```
> require(fBasics)
Loading required package: fBasics
Loading required package: timeDate
Loading required package: timeSeries


Rmetrics Package fBasics
Analysing Markets and calculating Basic Statistics
Copyright (C) 2005-2014 Rmetrics Association Zurich
Educational Software for Financial Engineering and Computational Science
Rmetrics is free software and comes with ABSOLUTELY NO WARRANTY.
https://www.rmetrics.org --- Mail to: info@rmetrics.org
>
> #---------------------------------------------------------------
> ### Part 1 ###
> da <- read.table("m-ge3dx8113.txt",header=T)
> head(da)
  PERMNO     date        ge    vwretd    ewretd    sprtrn
1  12060 19810130  0.000000 -0.040085  0.005615 -0.045742
2  12060 19810227  0.089796  0.015521  0.002150  0.013277
3  12060 19810331  0.014981  0.046184  0.072674  0.036033
4  12060 19810430 -0.020522 -0.011268  0.027885 -0.023456
5  12060 19810529  0.001905  0.013551  0.027187 -0.001657
6  12060 19810630 -0.046768 -0.010242 -0.013194 -0.010408
> # a) basic stats of raw data
> basicStats(da$ge)
               X..da.ge
nobs         396.000000
NAs            0.000000
Minimum       -0.272877
Maximum        0.251236
1. Quartile   -0.025779
3. Quartile    0.053870
Mean           0.012900
Median         0.008022
Sum            5.108405
SE Mean        0.003572
LCL Mean       0.005878
UCL Mean       0.019922
Variance       0.005051
Stdev          0.071073
Skewness      -0.226160
Kurtosis       1.373376
> basicStats(da$vwretd)
```

```
            X..da.vwretd
nobs          396.000000
NAs             0.000000
Minimum        -0.225363
Maximum         0.128496
1. Quartile    -0.016682
3. Quartile     0.039373
Mean            0.009698
Median          0.014381
Sum             3.840419
SE Mean         0.002263
LCL Mean        0.005249
UCL Mean        0.014147
Variance        0.002028
Stdev           0.045036
Skewness       -0.780736
Kurtosis        2.526277
> basicStats(da$ewretd)
            X..da.ewretd
nobs          396.000000
NAs             0.000000
Minimum        -0.272248
Maximum         0.225012
1. Quartile    -0.019678
3. Quartile     0.039903
Mean            0.011022
Median          0.015401
Sum             4.364730
SE Mean         0.002686
LCL Mean        0.005740
UCL Mean        0.016304
Variance        0.002858
Stdev           0.053461
Skewness       -0.499120
Kurtosis        3.259182
> basicStats(da$sprtrn)
            X..da.sprtrn
nobs          396.000000
NAs             0.000000
Minimum        -0.217630
Maximum         0.131767
1. Quartile    -0.017593
3. Quartile     0.035838
Mean            0.007594
```

```
Median       0.011063
Sum         3.007062
SE Mean      0.002207
LCL Mean     0.003254
UCL Mean     0.011933
Variance     0.001929
Stdev       0.043921
Skewness    -0.658830
Kurtosis    2.204877
> # b) Log returns of the raw data
> basicStats(exp(da$ge)-1)
          X..exp.da.ge..1
nobs          396.000000
NAs             0.000000
Minimum        -0.238814
Maximum         0.285613
1. Quartile    -0.025450
3. Quartile     0.055348
Mean            0.015527
Median          0.008055
Sum             6.148507
SE Mean         0.003609
LCL Mean        0.008431
UCL Mean        0.022622
Variance        0.005158
Stdev           0.071822
Skewness        0.113377
Kurtosis        1.049145
> basicStats(exp(da$vwretd)-1)
          X..exp.da.vwretd..1
nobs              396.000000
NAs                 0.000000
Minimum            -0.201774
Maximum             0.137117
1. Quartile        -0.016544
3. Quartile         0.040159
Mean                0.010756
Median              0.014485
Sum                 4.259201
SE Mean             0.002251
LCL Mean            0.006330
UCL Mean            0.015181
