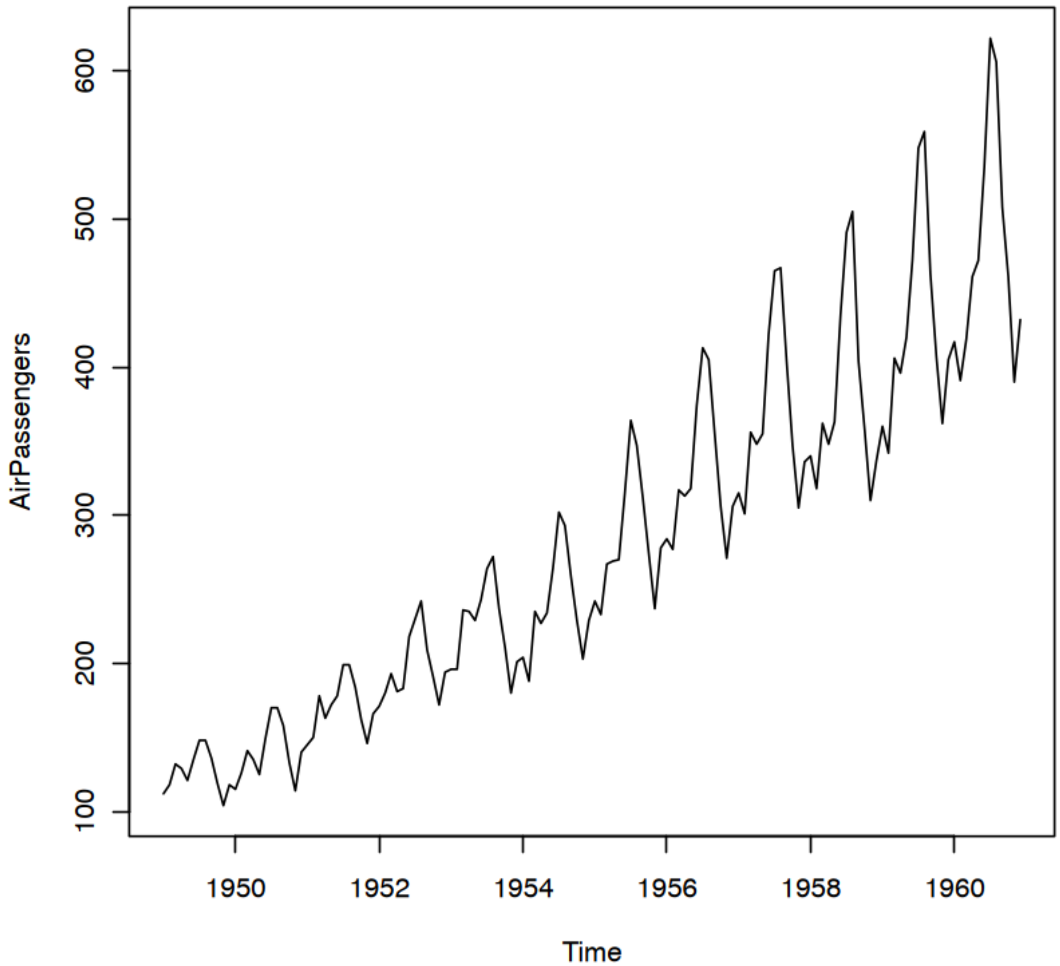
test1

November 26, 2024

[2]:
```r
library(astsa)

library(datasets)

## dataset
data("AirPassengers")
summary(AirPassengers)
plot(AirPassengers)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  104.0   180.0   265.5   280.3   360.5   622.0 
```



[3]:
```
data <- ts(AirPassengers)
sarima.for(data, xreg = data.frame(time = 1:length(data)), newxreg = data.
↪frame(time = 1:24 + length(data)), 24, 1, 0, 0, 0, 0, 1, 12)
```

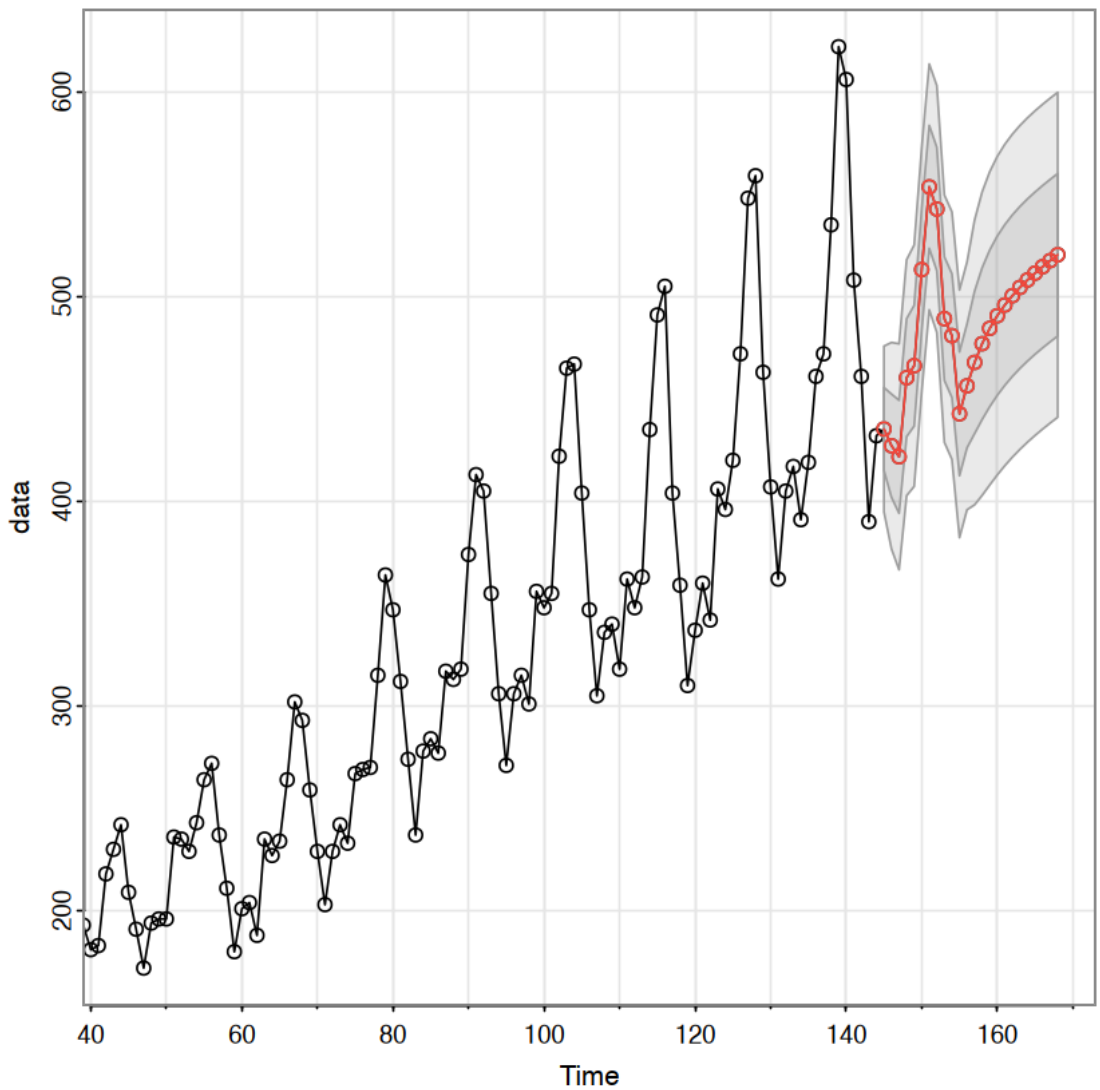
$pred A Time Series:

1. 435.296404448951 2. 427.132067810041 3. 421.740058102871 4. 460.354001689686
5. 466.215317459882 6. 513.152786883938 7. 553.530511194164 8. 542.737439328013
9. 489.167182676446 10. 480.895509645266 11. 442.72359302425 12. 456.381396774694
13. 467.826636698122 14. 476.9821798826 15. 484.434921417891 16. 490.621323438804
17. 495.865974101629 18. 500.410263123371 19. 504.433707206489 20. 508.069809330573
21. 511.417852990222 22. 514.551673333763 23. 517.526180083603 24. 520.382208501587

$se A Time Series:

