% select(Date, Production)), order = c(1, 0, 0))  
##   
## Coefficients:  
## ar1 intercept  
## 0.8302 136.2247  
## s.e. 0.0254 5.0195  
##   
## sigma^2 estimated as 352.7: log likelihood = -2072.03, aic = 4150.06

Aus\_tbl = Aus\_tbl %>% mutate(AR1\_resid = residuals(Aus\_AR1))  
  
ggplot(Aus\_tbl,aes(x=Date,y=AR1\_resid)) + geom\_line() + geom\_hline(yintercept=0)

## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.

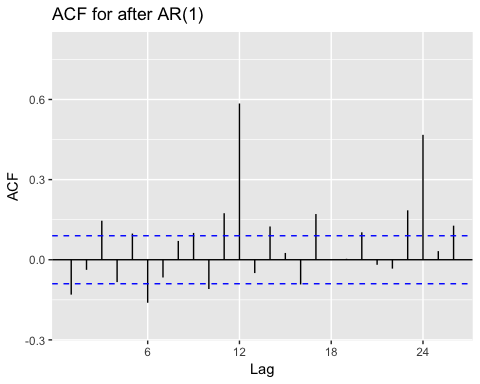


p\_ar1 = ggAcf(ts\_ts(Aus\_tbl %>% select(Date,AR1\_resid))) + ylim(c(-0.25,0.8))+ ggtitle("ACF for after AR(1)")

## [time]: 'Date'   
## [time]: 'Date'

p\_ar1

## Warning: Removed 1 rows containing missing values (geom\_segment).



Aus\_AR2 = arima(ts\_ts(Aus\_tbl %>% select(Date,Production)),order=c(2,0,0))

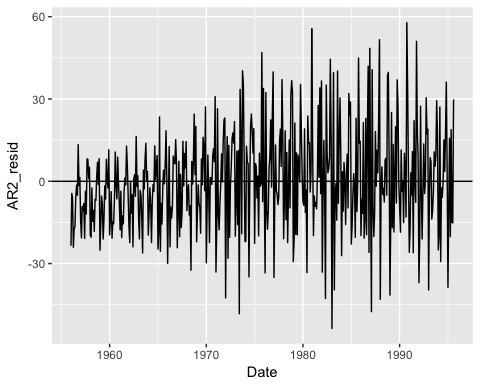
## [time]: 'Date'

print(Aus\_AR2)

##   
## Call:  
## arima(x = ts\_ts(Aus\_tbl %>% select(Date, Production)), order = c(2, 0, 0))  
##   
## Coefficients:  
## ar1 ar2 intercept  
## 0.6996 0.1572 135.8944  
## s.e. 0.0453 0.0454 5.8516  
##   
## sigma^2 estimated as 344: log likelihood = -2066.11, aic = 4140.23

Aus\_tbl = Aus\_tbl %>% mutate(AR2\_resid = residuals(Aus\_AR2))  
  
ggplot(Aus\_tbl,aes(x=Date,y=AR2\_resid)) + geom\_line() + geom\_hline(yintercept=0)

## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.

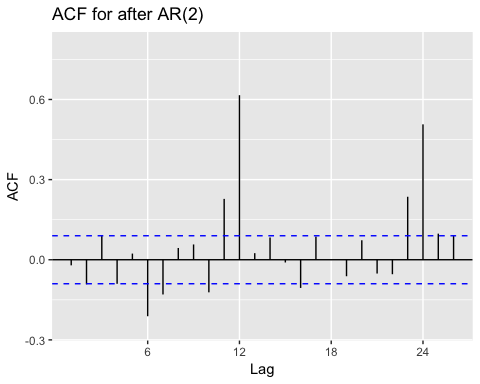


p\_ar2 = ggAcf(ts\_ts(Aus\_tbl %>% select(Date,AR2\_resid))) + ylim(c(-0.25,0.8))+ ggtitle("ACF for after AR(2)")