Variance            0.002007
Stdev               0.044799
```

```
Skewness           -0.530368
Kurtosis            1.728979
> basicStats(exp(da$ewretd)-1)
          X..exp.da.ewretd..1
nobs              396.000000
NAs                 0.000000
Minimum            -0.238335
Maximum             0.252338
1. Quartile        -0.019485
3. Quartile         0.040710
Mean                0.012513
Median              0.015520
Sum                 4.955338
SE Mean             0.002693
LCL Mean            0.007220
UCL Mean            0.017807
Variance            0.002871
Stdev               0.053581
Skewness           -0.111537
Kurtosis            2.686021
> basicStats(exp(da$sprtrn)-1)
          X..exp.da.sprtrn..1
nobs              396.000000
NAs                 0.000000
Minimum            -0.195577
Maximum             0.140842
1. Quartile        -0.017439
3. Quartile         0.036487
Mean                0.008583
Median              0.011124
Sum                 3.398995
SE Mean             0.002197
LCL Mean            0.004264
UCL Mean            0.012903
Variance            0.001912
Stdev               0.043724
Skewness           -0.422155
Kurtosis            1.550534
> # c) Test the Null Hypothesis
> t.test(da$ge)

          One Sample t-test

data:  da$ge
```

```
t = 3.6119, df = 395, p-value = 0.0003432
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.005878371 0.019921654
sample estimates:
 mean of x
0.01290001

> # e) obtain emperical density plot
> d1=density(da$ge)
> d2=density(da$sprtrn)
> par(mfcol=c(1,2))
> plot(d1$x,d1$y,xlab='returns',ylab='density',main= "GE",type='l')
> plot(d2$x,d2$y,xlab='returns', ylab='density', main='SP', type='l')
>
> #-----------------------------------------------------------------
> ### Part 2 ###
> ge=da$ge
> lr <- (exp(da$ge)-1)
> lr
  [1]  0.0000000000  0.0939510949  0.0150937777 -0.0203128569  0.0019068157 -
0.0456912285
  [7] -0.0140414857 -0.0805834547 -0.0035726030 -0.0113253795  0.1195233339 -
0.0357830907
 [13]  0.0934359653  0.0060180361  0.0208912293  0.0119042988 -0.0344795533
0.0429153368
 [19]  0.0339630082  0.1554466741  0.0046618327  0.1604745311  0.0909457712
0.0260084679
 [25]  0.0951693403  0.0494948992 -0.0241281476  0.0676671130 -0.0731634033
0.0760285555
 [31] -0.0889723218  0.0253797174  0.0419312893 -0.0187784461  0.1202221345
0.0266057786
 [37] -0.0699296030 -0.0427378537  0.0670565820  0.0114551130 -0.0418903018 -
0.0023502339
 [43]  0.0000000000  0.0817401560 -0.0066011163  0.0318878617 -0.0270192683
0.0228296980
 [49]  0.1365927827  0.0047080482 -0.0682837841  0.0000000000  0.0352800812
0.0208330401
 [55]  0.0370332579 -0.0475646040 -0.0472102321  0.0021763649  0.1489276166
0.1245804132
 [61] -0.0254437112  0.1070048113  0.0129866029 -0.0047656083  0.0210178277
0.0232123079
 [67] -0.0940444174  0.0801014803 -0.0754739027  0.0627607463  0.0925615663
0.0447054713
```

```
 [73]  0.1802166225  0.0316629347  0.0208187486 -0.0130254252  0.0048426881
0.0533957568
 [79]  0.0937968587  0.0563842582 -0.0106627456 -0.2052081321 -0.1001624213
0.0508065372
 [85]  0.0229217568  0.0083446239 -0.0996259582  0.0000000000  0.0410689273
0.0544043391
 [91] -0.0253150613 -0.0540409515  0.0865113858  0.0057806438  0.0320209838
0.0035623301
 [97]  0.0843774028 -0.0528179826 -0.0181934638  0.1033102731  0.1248469703 -
0.0463075128
[103]  0.1424464822 -0.0148456997 -0.0123313382 -0.0197150654  0.1270898987
0.0490688907
[109] -0.0342825670 -0.0004818839  0.0413167312 -0.0038834399  0.0871026088
0.0054598510
[115]  0.0348908891 -0.1257422165 -0.1071949223 -0.0448357846  0.0543083927
0.0589310967
[121]  0.1223985972  0.0730113056  0.0241848214  0.0162892464  0.1041557326 -
