Online Resource 01 - Supplementary appendices

How does the temperature vary over time? Evidence on the Stationary and Fractal nature of Temperature Fluctuations

John Dagsvik, Mariachiara Fortuna, Sigmund H. Moen

Affiliations:

John K. Dagsvik, Statistics Norway, Research Department;

Mariachiara Fortuna, freelance statistician, Turin;

Sigmund Hov Moen, Westerdals Oslo School of Arts, Communication and Technology.

Corresponding author:

John K. Dagsvik, E-mail: john.dagsvik@ssb.no

Mariachiara Fortuna, E-mail: mariachiara.fortuna1@gmail.com (reference for code and analysis)

```r
# require(tempFGN)
require(knitr)
require(dplyr)
require(ggplot2)
require(tidyr)
require(tempFGN)
```

```r
# DATA PATH
data_final_path <- file.path("data","final")
data_supporting_path <- file.path("data", "supporting")
data_moberg_path <- file.path("data", "moberg")

# OUTPUT PATH
output_supporting_path <- file.path("output", "supporting")
output_table_path <- file.path("output", "table")
output_figure_path <- file.path("output", "figure")
output_temporary_path <- file.path("output", "temporary")
output_manipuated_path <- file.path("output", "manipulated")
```

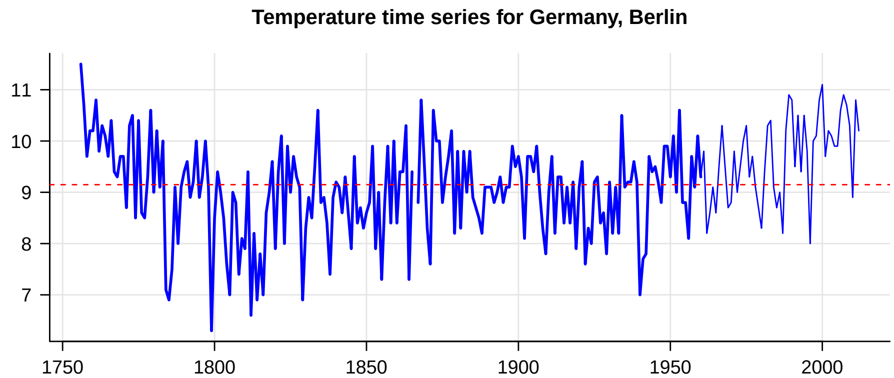
```r
# ACCESS TO SELECTED TIME SERIES
selected <- read.csv(file.path(data_supporting_path, "T0.SelInfo.csv"), sep=";", dec=",")
country_sel <- selected$Country
station_sel <- selected$Station
njs <- nrow(selected)
data_dir_sel <- file.path(data_final_path, country_sel, paste0(station_sel, ".txt"))
stationame_sel <- paste0(country_sel,", ",station_sel)

# ACCESS TO ALL THE TIME SERIES
all <- read.csv(file.path(data_supporting_path, "T0.TempInfo.csv"), sep=";", dec=",")
country_all <- all$Country
station_all <- all$Station
nja <- nrow(all)
data_dir_all <- file.path(data_final_path, country_all, paste0(station_all, ".txt"))
stationame_all <- paste0(country_all,", ",station_all)

# ACCESS TO MOBERG DATA
moberg <- read.table(file.path(data_moberg_path, "Moberg data.txt"),
                     header = T, na.strings = 99)
Year_m <- moberg[, 1]
Xj_m <- moberg[, 2]
Zj_m <- scale(Xj_m)
Yj_m <- cumsum(Zj_m)
```

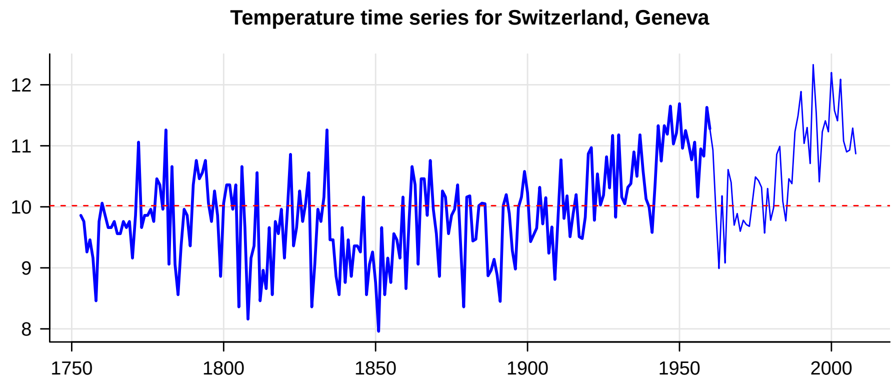
APPENDIX B - SUMMARY DATA INFORMATION

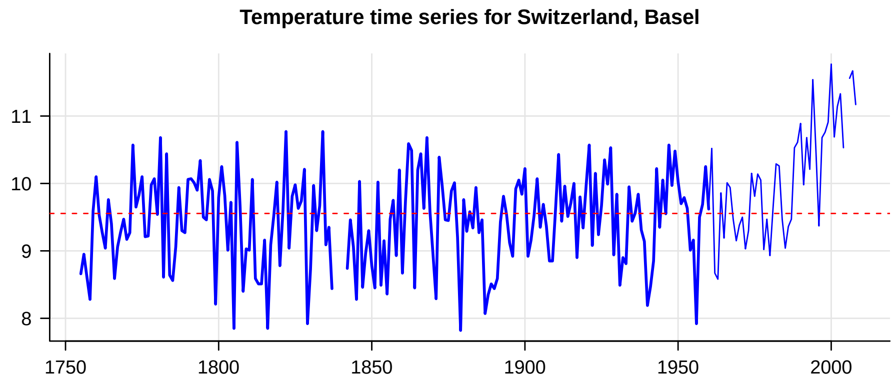
Figure B1. Plots of temperature series for 9 selected cities

```
pdf(file.path(output_figure_path, "FB1_selected_temperature_plot.pdf"),
     paper="a4", width=7, height=10)
par(mfrow=c(3,1))
for (j in 1:njs){
        data <- read.delim(data_dir_sel[j], header=F, na.strings=99)
        Xj <- data[,14]
        Year <- data[,1]
        temperaturePlot(Xj=Xj, Year=Year, break.val=1960,
                        main=paste("Temperature time series for",
                                   stationame_sel[j]), cex.axis=1.2,
                        cex.lab=1.2, cex.main=1.4)
        }
dev.off()
```

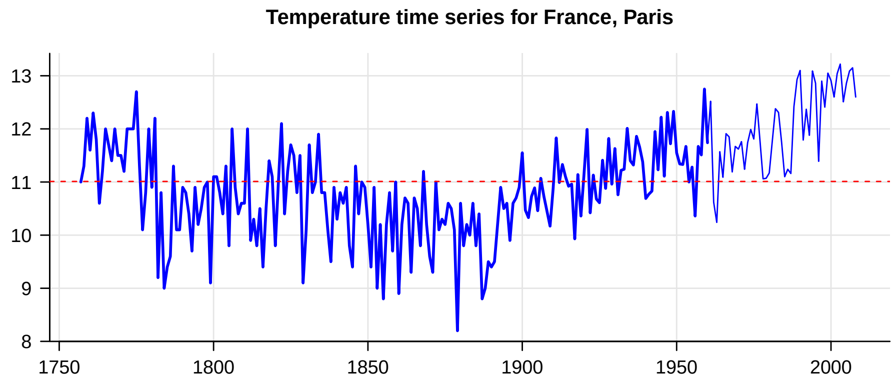
```
include_graphics(file.path(output_manipuated_path, "FB1_selected_temperature_plot",
                           "p01.pdf"))
```

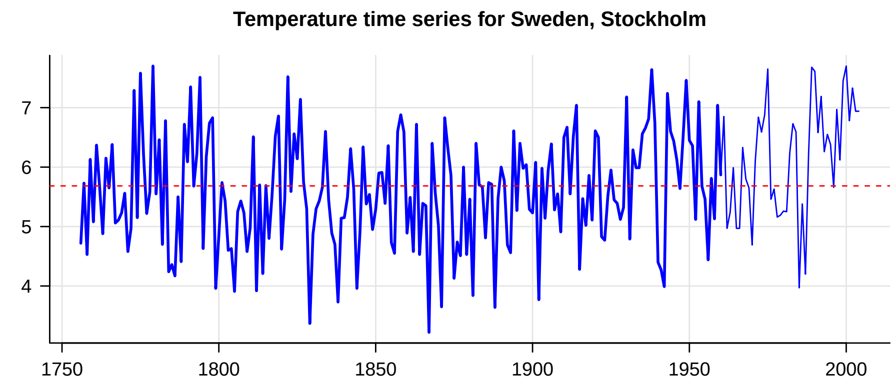


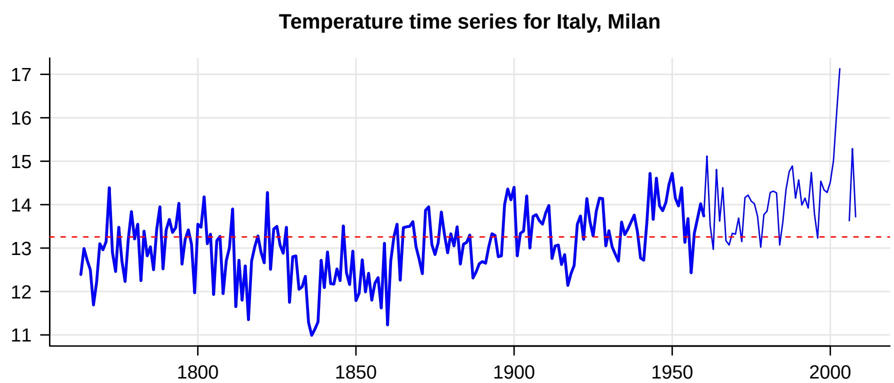





```
include_graphics(file.path(output_manipuated_path, "FB1_selected_temperature_plot",
                              "p02.pdf"))
```

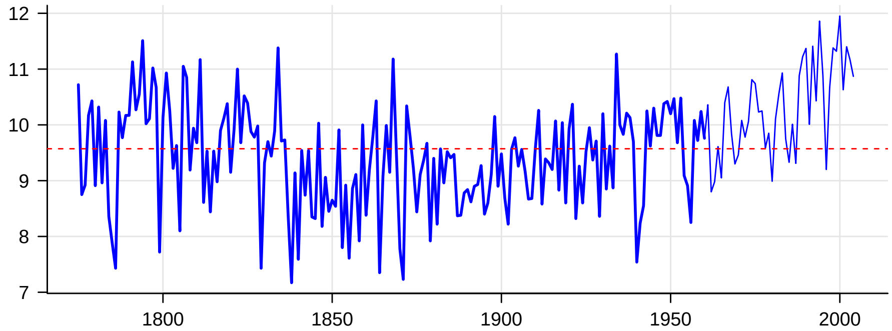




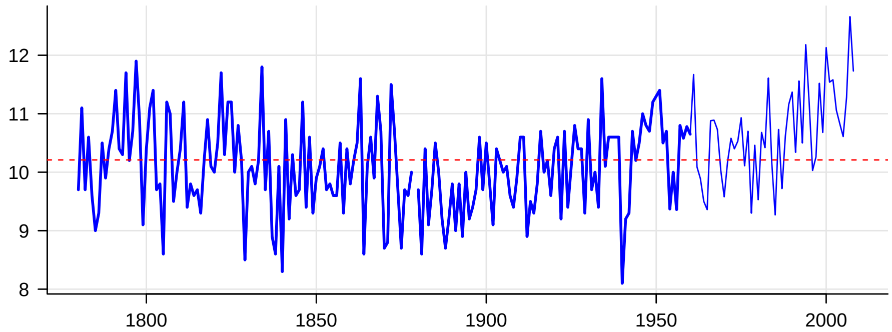



```
include_graphics(file.path(output_manipuated_path, "FB1_selected_temperature_plot",
                            "p03.pdf"))
```

Temperature time series for Czech Republic, Prague



Temperature time series for Hungary, Budapest



Temperature time series for Denmark, Copenhagen

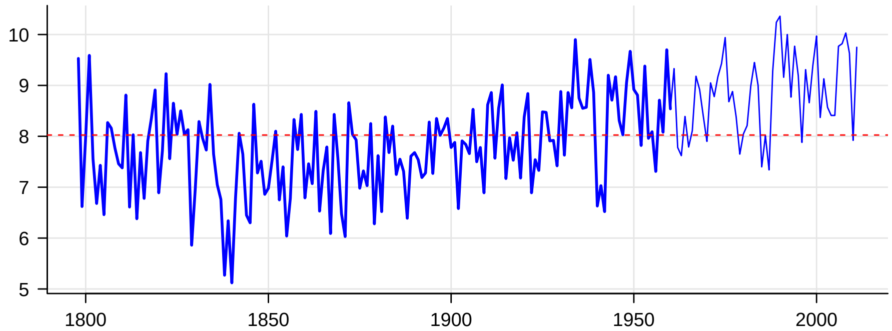


Table B1. Summary information about data

```
all_info <- NULL
for (j in 1:nja){
    data <- read.delim(data_dir_all[j], header=F, na.strings=99)
    dataM <- monthlyAdj(data, scale = T)
    # Compute information for a given station
    station_info <- dataM %>%
    summarize(Station = stationame_all[j],
            First_Year = min(year(Time)),
            Last_Year  = max(year(Time)),
            Years = Last_Year-First_Year+1,
            Nonmiss_Months = n(),
            Full_Lenght = 12*Years + month(max(Time)) - month(min(Time)),
            Missing_Months = Full_Lenght - Nonmiss_Months # Check it
            )
# Create the information data.frame
    all_info <- rbind(all_info, station_info)
}

# Write table
write.csv(all_info, file.path(output_table_path, "TB1_all_info_table.csv"), row.names=F)
```

```
TB1.allInfo <- read.csv(file.path(output_table_path, "TB1_all_info_table.csv"))%>%
  select(-Full_Lenght)
kable(TB1.allInfo,  digits=3, align='c',
      col.names = c(" Weather station", "First year", "Last year", "Years",
                    "Nonmissing months", "Missing months"))
```