1. 20.2369465077224 2. 25.2153420831165 3. 27.5854874380948 4. 28.8126935603087

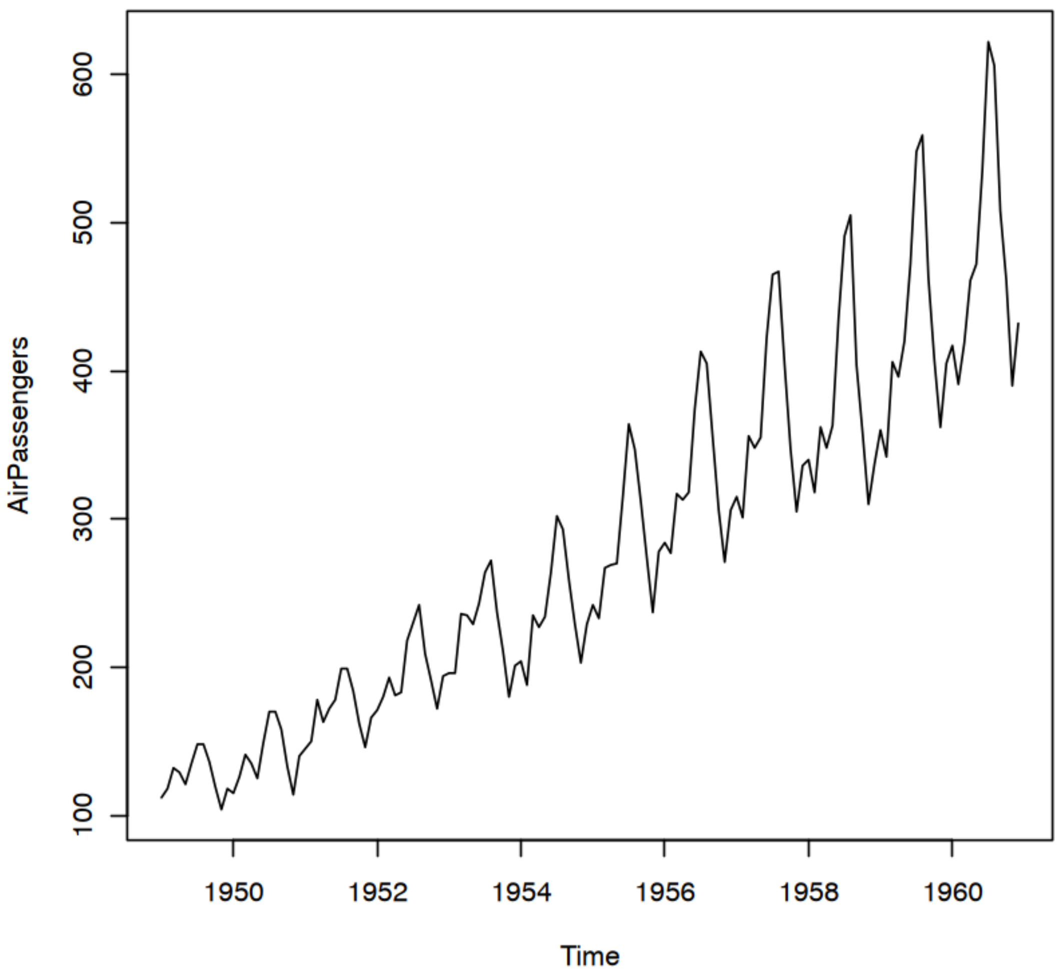
5. 29.4694463939624 6. 29.8264137556358 7. 30.0219320156396 8. 30.1293703559624 9. 30.1883551446478 10. 30.2204386342557 11. 30.2372840889865 12. 30.2449986195653 13. 34.7770323714392 14. 37.0461712998146 15. 38.2433623059643 16. 38.8896560011889 17. 39.2425241796212 18. 39.436325454688 19. 39.5431012580231 20. 39.6020310364572 21. 39.6345851062503 22. 39.6525779796707 23. 39.6625256114465 24. 39.6680261766189



[4]:
```r
DIR <- "/Users/gianlucamastrantonio/Dropbox (Politecnico di Torino Staff)/
↪didattica/DataSpaces/dispense/Time Series/"

# remotes::install_github("nickpoison/astsa/astsa_build")
library(astsa)
library(astsa)
library(datasets)
```

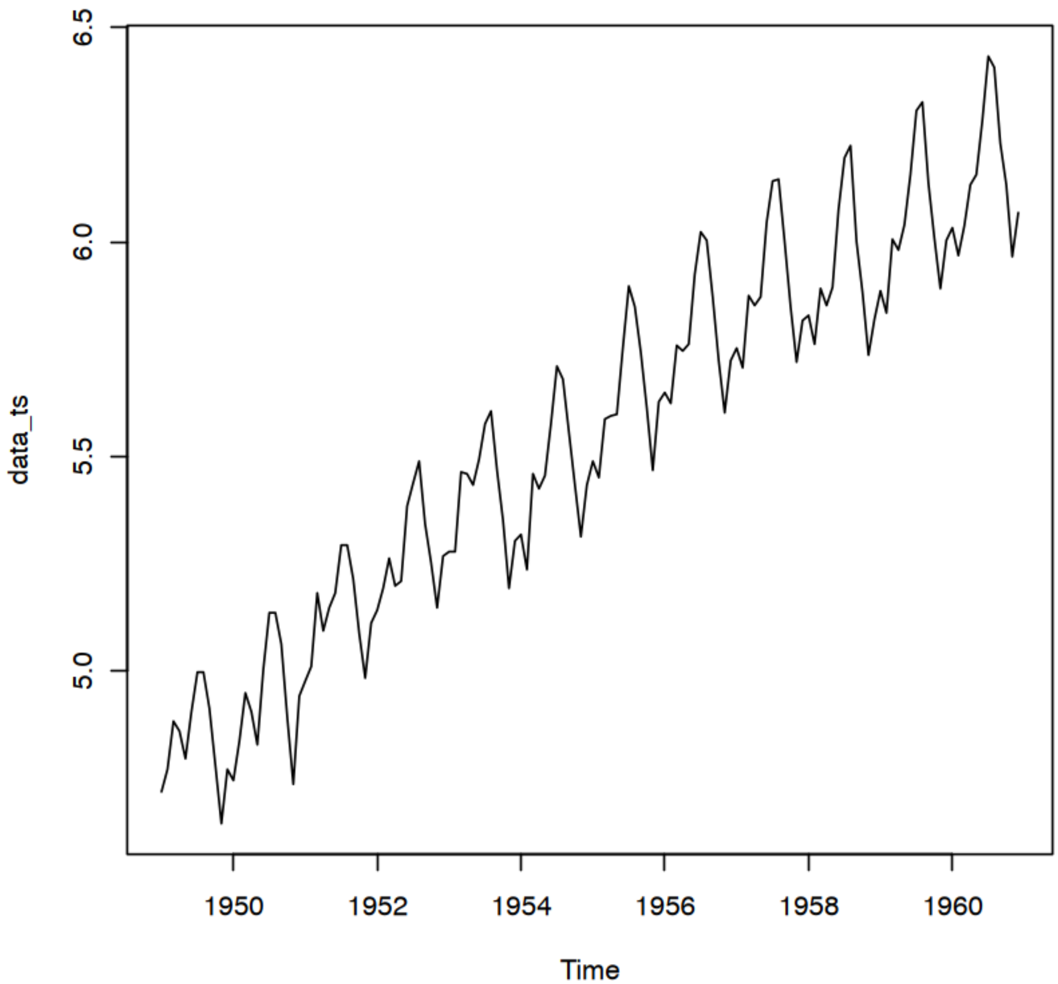
```r
## dataset
data("AirPassengers")
summary(AirPassengers)

# prendiamo lo stesso dataset precedente e confrontiao modelli ARIMA

# vediamo dei plot
plot(AirPassengers)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  104.0   180.0   265.5   280.3   360.5   622.0
```