## [time]: 'Date'   
## [time]: 'Date'

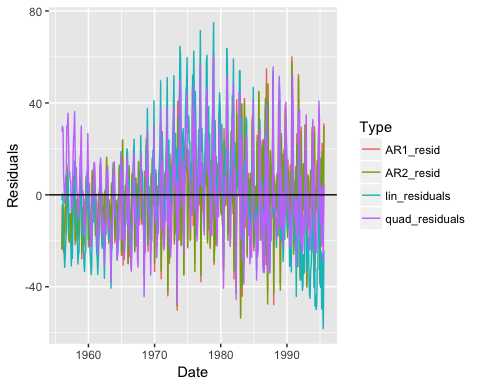
p\_ar2

## Warning: Removed 1 rows containing missing values (geom\_segment).



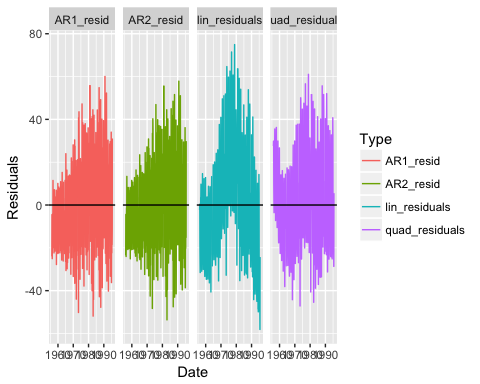
ggplot(Aus\_tbl %>% select(Date,lin\_residuals,quad\_residuals,AR1\_resid,AR2\_resid) %>%   
 gather(key=Type,value=Residuals,-Date),aes(x=Date,y=Residuals,col=Type)) + geom\_line() + geom\_hline(yintercept=0)

## Warning: attributes are not identical across measure variables;  
## they will be dropped



ggplot(Aus\_tbl %>% select(Date,lin\_residuals,quad\_residuals,AR1\_resid,AR2\_resid) %>%   
 gather(key=Type,value=Residuals,-Date),aes(x=Date,y=Residuals,col=Type)) + geom\_line() + facet\_grid(~Type) + geom\_hline(yintercept=0)

## Warning: attributes are not identical across measure variables;  
## they will be dropped

 Both Arima models show some tendendcy of heteroscedasticity. The model assuming a linear trend strongly suggests the seasonal component. The model assuming a quadratic trend makes effect of the seasonal component less pronounced.

grid.arrange(p\_lin,p\_quad, p\_ar1,p\_ar2,ncol=2)

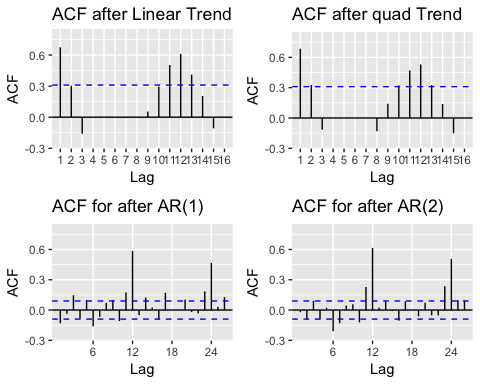
## Warning: Removed 6 rows containing missing values (geom\_segment).

## Warning: Removed 1 rows containing missing values (geom\_hline).

## Warning: Removed 5 rows containing missing values (geom\_segment).

## Warning: Removed 1 rows containing missing values (geom\_hline).

## Warning: Removed 1 rows containing missing values (geom\_segment).  
  
## Warning: Removed 1 rows containing missing values (geom\_segment).

 We can notice that both arima models bring the values of auto covariance function inside the confidence interval besides at every period of 12 (suggesting the existence of seasonal component).

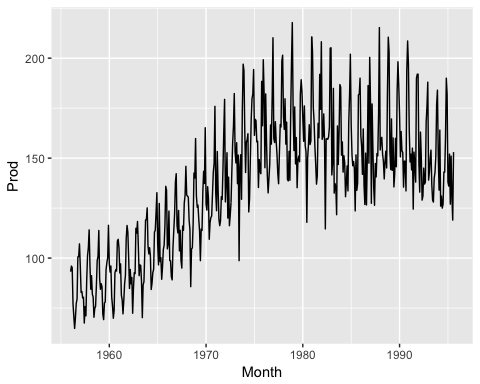
Aust\_ts = ts\_ts(australian\_beer)

## [time]: 'Date' [value]: 'Production'

Aust\_df = ts\_df(australian\_beer) %>% rename(Month = Date,  
 Prod= Production)

## [time]: 'Date' [value]: 'Production'

Aust\_tbl = as\_tbl\_time(Aust\_df, index = Month)  
  
ggplot(Aust\_tbl, aes(x=Month, y=Prod)) + geom\_line()



library(TTR)  
trend\_comp = ts\_df(SMA(Aust\_ts,n=12)) %>% rename(Month=time,SMA\_12=value)  
print(trend\_comp)