Weather station	First year	Last year	Years	Nonmissing months	Missing months
Argentina, Buenos Aires	1856	2006	151	1781	36
Australia, Adelaide	1881	2012	132	1567	27
Australia, Alice Springs	1881	2012	132	1564	30
Australia, Cap Otway	1865	2012	148	1731	56
Austria, Kremsmunster	1876	2009	134	1601	15
Austria, Vienna	1855	2009	155	1829	38
Belgium, Uccle	1833	2008	176	2108	11
Canada, Winnipeg	1881	2012	132	1575	19
Croatia, Zagreb	1861	2008	148	1765	11
Czech Republic, Prague	1775	2005	231	2764	11
Denmark, Copenhagen	1798	2011	214	2568	11
Denmark, Vestervig	1875	2012	138	1648	11
Egypt, Alexandria	1870	1990	121	1395	66
France, Nantes	1851	2009	159	1893	20
France, Paris	1757	2009	253	3030	11
Germany, Berlin	1756	2012	257	3083	12
Germany, Hohenpeissenberg	1781	2012	232	2782	13
Germany, Karlsruhe	1876	2008	133	1586	20
Greece, Athens	1858	2009	152	1814	14
Greenland, Illulisat	1873	2012	140	1674	16
Greenland, Ivittuut	1873	1960	88	1056	11
Hungary, Budapest	1780	2009	230	2753	13
Iceland, Djupivogur	1873	2009	137	1635	14

Weather station	First year	Last year	Years	Nonmissing months	Missing months
Iceland, Reykjavik	1870	2012	143	1711	11
India, Agra	1881	1987	107	1269	21
India, Allahabad	1881	2012	132	1517	70
India, Bombay	1881	2012	132	1569	18
India, Indore	1881	2012	132	1569	18
India, Madras	1881	2012	132	1569	18
India, Nagpur	1881	2011	131	1565	16
Israel, Jerusalem	1861	2004	144	1660	79
Italy, Bologna	1814	2009	196	2334	27
Italy, Milan	1763	2009	247	2943	30
Japan, Hiroshima	1881	2005	125	1489	22
Japan, Nagasaki	1881	2012	132	1582	11
Japan, Tokyo	1876	2012	137	1642	11
Kazakhstan, Kazalinsk	1881	1990	110	1307	16
Luxembourg, Luxembourg	1838	2008	171	2036	27
New Zealand, Wellington	1864	1989	126	1503	13
Norway, Andoya	1868	2012	145	1739	12
Norway, Bergen	1858	2012	155	1860	11
Norway, Bodo	1868	2012	145	1740	11
Norway, Dombas	1865	2012	148	1773	14
Norway, Karasjok	1876	2012	137	1644	11
Norway, Mandal	1861	2007	147	1760	11
Norway, Oksoy Lighthouse	1870	2012	143	1716	11
Norway, Ona	1868	2012	145	1717	34
Norway, Oslo	1816	2012	197	2364	11
Norway, Roros	1871	2012	142	1704	11
Norway, Tromso	1868	2012	145	1740	11
Norway, Utsira	1868	2012	145	1740	11
Norway, Vardo	1858	2008	151	1809	11
Pakistan, Lahore	1876	2012	137	1631	16
Portugal, Lisbon	1881	2009	129	1539	20
Romania, Sulina	1881	2009	129	1530	24
Russia, Archangelsk	1881	2012	132	1580	14
Russia, Sort	1881	1990	110	1294	29
Russia, St Petersburg	1881	2009	129	1543	12
Spain, Gibraltar	1852	2010	159	1850	65
Sweden, Bromma	1756	2011	256	3067	12
Sweden, Stockholm	1756	2004	249	2988	11
Sweden, Tullinge	1756	2011	256	3049	30
Sweden, Uppsala	1722	2012	291	3334	169
Switzerland, Basel	1755	2009	255	3014	52
Switzerland, Geneva	1753	2009	257	3077	13
UK, Aberdeen	1881	2012	132	1582	11
UK, Belfast	1881	2012	132	1576	12
UK, Cambridge	1871	2012	142	1702	11
UK, Durham	1847	2012	166	1989	12
UK, Edinbourg	1785	1993	209	2507	12
UK, London	1841	2004	164	1944	31
UK, Plymouth	1865	1993	129	1543	16
USA, Atlanta	1881	2012	132	1582	12
USA, Bismarck	1881	2012	132	1582	12
USA, Boise	1881	2012	132	1583	11

Weather station	First year	Last year	Years	Nonmissing months	Missing months
USA, Boston	1881	2012	132	1583	11
USA, Chattanooga	1881	2012	132	1583	11
USA, Cincinatti	1881	2012	132	1581	13
USA, Columbus	1881	2012	132	1583	11
USA, Concord	1881	2012	132	1583	11
USA, Des Moines	1881	2012	132	1583	11
USA, Detroit	1881	2012	132	1583	11
USA, Dodge City	1881	2012	132	1582	12
USA, Fargo	1881	2012	132	1583	11
USA, Galveston	1881	2012	132	1574	20
USA, Indianapolis	1881	2012	132	1583	11
USA, Jacksonville	1881	2012	132	1581	13
USA, Knoxville	1881	2012	132	1582	12
USA, Las Vegas	1875	1993	119	1428	11
USA, Madison	1881	2012	132	1583	11
USA, Marquette	1881	2012	132	1552	42
USA, Milwaukee	1881	2012	132	1583	11
USA, Mobile	1881	2012	132	1583	11
USA, Nashville	1881	2012	132	1582	12
USA, New Orleans	1874	2005	132	1580	11
USA, New York	1822	2007	186	2225	11

APPENDIX C - ESTIMATION AND TEST RESULTS

Table C1. Estimation results when using the characteristic function estimator and the Whittle method. Monthly data.

```
# Parameters table
Parameters <- matrix(0, nrow=nja, ncol=3)
colnames(Parameters) <- c("Hc", "Hw", "SE_Hw")
rownames(Parameters) <- stationame_all

pb <- txtProgressBar() # Start the progress bar

for (j in 1:nja){

  setTxtProgressBar(pb, j/nja) # Update the progress bar

  # Data reading
  data <- read.delim(data_dir_all[j], header=F, na.strings=99)
  Zm <- monthlyAdj(data, scale=T)$Zm

  # Estimation
  whittle <- WhittleEst(Zm)
  Parameters[j,1] <- estim.cf.H(Yj=Zm, FBM=F)
  Parameters[j,2] <- whittle$coefficients[1]
  Parameters[j,3] <- whittle$coefficients[2]
}
close(pb) # close the progress bar

# Saving data to csv
Parameters <- as.data.frame(Parameters)
write.csv(Parameters, file.path(output_table_path,
                                "TC1_monthly_H_all.csv"))
```

```
TC1.monthlyHAll <- read.csv(file.path(output_table_path, "TC1_monthly_H_all.csv"))
kable(TC1.monthlyHAll, digits=3, align='c', escape = F,
      col.names = c(" Weather station", "$H_c$", "$H_w$", "$SE(H_w)$"))
```

Weather station	H_c	H_w	$SE(H_w)$
Argentina, Buenos Aires	0.785	0.713	0.016
Australia, Adelaide	0.696	0.662	0.016
Australia, Alice Springs	0.700	0.683	0.017
Australia, Cap Otway	0.803	0.744	0.016
Austria, Kremsmunster	0.655	0.651	0.016
Austria, Vienna	0.684	0.659	0.015
Belgium, Uccle	0.660	0.643	0.014
Canada, Winnipeg	0.654	0.660	0.016
Croatia, Zagreb	0.654	0.650	0.015
Czech Republic, Prague	0.684	0.670	0.012
Denmark, Copenhagen	0.755	0.758	0.013
Denmark, Vestervig	0.725	0.763	0.016
Egypt, Alexandria	0.773	0.798	0.018
France, Nantes	0.643	0.643	0.015
France, Paris	0.733	0.672	0.012
Germany, Berlin	0.664	0.662	0.012
Germany, Hohenpeissenberg	0.617	0.605	0.012

Weather station	H_c	H_w	$SE(H_w)$
Germany, Karlsruhe	0.642	0.629	0.016
Greece, Athens	0.682	0.698	0.015
Greenland, Illulisat	0.738	0.725	0.016
Greenland, Ivittuut	0.782	0.751	0.020
Hungary, Budapest	0.627	0.645	0.012
Iceland, Djupivogur	0.762	0.736	0.016
Iceland, Reykjavik	0.731	0.711	0.016
India, Agra	0.731	0.753	0.019
India, Allahabad	0.699	0.694	0.017
India, Bombay	0.783	0.788	0.017
India, Indore	0.734	0.709	0.017
India, Madras	0.751	0.753	0.017
India, Nagpur	0.697	0.708	0.017
Israel, Jerusalem	0.685	0.699	0.016
Italy, Bologna	0.702	0.698	0.014
Italy, Milan	0.724	0.709	0.012
Japan, Hiroshima	0.731	0.733	0.017
Japan, Nagasaki	0.738	0.715	0.017
Japan, Tokyo	0.795	0.744	0.016
Kazakhstan, Kazalinsk	0.609	0.655	0.018
Luxembourg, Luxembourg	0.675	0.658	0.014
New Zealand, Wellington	0.774	0.752	0.017
Norway, Andoya	0.723	0.725	0.016
Norway, Bergen	0.678	0.687	0.015
Norway, Bodo	0.680	0.698	0.016
Norway, Dombas	0.633	0.664	0.015
Norway, Karasjok	0.655	0.679	0.016
Norway, Mandal	0.682	0.724	0.016
Norway, Oksoy Lighthouse	0.719	0.771	0.016
Norway, Ona	0.711	0.749	0.016
Norway, Oslo	0.693	0.724	0.014
Norway, Roros	0.667	0.695	0.016
Norway, Tromso	0.670	0.690	0.016
Norway, Utsira	0.735	0.768	0.016
Norway, Vardo	0.765	0.751	0.016
Pakistan, Lahore	0.649	0.693	0.016
Portugal, Lisbon	0.769	0.710	0.017
Romania, Sulina	0.653	0.699	0.017
Russia, Archangelsk	0.675	0.661	0.016
Russia, Sort	0.640	0.694	0.018
Russia, St Petersburg	0.697	0.696	0.017
Spain, Gibraltar	0.773	0.765	0.015
Sweden, Bromma	0.694	0.736	0.012
Sweden, Stockholm	0.681	0.721	0.012
Sweden, Tullinge	0.672	0.727	0.012
Sweden, Uppsala	0.690	0.718	0.011
Switzerland, Basel	0.625	0.622	0.012
Switzerland, Geneva	0.693	0.667	0.012
UK, Aberdeen	0.691	0.704	0.017
UK, Belfast	0.650	0.665	0.016
UK, Cambridge	0.678	0.673	0.016
UK, Durham	0.698	0.686	0.015

Weather station	H_c	H_w	$SE(H_w)$
UK, Edinbourg	0.644	0.670	0.013
UK, London	0.721	0.686	0.015
UK, Plymouth	0.624	0.676	0.017
USA, Atlanta	0.632	0.641	0.016
USA, Bismarck	0.655	0.640	0.016
USA, Boise	0.654	0.656	0.016
USA, Boston	0.693	0.670	0.016
USA, Chattanooga	0.637	0.647	0.016
USA, Cincinatti	0.656	0.645	0.016
USA, Columbus	0.629	0.631	0.016
USA, Concord	0.687	0.662	0.016
USA, Des Moines	0.626	0.632	0.016
USA, Detroit	0.659	0.653	0.016
USA, Dodge City	0.626	0.611	0.016
USA, Fargo	0.656	0.655	0.016
USA, Galveston	0.662	0.688	0.017
USA, Indianapolis	0.611	0.622	0.016
USA, Jacksonville	0.608	0.651	0.016
USA, Knoxville	0.624	0.630	0.016
USA, Las Vegas	0.643	0.647	0.017
USA, Madison	0.641	0.647	0.016
USA, Marquette	0.688	0.686	0.017
USA, Milwaukee	0.689	0.676	0.016
USA, Mobile	0.617	0.651	0.016
USA, Nashville	0.581	0.603	0.016
USA, New Orleans	0.696	0.695	0.017
USA, New York	0.745	0.699	0.014

Due to the fact that the monthly time series are quite long the estimates of the Hurst parameter are quite precise. From Table C1 we note that the difference between the characteristic function estimates and the Whittle estimates of the Hurst parameter are only significantly different in a few cases.