Applico la traasfornmazione logartmica

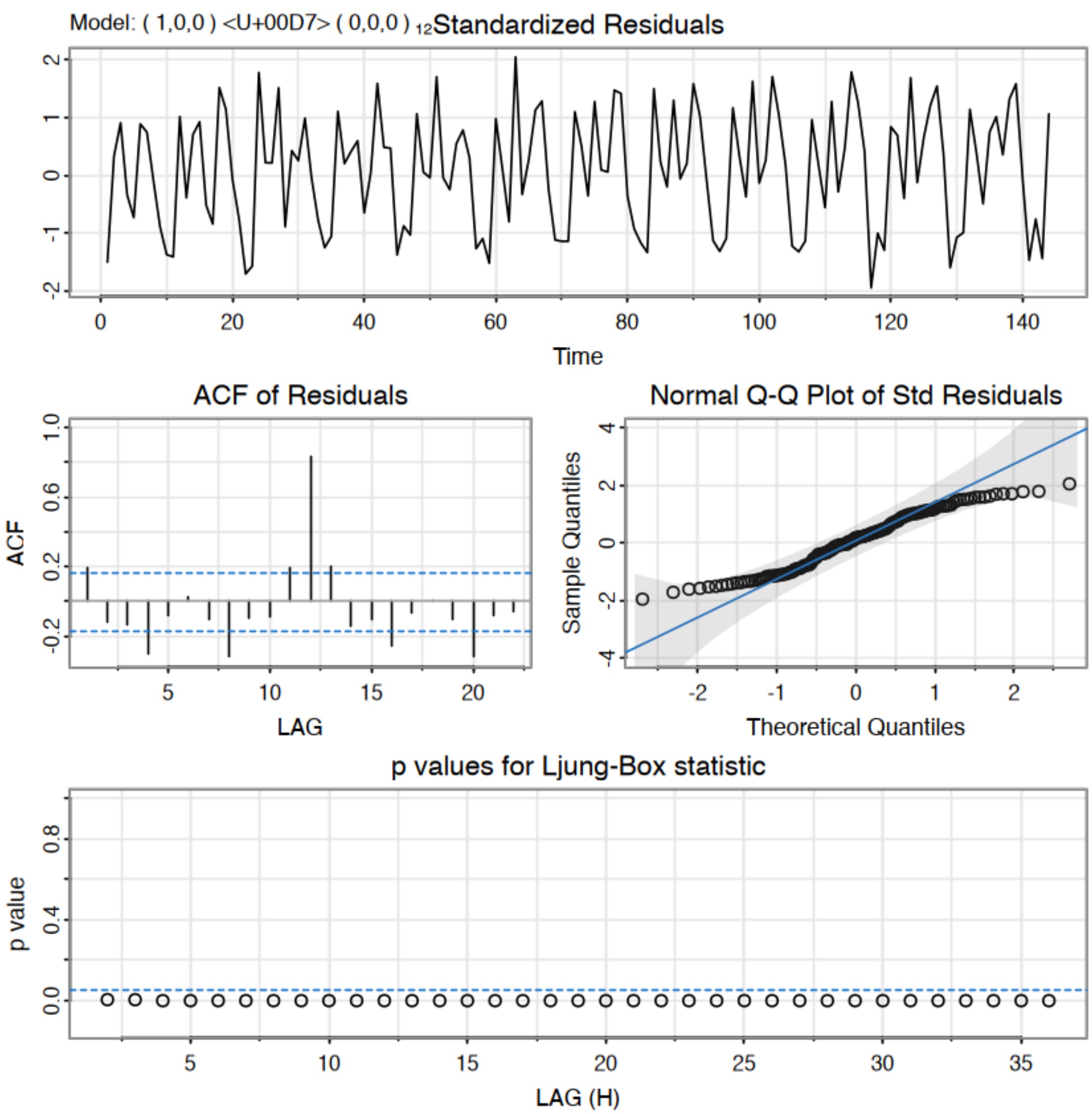
[5]:
```
data_ts = log(AirPassengers)
plot(data_ts)
```



[6]:
```
data_ts <- ts(data_ts)
sarima(data_ts,
1,# AR
0,# I
0,# MA
0,# AR_s
0,# I_s
0,# MA_s
12)
```

```
initial  value -0.829999
iter   2 value -2.248560
iter   3 value -2.250003
iter   4 value -2.250493
iter   5 value -2.250645
iter   6 value -2.251785
iter   7 value -2.252303
iter   8 value -2.252413
iter   9 value -2.252541
iter  10 value -2.252943
iter  11 value -2.253590
iter  12 value -2.253823
iter  13 value -2.253841
iter  14 value -2.253859
iter  15 value -2.253919
iter  16 value -2.254000
iter  17 value -2.254051
iter  18 value -2.254055
iter  19 value -2.254063
iter  20 value -2.254082
iter  21 value -2.254113
iter  22 value -2.254129
iter  23 value -2.254130
iter  24 value -2.254132
iter  25 value -2.254136
iter  26 value -2.254143
iter  27 value -2.254147
iter  28 value -2.254147
iter  29 value -2.254148
iter  30 value -2.254149
iter  31 value -2.254151
iter  32 value -2.254152
iter  32 value -2.254152
iter  32 value -2.254152
final  value -2.254152
converged
initial  value -2.222794
iter   2 value -2.224612
iter   3 value -2.228908
iter   4 value -2.229816
iter   5 value -2.230136
iter   6 value -2.230206
iter   7 value -2.230224
iter   8 value -2.231597
iter   9 value -2.231670
iter  10 value -2.231714
iter  11 value -2.231729
iter  12 value -2.231741
```

```
iter  13 value -2.231851
iter  14 value -2.231854
iter  15 value -2.231855
iter  16 value -2.231859
iter  17 value -2.231859
iter  18 value -2.231879
iter  19 value -2.231880
iter  20 value -2.231881
iter  21 value -2.231885
iter  22 value -2.231885
iter  23 value -2.231893
iter  23 value -2.231893
iter  23 value -2.231893
final  value -2.231893
converged
<><><><><><><><><><><><><><>

Coefficients:
      Estimate     SE t.value p.value
ar1     0.9781 0.0171 57.3534       0
xmean   5.4854 0.3181 17.2425       0

sigma^2 estimated as 0.01127042 on 142 degrees of freedom

AIC = -1.584242  AICc = -1.583651  BIC = -1.522371
```



[7]:

```
mod <- sarima(
  data_ts,
  1, # AR
  0, # I
  0, # MA
  1, # AR_s
  0, # I_s
  0, # MA_s
  12
)
```

```
initial  value -0.912435
iter   2 value -3.106051
```

```
iter   3 value -3.118288
iter   4 value -3.118682
iter   5 value -3.118841
iter   6 value -3.118922
iter   7 value -3.119519
iter   8 value -3.121222
iter   9 value -3.122354
iter  10 value -3.122715
iter  11 value -3.123006
iter  12 value -3.123424
iter  13 value -3.123719
iter  14 value -3.127010
iter  15 value -3.142954
iter  16 value -3.152645
iter  17 value -3.163801
iter  18 value -3.173404
iter  19 value -3.180734
iter  20 value -3.181914
iter  21 value -3.182794
iter  22 value -3.183073
iter  23 value -3.183084
iter  24 value -3.183338
iter  25 value -3.186227
iter  26 value -3.188607
iter  27 value -3.191853
iter  28 value -3.192674
iter  29 value -3.193119
iter  30 value -3.193122
iter  31 value -3.193131
iter  32 value -3.193137
iter  33 value -3.193138
iter  34 value -3.193154
iter  35 value -3.193178
iter  36 value -3.193223
iter  37 value -3.193225
iter  38 value -3.193248
iter  39 value -3.193253
iter  40 value -3.193254
iter  41 value -3.193267
iter  42 value -3.193278
iter  43 value -3.193290
iter  44 value -3.193291
iter  45 value -3.193292
iter  46 value -3.193293
iter  46 value -3.193293
iter  46 value -3.193293
final  value -3.193293
converged
```