## Month SMA\_12  
## 1 1956-12-01 86.04167  
## 2 1957-01-01 86.26667  
## 3 1957-02-01 85.16667  
## 4 1957-03-01 84.17500  
## 5 1957-04-01 84.41667  
## 6 1957-05-01 85.20833  
## 7 1957-06-01 85.43333  
## 8 1957-07-01 85.90000  
## 9 1957-08-01 85.38333  
## 10 1957-09-01 86.20000  
## 11 1957-10-01 86.24167  
## 12 1957-11-01 86.61667  
## 13 1957-12-01 87.20000  
## 14 1958-01-01 87.23333  
## 15 1958-02-01 87.36667  
## 16 1958-03-01 88.02500  
## 17 1958-04-01 88.18333  
## 18 1958-05-01 88.19167  
## 19 1958-06-01 88.43333  
## 20 1958-07-01 88.35833  
## 21 1958-08-01 88.75833  
## 22 1958-09-01 88.50833  
## 23 1958-10-01 88.30833  
## 24 1958-11-01 87.95000  
## 25 1958-12-01 87.92500  
## 26 1959-01-01 87.38333  
## 27 1959-02-01 87.38333  
## 28 1959-03-01 87.05000  
## 29 1959-04-01 87.35833  
## 30 1959-05-01 86.65000  
## 31 1959-06-01 86.55000  
## 32 1959-07-01 86.77500  
## 33 1959-08-01 86.95833  
## 34 1959-09-01 87.62500  
## 35 1959-10-01 87.54167  
## 36 1959-11-01 87.48333  
## 37 1959-12-01 87.70000  
## 38 1960-01-01 88.30833  
## 39 1960-02-01 89.02500  
## 40 1960-03-01 89.75833  
## 41 1960-04-01 89.33333  
## 42 1960-05-01 89.67500  
## 43 1960-06-01 89.73333  
## 44 1960-07-01 89.40833  
## 45 1960-08-01 90.61667  
## 46 1960-09-01 90.60833  
## 47 1960-10-01 90.25833  
## 48 1960-11-01 90.95000  
## 49 1960-12-01 90.36667  
## 50 1961-01-01 91.03333  
## 51 1961-02-01 90.99167  
## 52 1961-03-01 91.08333  
## 53 1961-04-01 91.15833  
## 54 1961-05-01 91.40833  
## 55 1961-06-01 91.59167  
## 56 1961-07-01 92.01667  
## 57 1961-08-01 91.55833  
## 58 1961-09-01 91.32500  
## 59 1961-10-01 92.69167  
## 60 1961-11-01 93.34167  
## 61 1961-12-01 93.64167  
## 62 1962-01-01 93.21667  
## 63 1962-02-01 92.57500  
## 64 1962-03-01 92.34167  
## 65 1962-04-01 92.81667  
## 66 1962-05-01 93.75000  
## 67 1962-06-01 93.77500  
## 68 1962-07-01 94.29167  
## 69 1962-08-01 94.75833  
## 70 1962-09-01 94.82500  
## 71 1962-10-01 95.24167  
## 72 1962-11-01 94.92500  
## 73 1962-12-01 95.36667  
## 74 1963-01-01 95.86667  
## 75 1963-02-01 96.40000  
## 76 1963-03-01 96.59167  
## 77 1963-04-01 97.35833  
## 78 1963-05-01 97.18333  
## 79 1963-06-01 97.00000  
## 80 1963-07-01 97.13333  
## 81 1963-08-01 96.75833  
## 82 1963-09-01 97.64167  
## 83 1963-10-01 97.99167  
## 84 1963-11-01 98.55000  
## 85 1963-12-01 99.11667  
## 86 1964-01-01 99.12500  
## 87 1964-02-01 100.03333  
## 88 1964-03-01 100.75000  
## 89 1964-04-01 101.14167  
## 90 1964-05-01 100.81667  
## 91 1964-06-01 102.25833  
## 92 1964-07-01 102.77500  
## 93 1964-08-01 103.29167  
## 94 1964-09-01 104.14167  
## 95 1964-10-01 103.70833  
## 96 1964-11-01 104.01667  
## 97 1964-12-01 104.65000  
## 98 1965-01-01 104.71667  
## 99 1965-02-01 104.25833  
## 100 1965-03-01 106.10000  
## 101 1965-04-01 105.86667  
## 102 1965-05-01 107.19167  
## 103 1965-06-01 107.35000  
## 104 1965-07-01 107.56667  
## 105 1965-08-01 108.38333  
## 106 1965-09-01 107.83333  
## 107 1965-10-01 108.02500  
## 108 1965-11-01 109.10833  
## 109 1965-12-01 109.21667  
## 110 1966-01-01 109.02500  
## 111 1966-02-01 109.90000  
## 112 1966-03-01 109.58333  
## 113 1966-04-01 109.63333  
## 114 1966-05-01 109.50000  
## 115 1966-06-01 109.60000  
## 116 1966-07-01 109.08333  
## 117 1966-08-01 109.16667  
## 118 1966-09-01 109.80000  
## 119 1966-10-01 110.29167  
## 120 1966-11-01 110.46667  
## 121 1966-12-01 111.15000  
## 122 1967-01-01 112.13333  
## 123 1967-02-01 112.59167  
## 124 1967-03-01 112.61667  
## 125 1967-04-01 113.01667  
## 126 1967-05-01 114.29167  
## 127 1967-06-01 114.95833  
## 128 1967-07-01 115.45000  
## 129 1967-08-01 116.35000  
## 130 1967-09-01 116.34167  
## 131 1967-10-01 116.79167  
## 132 1967-11-01 116.24167  
## 133 1967-12-01 116.55000  
## 134 1968-01-01 117.80833  
## 135 1968-02-01 119.34167  
## 136 1968-03-01 119.90000  
## 137 1968-04-01 121.17500  
## 138 1968-05-01 121.20833  
## 139 1968-06-01 120.13333  
## 140 1968-07-01 120.93333  
## 141 1968-08-01 120.02500  
## 142 1968-09-01 120.30833  
## 143 1968-10-01 121.55833  
## 144 1968-11-01 122.27500  
## 145 1968-12-01 123.43333  
## 146 1969-01-01 123.40833  
## 147 1969-02-01 122.94167  
## 148 1969-03-01 122.60833  
## 149 1969-04-01 122.65000  
## 150 1969-05-01 122.58333  
## 151 1969-06-01 123.66667  
## 152 1969-07-01 124.49167  
## 153 1969-08-01 125.21667  
## 154 1969-09-01 126.53333  
## 155 1969-10-01 126.60833  
## 156 1969-11-01 126.38333  
## 157 1969-12-01 126.83333  
## 158 1970-01-01 126.47500  
## 159 1970-02-01 126.35833  