Table C2. Estimates and test statistics based on annual data

```
Q_Hc <- NULL
Q_Hw <- NULL
Hc_vec <- NULL
Hw_vec <- NULL
Hw_se_vec <- NULL
for (j in 1:nja){
  data <- read.delim(data_dir_all[j], header=F, na.strings=99)
  Xj <- data[,14]
  Zj <- scale(Xj[!is.na(Xj)])
  TT <- length(Zj)
  whittle <- WhittleEst(Zj)
  Hc <- estim.cf.H(Yj=Zj, FBM=F)
  Hw <- whittle$coefficients[1]
  Hw_se <- whittle$coefficients[2]
  Q_Hc <- c(Qstat(Zj, H=Hc, TT=TT), Q_Hc)
  Q_Hw <- c(Qstat(Zj, H=Hw, TT=TT), Q_Hw)
```

```r
  Hc_vec <- c(Hc_vec, Hc)
  Hw_vec <- c(Hw_vec, Hw)
  Hw_se_vec <- c(Hw_se_vec, Hw_se)
}

#star <- ifelse(Q>1.96|Q<(-1.96), "*", "")
annual_Qt_all <- data.frame(City=stationame_all, Hc=Hc_vec, Q_Hc=Q_Hc,
                            Hw=Hw_vec, Hw_se=Hw_se_vec, Q_Hw=Q_Hw)

write.csv(annual_Qt_all, file.path(output_table_path,
                                   "TC2_annual_Qt_all.csv"), row.names=F)
```

```r
TC2.annualQtAll <- read.csv(file.path(output_table_path,
                                      "TC2_annual_Qt_all.csv"))
kable(TC2.annualQtAll, digits=3, align='c', escape = F,
      col.names = c(" Weather station", "$H_c$", "$Q(H_c)$",
                   "$H_w$", "$SE(H_w)$", "$Q(H_w)$"))
```

Weather station	H_c	$Q(H_c)$	H_w	$SE(H_w)$	$Q(H_w)$
Argentina, Buenos Aires	0.951	4.960	0.938	0.055	-0.222
Australia, Adelaide	0.882	2.511	0.781	0.058	0.336
Australia, Alice Springs	0.647	-0.040	0.708	0.058	0.062
Australia, Cap Otway	0.905	0.222	0.869	0.059	0.157
Austria, Kremsmunster	0.728	-0.679	0.782	0.058	0.273
Austria, Vienna	0.811	-0.527	0.806	0.055	-0.270
Belgium, Uccle	0.740	-0.086	0.739	0.050	0.011
Canada, Winnipeg	0.713	0.346	0.728	0.058	0.174
Croatia, Zagreb	0.723	0.888	0.780	0.055	-0.144
Czech Republic, Prague	0.745	0.442	0.716	0.043	0.012
Denmark, Copenhagen	0.817	0.092	0.753	0.045	0.007
Denmark, Vestervig	0.699	0.093	0.733	0.056	0.008
Egypt, Alexandria	0.882	0.224	0.862	0.064	0.010
France, Nantes	0.738	-0.494	0.720	0.052	0.120
France, Paris	0.873	0.574	0.802	0.042	-0.010
Germany, Berlin	0.726	-0.053	0.712	0.041	-0.041
Germany, Hohenpeissenberg	0.701	1.053	0.684	0.043	-0.338
Germany, Karlsruhe	0.728	0.209	0.819	0.059	0.172
Greece, Athens	0.754	0.863	0.788	0.054	0.094
Greenland, Illulisat	0.806	0.839	0.805	0.057	0.000
Greenland, Ivittuut	0.797	-0.202	0.804	0.071	-0.273
Hungary, Budapest	0.682	0.288	0.663	0.043	-0.115
Iceland, Djupivogur	0.852	0.084	0.841	0.058	0.332
Iceland, Reykjavik	0.889	0.996	0.885	0.057	0.146
India, Agra	0.802	-0.281	0.844	0.066	0.184
India, Allahabad	0.706	-1.135	0.807	0.059	-0.839
India, Bombay	0.793	-0.072	0.887	0.059	0.087
India, Indore	0.820	-0.303	0.899	0.059	-0.534
India, Madras	0.775	-0.511	0.906	0.059	-0.328
India, Nagpur	0.610	-0.424	0.727	0.058	-0.158
Israel, Jerusalem	0.702	0.022	0.654	0.057	-0.060
Italy, Bologna	0.819	0.602	0.845	0.048	-0.723
Italy, Milan	0.851	-1.014	0.826	0.043	-0.281
Japan, Hiroshima	0.798	-0.165	0.738	0.059	-0.267

Weather station	H_c	$Q(H_c)$	H_w	$SE(H_w)$	$Q(H_w)$
Japan, Nagasaki	0.823	0.026	0.761	0.058	0.014
Japan, Tokyo	0.926	-0.235	0.851	0.058	-0.086
Kazakhstan, Kazalinsk	0.611	-0.250	0.563	0.061	-0.216
Luxembourg, Luxembourg	0.815	-1.068	0.825	0.051	1.062
New Zealand, Wellington	0.810	-0.618	0.919	0.060	-0.296
Norway, Andoya	0.773	0.049	0.761	0.055	-0.013
Norway, Bergen	0.783	-0.377	0.717	0.053	0.206
Norway, Bodo	0.700	-0.289	0.682	0.054	-0.130
Norway, Dombas	0.679	3.476	0.637	0.053	3.275
Norway, Karasjok	0.655	-0.386	0.652	0.056	0.627
Norway, Mandal	0.620	-0.499	0.625	0.054	0.368
Norway, Oksoy Lighthouse	0.666	-0.126	0.672	0.055	0.156
Norway, Ona	0.674	-0.139	0.702	0.056	0.030
Norway, Oslo	0.692	0.464	0.699	0.047	-0.130
Norway, Roros	0.727	-0.333	0.688	0.055	-0.229
Norway, Tromso	0.616	-0.318	0.641	0.054	0.005
Norway, Utsira	0.738	-0.094	0.753	0.055	-0.040
Norway, Vardo	0.724	-0.053	0.770	0.054	-0.017
Pakistan, Lahore	0.659	0.047	0.743	0.057	0.020
Portugal, Lisbon	0.933	0.211	0.931	0.060	-0.221
Romania, Sulina	0.591	0.121	0.631	0.057	-0.121
Russia, Archangelsk	0.707	1.187	0.746	0.058	-0.369
Russia, Sort	0.594	0.722	0.581	0.062	0.431
Russia, St Petersburg	0.670	-0.161	0.706	0.058	7.045
Spain, Gibraltar	0.787	-0.075	0.855	0.056	0.283
Sweden, Bromma	0.688	0.202	0.690	0.041	-0.043
Sweden, Stockholm	0.614	5.121	0.632	0.041	-0.940
Sweden, Tullinge	0.624	1.523	0.622	0.041	-0.312
Sweden, Uppsala	0.715	1.135	0.710	0.040	-0.291
Switzerland, Basel	0.664	0.558	0.720	0.042	-0.753
Switzerland, Geneva	0.845	-0.537	0.818	0.042	0.563
UK, Aberdeen	0.771	0.268	0.767	0.058	-0.245
UK, Belfast	0.707	-0.576	0.727	0.058	0.252
UK, Cambridge	0.773	-1.596	0.781	0.056	4.131
UK, Durham	0.771	-0.656	0.761	0.052	3.554
UK, Edinbourg	0.605	-1.430	0.626	0.045	2.282
UK, London	0.798	-0.543	0.809	0.053	1.394
UK, Plymouth	0.559	0.268	0.671	0.058	1.633
USA, Atlanta	0.766	2.043	0.725	0.058	1.774
USA, Bismarck	0.749	1.135	0.761	0.058	0.631
USA, Boise	0.725	-0.017	0.698	0.057	-0.281
USA, Boston	0.728	0.389	0.724	0.058	0.541
USA, Chattanooga	0.744	0.715	0.695	0.057	0.684
USA, Cincinatti	0.758	-0.108	0.718	0.058	0.685
USA, Columbus	0.705	-1.449	0.702	0.057	0.248
USA, Concord	0.790	0.012	0.729	0.058	-0.255
USA, Des Moines	0.621	0.042	0.623	0.056	-0.247
USA, Detroit	0.707	1.846	0.663	0.057	-1.948
USA, Dodge City	0.648	0.123	0.715	0.058	-0.191
USA, Fargo	0.738	2.083	0.725	0.058	0.948
USA, Galveston	0.674	-0.566	0.666	0.057	-0.104
USA, Indianapolis	0.667	1.495	0.658	0.057	-0.726

Weather station	H_c	$Q(H_c)$	H_w	$SE(H_w)$	$Q(H_w)$
USA, Jacksonville	0.664	0.212	0.618	0.056	-0.423
USA, Knoxville	0.744	-0.773	0.680	0.057	0.267
USA, Las Vegas	0.707	-0.310	0.694	0.060	-0.100
USA, Madison	0.673	-0.336	0.682	0.057	-0.347
USA, Marquette	0.694	-0.026	0.716	0.058	-0.167
USA, Milwaukee	0.683	-1.088	0.755	0.058	-0.247
USA, Mobile	0.678	3.408	0.672	0.057	0.667
USA, Nashville	0.609	-0.240	0.625	0.057	0.360
USA, New Orleans	0.861	4.034	0.812	0.058	-0.570
USA, New York	0.907	4.772	0.843	0.049	2.002

From the results in Table C2 we note that the estimates of the Hurst parameter based on annual data are, on average, higher than the corresponding estimates based on monthly data. Furthermore, we see that data from 9 weather stations reject the FGN hypothesis when using the characteristic function estimate of the Hurst parameter whereas data from 6 weather stations reject the FGN when using the Whittle estimate of the Hurst parameter.

Table C3. Estimates and test statistics based on Moberg et al. (2005) time series

```r
TT <- length(Xj_m)
mu <- mean(Xj_m)
sd <- sd(Xj_m)
mu_c <- estim.cf.mu(Zj=Xj_m)
sigma_c <- estim.cf.sigma(Yj=Xj_m, FBM=F)
Hc <- estim.cf.H(Yj=Zj_m, FBM=F)
Hw <- estim.w.H(Zj_m)
Hw_SE <- WhittleEst(Zj_m)$coefficient[2]
Q.Hc <- Qstat(Zj_m, H=Hc, TT=TT)
Q.Hw <- Qstat(Zj_m, H=Hw, TT=TT)


moberg_estimates <- data.frame(c(mu, sd, mu_c, sigma_c, Hc, Hw, Hw_SE, Q.Hc, Q.Hw))
rownames(moberg_estimates) <- c("Mu", "Sigma", "Mu_c", "Sigma_c",
                                 "Hc", "Hw", "Hw_SE", "Q.Hc", "Q.Hw")
colnames(moberg_estimates) <- "Estimates"

Hc <- estim.cf.H(Yj=Zj_m, FBM=F)
Hw <- estim.w.H(Zj_m)
Hw_SE <- WhittleEst(Zj_m)$coefficient[2]
Q.Hc <- Qstat(Zj_m, H=Hc, TT=TT)
Q.Hw <- Qstat(Zj_m, H=Hw, TT=TT)

write.csv(moberg_estimates, file.path(output_table_path,
                                "TC3_moberg_estimates.csv"))
```

```r
TC3.mobergEstimates <- read.csv(file.path(output_table_path, "TC3_moberg_estimates.csv"))

TC3.mobergEstimates[,1] <- c("$\\mu$", "$\\sigma$", "$\\mu_c$", "$\\sigma_c$",
                   "$H_c$", "$H_w$", "$SE(H_w)$", "$Q(H_c)$",  "$Q(H_w)$")
kable(TC3.mobergEstimates, digits=3, align='c' , escape = F,
```

```
      col.names = c("Parameters and statistics", "Value"))
```

Parameters and statistics	Value
μ	-0.354
σ	0.220
μ_c	-0.354
σ_c	0.051
H_c	0.917
H_w	0.990
$SE(H_w)$	0.015
$Q(H_c)$	-11.205
$Q(H_w)$	104.220

The results of Table C3 show that the FGN model is rejected for the Moberg data when the respective estimated Hurst parameters are used.

Table C4. Chi-square statistics based on the data of Moberg et al. (2009)

```
Hvec <- seq(0.92, 0.98, by=0.01)

Qt <- NULL
  for (Hval in Hvec){
    Qt <- c(Qt, Qstat(Zj_m, H=Hval, TT=TT))
  }

moberg_Qt_grid <- data.frame(H=Hvec, Qt)

write.csv(moberg_Qt_grid, row.names=F, file.path(output_table_path,
                                "TC4_moberg_Qt_grid.csv"))
```

```
TC4.mobergQtGrid <- read.csv(file.path(output_table_path,
                                       "TC4_moberg_Qt_grid.csv"))
kable(TC4.mobergQtGrid, digits=3, align='c',
      col.names = c("H", "Q(H)"))
```

H	Q(H)
0.92	-10.595
0.93	-8.332
0.94	-5.274
0.95	-0.946
0.96	5.599
0.97	16.575
0.98	38.621

The results of Table C4 shows that the power of the Q test is high (conditional on the FGN model). In particular, when H = 0.95 then Q(H) $\in$ (-1.96, 1.96) whereas when H equals 0.94 or 0.96 (or further away from 0.95) then Q(H) $\notin$ (-1.96, 1.96) which means rejection of FGN.