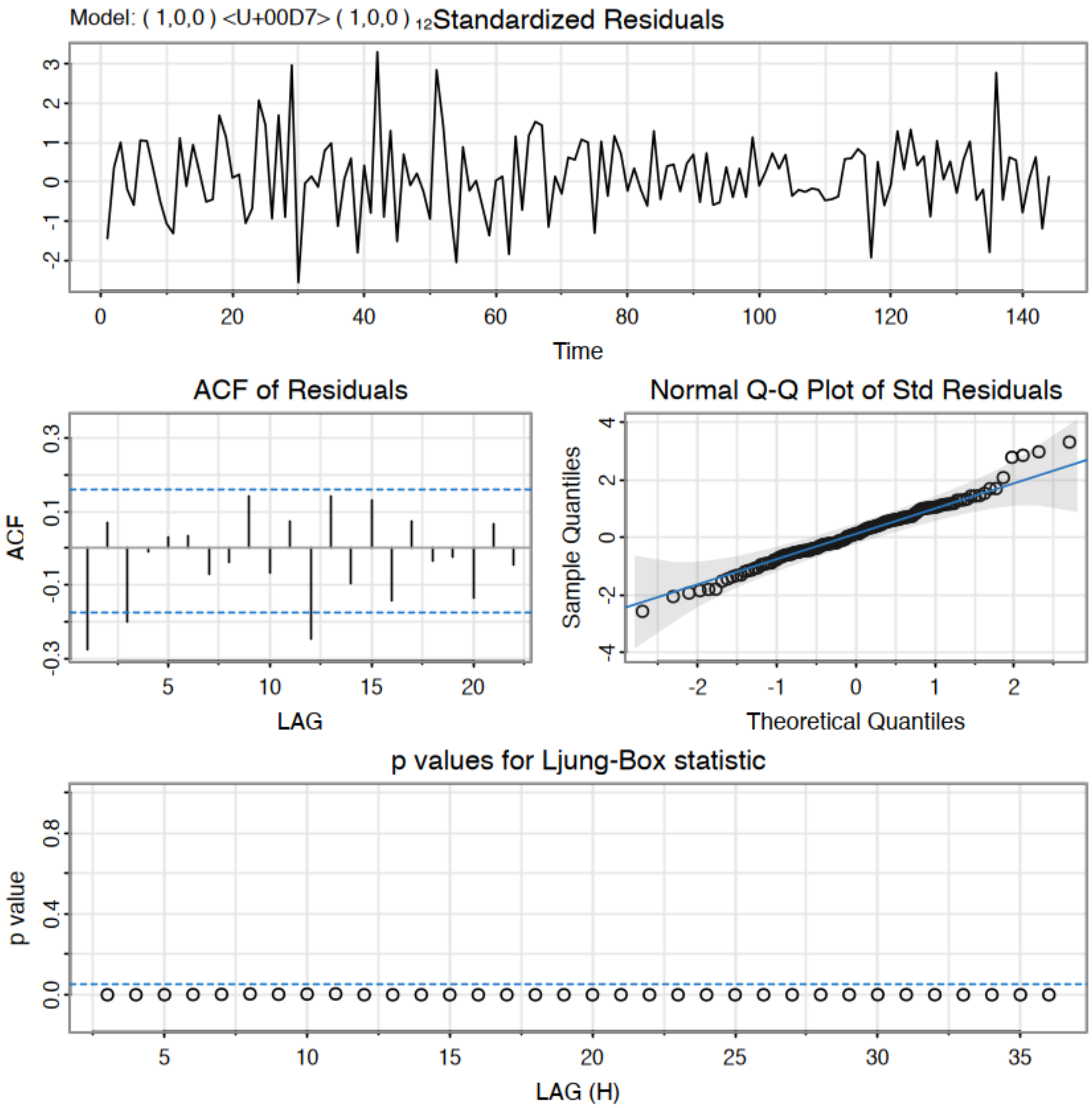
```
initial  value -2.333650
iter   2 value -2.982095
iter   3 value -2.988231
iter   4 value -3.017026
iter   5 value -3.018893
iter   6 value -3.020718
iter   7 value -3.021906
iter   8 value -3.024465
iter   9 value -3.024555
iter  10 value -3.024714
iter  11 value -3.024751
iter  12 value -3.024864
iter  13 value -3.024942
iter  14 value -3.024985
iter  15 value -3.025017
iter  16 value -3.025037
iter  17 value -3.025210
iter  18 value -3.025257
iter  19 value -3.025364
iter  20 value -3.025422
iter  21 value -3.025507
iter  22 value -3.025607
iter  23 value -3.025686
iter  24 value -3.025814
iter  25 value -3.025843
iter  26 value -3.025957
iter  27 value -3.026058
iter  28 value -3.026131
iter  29 value -3.026181
iter  30 value -3.026214
iter  31 value -3.026529
iter  32 value -3.026567
iter  33 value -3.026660
iter  34 value -3.026705
iter  35 value -3.026822
iter  36 value -3.027039
iter  37 value -3.027199
iter  38 value -3.027353
iter  39 value -3.027372
iter  40 value -3.027471
iter  41 value -3.027632
iter  42 value -3.027950
iter  43 value -3.028316
iter  44 value -3.028456
iter  45 value -3.029088
iter  46 value -3.029223
iter  47 value -3.029538
iter  48 value -3.029624
```

```
iter  49 value -3.029652
iter  50 value -3.029763
iter  51 value -3.029803
iter  52 value -3.030163
iter  53 value -3.030200
iter  54 value -3.030426
iter  55 value -3.030635
iter  56 value -3.031067
iter  57 value -3.031960
iter  58 value -3.032545
iter  59 value -3.033705
iter  60 value -3.033902
iter  61 value -3.034408
iter  62 value -3.034738
iter  63 value -3.034781
iter  64 value -3.034848
iter  65 value -3.034883
iter  66 value -3.035200
iter  67 value -3.035299
iter  68 value -3.035548
iter  69 value -3.035819
iter  70 value -3.036228
iter  71 value -3.036968
iter  72 value -3.037472
iter  73 value -3.038425
iter  74 value -3.038632
iter  75 value -3.038890
iter  76 value -3.039009
iter  77 value -3.039086
iter  78 value -3.039114
iter  79 value -3.039137
iter  80 value -3.039286
iter  81 value -3.039349
iter  82 value -3.039634
iter  83 value -3.040038
iter  84 value -3.040541
iter  85 value -3.040854
iter  86 value -3.041001
iter  87 value -3.041532
iter  88 value -3.041584
iter  89 value -3.041587
iter  90 value -3.041588
iter  91 value -3.041589
iter  92 value -3.041591
iter  93 value -3.041591
iter  94 value -3.041593
iter  95 value -3.041594
iter  96 value -3.041598
```

```
iter  97 value -3.041602
iter  98 value -3.041608
iter  99 value -3.041613
iter 100 value -3.041614
final  value -3.041614
stopped after 100 iterations
```

```
Warning message in arima(xdata, order = c(p, d, q), seasonal = list(order = c(P,
:
"possibile errore di convergenza: optim ha restituito codice = 1"
```

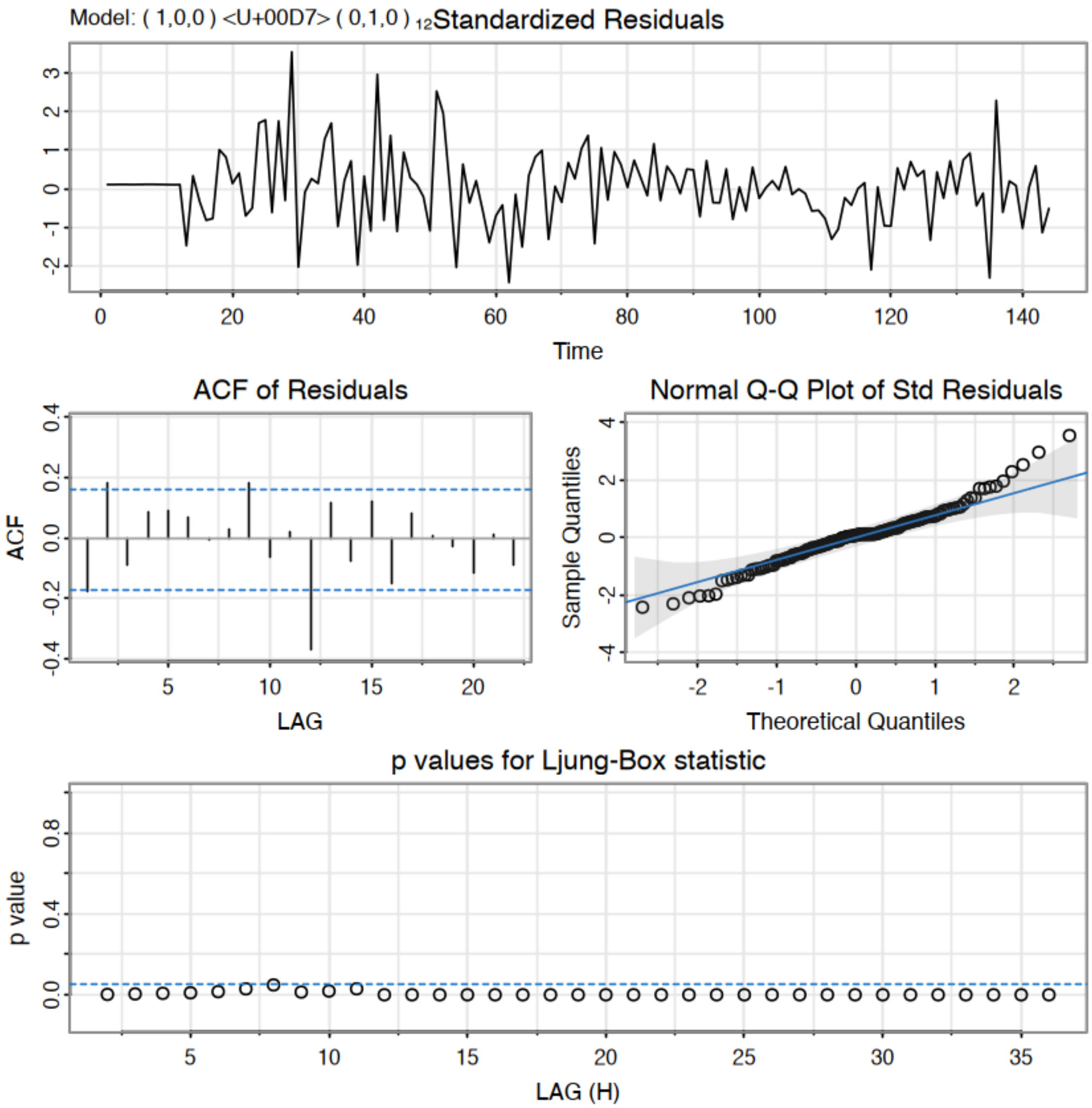
```
<><><><><><><><><><><><><><>

Coefficients:
      Estimate     SE t.value p.value
ar1     0.9488 0.0220 43.1396       0
sar1    0.9094 0.0271 33.5774       0
xmean   5.5325 0.4599 12.0306       0

sigma^2 estimated as 0.001921555 on 141 degrees of freedom

AIC = -3.189796  AICc = -3.188606  BIC = -3.107301
```