## 160 1970-03-01 127.12500  
## 161 1970-04-01 128.00833  
## 162 1970-05-01 127.66667  
## 163 1970-06-01 129.25833  
## 164 1970-07-01 129.74167  
## 165 1970-08-01 130.34167  
## 166 1970-09-01 130.27500  
## 167 1970-10-01 130.23333  
## 168 1970-11-01 131.07500  
## 169 1970-12-01 131.96667  
## 170 1971-01-01 132.44167  
## 171 1971-02-01 132.41667  
## 172 1971-03-01 133.88333  
## 173 1971-04-01 134.21667  
## 174 1971-05-01 135.06667  
## 175 1971-06-01 134.93333  
## 176 1971-07-01 134.79167  
## 177 1971-08-01 135.60000  
## 178 1971-09-01 135.35000  
## 179 1971-10-01 135.47500  
## 180 1971-11-01 136.79167  
## 181 1971-12-01 137.08333  
## 182 1972-01-01 136.70833  
## 183 1972-02-01 137.93333  
## 184 1972-03-01 137.88333  
## 185 1972-04-01 136.71667  
## 186 1972-05-01 138.45833  
## 187 1972-06-01 138.45833  
## 188 1972-07-01 138.69167  
## 189 1972-08-01 138.45000  
## 190 1972-09-01 139.64167  
## 191 1972-10-01 140.74167  
## 192 1972-11-01 140.99167  
## 193 1972-12-01 141.23333  
## 194 1973-01-01 143.31667  
## 195 1973-02-01 144.08333  
## 196 1973-03-01 144.50000  
## 197 1973-04-01 145.93333  
## 198 1973-05-01 146.85000  
## 199 1973-06-01 145.39167  
## 200 1973-07-01 147.42500  
## 201 1973-08-01 149.41667  
## 202 1973-09-01 148.23333  
## 203 1973-10-01 149.60833  
## 204 1973-11-01 152.17500  
## 205 1973-12-01 153.14167  
## 206 1974-01-01 154.05833  
## 207 1974-02-01 153.65833  
## 208 1974-03-01 153.67500  
## 209 1974-04-01 155.50833  
## 210 1974-05-01 156.40000  
## 211 1974-06-01 158.43333  
## 212 1974-07-01 157.11667  
## 213 1974-08-01 156.98333  
## 214 1974-09-01 160.31667  
## 215 1974-10-01 160.78333  
## 216 1974-11-01 159.54167  
## 217 1974-12-01 159.57500  
## 218 1975-01-01 159.35000  
## 219 1975-02-01 161.56667  
## 220 1975-03-01 162.47500  
## 221 1975-04-01 162.38333  
## 222 1975-05-01 162.07500  
## 223 1975-06-01 163.09167  
## 224 1975-07-01 164.70000  
## 225 1975-08-01 164.14167  
## 226 1975-09-01 161.87500  
## 227 1975-10-01 162.60000  
## 228 1975-11-01 161.27500  
## 229 1975-12-01 161.68333  
## 230 1976-01-01 163.45833  
## 231 1976-02-01 161.44167  
## 232 1976-03-01 162.55000  
## 233 1976-04-01 162.60000  
## 234 1976-05-01 161.19167  
## 235 1976-06-01 160.96667  
## 236 1976-07-01 160.15833  
## 237 1976-08-01 160.45833  
## 238 1976-09-01 162.49167  
## 239 1976-10-01 159.87500  
## 240 1976-11-01 161.05833  
## 241 1976-12-01 161.97500  
## 242 1977-01-01 160.06667  
## 243 1977-02-01 161.11667  
## 244 1977-03-01 159.95833  
## 245 1977-04-01 159.93333  
## 246 1977-05-01 160.80000  
## 247 1977-06-01 161.60000  
## 248 1977-07-01 161.40000  
## 249 1977-08-01 161.86667  
## 250 1977-09-01 161.88333  
## 251 1977-10-01 162.60000  
## 252 1977-11-01 164.11667  
## 253 1977-12-01 163.39167  
## 254 1978-01-01 164.30000  
## 255 1978-02-01 164.85000  
## 256 1978-03-01 165.80833  
## 257 1978-04-01 165.69167  
## 258 1978-05-01 167.02500  
## 259 1978-06-01 166.78333  
## 260 1978-07-01 166.90000  
## 261 1978-08-01 166.96667  
## 262 1978-09-01 164.64167  
## 263 1978-10-01 165.18333  
## 264 1978-11-01 165.16667  
## 265 1978-12-01 166.52500  
## 266 1979-01-01 166.77500  
## 267 1979-02-01 165.89167  
## 268 1979-03-01 165.55000  
## 269 1979-04-01 164.72500  
## 270 1979-05-01 164.08333  
## 271 1979-06-01 163.74167  
## 272 1979-07-01 164.59167  
## 273 1979-08-01 164.39167  
## 274 1979-09-01 165.16667  
## 275 1979-10-01 166.00833  
## 276 1979-11-01 165.24167  
## 277 1979-12-01 162.35000  
## 278 1980-01-01 162.04167  
## 279 1980-02-01 162.42500  
## 280 1980-03-01 162.46667  
## 281 1980-04-01 163.22500  
## 282 1980-05-01 162.63333  
## 283 1980-06-01 161.19167  
## 284 1980-07-01 161.27500  
## 285 1980-08-01 161.74167  
## 286 1980-09-01 163.28333  
## 287 1980-10-01 161.16667  
## 288 1980-11-01 158.61667  
## 289 1980-12-01 160.92500  
## 290 1981-01-01 163.72500  
## 291 1981-02-01 165.12500  
## 292 1981-03-01 164.50833  
## 293 1981-04-01 164.48333  
## 294 1981-05-01 163.99167  
## 295 1981-06-01 165.58333  
## 296 1981-07-01 164.85833  
## 297 1981-08-01 165.75833  
## 298 1981-09-01 165.21667  
## 299 1981-10-01 168.14167  
## 300 1981-11-01 169.45833  
## 301 1981-12-01 169.24167  
## 302 1982-01-01 165.55833  
## 303 1982-02-01 164.38333  
## 304 1982-03-01 164.66667  
## 305 1982-04-01 164.87500  
## 306 1982-05-01 162.15000  
## 307 1982-06-01 164.03333  
## 308 1982-07-01 165.58333  
## 309 1982-08-01 164.91667  
## 310 1982-09-01 164.95833  
## 311 1982-10-01 162.75833  
## 312 1982-11-01 165.30833  