Table C5. Stationarity test. Moberg data

```r
Stationary_moberg <- matrix(0, nrow=2, ncol=3)
colnames(Stationary_moberg) <- c("test.stat","test.res","test.criterion")
rownames(Stationary_moberg) <- c("Sign: 0.05", "Sign: 0.1")


sign_level <- c(0.05, 0.1)

for (j in 1:2) {
        station_res <- unsys.station.test(Xj_m, M=2000,
                                          sig.lev = sign_level[j])
        Stationary_moberg[j,1] <- station_res$test.stat
        Stationary_moberg[j,2] <- station_res$test.res
        Stationary_moberg[j,3] <- station_res$test.criterion
        }

Stationary_moberg <- as.data.frame(Stationary_moberg)


# sum(Stationary_all_annual$test.res)

write.csv(Stationary_moberg,
          file.path(output_table_path, "TC5_Stationarity_moberg.csv"))
```

```r
TC5.Stationarity_moberg <- read.csv(file.path(output_table_path,
                                              "TC5_Stationarity_moberg.csv"))
rownames(TC5.Stationarity_moberg) <- c("Significance level: 0.05",
                                       "Significance level: 0.1")
TC5.Stationarity_moberg$X <- NULL

TC5.Stationarity_moberg$test.res <- replace(TC5.Stationarity_moberg$test.res,
                                            TC5.Stationarity_moberg$test.res==0,
                                            "no rejection")



kable(TC5.Stationarity_moberg, digits=3, align='c',
      col.names = c( "Test statistic", "Test result", "Test criterion"))
```

	Test statistic	Test result	Test criterion
Significance level: 0.05	3.397	no rejection	5.494
Significance level: 0.1	3.430	no rejection	5.350

Table C6. Stationarity test. Annual data

```
Stationary_all_annual <- matrix(0, nrow=nja, ncol=3)
colnames(Stationary_all_annual) <- c("test.stat","test.res","test.criterion")
rownames(Stationary_all_annual) <- paste0(all$Country, ",", all$Station)

for (j in 1:nja) {
        #--- 1. Data reading
        data <- read.delim(data_dir_all[j], header=F, na.strings=99)
          Xj <- data[,14]
          Zj <- Xj[!is.na(Xj)]
        #--- 2. Estimation
        station_res <- unsys.station.test(Zj, M=2000, sig.lev = 0.01)
        Stationary_all_annual[j,1] <- station_res$test.stat
        Stationary_all_annual[j,2] <- station_res$test.res
        Stationary_all_annual[j,3] <- station_res$test.criterion
        }

Stationary_all_annual <- as.data.frame(Stationary_all_annual)

# sum(Stationary_all_annual$test.res)

write.csv(Stationary_all_annual,
          file.path(output_table_path, "TC6_Stationarity_all_annual.csv"))
```

```
TC6.Stationarity_all_annual <- read.csv(file.path(output_table_path,
                                                   "TC6_Stationarity_all_annual.csv"))

TC6.Stationarity_all_annual <- TC6.Stationarity_all_annual %>%
  mutate(test.res = case_when(test.res == 0 ~ "no rejection",
                              test.res == 1 ~ "rejection"))

kable(TC6.Stationarity_all_annual, digits=3, align='c',
      col.names = c(" Weather station", "Test statistic",
                    "Test result", "Test criterion"))
```

Weather station	Test statistic	Test result	Test criterion
Argentina,Buenos Aires	3.086	no rejection	5.865
Australia,Adelaide	3.282	no rejection	5.878
Australia,Alice Springs	3.947	no rejection	5.802
Australia,Cap Otway	3.719	no rejection	5.788
Austria,Kremsmunster	3.229	no rejection	5.742
Austria,Vienna	2.722	no rejection	5.747
Belgium,Uccle	3.905	no rejection	5.941
Canada,Winnipeg	3.593	no rejection	5.628
Croatia,Zagreb	4.149	no rejection	5.813
Czech Republic,Prague	4.066	no rejection	5.971
Denmark,Copenhagen	3.912	no rejection	5.957
Denmark,Vestervig	3.926	no rejection	5.864
Egypt,Alexandria	5.144	no rejection	5.493
France,Nantes	3.805	no rejection	5.855
France,Paris	4.139	no rejection	5.952
Germany,Berlin	4.309	no rejection	5.976

Weather station	Test statistic	Test result	Test criterion
Germany,Hohenpeissenberg	3.038	no rejection	5.987
Germany,Karlsruhe	2.600	no rejection	5.717
Greece,Athens	3.658	no rejection	5.581
Greenland,Illulisat	3.716	no rejection	5.865
Greenland,Ivittuut	3.042	no rejection	5.740
Hungary,Budapest	3.673	no rejection	5.941
Iceland,Djupivogur	6.377	rejection	5.849
Iceland,Reykjavik	3.259	no rejection	5.717
India,Agra	4.363	no rejection	5.828
India,Allahabad	2.495	no rejection	5.822
India,Bombay	4.361	no rejection	5.878
India,Indore	2.763	no rejection	5.892
India,Madras	4.714	no rejection	5.718
India,Nagpur	3.669	no rejection	5.869
Israel,Jerusalem	3.665	no rejection	5.821
Italy,Bologna	4.363	no rejection	5.907
Italy,Milan	4.002	no rejection	5.935
Japan,Hiroshima	2.940	no rejection	5.333
Japan,Nagasaki	2.960	no rejection	5.637
Japan,Tokyo	2.550	no rejection	5.569
Kazakhstan,Kazalinsk	3.289	no rejection	5.751
Luxembourg,Luxembourg	3.368	no rejection	5.902
New Zealand,Wellington	3.390	no rejection	5.606
Norway,Andoya	4.493	no rejection	5.914
Norway,Bergen	2.312	no rejection	5.861
Norway,Bodo	4.287	no rejection	5.896
Norway,Dombas	5.485	no rejection	5.867
Norway,Karasjok	3.920	no rejection	5.821
Norway,Mandal	3.929	no rejection	5.924
Norway,Oksoy Lighthouse	4.187	no rejection	5.873
Norway,Ona	3.902	no rejection	5.894
Norway,Oslo	3.286	no rejection	5.928
Norway,Roros	3.352	no rejection	5.832
Norway,Tromso	3.834	no rejection	5.871
Norway,Utsira	2.863	no rejection	5.868
Norway,Vardo	2.742	no rejection	5.867
Pakistan,Lahore	2.830	no rejection	5.889
Portugal,Lisbon	4.938	no rejection	5.784
Romania,Sulina	2.957	no rejection	5.396
Russia,Archangelsk	3.720	no rejection	5.683
Russia,Sort	3.222	no rejection	5.437
Russia,St Petersburg	3.695	no rejection	5.852
Spain,Gibraltar	5.704	no rejection	5.867
Sweden,Bromma	3.244	no rejection	5.963
Sweden,Stockholm	3.083	no rejection	5.955
Sweden,Tullinge	3.508	no rejection	5.950
Sweden,Uppsala	3.182	no rejection	5.920
Switzerland,Basel	4.411	no rejection	5.961
Switzerland,Geneva	4.266	no rejection	5.955
UK,Aberdeen	2.114	no rejection	5.818
UK,Belfast	2.766	no rejection	5.846
UK,Cambridge	2.815	no rejection	5.863

Weather station	Test statistic	Test result	Test criterion
UK,Durham	2.867	no rejection	5.930
UK,Edinbourg	3.406	no rejection	5.914
UK,London	4.037	no rejection	5.909
UK,Plymouth	4.379	no rejection	5.852
USA,Atlanta	3.985	no rejection	5.897
USA,Bismarck	3.510	no rejection	5.635
USA,Boise	3.839	no rejection	5.567
USA,Boston	3.373	no rejection	5.828
USA,Chattanooga	3.835	no rejection	5.863
USA,Cincinatti	4.886	no rejection	5.860
USA,Columbus	3.413	no rejection	5.831
USA,Concord	2.573	no rejection	5.911
USA,Des Moines	2.397	no rejection	5.799
USA,Detroit	3.542	no rejection	5.841
USA,Dodge City	2.791	no rejection	5.799
USA,Fargo	2.176	no rejection	5.584
USA,Galveston	2.841	no rejection	5.879
USA,Indianapolis	3.422	no rejection	5.825
USA,Jacksonville	3.743	no rejection	5.800
USA,Knoxville	2.886	no rejection	5.866
USA,Las Vegas	2.441	no rejection	5.746
USA,Madison	2.792	no rejection	5.842
USA,Marquette	3.198	no rejection	5.843
USA,Milwaukee	2.621	no rejection	5.819
USA,Mobile	2.741	no rejection	5.887
USA,Nashville	3.836	no rejection	5.858
USA,New Orleans	2.708	no rejection	5.859
USA,New York	4.448	no rejection	5.617

From Table C6 we note that only in one case (Djupivogur, Iceland) do the data reject the stationarity hypothesis.

Table C7. Stationarity test. Monthly data

```
Stationarity_all <- matrix(0, nrow=nja, ncol=3)
colnames(Stationarity_all) <- c("test.stat","test.res","test.criterion")
rownames(Stationarity_all) <- paste0(all$Country, ",", all$Station)

for (j in 1:nja) {
        #--- 1. Data reading
        data <- read.delim(data_dir_all[j], header=F, na.strings=99)
        Zm <- monthlyAdj(data, scale=T)$Zm

        #--- 2. Estimation
        station_res <- unsys.station.test(Zm, M=2000, sig.lev = 0.01)
        Stationarity_all[j,1] <- station_res$test.stat
        Stationarity_all[j,2] <- station_res$test.res
        Stationarity_all[j,3] <- station_res$test.criterion
        }

Stationarity_all <- as.data.frame(Stationarity_all)

# sum(Stationarity_all$test.res)

write.csv(Stationarity_all, file.path(output_table_path,
                                "TC7.Stationarity_all.csv"))
```

```
TC7.Stationarity_all <- read.csv(file.path(output_table_path, "TC7_Stationarity_all.csv"))

TC7.Stationarity_all <- TC7.Stationarity_all %>%
  mutate(test.res = case_when(test.res == 0 ~ "no rejection",
                        test.res == 1 ~ "rejection"))

kable(TC7.Stationarity_all, digits=3, align='c',
      col.names = c(" Weather station", "Test statistic", "Test result",
                    "Test criterion"))
```

Weather station	Test statistic	Test result	Test criterion
Argentina,Buenos Aires	5.757	no rejection	5.827
Australia,Adelaide	6.218	rejection	6.061
Australia,Alice Springs	3.191	no rejection	6.050
Australia,Cap Otway	4.632	no rejection	6.040
Austria,Kremsmunster	5.184	no rejection	5.924
Austria,Vienna	5.872	no rejection	6.011
Belgium,Uccle	3.473	no rejection	6.036
Canada,Winnipeg	3.152	no rejection	5.976
Croatia,Zagreb	4.593	no rejection	5.997
Czech Republic,Prague	6.596	rejection	6.113
Denmark,Copenhagen	4.761	no rejection	6.113
Denmark,Vestervig	3.951	no rejection	5.906
Egypt,Alexandria	5.577	no rejection	5.903
France,Nantes	6.010	no rejection	6.043
France,Paris	6.605	rejection	6.125
Germany,Berlin	4.873	no rejection	6.139
Germany,Hohenpeissenberg	5.198	no rejection	6.085