[8]:
```
mod1 <- sarima(
  data_ts,
  1, # AR
  0, # I
  0, # MA
  0, # AR_s
  1, # I_s
  0, # MA_s
  12
)
```

```
initial  value -2.795206
iter   2 value -3.163852
```

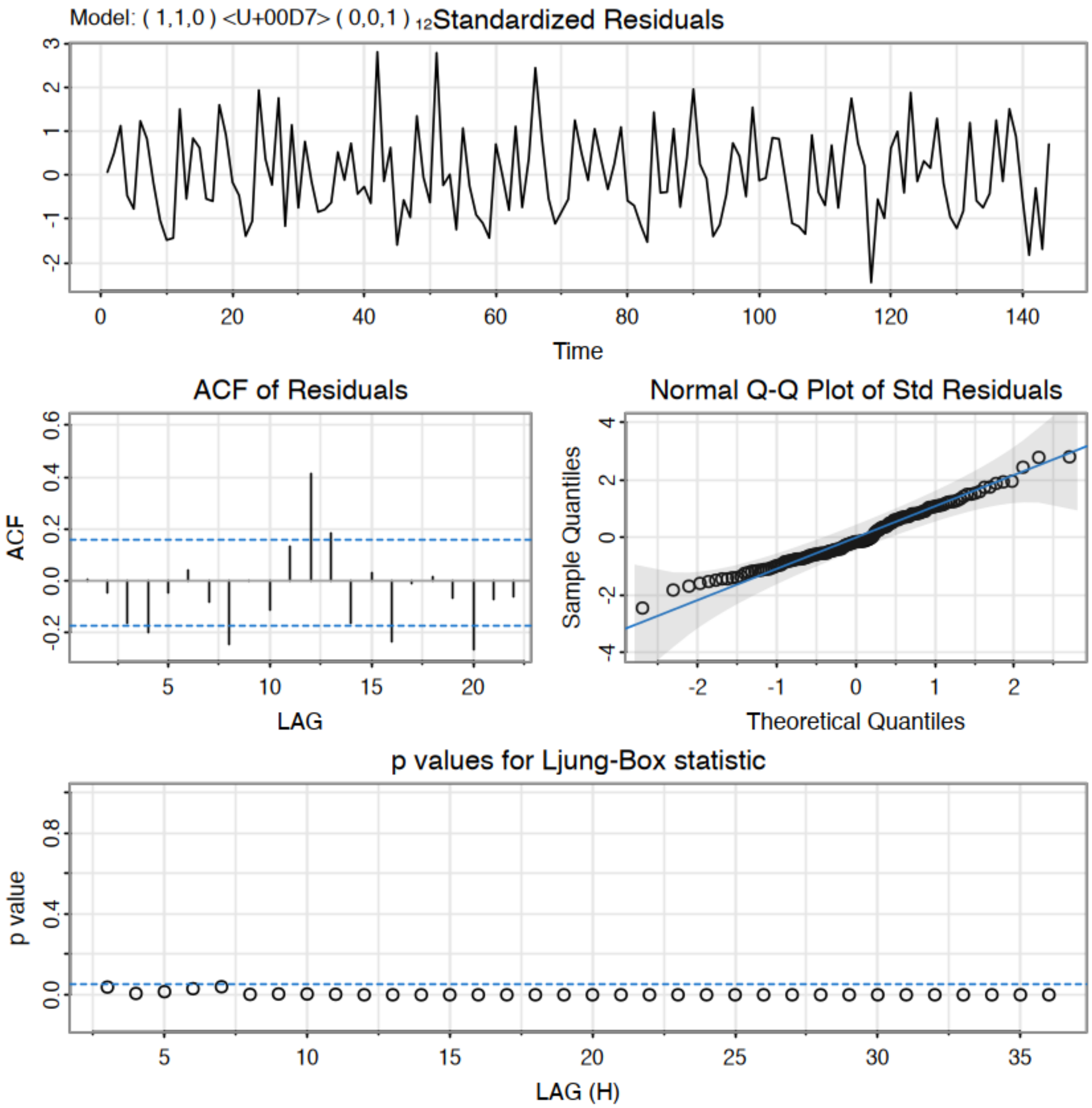
```
iter   3 value -3.163901
iter   4 value -3.163932
iter   5 value -3.163935
iter   6 value -3.163944
iter   7 value -3.163944
iter   8 value -3.163944
iter   8 value -3.163944
iter   8 value -3.163944
final  value -3.163944
converged
initial  value -3.155776
iter   2 value -3.155917
iter   3 value -3.156145
iter   4 value -3.156193
iter   5 value -3.156229
iter   6 value -3.156234
iter   7 value -3.156234
iter   8 value -3.156234
iter   8 value -3.156234
iter   8 value -3.156234
final  value -3.156234
converged
<><><><><><><><><><><><><><>

Coefficients:
         Estimate     SE t.value p.value
ar1        0.7255 0.0601 12.0676       0
constant   0.0097 0.0011  8.8101       0

sigma^2 estimated as 0.001803314 on 130 degrees of freedom

AIC = -3.429137  AICc = -3.428432  BIC = -3.363619
```



[9]:
```
mod2 <- sarima(
  data_ts,
  1, # AR
  1, # I
  0, # MA
  0, # AR_s
  0, # I_s
  1, # MA_s
  12
)
```

```
initial  value -2.239650
iter   2 value -2.556407
```

```
iter   3 value -2.594623
iter   4 value -2.603738
iter   5 value -2.604535
iter   6 value -2.604618
iter   7 value -2.604619
iter   7 value -2.604619
iter   7 value -2.604619
final  value -2.604619
converged
initial  value -2.604052
iter   2 value -2.604254
iter   3 value -2.605169
iter   4 value -2.605170
iter   5 value -2.605170
iter   5 value -2.605170
iter   5 value -2.605170
final  value -2.605170
converged
<><><><><><><><><><><><><><>

Coefficients:
         Estimate     SE t.value p.value
ar1        0.1053 0.0834  1.2629  0.2087
sma1       0.7401 0.0607 12.1952  0.0000
constant   0.0078 0.0112  0.6984  0.4861

sigma^2 estimated as 0.005107866 on 140 degrees of freedom

AIC = -2.316519  AICc = -2.315312  BIC = -2.233642
```