## 313 1982-12-01 165.05833  
## 314 1983-01-01 163.57500  
## 315 1983-02-01 162.49167  
## 316 1983-03-01 163.55833  
## 317 1983-04-01 161.40000  
## 318 1983-05-01 163.29167  
## 319 1983-06-01 161.28333  
## 320 1983-07-01 158.11667  
## 321 1983-08-01 158.67500  
## 322 1983-09-01 157.51667  
## 323 1983-10-01 157.29167  
## 324 1983-11-01 155.77500  
## 325 1983-12-01 154.13333  
## 326 1984-01-01 154.95833  
## 327 1984-02-01 155.79167  
## 328 1984-03-01 152.30000  
## 329 1984-04-01 153.85833  
## 330 1984-05-01 154.71667  
## 331 1984-06-01 154.31667  
## 332 1984-07-01 155.63333  
## 333 1984-08-01 153.96667  
## 334 1984-09-01 152.86667  
## 335 1984-10-01 153.29167  
## 336 1984-11-01 152.88333  
## 337 1984-12-01 154.25833  
## 338 1985-01-01 155.50833  
## 339 1985-02-01 154.94167  
## 340 1985-03-01 155.20833  
## 341 1985-04-01 154.96667  
## 342 1985-05-01 154.72500  
## 343 1985-06-01 154.13333  
## 344 1985-07-01 155.30833  
## 345 1985-08-01 154.29167  
## 346 1985-09-01 154.60833  
## 347 1985-10-01 155.75000  
## 348 1985-11-01 155.75833  
## 349 1985-12-01 154.75833  
## 350 1986-01-01 154.31667  
## 351 1986-02-01 154.66667  
## 352 1986-03-01 154.30833  
## 353 1986-04-01 155.66667  
## 354 1986-05-01 154.95833  
## 355 1986-06-01 155.22500  
## 356 1986-07-01 155.30000  
## 357 1986-08-01 154.69167  
## 358 1986-09-01 155.75000  
## 359 1986-10-01 156.14167  
## 360 1986-11-01 153.26667  
## 361 1986-12-01 154.13333  
## 362 1987-01-01 155.46667  
## 363 1987-02-01 153.12500  
## 364 1987-03-01 156.05833  
## 365 1987-04-01 155.20833  
## 366 1987-05-01 155.12500  
## 367 1987-06-01 155.09167  
## 368 1987-07-01 154.65833  
## 369 1987-08-01 155.82500  
## 370 1987-09-01 156.00833  
## 371 1987-10-01 153.08333  
## 372 1987-11-01 155.14167  
## 373 1987-12-01 156.38333  
## 374 1988-01-01 154.45833  
## 375 1988-02-01 157.11667  
## 376 1988-03-01 155.72500  
## 377 1988-04-01 155.51667  
## 378 1988-05-01 156.61667  
## 379 1988-06-01 157.71667  
## 380 1988-07-01 157.79167  
## 381 1988-08-01 158.86667  
## 382 1988-09-01 158.26667  
## 383 1988-10-01 160.97500  
## 384 1988-11-01 164.16667  
## 385 1988-12-01 163.16667  
## 386 1989-01-01 163.10000  
## 387 1989-02-01 161.85000  
## 388 1989-03-01 162.61667  
## 389 1989-04-01 161.93333  
## 390 1989-05-01 162.90833  
## 391 1989-06-01 162.57500  
## 392 1989-07-01 162.04167  
## 393 1989-08-01 162.57500  
## 394 1989-09-01 162.62500  
## 395 1989-10-01 162.60833  
## 396 1989-11-01 161.58333  
## 397 1989-12-01 160.20833  
## 398 1990-01-01 161.76667  
## 399 1990-02-01 162.29167  
## 400 1990-03-01 161.76667  
## 401 1990-04-01 162.60000  
## 402 1990-05-01 162.00000  
## 403 1990-06-01 161.99167  
## 404 1990-07-01 162.55000  
## 405 1990-08-01 161.59167  
## 406 1990-09-01 160.58333  
## 407 1990-10-01 161.46667  
## 408 1990-11-01 162.33333  
## 409 1990-12-01 163.20833  
## 410 1991-01-01 162.57500  
## 411 1991-02-01 162.36667  
## 412 1991-03-01 161.42500  
## 413 1991-04-01 160.62500  
## 414 1991-05-01 160.80000  
## 415 1991-06-01 159.88333  
## 416 1991-07-01 160.25833  
## 417 1991-08-01 160.05833  
## 418 1991-09-01 160.42500  
## 419 1991-10-01 160.08333  
## 420 1991-11-01 158.70000  
## 421 1991-12-01 158.25833  
## 422 1992-01-01 156.80833  
## 423 1992-02-01 155.55000  
## 424 1992-03-01 156.46667  
## 425 1992-04-01 156.95833  
## 426 1992-05-01 154.79167  
## 427 1992-06-01 155.33333  
## 428 1992-07-01 154.66667  
## 429 1992-08-01 153.91667  
## 430 1992-09-01 153.91667  
## 431 1992-10-01 152.08333  
## 432 1992-11-01 150.75000  
## 433 1992-12-01 150.41667  
## 434 1993-01-01 149.75000  
## 435 1993-02-01 150.58333  
## 436 1993-03-01 149.50000  
## 437 1993-04-01 149.83333  
## 438 1993-05-01 150.50000  
## 439 1993-06-01 150.33333  
## 440 1993-07-01 148.91667  
## 441 1993-08-01 149.16667  
## 442 1993-09-01 149.58333  
## 443 1993-10-01 148.16667  
## 444 1993-11-01 148.25000  
## 445 1993-12-01 147.91667  
## 446 1994-01-01 148.91667  
## 447 1994-02-01 148.16667  
## 448 1994-03-01 149.33333  
## 449 1994-04-01 147.00000  
## 450 1994-05-01 146.50000  
## 451 1994-06-01 146.16667  
## 452 1994-07-01 146.08333  
## 453 1994-08-01 146.33333  
## 454 1994-09-01 146.33333  
## 455 1994-10-01 147.08333  
## 456 1994-11-01 148.16667  
## 457 1994-12-01 148.00000  
## 458 1995-01-01 146.91667  
## 459 1995-02-01 147.08333  
## 460 1995-03-01 146.08333  
## 461 1995-04-01 146.16667  
## 462 1995-05-01 147.83333  
## 463 1995-06-01 148.25000  
## 464 1995-07-01 147.58333  
## 465 1995-08-01 148.41667