Weather station	Test statistic	Test result	Test criterion
Germany,Karlsruhe	5.494	no rejection	5.891
Greece,Athens	5.590	no rejection	6.003
Greenland,Illulisat	4.709	no rejection	5.986
Greenland,Ivittuut	4.249	no rejection	5.988
Hungary,Budapest	4.918	no rejection	6.100
Iceland,Djupivogur	11.165	rejection	5.973
Iceland,Reykjavik	7.734	rejection	6.007
India,Agra	6.462	rejection	6.040
India,Allahabad	2.863	no rejection	5.799
India,Bombay	4.242	no rejection	5.617
India,Indore	5.282	no rejection	6.044
India,Madras	3.034	no rejection	5.936
India,Nagpur	4.462	no rejection	5.997
Israel,Jerusalem	6.134	rejection	6.032
Italy,Bologna	4.077	no rejection	6.071
Italy,Milan	6.805	rejection	6.087
Japan,Hiroshima	3.005	no rejection	6.072
Japan,Nagasaki	3.650	no rejection	6.066
Japan,Tokyo	4.694	no rejection	5.955
Kazakhstan,Kazalinsk	5.152	no rejection	5.884
Luxembourg,Luxembourg	4.309	no rejection	6.060
New Zealand,Wellington	6.178	rejection	5.738
Norway,Andoya	4.066	no rejection	5.984
Norway,Bergen	3.425	no rejection	5.999
Norway,Bodo	3.316	no rejection	5.996
Norway,Dombas	4.461	no rejection	5.920
Norway,Karasjok	2.625	no rejection	5.992
Norway,Mandal	3.047	no rejection	5.827
Norway,Oksoy Lighthouse	4.834	no rejection	5.869
Norway,Ona	3.389	no rejection	5.994
Norway,Oslo	2.952	no rejection	6.035
Norway,Roros	2.861	no rejection	5.883
Norway,Tromso	2.613	no rejection	5.954
Norway,Utsira	4.547	no rejection	5.975
Norway,Vardo	5.063	no rejection	6.066
Pakistan,Lahore	5.674	no rejection	5.979
Portugal,Lisbon	4.202	no rejection	5.798
Romania,Sulina	4.191	no rejection	6.023
Russia,Archangelsk	4.255	no rejection	5.711
Russia,Sort	3.307	no rejection	5.849
Russia,St Petersburg	3.804	no rejection	5.945
Spain,Gibraltar	5.015	no rejection	5.984
Sweden,Bromma	5.945	no rejection	6.122
Sweden,Stockholm	6.566	rejection	6.129
Sweden,Tullinge	6.485	rejection	6.118
Sweden,Uppsala	4.344	no rejection	6.122
Switzerland,Basel	4.622	no rejection	6.095
Switzerland,Geneva	5.076	no rejection	6.121
UK,Aberdeen	3.624	no rejection	5.806
UK,Belfast	4.199	no rejection	6.032
UK,Cambridge	4.162	no rejection	6.039
UK,Durham	3.426	no rejection	6.049

Weather station	Test statistic	Test result	Test criterion
UK,Edinbourg	5.059	no rejection	6.084
UK,London	4.491	no rejection	6.072
UK,Plymouth	6.162	rejection	6.026
USA,Atlanta	3.036	no rejection	6.061
USA,Bismarck	6.314	rejection	6.041
USA,Boise	5.216	no rejection	5.993
USA,Boston	3.340	no rejection	6.044
USA,Chattanooga	4.418	no rejection	6.034
USA,Cincinatti	3.210	no rejection	6.030
USA,Columbus	3.712	no rejection	6.045
USA,Concord	3.177	no rejection	6.024
USA,Des Moines	4.699	no rejection	6.041
USA,Detroit	3.798	no rejection	6.058
USA,Dodge City	4.666	no rejection	6.050
USA,Fargo	4.005	no rejection	6.053
USA,Galveston	3.254	no rejection	6.054
USA,Indianapolis	4.700	no rejection	6.053
USA,Jacksonville	4.632	no rejection	5.996
USA,Knoxville	4.195	no rejection	6.054
USA,Las Vegas	3.920	no rejection	6.040
USA,Madison	3.556	no rejection	6.046
USA,Marquette	4.005	no rejection	5.983
USA,Milwaukee	3.842	no rejection	6.044
USA,Mobile	6.051	rejection	5.901
USA,Nashville	4.777	no rejection	6.029
USA,New Orleans	4.446	no rejection	5.743
USA,New York	3.618	no rejection	6.025

Table C7 shows that stationarity (based on the default option of Cho's test) is rejected for data from 14 weather stations when monthly time series are used.

Table C8. Estimation of H using the Wavelet Lifting estimator. Monthly data

```r
# Wavelet estimation for all time series
Parameters <- data.frame()
for (j in 1:nja) {
        #--- 1. Data reading
        data <- read.delim(data_dir_all[j], header=F, na.strings=99)
        Zm <- monthlyAdj(data, scale=T)$Zm
        timeindex <- which(!is.na(data[,2:13]))
        nmiss <- sum(is.na(Zm))
        #--- 2. Estimation
        Hvec <- c(liftHurst(Zm, grid=timeindex), nmiss)
        Parameters <- rbind(Parameters, Hvec)
        }

colnames(Parameters) <- c("Beta", "Hl", "Sd_Hl", "Lo_Hl", "Hi_Hl", "Missing")
Parameters <- cbind(stationame_all, Parameters)
saveRDS(Parameters, file.path(output_supporting_path, "wavelet_H_all.rds"))

# Comparison between Wavelet and Whittle estimator
Hl_tbl <- readRDS(file.path(output_supporting_path, "wavelet_H_all.rds"))
Hw_tbl <- read.csv(file.path(output_table_path, "TC1_monthly_H_all.csv")) %>%
  rename(stationame_all = X)
H_comparison_tbl <- Hl_tbl %>%
  left_join(Hw_tbl) %>%
  select(-c(Beta, Missing)) %>%
  mutate(Lo_Hw = Hw - 3*SE_Hw,
         Hi_Hw = Hw + 3*SE_Hw,
         overlap = case_when(Hi_Hl < Lo_Hw | Hi_Hw < Lo_Hl ~ 0,
                                TRUE ~ 1))

# Q statistics for Whittle and Wavelet estimator for all time series
H_comparison_tbl$Q_Hl <- NA
H_comparison_tbl$Q_Hw <- NA
for (j in 1:nja){
 data <- read.delim(data_dir_all[j], header=F, na.strings=99)
 Zm <- monthlyAdj(data, scale=T)$Zm
 TT <- length(Zm)
 Hl <- H_comparison_tbl[j, "Hl"]
 Hw <- H_comparison_tbl[j, "Hw"]
 H_comparison_tbl[j, "Q_Hl"] <- Qstat(Zm, H=Hl, TT=TT)
 H_comparison_tbl[j, "Q_Hw"] <- Qstat(Zm, H=Hw, TT=TT)
}

# Saving Whittle vs Wavalet estimation table
write.csv(H_comparison_tbl,
          file.path(output_table_path, "TC8_monthly_all_H_wavelet.csv"))
```

```r
H_comparison_tbl <- read.csv(
          file.path(output_table_path, "TC8_monthly_all_H_wavelet.csv"))
```

```r
H_comparison_tbl %>%
  select(stationame_all, Hl, Q_Hl) %>%
  kable(digits=3, align='c',
```

```r
col.names = c("Weather station", "$H_{wav}$", "$Q(H_{wav})$"))
```

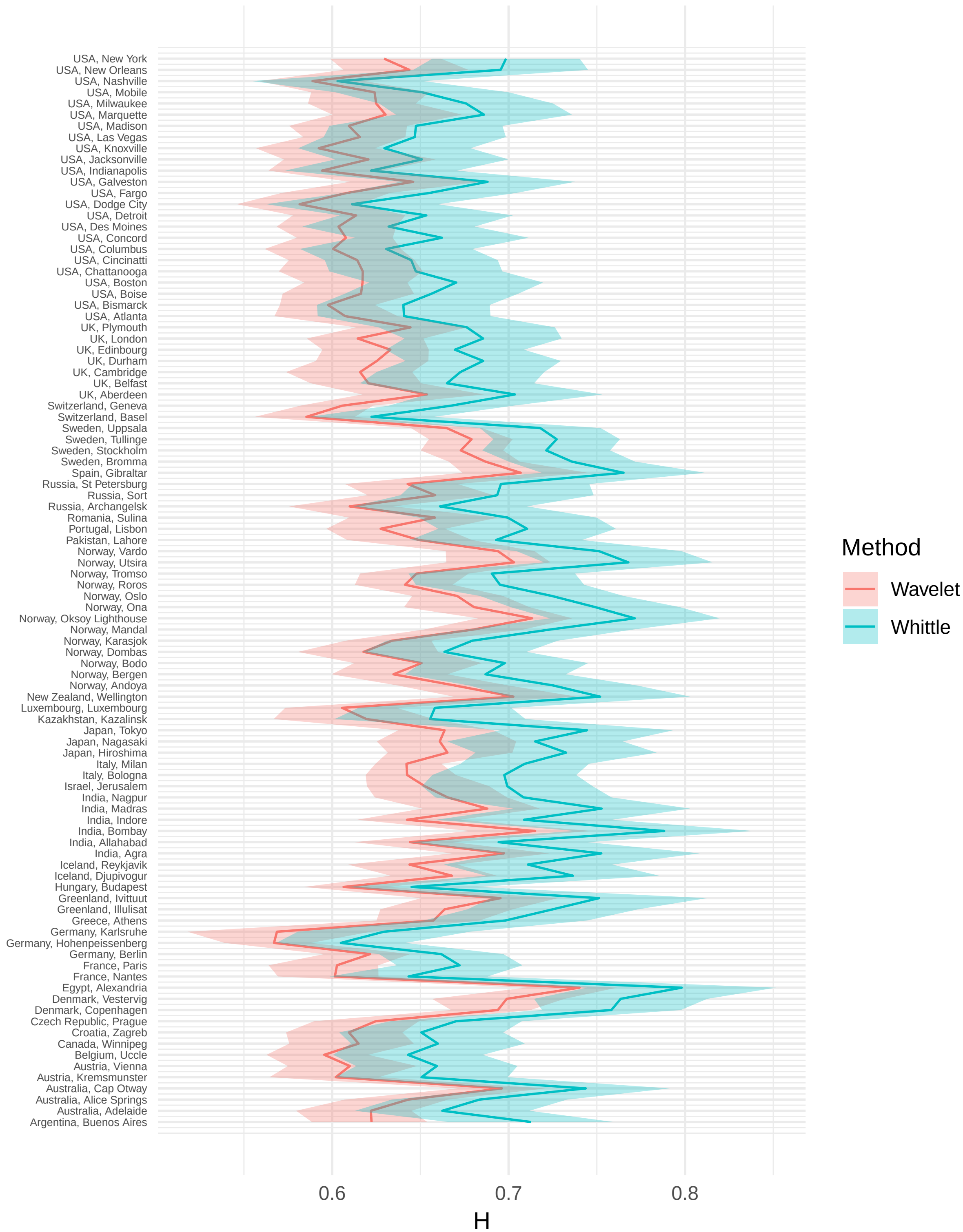
Weather station	H_{wav}	$Q(H_{wav})$
Argentina, Buenos Aires	0.622	-5.299
Australia, Adelaide	0.622	-1.906
Australia, Alice Springs	0.643	-1.490
Australia, Cap Otway	0.696	-4.644
Austria, Kremsmunster	0.602	-1.606
Austria, Vienna	0.610	-2.172
Belgium, Uccle	0.595	-1.695
Canada, Winnipeg	0.615	-1.349
Croatia, Zagreb	0.609	-1.449
Czech Republic, Prague	0.625	-2.273
Denmark, Copenhagen	0.694	-4.464
Denmark, Vestervig	0.699	-2.392
Egypt, Alexandria	0.740	-3.498
France, Nantes	0.602	-1.312
France, Paris	0.603	-4.348
Germany, Berlin	0.621	-1.907
Germany, Hohenpeissenberg	0.567	-1.001
Germany, Karlsruhe	0.569	-1.383
Greece, Athens	0.657	-1.622
Greenland, Illulisat	0.664	-2.943
Greenland, Ivittuut	0.695	-2.970
Hungary, Budapest	0.606	-1.260
Iceland, Djupivogur	0.668	-4.356
Iceland, Reykjavik	0.644	-3.351
India, Agra	0.697	-2.155
India, Allahabad	0.644	-1.767
India, Bombay	0.715	-4.660
India, Indore	0.642	-2.954
India, Madras	0.688	-3.555
India, Nagpur	0.665	-1.588
Israel, Jerusalem	0.652	-1.589
Italy, Bologna	0.642	-3.037
Italy, Milan	0.642	-4.870
Japan, Hiroshima	0.665	-2.826
Japan, Nagasaki	0.661	-2.904
Japan, Tokyo	0.664	-5.614
Kazakhstan, Kazalinsk	0.619	-0.592
Luxembourg, Luxembourg	0.606	-2.034
New Zealand, Wellington	0.703	-3.066
Norway, Andoya	0.669	-2.437
Norway, Bergen	0.635	-1.900
Norway, Bodo	0.651	-1.500
Norway, Dombas	0.618	-1.215
Norway, Karasjok	0.634	-1.152
Norway, Mandal	0.680	-0.963
Norway, Oksoy Lighthouse	0.713	-1.528
Norway, Ona	0.680	-2.199
Norway, Oslo	0.671	-2.014
Norway, Roros	0.641	-1.650
Norway, Tromso	0.648	-1.162