[10]:
```
mod1$fit$aic
mod2$fit$aic
```

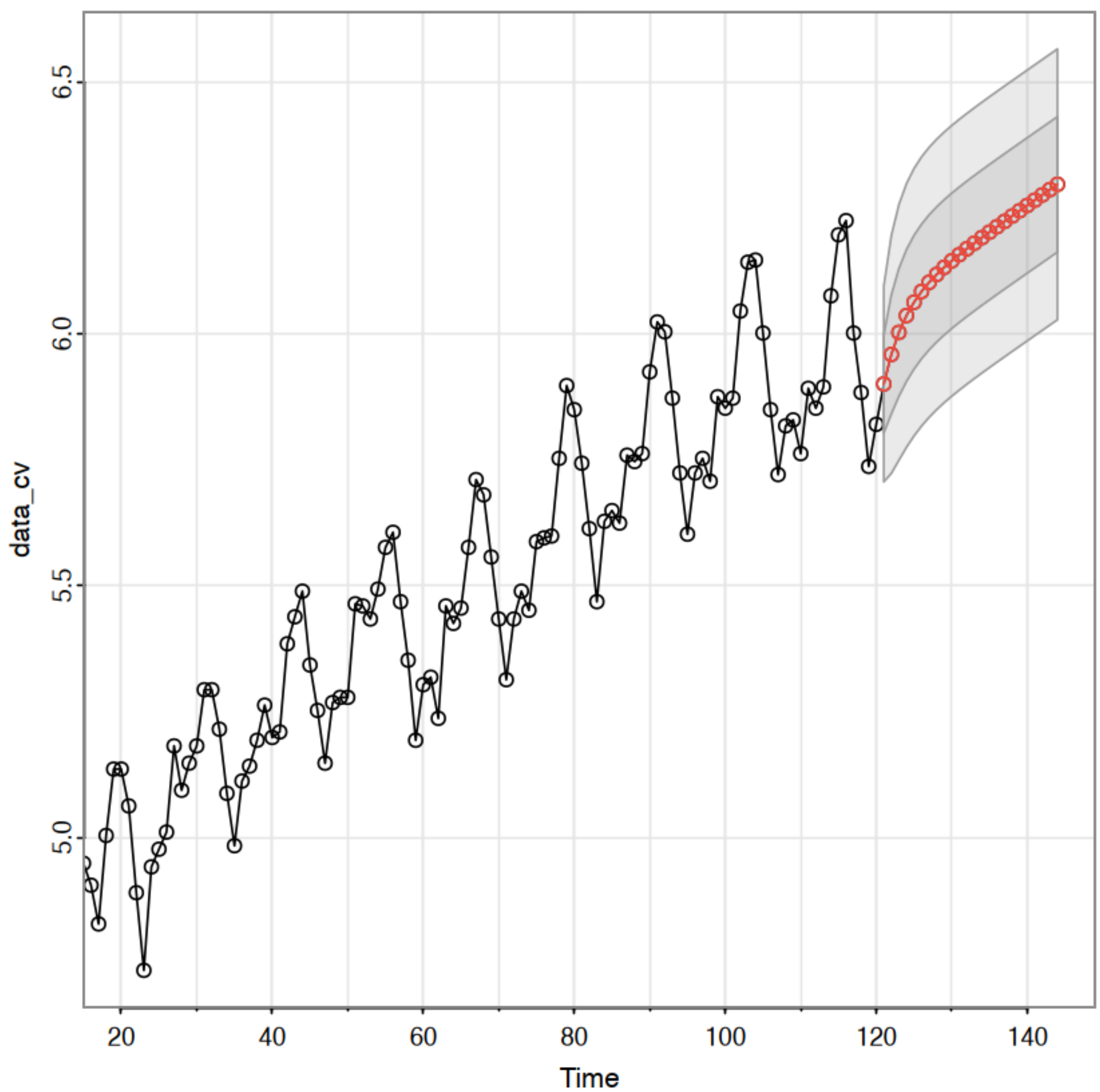
-452.646059544319

-331.262213554599

[11]:
```
n = length(data_ts)
n
data_cv <- ts(data_ts[1:120])
#data_ts

prev <- mod2 <- sarima.for(
  data_cv,
```

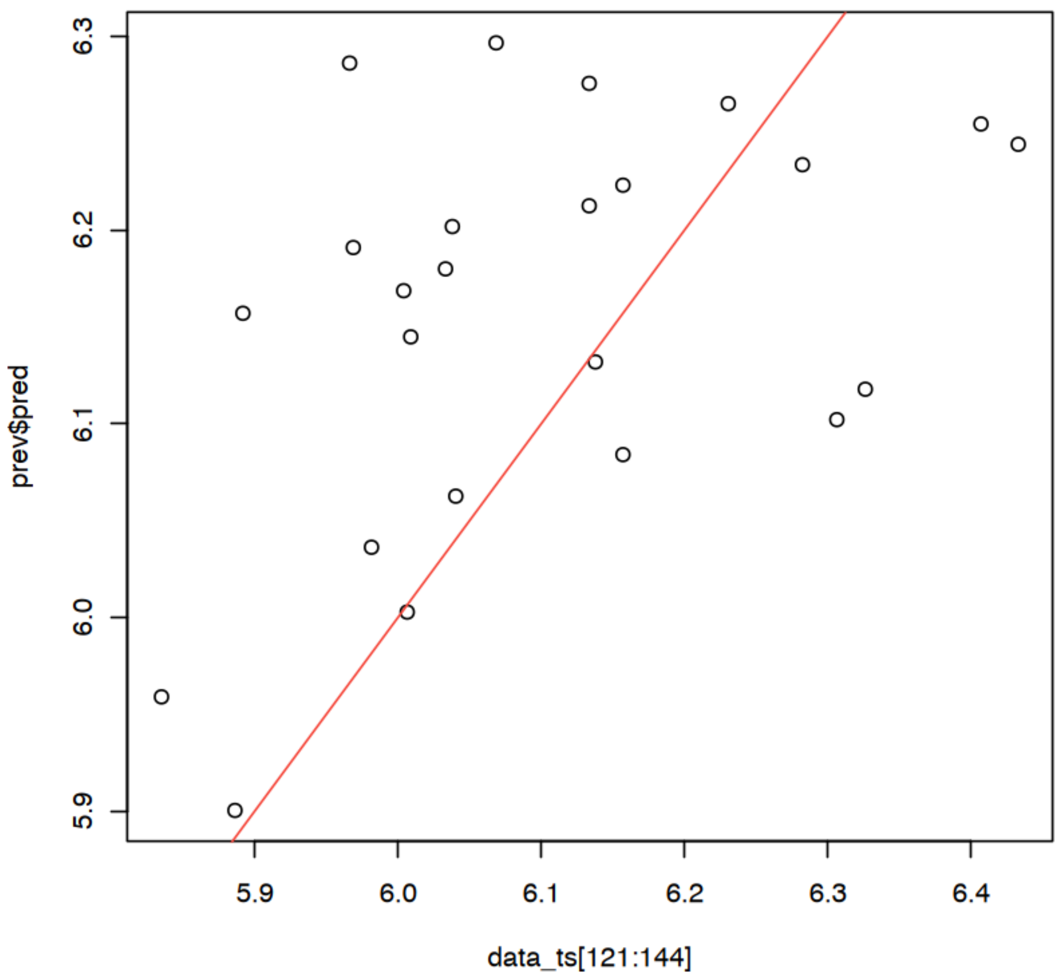
```
  xreg = data.frame(time = 1:120),
  newxreg = data.frame(time = 121:144),
  24,
  1, # AR
  0, # I
  0, # MA
  0, # AR_s
  0, # I_s
  0, # MA_s
  12
)
```

144



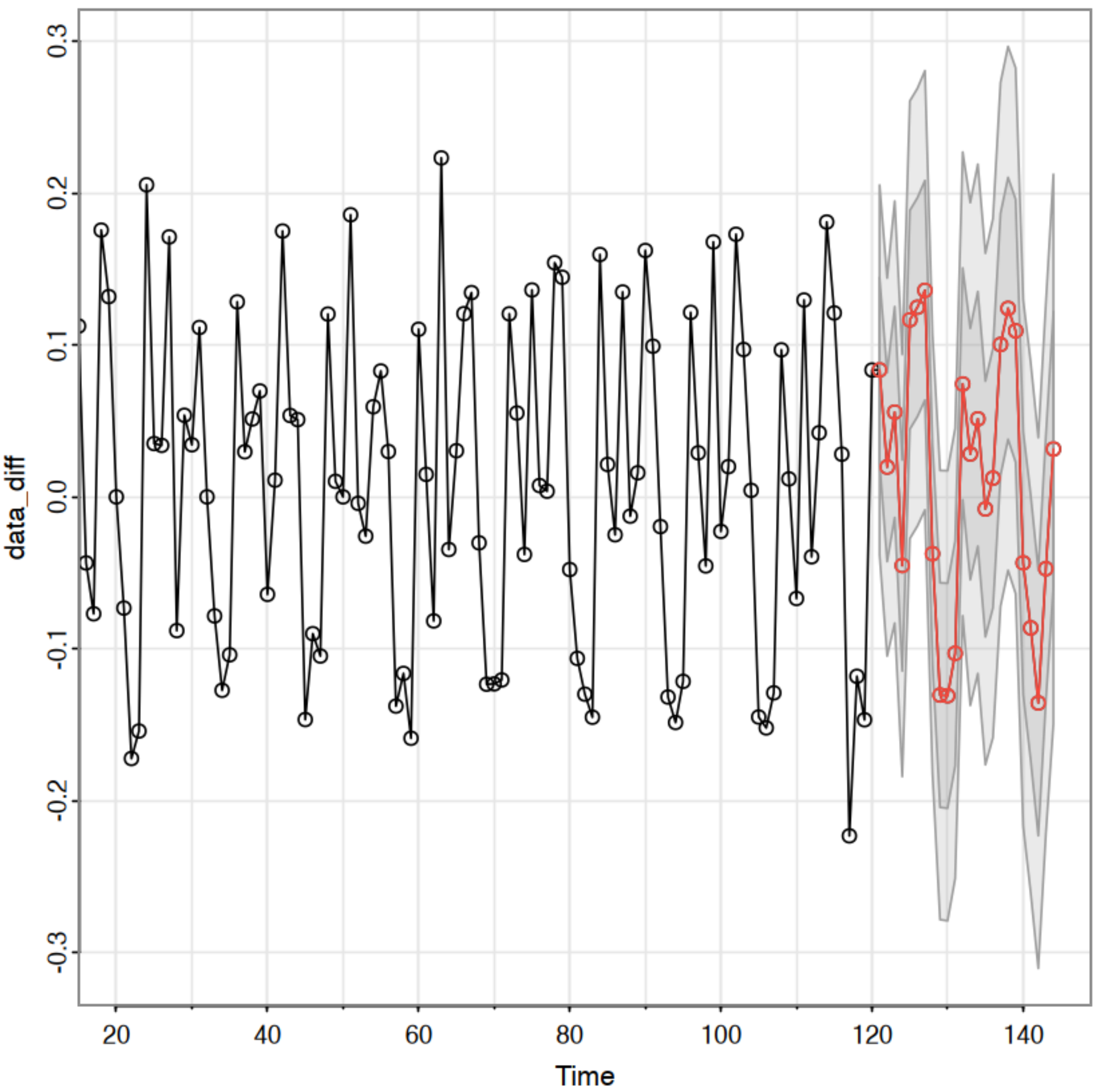
[12]:
```
plot(data_ts[121:144],prev$pred)
abline(a=0, b = 1, col=2)

mean((data_ts[121:144] - prev$pred)^2)
```