print(australian\_beer)

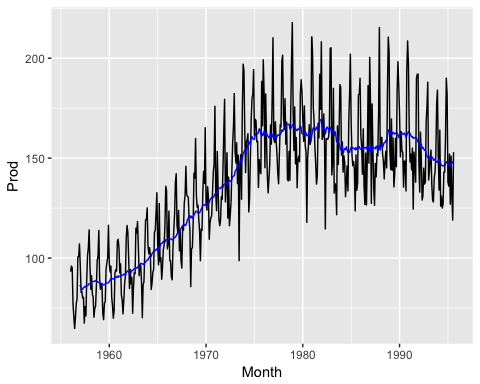
## # A tibble: 476 x 2  
## Date Production  
## <date> <dbl>  
## 1 1956-01-01 93.2  
## 2 1956-02-01 96.0  
## 3 1956-03-01 95.2  
## 4 1956-04-01 77.1  
## 5 1956-05-01 70.9  
## 6 1956-06-01 64.8  
## 7 1956-07-01 70.1  
## 8 1956-08-01 77.3  
## 9 1956-09-01 79.5  
## 10 1956-10-01 101.   
## # ... with 466 more rows

Aust\_tbl = full\_join(Aust\_tbl,trend\_comp) %>%  
 mutate(SMA\_resid=Prod-SMA\_12)