Weather station	H_{wav}	$Q(H_{wav})$
Norway, Utsira	0.703	-2.522
Norway, Vardo	0.694	-3.376
Pakistan, Lahore	0.649	-1.011
Portugal, Lisbon	0.627	-4.469
Romania, Sulina	0.658	-0.876
Russia, Archangelsk	0.610	-1.618
Russia, Sort	0.658	-0.460
Russia, St Petersburg	0.643	-1.817
Spain, Gibraltar	0.707	-4.289
Sweden, Bromma	0.687	-2.110
Sweden, Stockholm	0.673	-1.558
Sweden, Tullinge	0.679	-1.350
Sweden, Uppsala	0.665	-2.545
Switzerland, Basel	0.585	-1.229
Switzerland, Geneva	0.606	-3.239
UK, Aberdeen	0.654	-1.924
UK, Belfast	0.621	-1.307
UK, Cambridge	0.616	-2.031
UK, Durham	0.625	-2.584
UK, Edinbourg	0.633	-0.943
UK, London	0.614	-3.479
UK, Plymouth	0.644	-0.635
USA, Atlanta	0.607	-0.909
USA, Bismarck	0.598	-1.234
USA, Boise	0.616	-1.300
USA, Boston	0.617	-1.920
USA, Chattanooga	0.617	-0.855
USA, Cincinatti	0.614	-1.090
USA, Columbus	0.601	-0.866
USA, Concord	0.608	-1.973
USA, Des Moines	0.604	-0.734
USA, Detroit	0.614	-1.196
USA, Dodge City	0.582	-0.776
USA, Fargo	0.608	-1.420
USA, Galveston	0.646	-1.334
USA, Indianapolis	0.594	-0.666
USA, Jacksonville	0.621	-0.723
USA, Knoxville	0.592	-0.926
USA, Las Vegas	0.616	-0.933
USA, Madison	0.609	-1.009
USA, Marquette	0.630	-2.118
USA, Milwaukee	0.625	-1.860
USA, Mobile	0.624	-0.638
USA, Nashville	0.589	-0.382
USA, New Orleans	0.644	-2.132
USA, New York	0.629	-4.211

Figure C1. Comparison between the Wavelet Lifting and the Whittle estimator

Wavelet Lifting vs Whittle estimates of H, with 95% confidence bands

```r
# Long data format
H_comparison_tbl_long <- data.frame(
  City = rep(1:96, 2),
    City_label = rep(H_comparison_tbl$stationame_all, 2),
    Method = rep(c("Wavelet", "Whittle"), each = 96),
    H = c(H_comparison_tbl$Hl, H_comparison_tbl$Hw),
    Low = c(H_comparison_tbl$Lo_Hl, H_comparison_tbl$Lo_Hw),
    High = c(H_comparison_tbl$Hi_Hl, H_comparison_tbl$Hi_Hw)
  )

# Plot
H_comparison_tbl_long %>%
  ggplot() +
  geom_line(aes(x = City, y = H, col = Method)) +
  geom_ribbon(aes(x = City, ymin = Low, ymax = High,
                fill = Method), alpha = 0.3) +
  scale_x_continuous(breaks=1:96,
                     labels=stationame_all) +
  theme_minimal() +
   theme(#axis.text.x = element_text(angle = 90, size = 2),
         axis.text.y = element_text(size = 5)) +
  coord_flip() +
  labs(# title = "Whittle vs Wavelet Lifting estimates of H",
       # subtitle = "with 95% confidence intervals",
       x = "")
```




```r
ggsave(file.path(output_figure_path, "FC1_H_wavelet_whittle_plot.png"),
       width = 5, height = 7)
```

APPENDIX D - PROPERTIES OF THE ESTIMATORS

Table D1, D2, D3. Tables D1, D2, D3. Properties of different estimators of the FGN model. Bootstrap simulations

The following tables show the results of the bootstrap simultations for different estimators of μ, σ, H and α, given the FGN model with H equal to 0.7, 0.8, 0.9, 0.95.

Each bootstrap estimate is based on 1,000 simulated FGN series of length 2,000.

```
#--- 1. Parameters choice
Hvec <- c(0.7, 0.8, 0.9, 0.95)
N <- 1000
Tlenght <- 2000
estimator_names <- c("Mu_c", "Mu_ML", "Sigma_c", "Sigma_ML", "H_c", "H_ML",
                     "Alpha_c", "Q_Stat", "Mean", "Sd")

start <- Sys.time()

#--- 2. Bootstrap estimation
for (j in 1:4){
          H <- Hvec[j]
          start <- Sys.time()
          print(paste0("----- H = ", H, " ----- ", Sys.time()))
          Parameters <- matrix(0, nrow=N, ncol=10)
          colnames(Parameters) <- estimator_names
          #--- 3a. Estimation for given H
          for (i in 1:N){
                         Zjsim <- simFGN0(Tlenght, H)
                         reg <- estim.cf.reg(Yj=Zjsim, FBM=F)
                         Hw <- estim.w.H(Zjsim)
                         mu_ML <- FgnMean(Zjsim, H=Hw, sigma=1)
                         Parameters[i,1] <- estim.cf.mu(Zj=Zjsim)
                         Parameters[i,2] <- mu_ML
                         Parameters[i,3] <- reg["Sigma"]
                         Parameters[i,4] <- FgnVar(Zjsim, mu_ML)
                         Parameters[i,5] <- reg["H"]
                         Parameters[i,6] <- Hw
                         Parameters[i,7] <- estim.cf.alpha(Yj=Zjsim, FBM=T)
                         Parameters[i,8] <- Qstat(Zjsim, H=H, TT=Tlenght)
                         Parameters[i,9] <- mean(Zjsim)
                         Parameters[i,10] <- sd(Zjsim)
          }
          write.csv(Parameters, file.path(output_supporting_path,
                                 paste0("TD1.2.3_Full_Estimator_Tab_H=", H,".csv")))

          print(Sys.time() - start)
}
```

```
#--- 1. Matrix building
Hvec <- c(0.7, 0.8, 0.9, 0.95)
estimator_names <- c("Mu_c", "Mu_ML", "Sigma_c", "Sigma_ML", "H_c", "H_ML",
                     "Alpha_c", "Q_Stat")
Boots_Estim <- matrix(0, nrow=4, ncol=8)
 colnames(Boots_Estim) <- estimator_names
```

```
 rownames(Boots_Estim) <- Hvec
Boots_SE <- Boots_Estim

# 2. Estimation computing
for (j in 1:4){
      H <- Hvec[j]
      Parameters <- read.csv(file.path(output_supporting_path,
                                 paste0("TD1.2.3_Full_Estimator_Tab_H=", H,".csv")))[,2:9]
                 estim <- apply(Parameters, 2, mean)
                 SE <- apply(Parameters, 2, sd)
                 Boots_Estim[j,] <- estim
                 Boots_SE[j,] <- SE
      }


Boots_Estim <- as.data.frame(Boots_Estim)
Boots_SE <- as.data.frame(Boots_SE)

write.csv(Boots_Estim, file.path(output_supporting_path,
                                 "TD1.2.3_properties_boots_estim.csv"))
write.csv(Boots_SE, file.path(output_supporting_path,
                                 "TD1.2.3_properties_boots_SE.csv"))
```

Table D1. Results of the bootstrap simulations for estimators of μ and σ

- μ_c is the characteristic function estimator for the mean, and $SE(\mu_c)$ is its bootstrap simulation standard error
- μ_{ML} is the maximum likelihood estimator for the mean, and $SE(\mu_{ML})$ is its bootstrap standard error
- σ_c is the characteristic function estimator for the standard deviation, and $SE(\sigma_{ML})$ is its bootstrap standard error
- σ_{ML} is the maximum likelihood estimator for the standard deviation, and $SE(\sigma_{ML})$ is its bootstrap standard error

```
#=== TD1 READING and BUILDING
TD123.bootsMuSigmaEstim <- read.csv(file.path(output_supporting_path,
                                        "TD1.2.3_properties_boots_estim.csv"))
TD123.bootsMuSigmaSE <- read.csv(file.path(output_supporting_path,
                                        "TD1.2.3_properties_boots_SE.csv"))
TD1a <- data.frame(
  H = TD123.bootsMuSigmaEstim[,1],
  Mu_c = TD123.bootsMuSigmaEstim$Mu_c,
  Mu_c_SE = TD123.bootsMuSigmaSE$Mu_c,
  Mu_ML = TD123.bootsMuSigmaEstim$Mu_ML,
  Mu_ML_SE = TD123.bootsMuSigmaSE$Mu_ML
  )

TD1b <- data.frame(
  H = TD123.bootsMuSigmaEstim[,1],
  Sigma_c = TD123.bootsMuSigmaEstim$Sigma_c,
  Sigma_c_SE = TD123.bootsMuSigmaSE$Sigma_c,
  Sigma_ML = TD123.bootsMuSigmaEstim$Sigma_ML,
```

```r
  Sigma_ML_SE = TD123.bootsMuSigmaSE$Sigma_ML
  )


kable(TD1a, digits=3, align='c', escape = F,
      col.names = c("H", "$\\mu_c$", "$SE(\\mu_c)$",
                    "$\\mu_{ML}$", "$SE(\\mu_{ML})$"))
```

H	μ_c	$SE(\mu_c)$	μ_{ML}	$SE(\mu_{ML})$
0.70	0.002	0.102	0.002	0.101
0.80	-0.001	0.222	-0.001	0.220
0.90	-0.012	0.485	-0.011	0.482
0.95	-0.024	0.700	-0.021	0.698

```r
kable(TD1b, digits=3, align='c', escape = F,
      col.names = c("H", "$\\sigma_c$", "$SE(\\sigma_c)$",
                    "$\\sigma_{ML}$", "$SE(\\sigma_{ML})$"))
```

H	σ_c	$SE(\sigma_c)$	σ_{ML}	$SE(\sigma_{ML})$
0.70	0.994	0.039	0.990	0.039
0.80	0.956	0.053	0.950	0.056
0.90	0.793	0.080	0.789	0.086
0.95	0.535	0.073	0.534	0.077

The results in Table D1 show that the standard errors of the respective estimators for the mean increase substantially when H increases. Also the estimators for σ become severely downward biased.

Table D2. Bootstrap simulations results for the H parameter

- H_c is the characteristic function estimator for the H parameter, and $SE(H_c)$ is its bootstrap standard error
- H_w is the Whittle estimator for the the H parameter, and $SE(\mu_{ML})$ is its bootstrap standard error

```r
#=== TD2 READING and BUILDING

TD2 <- data.frame(
  H = TD123.bootsMuSigmaEstim[,1],
  H_c = TD123.bootsMuSigmaEstim$H_c,
  H_c_SE = TD123.bootsMuSigmaSE$H_c,
  H_ML = TD123.bootsMuSigmaEstim$H_ML,
  H_ML_SE = TD123.bootsMuSigmaSE$H_ML
  )


kable(TD2, digits=3, align='c', escape = F,
      col.names = c("H", "$H_c$", "$SE(H_c)$",
                    "$H_{ML}$", "$SE(H_{ML})$"))
```

H	H_c	$SE(H_c)$	H_{ML}	$SE(H_{ML})$
0.70	0.693	0.019	0.700	0.015
0.80	0.781	0.021	0.799	0.015
0.90	0.860	0.023	0.899	0.015
0.95	0.891	0.021	0.947	0.014

Table D2 shows that the characteristic function estimator becomes downward biased as H increases.

Table D3. Bootstrap simulations results for the α parameter of the stable distribution

α_c is the characteristic function estimator for the α parameter, and $SE(\alpha_c)$ is its bootstrap standard error

```
#=== TD3 READING and BUILDING

TD3 <- data.frame(
  H = TD123.bootsMuSigmaEstim[,1],
  Alpha_c = TD123.bootsMuSigmaEstim$Alpha_c,
  Alpha_c_SE = TD123.bootsMuSigmaSE$Alpha_c
  )

# rownames(TD3) <- TD123.bootsMuSigmaEstim[,1]

kable(TD3, digits=3, align='c', escape = F,
      col.names = c("H", "$\\alpha_c$", "$SE(\\alpha_c)$"))
```

H	α_c	$SE(\alpha_c)$
0.70	2	0.003
0.80	2	0.002
0.90	2	0.001
0.95	2	0.001

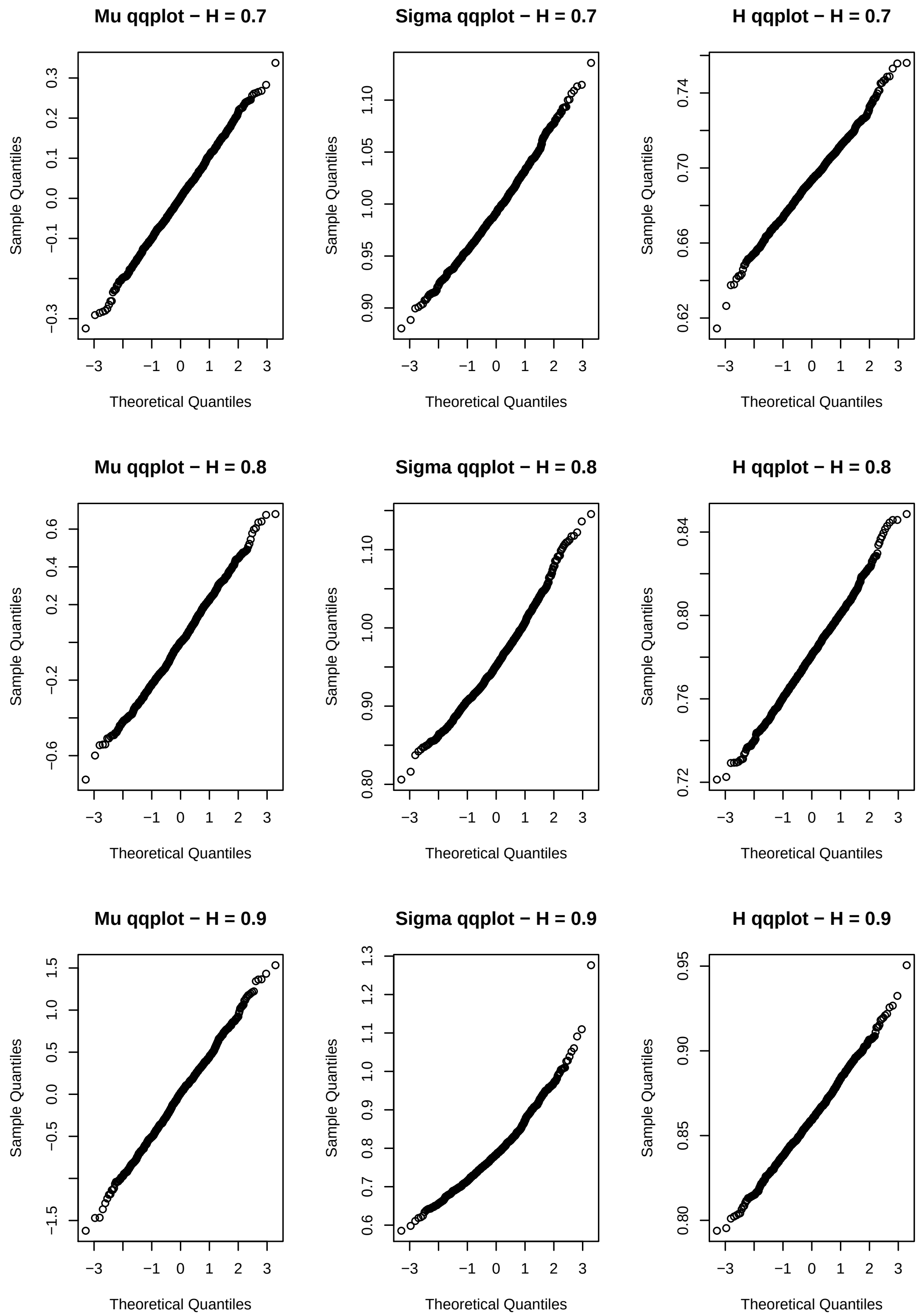
The results in Table D3 show that the characteristic function estimator in this case performs very well at any level of H.