0.0237953299801296



[13]:
```
data_diff <- diff(data_cv)
#plot(data_diff)


prev <- mod2 <- sarima.for(
  data_diff,
```

```
  24,
  10, # AR
  0, # I
  0, # MA
  0, # AR_s
  0, # I_s
  0, # MA_s
  12
)
```



1 GP

Simuliamo un GP con

$$C() = \sigma^2 \exp(-\phi h)$$

o

$$C() = \sigma^2 \exp(-\frac{h}{\phi})$$

e media 0.

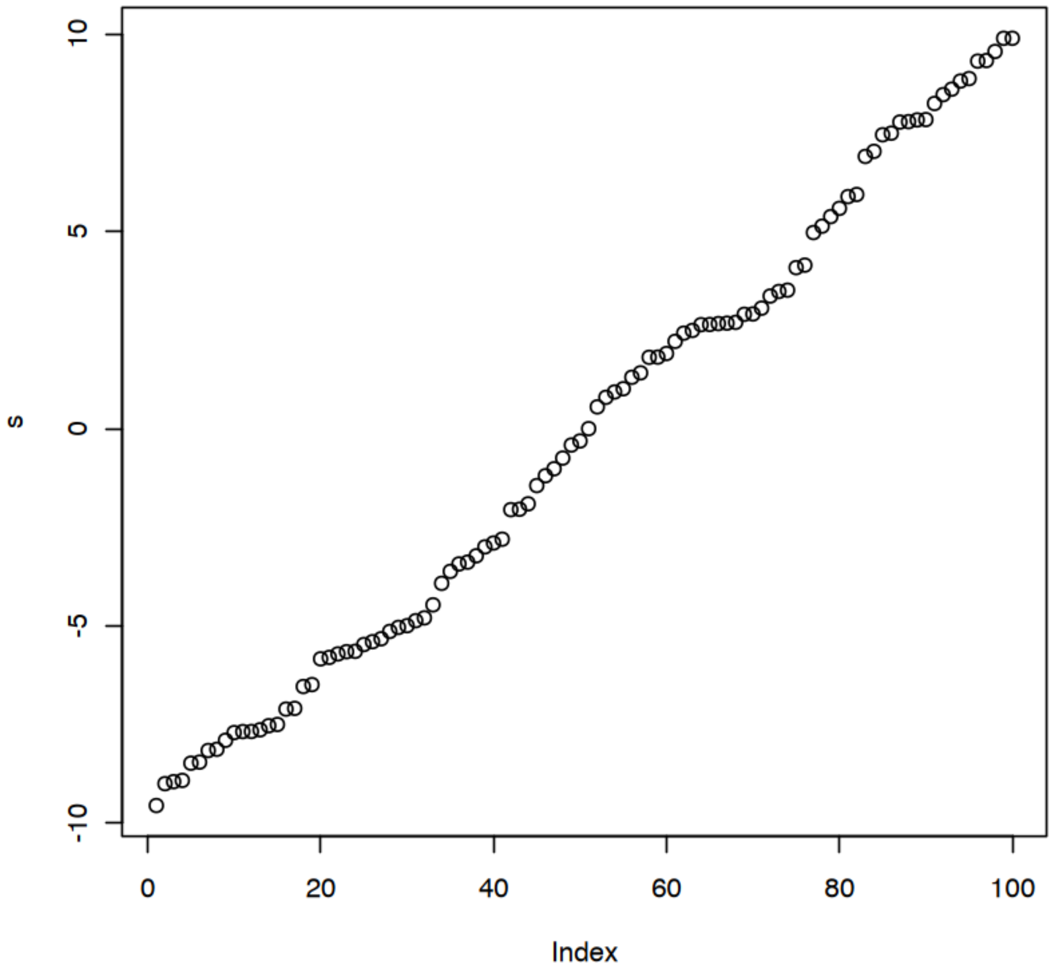
Asssumiamo di aver opsservato iol processo nei punti

$$s_1, \dots s_n$$

con

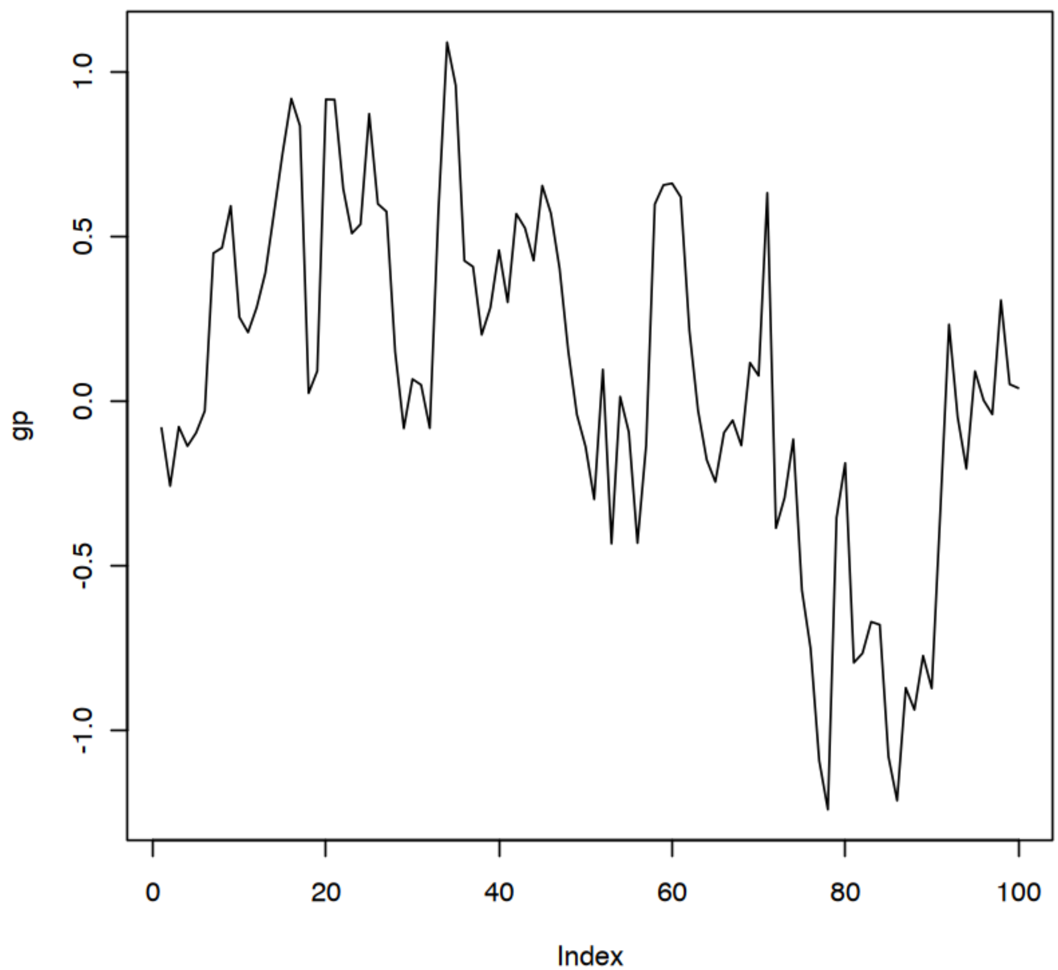
$$s_i \sim R$$

[14]:
```
n = 100
s = runif(n, -10,10)

s = s[order(s)]
plot(s)
```



[18]:
```
mean_function = rep(0, n)
Sigma = matrix(NA, nrow=n , ncol= n)
sigma2 = 1
phi = 3/10

for(i in 1:n)
{
  for(j in 1:n)
  {
    Sigma[i, j] = sigma2 * exp(-phi * abs(s[i] - s[j]))

  }
}
```

```r
Sigma_chol = t(chol(Sigma))


gp <- mean_function + Sigma_chol%*% matrix(rnorm(n,0,1), ncol=1)

plot(gp, type = "l")
```