0.0411982968
[127] -0.0100838140  0.0224318946 -0.0644718099 -0.0053904193 -0.0597354433
0.2092024378
[133] -0.0162072264  0.0458709675 -0.0291380560  0.0116179704 -0.0032576822
0.0255253317
[139] -0.0159484548 -0.0321517785  0.0675913113 -0.0189864431  0.0883807051
0.0352003677
[145]  0.0073367833 -0.0229544444  0.0692142148  0.0169724223  0.0237250657
0.0399129420
[151]  0.0291374251 -0.0025347820 -0.0176042031  0.0118031134  0.0142759417
0.0761533821
[157]  0.0277932211 -0.0218008502 -0.0432144516 -0.0463895269  0.0456293992 -
0.0540617624
[163]  0.0837518975 -0.0123303505 -0.0251064564  0.0157060638 -0.0571272999
0.1248042269
[169]  0.0098522167  0.0651408032 -0.0061907578  0.0377314162  0.0363594053 -
0.0207301147
[175]  0.0476640798 -0.0021167565  0.0939182769 -0.0078123239  0.0631806197
0.0827228221
[181]  0.0681968072 -0.0161550840  0.0382639090 -0.0079938777  0.0737927422
0.0495253349
[187] -0.0455032111  0.0106947847  0.1054704600  0.0652260179  0.0778140907 -
0.0433129954
[193]  0.0478872560 -0.0060208019 -0.0297310715  0.1256807908  0.0918112343
0.0796155440
[199]  0.0863744940 -0.1022314650  0.0964393774 -0.0492508251  0.1538832579 -
0.0027033394
```

```
[205]  0.0578282652  0.0032312091  0.1189368574 -0.0115359448 -0.0210522415
0.0941250470
[211] -0.0124389882 -0.1001444243 -0.0017175234  0.1049101285  0.0334027521
0.1416824411
[217]  0.0285869838 -0.0425712756  0.1121476880 -0.0463494744 -0.0343887901
0.1176720552
[223] -0.0317848945  0.0308564896  0.0605259878  0.1535648301 -0.0388915258
0.2121455849
[229] -0.1276032762 -0.0093113781  0.1957038407  0.0104967079  0.0051803720
0.0059486232
[235] -0.0265443379  0.1513647755 -0.0146033219 -0.0486155125 -0.0913370085 -
0.0291264056
[241] -0.0400115252  0.0113731885 -0.0918484418  0.1727377821  0.0097320514 -
0.0050890069
[247] -0.0991441292 -0.0562849385 -0.0845467672 -0.0210130828  0.0590814756
0.0467749905
[253] -0.0704958434  0.0420438240 -0.0269044498 -0.1457599195 -0.0129109295 -
0.0594928237
[259]  0.1145313470 -0.0616807162 -0.1617601481  0.0246396604  0.0768703390 -
0.0907480041
[265] -0.0484775517  0.0486850017  0.0621455859  0.1675435362 -0.0251454514
0.0059405756
[271] -0.0083330857  0.0405329148  0.0146482493 -0.0264800877 -0.0116515883
0.0914946549
[277]  0.0893040434 -0.0266961851 -0.0596479946 -0.0185026841  0.0398380709
0.0487070244
[283]  0.0265821669 -0.0137397362  0.0306524002  0.0162109952  0.0370104434
0.0392112376
[289] -0.0100857938 -0.0194591777  0.0247329069  0.0038895447  0.0077649924 -
0.0431742658
[295] -0.0043196434 -0.0254671003  0.0083657994  0.0071534647  0.0548272400 -
0.0116891448
[301] -0.0635142749  0.0113620634  0.0598294511 -0.0054481050 -0.0094946386 -
0.0301831115
[307] -0.0081585370  0.0427995797  0.0447169631 -0.0053675430  0.0048537414
0.0646456279
[313] -0.0306931010 -0.0235736950  0.0129734342  0.0433336298  0.0197250172
0.0264210062
[319]  0.0126179429  0.0028420309  0.0749691892 -0.0057802298 -0.0673525345 -
0.0234858127
[325] -0.0450812303 -0.0525830523  0.1238687790 -0.1099298508 -0.0587542357 -
0.1140493237
[331]  0.0617813325 -0.0066934981 -0.0782626815 -0.2093516706 -0.1130254595 -
0.0377095973
```

```
[337] -0.2221604422 -0.2388135922  0.2068504111  0.2856134533  0.0678133934 -
0.1158645464
[343]  0.1541279071  0.0380178696  0.2074238013 -0.1232619325  0.1313617549 -
0.0481168564
[349]  0.0648021423  0.0049873959  0.1425355965  0.0369295598 -0.1246101947 -
0.1058904749
[355]  0.1251225916 -0.0967329203  0.1394264252 -0.0140543031 -0.0117899474
0.1785030130
[361]  0.1064403828  0.0467394007 -0.0407341244  0.0201503312 -0.0388348187 -
0.0315679901
[367] -0.0491234162 -0.0854617629 -0.0560036695  0.1028491858 -0.0467470638