Figure D1 and D2. Graphical test of asymptotic normality of the characteristic function estimators

Bootstrap estimates based on 1,000 simulated FGN series of length 2,000.

```
# 3x3 plot matrix
pdf(file.path(output_figure_path, "FD1_estimators_qqplot.pdf"),
    width=7, height=10)
par(mfrow=c(3, 3))
# Testing for different H values
  for(H in c(0.7, 0.8, 0.9)){
    # Data reading: estimators value during bootstrap simulations
    Parameters <- read.csv(file.path(output_supporting_path,
                               paste0("TD1.2.3_Full_Estimator_Tab_H=", H,".csv")))
    # Q-qplot
    qqnorm(Parameters$mu_c, main=paste("Mu qqplot - H =", H))
    qqnorm(Parameters$Sigma_c, main=paste("Sigma qqplot - H =", H))
    qqnorm(Parameters$H_c, main=paste("H qqplot - H =", H))
  }
dev.off()
```

```
include_graphics(file.path(output_figure_path, "FD1_estimators_qqplot.pdf"))
```




```
pdf(file.path(output_figure_path, "FD2_estimators_hist.pdf"),
    width = 7, height = 10)
# 3x3 plot matrix
par(mfrow=c(3, 3))
# Testing for different H values
for(H in c(0.7, 0.8, 0.9)){
```

```
  # Data reading: estimators value during bootstrap simulations
  Parameters <- read.csv(file.path(output_supporting_path,
                                   paste0("TD1.2.3_Full_Estimator_Tab_H=", H,".csv")))
  # Histograms
  hist(Parameters$Mu_c, breaks = 25, col ="black",
       main=paste("Mu distribution - H =", H), xlab=NULL)
  abline(v=0, col=2)
  hist(Parameters$Sigma_c, breaks = 25, col ="black",
       main=paste("Sigma distribution - H =", H), xlab=NULL)
  abline(v=1, col=2)
  hist(Parameters$H_c, breaks = 25, col ="black",
       main=paste("H distribution - H =", H), xlab=NULL)
  abline(v=H, col=2)
}
dev.off()
```

```
include_graphics(file.path(output_figure_path, "FD2_estimators_hist.pdf"))
```

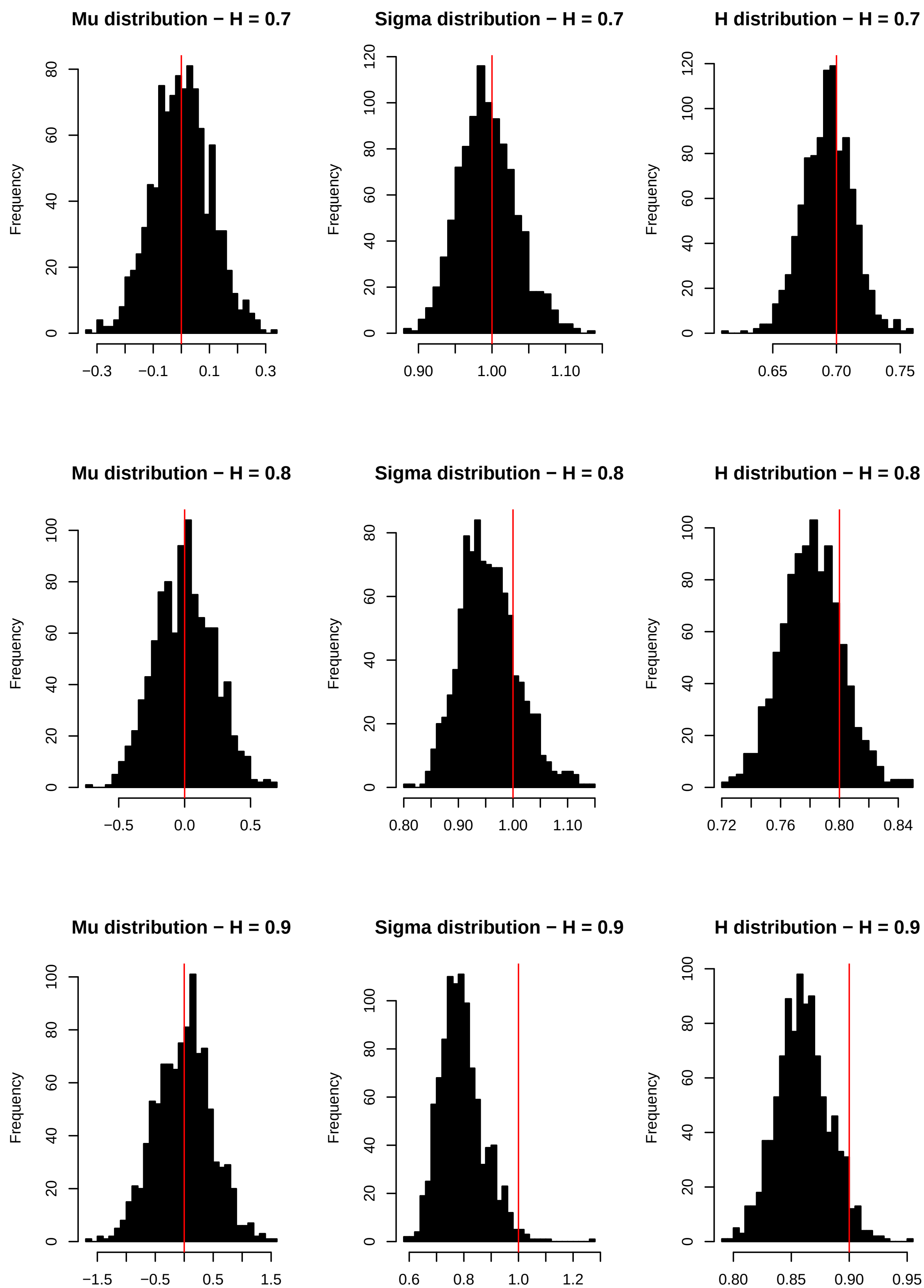

Mu distribution − H = 0.7
Sigma distribution − H = 0.7
H distribution − H = 0.7
Mu distribution − H = 0.8
Sigma distribution − H = 0.8
H distribution − H = 0.8
Mu distribution − H = 0.9
Sigma distribution − H = 0.9
H distribution − H = 0.9
Frequency
−0.3 −0.1 0.1 0.3
0.90 1.00 1.10
0.65 0.70 0.75
−0.5 0.0 0.5
0.80 0.90 1.00 1.10
0.72 0.76 0.80 0.84
−1.5 −0.5 0.5 1.5
0.6 0.8 1.0 1.2
0.80 0.85 0.90 0.95

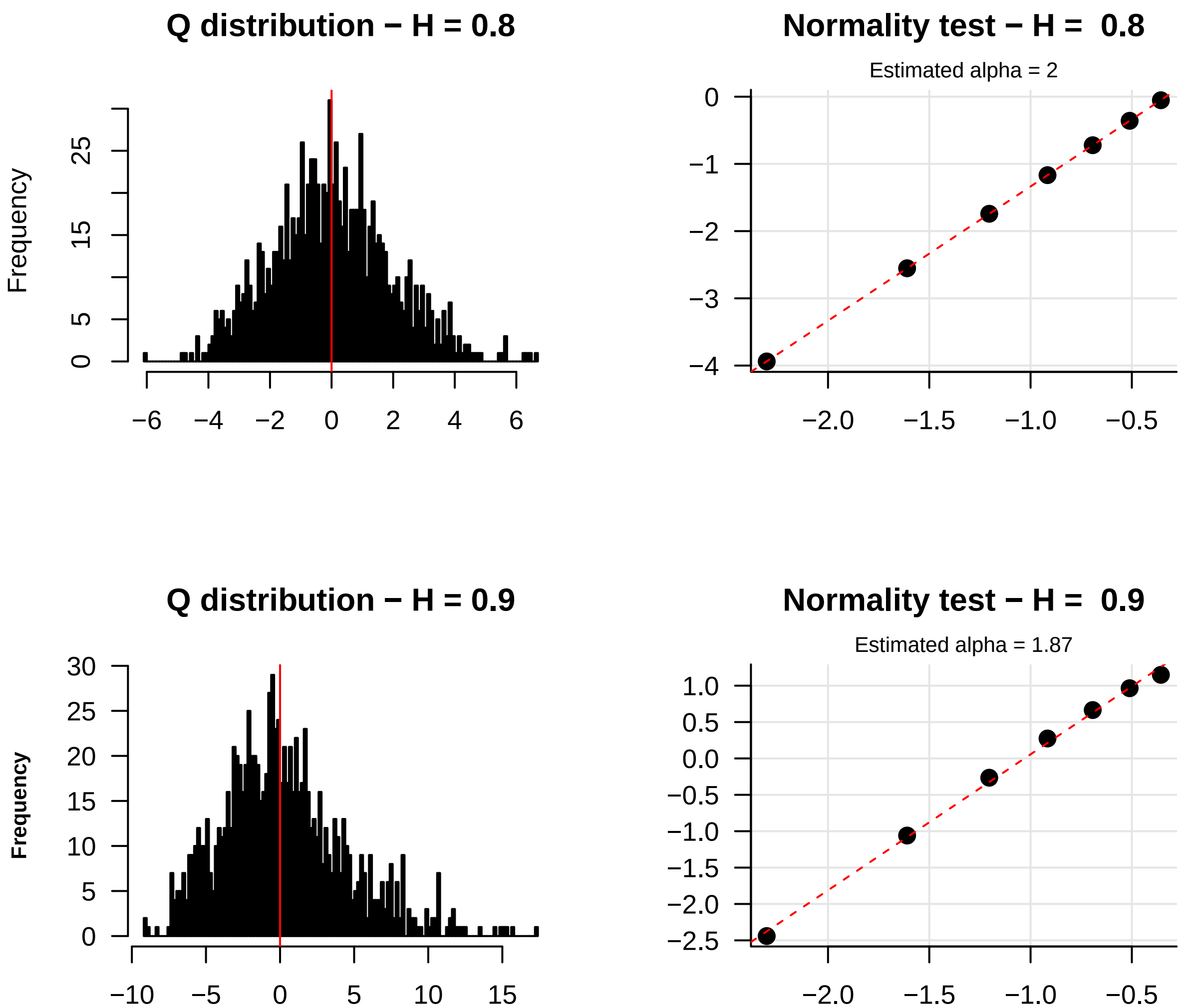
Figure D3. Graphical test of the distribution of the Chi-Square statistics Q when estimated H values are inserted

Bootstrap estimates based on 1,000 simulated FGN series of length 2,000.