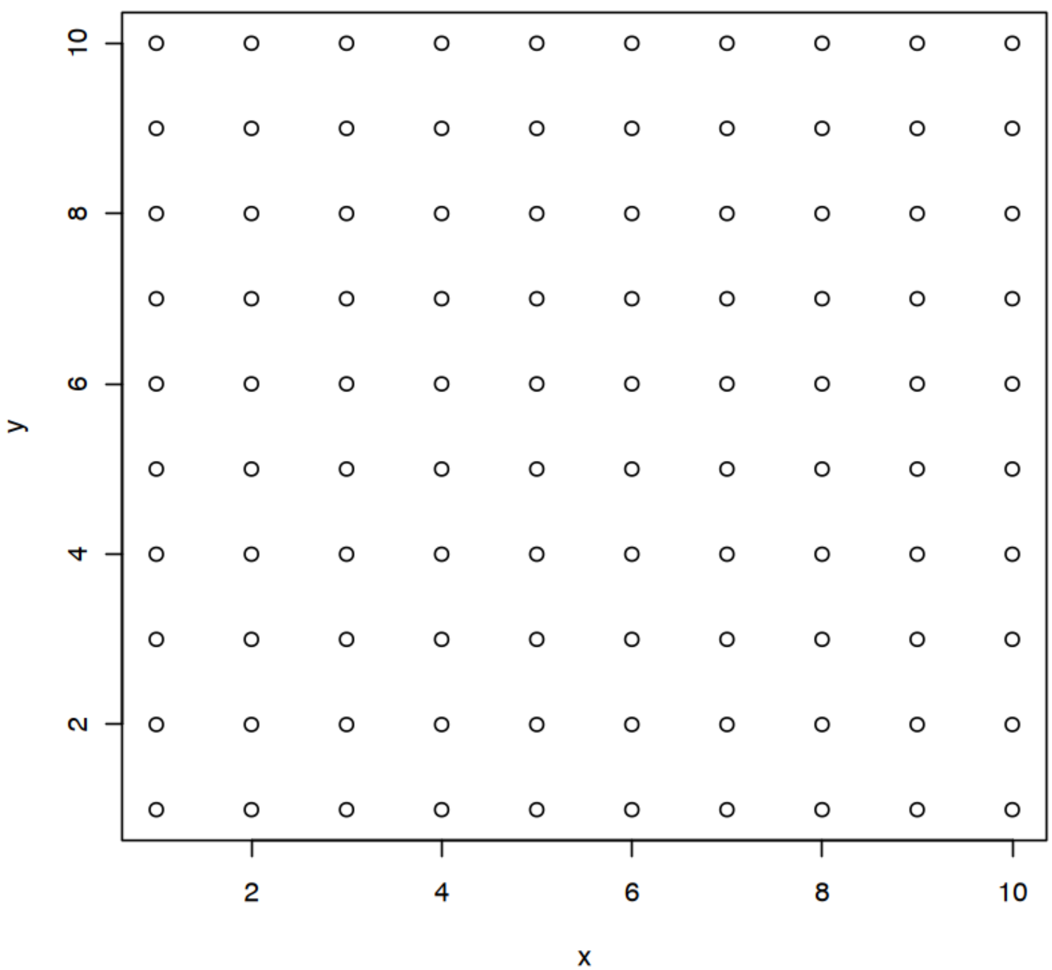


[ ]:
```

```

[ ]:
```

```

Simulazione di un processo gaussiano definito su $\mathcal{R}^2$ con funzione di covarianza esponenziale

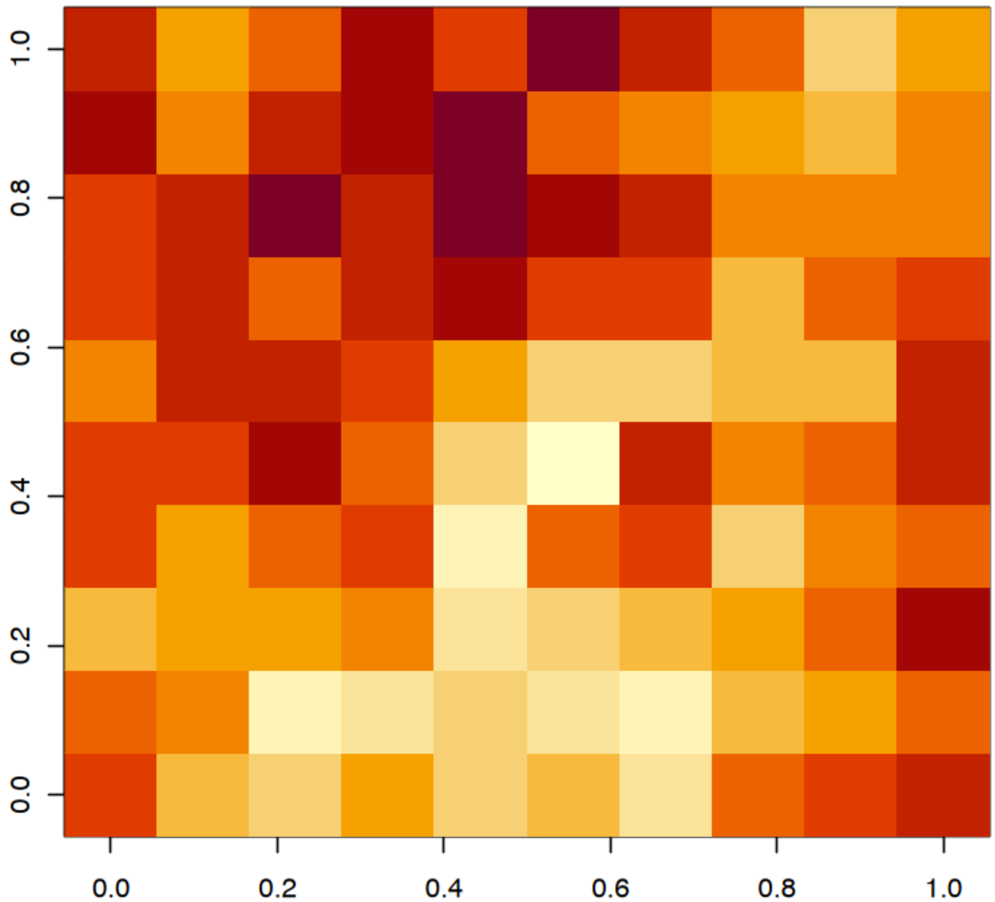
[19]:
```
n_grid = 10
x_grid = seq(1, n_grid, by = 1)
y_grid = seq(1, n_grid, by = 1)


data_grid = data.frame(x = rep(x_grid, times = n_grid), y = rep(y_grid, each =␣
 ↪n_grid))

plot(data_grid)
```



[21]:
```
n = n_grid^2
mu = rep(0, n)
```

```
sigma2 = 0.2
phi = 3/5
Sigma = matrix(NA, ncol=n, nrow=n)
dist_mat = as.matrix(dist(data_grid))
for(i in 1:n)
{
  for(j in 1:n)
  {
    Sigma[i,j] = sigma2 * exp(- phi * sqrt((data_grid[i, 1] - data_grid[j,
 ↪1])^2 + (data_grid[i, 2] - data_grid[j, 2])^2))
  }
}



gp2 <- mu + t(chol(Sigma))%*%matrix(rnorm(n,0,1), ncol=1)
```

[24]:
```
image(matrix(gp2, ncol=n_grid))
```


0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0