0.1461310889
[373]  0.0456806363  0.0276328979  0.0550023559 -0.0241193648 -0.0247154457
0.1058076799
[379] -0.0043096865 -0.0019261426  0.1104662809 -0.0700784024  0.0033295306
0.0023688012
[385]  0.0633858334  0.0520261801 -0.0042977382 -0.0352632377  0.0472932724
0.0025763130
[391]  0.0522008309 -0.0492194498  0.0414583599  0.0987597020  0.0200921843
0.0614543542
> t.test(lr)

        One Sample t-test

data:  lr
t = 4.3019, df = 395, p-value = 2.137e-05
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.00843089 0.02262218
sample estimates:
 mean of x
0.01552653

> skewness(ge)
[1] -0.2261597
attr(,"method")
[1] "moment"
> tm3=skewness(ge)/sqrt(6/length(ge))
> tm3
[1] -1.83733
attr(,"method")
[1] "moment"
>
> kurtosis(ge)
```

```
[1] 1.373376
attr(,"method")
[1] "excess"
> tk=kurtosis(ge)/sqrt(24/length(ge))
> tk
[1] 5.578679
attr(,"method")
[1] "excess"
>
> #--------------------------------------------------------------
> ### Part 3 ###
> require(forecast)
Loading required package: forecast
> suppressMessages(require(fpp))
>
> # a) Make a plot of the data
> plot(visitors)
> plot
standardGeneric for "plot" defined from package "graphics"

function (x, y, ...)
standardGeneric("plot")
<environment: 0x108c1d6d0>
Methods may be defined for arguments: x, y
Use  showMethods("plot")  for currently available ones.
>
> # b) forecast the next two years using Holt-Winters' multiplicative method
> aust <- window(visitors)
> fit_multi <- hw(aust,seasonal="multiplicative")
> print(fit_multi)
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
May 2005       369.3175 343.3002 395.3348 329.5275 409.1076
Jun 2005       395.5080 365.2767 425.7393 349.2733 441.7427
Jul 2005       485.9444 446.0391 525.8497 424.9145 546.9743
Aug 2005       436.7465 398.5070 474.9859 378.2643 495.2287
Sep 2005       422.9069 383.6657 462.1481 362.8927 482.9211
Oct 2005       478.2627 431.4628 525.0627 406.6885 549.8370
Nov 2005       502.5833 450.9301 554.2365 423.5865 581.5800
Dec 2005       615.6455 549.4181 681.8728 514.3595 716.9314
Jan 2006       461.1564 409.3845 512.9284 381.9781 540.3348
Feb 2006       511.8202 452.0068 571.6335 420.3436 603.2968
Mar 2006       498.9206 438.3614 559.4798 406.3033 591.5378
Apr 2006       443.9647 388.1032 499.8261 358.5320 529.3974
May 2006       383.5190 333.5830 433.4550 307.1484 459.8896
```

```
Jun 2006       410.6680 355.4225 465.9134 326.1774 495.1585
Jul 2006       504.5116 434.4881 574.5350 397.4199 611.6032
Aug 2006       453.3808 388.5399 518.2217 354.2152 552.5464
Sep 2006       438.9632 374.3497 503.5767 340.1454 537.7811
Oct 2006       496.3635 421.2456 571.4814 381.4806 611.2464
Nov 2006       521.5446 440.4747 602.6146 397.5588 645.5305
Dec 2006       638.7996 536.9011 740.6982 482.9592 794.6400
Jan 2007       478.4461 400.1915 556.7008 358.7660 598.1263
Feb 2007       530.9496 441.9744 619.9248 394.8738 667.0255
Mar 2007       517.5100 428.7206 606.2994 381.7183 653.3017
Apr 2007       460.4553 379.6266 541.2840 336.8384 584.0721
> plot(fit_multi)
> plot
standardGeneric for "plot" defined from package "graphics"

function (x, y, ...)
standardGeneric("plot")
<environment: 0x108c1d6d0>
Methods may be defined for arguments: x, y
Use  showMethods("plot")  for currently available ones.
>
> # d) compare with exponential or damped and compare
> fit_multi_damped <- hw(aust,seasonal="multiplicative",damped=TRUE)
> plot(forecast(fit_multi_damped))
> plot
standardGeneric for "plot" defined from package "graphics"

function (x, y, ...)
standardGeneric("plot")
<environment: 0x108c1d6d0>
Methods may be defined for arguments: x, y
Use  showMethods("plot")  for currently available ones.