H estimated by the Whittle method.

```
#--- 1. Parameters choice
Hvec <- c(0.8, 0.9)
N <- 1000
Tlenght <- 2000
estimator_names <- c("Hw", "Q.Hw")

start <- Sys.time()

#--- 2. Bootstrap estimation
for (H in Hvec){
          start <- Sys.time()
          print(paste0("----- H = ", H, " ----- ", Sys.time()))
          Parameters <- matrix(0, nrow=N, ncol=2)
          colnames(Parameters) <- estimator_names
          #--- 3a. Estimation for given H
          for (i in 1:N){
                         Zjsim <- simFGN0(Tlenght, H)
                         Hw <- estim.w.H(Zjsim)
                         mu_ML <- FgnMean(Zjsim, H=Hw, sigma=1)
                         Parameters[i,1] <- Hw
                         Parameters[i,2] <- Qstat(Zjsim, H=Hw, TT=Tlenght)
          }
          write.csv(Parameters, file.path(output_supporting_path,
                                paste0("FD3_Q.Hw_Estimation_H=", H,".csv")))

          print(Sys.time() - start)
}
```

```
# 3x3 plot matrix
# png(file.path(output_figure_path, "FD3_Qstat_distribution_Hw.png"),
#     width = 960, height = 950)
par(mfrow=c(2, 2))
# Testing for different H values
  for(H in c(0.8, 0.9)){
    # Data reading: estimators value during bootstrap simulations
    Q_Stat <- read.csv(file.path(output_supporting_path,
                                paste0("FD3_Q.Hw_Estimation_H=", H,".csv")))$Q.Hw
    # Q-qplot
   hist(Q_Stat, breaks = 100, col ="black",
        main=paste("Q distribution - H =", H), xlab=NULL)
   abline(v=0, col=2)
   fgtNormality(Yj=cumsum(Q_Stat), xmax=0.7, main=paste("- H = ", H), cex.dots=1.5,
                    cex.axis=1,cex.main=1.2)$plot
  }
```




```
# dev.off()
```

```
# include_graphics(file.path(output_figure_path, "FD3_Qstat_distribution_Hw.png"))
```

Figure D3 shows that Q is approximately normally distributed when H = 0.8 whereas the distribution becomes skew to the right when H = 0.9. (similar to a stable distribution that is totally skew to the right)

Figure D4. Theoretical vs unbiased bootstrap autocorrelations

```
unbiased_autocorr_tbl <- read.csv(
  file.path(output_supporting_path, "FD4_Unbiased_autocorrelation_tbl.csv"))

unbiased_autocorr_long2 <- unbiased_autocorr_tbl %>%
  select(Lag, Theoretical, Simulated = simulated) %>%
  gather("Autocorrelation", "Value", -Lag)

unbiased_autocorr_long2 %>%
```

```
  ggplot(aes(x = Lag, y = Value, colour = Autocorrelation)) +
  ylim(0, 1) +
  geom_line()  +
  geom_point() +
  scale_colour_manual(values = c("orange", "blue"),
                      labels = c("Simulated", "FGN model")) +
  theme_minimal() +
  theme(legend.position = "bottom") +
  labs(title = "Bootstrap vs FGN autocorrelations",
       subtitle = "H = 0.95 - with bias correction formula",
       y = "Autocorrelation")

ggsave(file.path(output_figure_path, "FD4_bootstrap_autocorrelation.png"))
```

```
include_graphics(file.path(output_figure_path, "FD4_bootstrap_autocorrelation.png"))
```

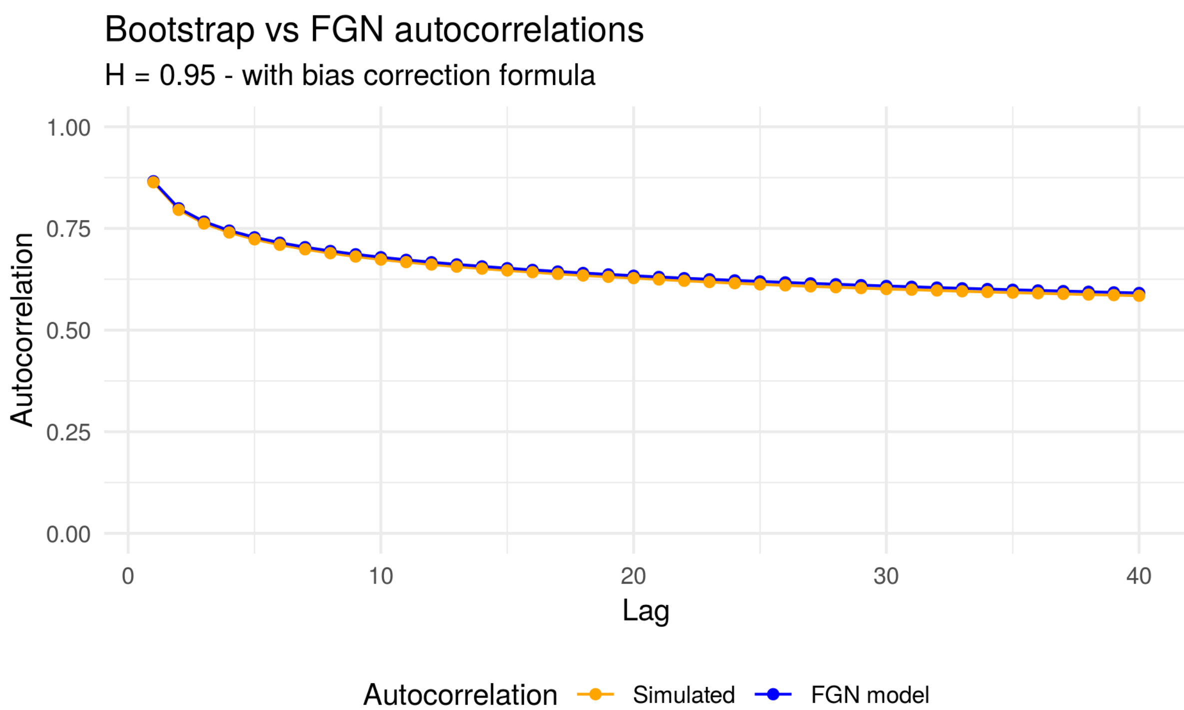


Figure D5. Impact of missing data on the Q statistics

Simulation of 1,000 time series of length 2,000 and H = 0.8.

Allowing for 70 randomly dispersed data points.

```r
# Not run - About 25 hours

# Fgn simulation
db <- sim.multiFGN(N = 1000, Tj = 2000, H = 0.8)
NNA <- 70

# Na imputation
set.seed(NULL)
na_in_vector <- function(x) {
  rnd_places <- sample(1:length(x), size = NNA)
  x[rnd_places] <- NA_real_
  x
}
db_with_NA <- map_dfc(db, na_in_vector)

# NA removal
remove_NA <- function(x) {
  na.omit(x)
}
db_without_NA <- map_dfc(db_with_NA, remove_NA)

# Q statistics computation
Q_vec <- apply(db_without_NA, 2, Qstat, H = 0.8, TT = 2000 - NNA)

saveRDS(Q_vec, file = file.path(output_supporting_path, "Q_H08_missing.rds"))
```

```r
Q_vec <- readRDS(file.path(output_supporting_path, "Q_H08_missing.rds"))
```

Bootstrap mean of Q: 0.244.

Bootstrap standard deviation of Q: 1.021.

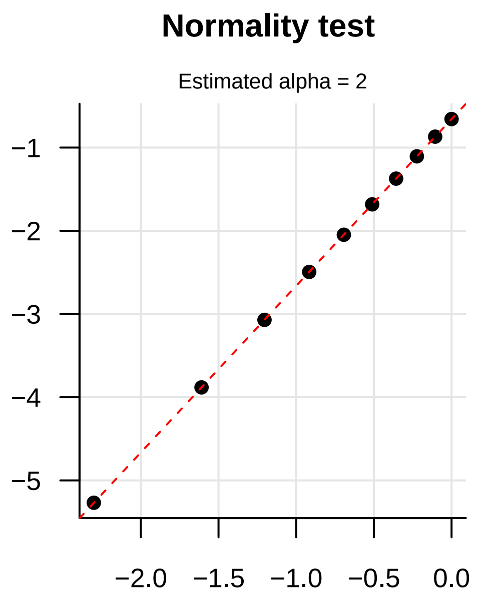
Normality of Q:

```r
png(file.path(output_figure_path, "FD5_Q_performance_with_missing_data.png"),
    width = 700, height = 480)
par(mfrow=c(1, 2))
fgtNormality(Yj=cumsum(Q_vec), xmax=1, main=NULL, cex.axis=0.8,
                         cex.lab=0.8, cex.main=1, cex.dots=1, lwd=1)$plot
qqnorm(Q_vec)
dev.off()
```

```r
par(mfrow=c(1, 2))
fgtNormality(Yj=cumsum(Q_vec), xmax=1, main=NULL, cex.axis=0.8,
                         cex.lab=0.8, cex.main=1, cex.dots=1, lwd=1)$plot
```

```
## NULL
```

```r
qqnorm(Q_vec)
```



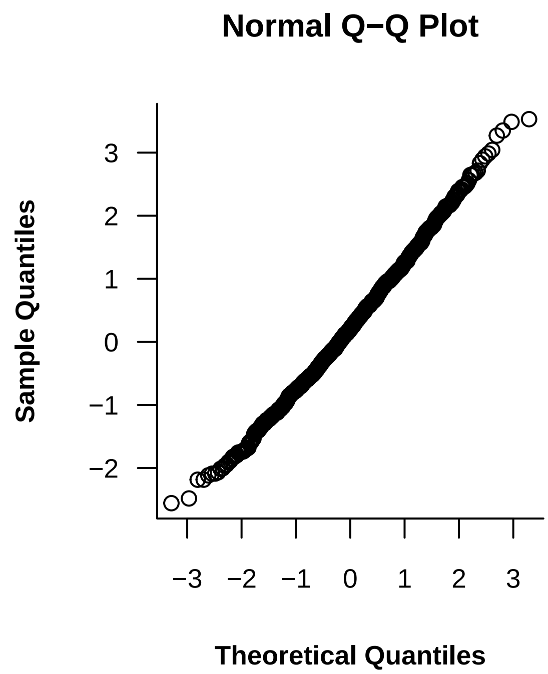


Figure D6. Performance of the Q statistics when H = 0.95

Simulation of 1,000 time series of length 2,000 and H = 0.95.

```
db <- sim.multiFGN(N = 1000, Tj = 2000, H = 0.95)

Q_vec <- apply(db, 2, Qstat, H = 0.95, TT = 2000)

saveRDS(Q_vec, file = file.path(output_supporting_path, "Q_H095.rds"))
```

```
Q_095 <- readRDS(file.path(output_supporting_path, "Q_H095.rds"))
```

Bootstrap mean of Q: -0.006.

Bootstrap standard deviation of Q: 0.963.

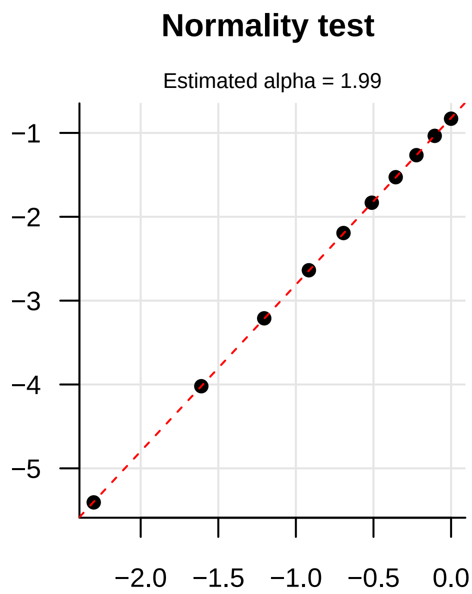
Normality of Q:

```
png(file.path(output_figure_path, "FD5_Q_performance_with_missing_data.png"),
    width = 700, height = 480)
par(mfrow=c(1, 2))
fgtNormality(Yj=cumsum(Q_095), xmax=1, main=NULL, cex.axis=0.8,
                         cex.lab=0.8, cex.main=1, cex.dots=1, lwd=1)$plot
qqnorm(Q_095)
dev.off()
```

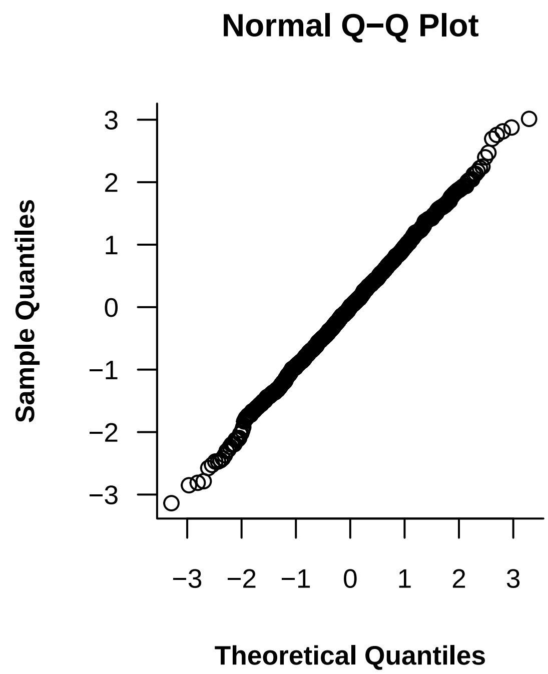
```
par(mfrow=c(1, 2))
fgtNormality(Yj=cumsum(Q_095), xmax=1, cex.axis=0.8, main = "",
                         cex.lab=0.8, cex.main=1, cex.dots=1, lwd=1)$plot
```

```
## NULL
```

```
qqnorm(Q_095)
```



Normality test
Estimated alpha = 1.99
−1
−2
−3
−4
−5
−2.0
−1.5
−1.0
−0.5
0.0



Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3