## Joining, by = "Month"

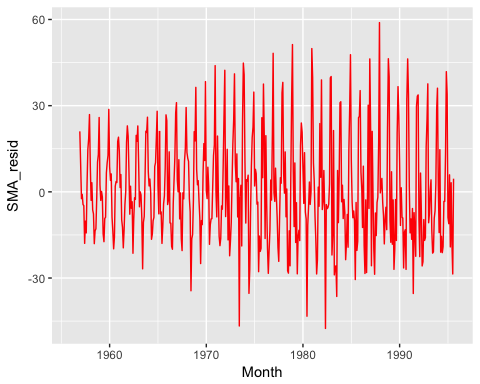
ggplot(Aust\_tbl,aes(x=Month,y=Prod)) + geom\_line() +   
 geom\_line(aes(y=SMA\_12),color="blue")

## Warning: Removed 11 rows containing missing values (geom\_path).



ggplot(Aust\_tbl,aes(x=Month,y=SMA\_resid)) + geom\_line(col="red")

## Warning: Removed 11 rows containing missing values (geom\_path).

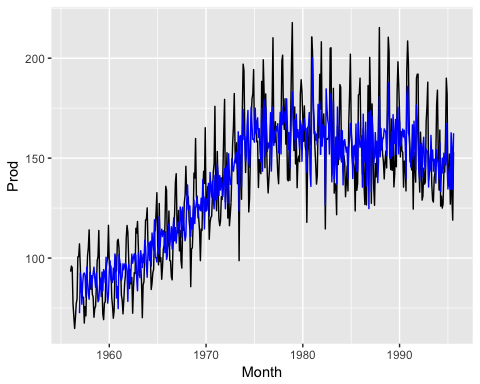


SMA\_resid\_ts = ts(Aust\_tbl%>% filter(!is.na(SMA\_resid)) %>% pull(SMA\_resid),   
 start=c(1956,12),frequency=12)  
   
 season\_comp = season(SMA\_resid\_ts,d=12)  
  
SMA\_resid\_tbl = ts\_df(SMA\_resid\_ts) %>% rename(Month=time,  
 SMA\_resid=value) %>%   
 mutate(seasonal = season\_comp) %>% as\_tbl\_time(index=Month)  
  
Aust\_tbl = full\_join(Aust\_tbl,SMA\_resid\_tbl)

## Joining, by = c("Month", "SMA\_resid")

Aust\_tbl = Aust\_tbl %>% mutate(deseason=Prod-seasonal)  
  
  
ggplot(Aust\_tbl,aes(x=Month,y=Prod)) + geom\_line() +   
 geom\_line(aes(y=deseason),color="blue")

## Warning: Removed 11 rows containing missing values (geom\_path).



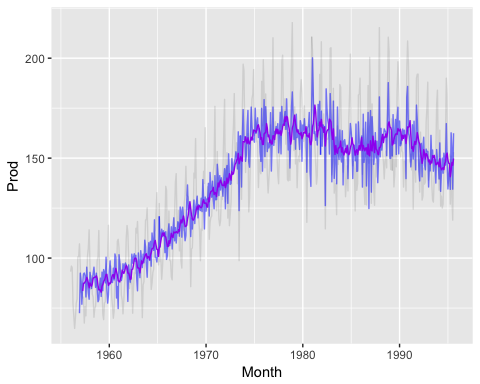
deseason\_ts = ts(Aust\_tbl %>% filter(!is.na(deseason)) %>% pull(deseason),  
 start=c(1956,12),frequency=12)  
  
deseason\_trend = ts\_df(SMA(deseason\_ts,n=5)) %>% rename(Month=time,de\_SMA\_5 = value)  
  
Aust\_tbl = full\_join(Aust\_tbl,deseason\_trend)

## Joining, by = "Month"

## Joining, by = "Month"  
  
Aust\_tbl = Aust\_tbl %>% mutate(Final\_resid=Prod-de\_SMA\_5-seasonal)  
  
ggplot(Aust\_tbl,aes(x=Month,y=Prod)) +geom\_line(alpha=0.1) + geom\_line(aes(y=deseason),color="blue",alpha=0.5) + geom\_line(aes(y=de\_SMA\_5),color="purple")

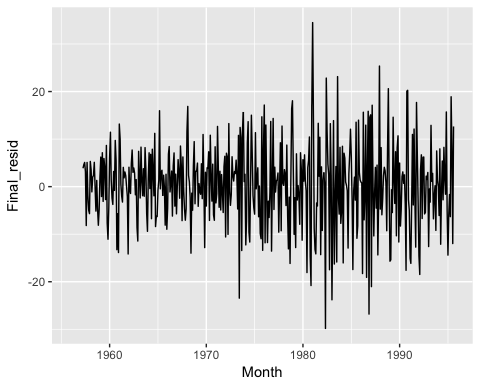
## Warning: Removed 11 rows containing missing values (geom\_path).