>
> fit_multi_exp <- hw(aust,seasonal="multiplicative",exponential=TRUE)
> plot(forecast(fit_multi_exp))
> plot
standardGeneric for "plot" defined from package "graphics"

function (x, y, ...)
standardGeneric("plot")
<environment: 0x108c1d6d0>
Methods may be defined for arguments: x, y
Use  showMethods("plot")  for currently available ones.
>
```

```
> fit_multi_exp_damped <- hw(aust,seasonal="multiplicative",
+                        exponential=TRUE,damped=TRUE)
> plot(forecast(fit_multi_exp_damped))
> plot
standardGeneric for "plot" defined from package "graphics"

function (x, y, ...)
standardGeneric("plot")
<environment: 0x108c1d6d0>
Methods may be defined for arguments: x, y
Use  showMethods("plot")  for currently available ones.
>
> accuracy(fit_multi)
                  ME    RMSE      MAE        MPE     MAPE      MASE      ACF1
Training set -0.9498442 14.8295 10.96716 -0.8150922 4.271167 0.4050069 0.2223887
> accuracy(fit_multi_damped)
                 ME    RMSE      MAE        MPE     MAPE      MASE       ACF1
Training set 0.9123468 14.44801 10.64909 0.07071844 4.064322 0.3932608 0.01740636
> accuracy(fit_multi_exp_damped)
                 ME    RMSE      MAE        MPE     MAPE      MASE       ACF1
Training set 0.7230142 14.45533 10.72791 0.03798703 4.090931 0.3961716 0.01218167
>
> #---------------------------------------------------------------
> ### Part 4 ###
>
> # a)
> fit_multi <- hw(aust,seasonal="multiplicative")
> plot(fit_multi)
> plot
standardGeneric for "plot" defined from package "graphics"

function (x, y, ...)
standardGeneric("plot")
<environment: 0x108c1d6d0>
Methods may be defined for arguments: x, y
Use  showMethods("plot")  for currently available ones.
> hist(residuals(fit_multi),nclass=20)
> plot
standardGeneric for "plot" defined from package "graphics"

function (x, y, ...)
standardGeneric("plot")
<environment: 0x108c1d6d0>
Methods may be defined for arguments: x, y
```

```
Use  showMethods("plot")  for currently available ones.
> plot(residuals(fit_multi))
> plot
standardGeneric for "plot" defined from package "graphics"

function (x, y, ...)
standardGeneric("plot")
<environment: 0x108c1d6d0>
Methods may be defined for arguments: x, y
Use  showMethods("plot")  for currently available ones.
> accuracy(fit_multi)
                     ME    RMSE      MAE        MPE     MAPE      MASE      ACF1
Training set -0.9498442 14.8295 10.96716 -0.8150922 4.271167 0.4050069 0.2223887
>
> # b)
> fit_mam <- ets(visitors, model="ZZZ")
> plot(forecast(fit_mam))
> hist(residuals(fit_mam),nclass=20)
> plot(residuals(fit_mam))
> accuracy(fit_mam)
                    ME     RMSE      MAE        MPE     MAPE      MASE         ACF1
Training set -1.536043 15.86105 11.53405 -0.7017724 4.076346 0.4259416 -0.004687451
>
> # c)
> fit_ana_box <- ets(visitors,additive.only=TRUE,lambda=TRUE)
> plot(forecast(fit_ana_box))
> hist(residuals(fit_ana_box),nclass=20)
> plot(residuals(fit_ana_box))
> accuracy(fit_ana_box)
                   ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
Training set 2.346807 17.51126 13.18528 0.5506054 5.103531 0.4869199 0.02105629
>
> # d)
> fit_naive <- snaive(visitors,lambda=TRUE)
> plot(forecast(fit_naive))
> hist(residuals(fit_naive),nclass=20)
> plot(residuals(fit_naive))
> accuracy(fit_naive)
                   ME     RMSE      MAE      MPE     MAPE MASE      ACF1
Training set 18.22368 32.56941 27.07895 7.011798 10.12935    1 0.6600405
>
> # e)
> fit_stld <- stlf(visitors,method="ets",lambda=TRUE)
> plot(forecast(fit_stld))
```

```
>
> hist(residuals(fit_stld),nclass=20)
> plot(residuals(fit_stld))
> accuracy(fit_stld)
                     ME     RMSE      MAE       MPE     MAPE      MASE        ACF1
Training set -0.3615751 12.17064 9.129055 -0.226499 3.252608 0.3371274 -0.02051013
```