## Warning: Removed 15 rows containing missing values (geom\_path).

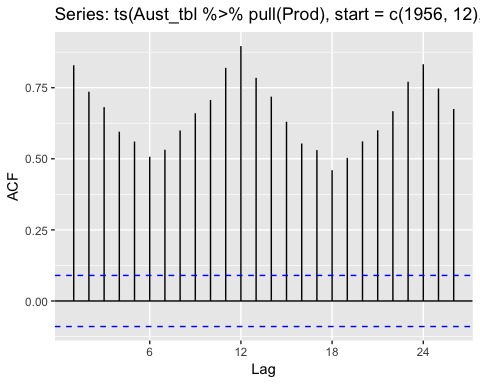


ggplot(Aust\_tbl,aes(x=Month,y=Final\_resid)) + geom\_line()

## Warning: Removed 15 rows containing missing values (geom\_path).

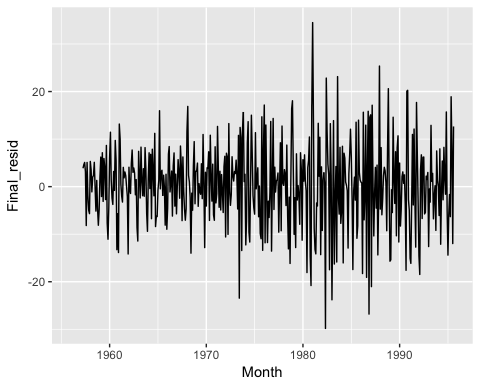


ggAcf(ts(Aust\_tbl %>% pull(Prod), start=c(1956,12), freq=12))

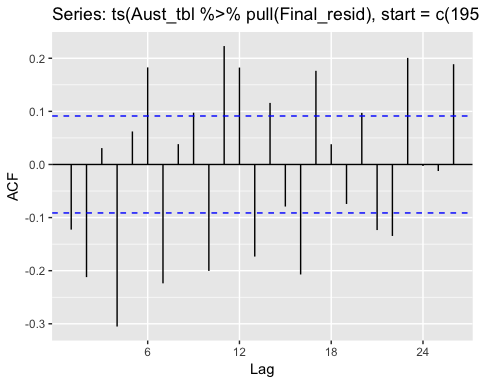


ggplot(Aust\_tbl,aes(x=Month,y=Final\_resid)) + geom\_line()

## Warning: Removed 15 rows containing missing values (geom\_path).



ggAcf(ts(Aust\_tbl %>% pull(Final\_resid), start=c(1956,12),freq=12))



The acf plot suggests that residuals seems quite stationary. It is something additional, but we might conduct a statistical test.

library(aTSA)

##   
## Attaching package: 'aTSA'

## The following object is masked from 'package:forecast':  
##   
## forecast

## The following object is masked from 'package:itsmr':  
##   
## forecast

## The following object is masked from 'package:graphics':  
##   
## identify

# ADF test for the residual  
  
adf.test(Aust\_tbl$Final\_resid)

## Augmented Dickey-Fuller Test   
## alternative: stationary   
##   
## Type 1: no drift no trend   
## lag ADF p.value  
## [1,] 0 -24.2 0.01  
## [2,] 1 -20.2 0.01  
## [3,] 2 -15.1 0.01  
## [4,] 3 -18.4 0.01  
## [5,] 4 -14.6 0.01  
## [6,] 5 -11.9 0.01  
## Type 2: with drift no trend   
## lag ADF p.value  
## [1,] 0 -24.2 0.01  
## [2,] 1 -20.2 0.01  
## [3,] 2 -15.1 0.01  
## [4,] 3 -18.5 0.01  
## [5,] 4 -14.7 0.01  
## [6,] 5 -12.0 0.01  
## Type 3: with drift and trend   
## lag ADF p.value  
## [1,] 0 -24.2 0.01  
## [2,] 1 -20.3 0.01  
## [3,] 2 -15.2 0.01  
## [4,] 3 -18.6 0.01  
## [5,] 4 -14.9 0.01  
## [6,] 5 -12.1 0.01  
## ----   
## Note: in fact, p.value = 0.01 means p.value <= 0.01

It seems that we can reject the null, suggesting that the series is stationary.